JAI 2017

Jornadas de Atualização em Informática

Organização

Flávia C. Delicato

Paulo F. Pires

Ismar Frango Silveira

XXXVI JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA

De 02 a 06 de Julho de 2017

São Paulo – SP

ANAIS

Sociedade Brasileira de Computação – SBC

Organizadores

Flávia C. Delicato

Paulo F. Pires

Ismar Frango Silveira

Realização

Sociedade Brasileira de Computação – SBC

Editores

Flávia C. Delicato (Universidade Federal do Rio de Janeiro)
Paulo F. Pires (Universidade Federal do Rio de Janeiro)
Ismar Frango Silveira (Universidade Presbiteriana Mackenzie)

Título - Jornadas de Atualização em Informática 2017

Local - Porto Alegre/RS

Ano de Publicação - 2017

Edição - 1ª

Editora - Sociedade Brasileira de Computação - SBC

Organizadores - Flávia C. Delicato (UFRJ), Paulo F. Pires (UFRJ), Ismar Frango Silveira (UPM)

ISBN: 978-85-7669-374-1

Comitê de Organização

Coordenação Geral

Flávia Coimbra Delicato (UFRJ)
Paulo de Figueiredo Pires (UFRJ)

Coordenação Local

Ismar Frango Silveira (Mackenzie)

Comitê de Programa

Alfredo Goldman, IME-USP
André Carlos Ponce de Leon Ferreira de Carvalho, USP
Bruno Schulze, LNCC
Cecilia Rubira, UNICAMP
Celina de Figueiredo, UFRJ
Cláudia Linhares Sales, UFC
Claudia Werner, UFRJ
Fabio Kon, IME-USP
Fabio Protti, UFF
Fabiola Greve, UFBA
Jose Carlos Maldonado, ICMC-USP
Jose Marcos Nogueira, UFMG
Karin Breitman, Dell EMC
Luci Pirmez, UFRJ
Marta Mattoso, UFRJ
Philippe O. A. Navaux, UFRGS
Rodolfo Azevedo, UNICAMP
Thais Vasconcelos Batista, UFRN

Índice

Prefácio

A Jornada de Atualização em Informática (JAI), promovida pela Sociedade Brasileira de Computação, é um dos mais importantes eventos acadêmicos de atualização científica e tecnológica da comunidade de Computação do Brasil. A JAI compreende trabalhos de pesquisadores sêniores da nossa comunidade, oferecendo uma oportunidade única para acadêmicos e profissionais atualizarem-se em temas diversos, interagindo com líderes das mais variadas áreas de pesquisa no Brasil. A JAI é tradicionalmente oferecida no escopo do Congresso da Sociedade Brasileira de Computação (CSBC). Neste ano de 2017, temos a sua 36ª edição, que irá ocorrer como parte do XXXVII CSBC na cidade de São Paulo, de 2 a 6 de julho, na Universidade Presbiteriana Mackenzie, Campus Higienópolis.

Gostaríamos de agradecer imensamente aos autores que submeteram propostas para a edição de 2017 da JAI, bem como aos membros do Comitê de Programa, os quais – autores e avaliadores – contribuíram para a seleção de um conjunto de textos que tratam de temas atuais, avançados e de grande relevância técnico-científica. Agradecemos também a atenção e o apoio da organização geral do CSBC 2017 e da Diretoria da SBC que propiciaram todas as condições para a realização desta edição da JAI.

É importante destacar que a JAI favorece a participação de pesquisadores renomados, nacional e internacionalmente, com a colaboração de talentos emergentes, propiciando um ambiente rico para a evolução e disseminação do conhecimento no âmbito do CSBC.

Nesta edição da JAI, são apresentados seis textos, sendo três deles cursos convidados, a partir de sugestões da comunidade organizadora do evento. Os textos abordam um amplo espectro de temas atuais e relevantes para a computação e suas aplicações, e certamente contribuirão para o desenvolvimento de soluções complexas e multidisciplinares relacionadas aos Grandes Desafios da Computação.

O texto “Simulação de Multidões - Aplicações em Segurança e Conforto de Pessoas” introduz um tema muito importante para várias aplicações atuais, como entretenimento e segurança. Particularmente em segurança, o planejamento de evacuação de ambientes públicos é uma tarefa relevante e complexa a ser realizada como parte do projeto de construção de um novo empreendimento. O texto discorre sobre vários tópicos atuais da área de pesquisa em simulação de multidões, com foco especificamente em sistemas de evacuação de ambientes. Além do embasamento teórico e de discutir vários aspectos relacionados à análise do comportamento de uma multidão, o curso inclui uma atraente parte prática, onde são abordadas ferramentas existentes para simular multidões.

Já o texto “A Aplicação de Métodos Qualitativos em Computação” aborda uma temática totalmente diferente dentro da área de computação. A ampla adoção da tecnologia na vida das pessoas tem gerado uma crescente necessidade e um maior interesse em profissionais e pesquisadores da área de computação em conhecer aspectos humanos, psicológicos, e socioculturais dos usuários. Para tal, tem-se cada vez mais adotado, em computação, métodos de pesquisa qualitativos tradicionalmente usados em pesquisas nas ciências humanas e sociais. Assim, o objetivo deste texto é apresentar uma visão geral de métodos qualitativos de pesquisa e suas principais características, discutindo suas diferenças em relação a métodos quantitativos.

O texto “Preservação de Privacidade de Dados: Fundamentos, Técnicas e Aplicações” trata de um tema que é um constante desafio na área de computação e comunicação, ainda mais nos tempos atuais de dados abertos, comunicação sem fio e conectividade ubíqua: a privacidade dos dados. Tornar dados acessíveis ao público é muitas vezes necessário para realizar importantes análises, prever tendências e detectar padrões. No entanto, esses dados podem conter informações que identificam unicamente indivíduos, causando assim uma violação de privacidade. Manter a utilidade dos dados para que análises sejam realizadas e simultaneamente garantir a privacidade dos indivíduos é um problema que tem recebido muita atenção nos últimos anos. Este texto apresenta os principais conceitos em torno da preservação de privacidade de dados, além das técnicas para assegurar que indivíduos não possam ser reidentificados a partir do compartilhamento de suas informações. Da mesma forma que no texto do primeiro curso, também haverá um caráter prático, com a demonstração de aplicações em cenários reais que utilizam as técnicas apresentadas.

O texto “Computação aplicada a Cidades Inteligentes: Como dados, serviços e aplicações podem melhorar a qualidade de vida nas cidades”, como o próprio nome diz, trará uma visão geral e abrangente dessa importante e atual aplicação das Tecnologias de Informação e Comunicação. É notório o crescimento da população urbana e com isso a aceleração dos problemas de infraestrutura pública. Tornar as cidades mais inteligentes pode ajudar a melhorar os serviços urbanos aumentando a qualidade de vida de seus cidadãos. No curso, os autores discorrem sobre como plataformas de software podem ser usadas para facilitar a criação e integração de aplicações robustas para cidades inteligentes, apresentando os principais desafios técnicos e científicos que precisam ser enfrentados antes que essas plataformas possam ser amplamente utilizadas.

Os últimos dois textos abordam dois temas que têm atraído imensamente a atenção da comunidade científica e da indústria nos últimos anos: *Big Data* e *Deep Learning*. A disponibilização crescente de dados em grandes volumes nas ciências, indústria, governo e redes sociais está transformando o modo pelo qual essas áreas se desenvolvem. No centro dessas mudanças estão algoritmos construídos especialmente para lidar com os desafios em *Big Data* e sua adaptação para os modelos do tipo Map-Reduce. Nesse contexto, o texto “Algoritmos e Modelos de Programação em Big Data” apresenta os principais algoritmos de apoio ao processamento de dados em grande volume, bem como alguns exemplos de algoritmos que implementam novas estratégias para a busca de padrões em *Big Data.* Por outro lado, as chamadas *Redes Neurais Profundas* vêm provocando recentemente uma revolução na indústria de TI, abrindo diversos horizontes e possibilidades para as mais diversas áreas e aplicações. Graças ao investimento de grandes empresas, inúmeras ferramentas e plataformas vêm se tornando acessíveis para aplicar a técnica de *Deep Learning* na solução de diversos problemas. No texto “*Deep Learning* - Teoria e Prática” são apresentados conceitos básicos de redes neurais profundas, algumas bibliotecas e ferramentas, arquiteturas de GPUs e como as mesmas são capazes de viabilizar essa emergente área da computação.

Desejamos a todos os participantes do CSBC uma excelente Jornada de Atualização de Informática e reiteramos nossos imensos agradecimentos à comunidade da computação no Brasil. Esperamos que a compilação desses ricos textos contribua para motivar soluções originais, inovadoras, e comprometidas com a evolução da riqueza social. Uma sociedade que pretende participar da construção do futuro da humanidade, não somente como expectadora, necessita construir e nutrir, incessantemente, o espírito criativo e crítico da pesquisa.

Flávia C. Delicato (UFRJ) e Paulo de F. Pires (UFRJ) - *Coordenação Geral*
Ismar Frango Silveira (Mackenzie) - *Coordenação Local*

Apresentação dos Autores

Carla Faria Leitão é graduada em Psicologia pela Universidade do Estado do Rio de Janeiro (1988), especialista em Saúde Mental pela Universidade Federal do Rio de Janeiro(1993) e Mestre (1995) e Doutora (2003) em Semiotic Engineering Research Group (SERG), no Departamento de Informática da PUC-Rio. Sua pesquisa concentra-se na área de Interação Humano-Computador (IHC) e na investigação dos impactos culturais, sociais e psicológicos da experiência com Tecnologias da Informação e Comunicação (TICs). Uma das pioneiras na contribuição interdisciplinar da psicologia à área de IHC no Brasil, é coautora de inúmeros artigos e de dois livros internacionalmente publicados sobre métodos qualitativos de investigação em Engenharia Semiótica, uma teoria interdisciplinar de base semiótica para a análise científica de TICs. Seus principais interesses na área são: abordagens teórico-metodológicas para o estudo dos impactos de TICs, interdisciplinaridade e desenvolvimento de raciocínio computacional.

Cristina Nader Vasconcelos possui graduação no curso de Bacharel em Informática pela Universidade Federal do Rio de Janeiro (2003), mestrado em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (2005) e doutorado (2009) pela mesma instituição. Tem experiência na área de Ciência da Computação, com ênfase em Computação Visual, atuando principalmente nos seguintes temas: Visão Computacional e Processamento de Imagens, Reconhecimento de Padrões e Computação Gráfica. Suas contribuições principais incluem temas de processamento genérico paralelo em hardware gráfico aplicados a tarefas de visão computacional, métodos de otimização discreta em grafos para computação visual, processamento, gerenciamento e formatos de imagens e vídeo, estruturas de dados espaciais. Desde 2014, tem atuado na divulgação e formação de pesquisadores na área de aprendizado profundo. Hoje é professora adjunta da Universidade Federal Fluminense no Instituto de Computação.

Eduardo Felipe Zambom Santana é bacharel (2007) e mestre (2010) em Ciência da Computação pela Universidade Federal de São Carlos. Atualmente é aluno de doutorado no IME-USP e professor na Universidade Anhembi Morumbi. Atua nas áreas de Sistemas Distribuídos, Cidades Inteligentes e Simulação. Tem mais de 10 anos de experiência como programador e arquiteto de sistemas. Nos últimos anos tem realizado pesquisas na área de cidades inteligentes principalmente em simulações de cenários de mobilidade urbana e na identificação de requisitos funcionais e não-funcionais para o desenvolvimento de uma plataforma de software para cidades inteligentes.

Esteban W. Gonzalez Clua é professor da Universidade Federal Fluminense e coordenador do UFF Medialab. Em 2009 e em 2013 foi Jovem Cientista do Nosso Estado pela FAPERJ. Possui graduação em Computação pela Universidade de São Paulo, mestrado e doutorado em Informática pela PUC-Rio. Sua atuação está especialmente focada nas áreas de Video Games, Realidade Virtual, GPUs e visualização. É um dos fundadores do SBGames (Simpósio Brasileiro de Games e Entretenimento Digital). Em 2015 foi nomeado NVIDIA Fellow. Em 2007 recebeu o prêmio da ABRAGAMES como o maior contribuidor da academia para a indústria de

jogos digitais no Brasil. Esteban é membro do comitê de programa das mais importantes conferencias na área de entretenimento digital. Atualmente é coordenador do Centro de Excelência da NVIDIA no Brasil, que funciona no instituto de computação da Universidade Federal Fluminense. Esteban é do conselho de inovação da Secretaria da Cultura do Estado do Rio de Janeiro, membro da comissão permanente do Rio Criativo, Membro do Fórum permanente de Inovação e Tecnologia da Assembleia Legislativa do Rio de Janeiro e membro do conselho da Agência de Inovação da UFF.

Fabio Kon é Professor Titular de ciência da computação do IME-USP, coordenador adjunto de pesquisa para inovação da FAPESP e Editor-Chefe do SpringerOpen Journal of Internet Services and Applications. Atua nas áreas de Empreendedorismo Digital, Software Livre, Sistemas Distribuídos e Cidades Inteligentes. Além de forte atuação como pesquisador, tendo publicado vários artigos internacionais de alto impacto e recebido o ACM Middleware 10-Year Best Paper Award, Fabio é conselheiro voluntário de várias startups de software, inclusive na área de Cidades Inteligentes. Em 2016 Fabio foi agraciado com o título de ACM Distinguished Scientist.

Fabio Porto é Pesquisador Senior do Laboratório Nacional de Computação Científica e Coordenador do Laboratório Data Extreme Lab (DEXL). Possui Doutorado e Mestrado em Informática pela PUC-Rio, em 2001 e 1997, respectivamente. Foi pesquisador sênior da EPFL, em Lausanne, entre 2004 e 2009. Foi Coordenador geral do SBBD 20015 e é um dos Coordenadores gerais da Conferência Internacional Very Large Data Bases, a realizar-se em 2018. Seu interesse de pesquisa principal está no desenvolvimento de técnica e algoritmos para análise e gerência de grandes volumes de dados. Participa de colaborações internacionais, incluindo os projetos MUSIC, do qual é coordenador, em parceria com o INRIA, França, e do projeto HPC4e, de cooperação com a EU. No nível nacional, tem colaborações com diversas instituições, incluindo : Observatório Nacional, FIOCRUZ (ICICT e CIDACT), Comitê Olímpico Brasileiro e DELL-EMC.

Felipe Timbó Brito possui graduação em Computação pela Universidade Federal do Ceará (2010). Também é Mestre em Ciência da Computação pela Universidade Federal do Ceará (2015) com ênfase em privacidade de dados de trajetória. Atualmente é doutorando em Computação¸ também pela Universidade Federal do Ceará e atua como líder de projetos de pesquisa e desenvolvimento em Computação no Laboratório de Sistemas e Banco de Dados (LSBD/UFC). Possui publicações internacionais em diversas áreas, incluindo privacidade de dados. Suas áreas de interesse são mineração e privacidade de dados.

Javam de Castro Machado possui doutorado em Informática pela Universite de Grenoble (1995). É Professor Titular do Departamento de Computação da Universidade Federal do Ceará, foi vice-diretor do Centro de Ciências da mesma Universidade e atualmente coordena o Laboratório de Sistemas e Banco de Dados (LSBD/UFC). É também coordenador de vários projetos de pesquisa e desenvolvimento em Computação, além de atuar como pesquisador do Programa de Mestrado e Doutorado em Ciência da Computação da UFC. Participa de projetos de cooperação internacional com universidades Europeias e tem vários artigos publicados em veículos nacionais e internacionais, inclusive no tema de privacidade de dados em nuvem e privacidade de

dados móveis. No momento suas áreas de interesse são privacidade de dados, sistemas de banco de dados e computação em nuvens, além de sistemas distribuídos.

Raquel Oliveira Prates possui graduação em Ciência da Computação pela Universidade Federal de Minas Gerais (1991), mestrado em Informática pela PUC-Rio (1994) e doutorado em Informática pela PUC-Rio (1998). É professora associada da Universidade Federal de Minas Gerais desde 2006. Fez pós-doutorado na Pennsylvania State University, no College of Information Systems and Technology de agosto de 2014 a julho de 2015. Sua pesquisa é em Interação Humano Computador e Sistemas Colaborativos, atuando principalmente nos seguintes temas: engenharia semiótica, interação humano-computador, avaliação de interfaces, comunicabilidade, design de interfaces e programação por usuário final. É coordenadora da Comissão Especial de Sistemas Colaborativos (CESC) desde 2015 e foi Coordenadora da Comissão Especial de Interação Humano-Computador (CEIHC) da SBC de 2006 a 2010. Atualmente ela é também representante da SBC no Comitê Técnico de IHC (TC13) da IFIP TC13. Foi membro do Comitê Executivo do SIGCHI de 2001 a 2004 e participa como membro de diversos comitês de programa nacionais e internacionais de conferências nas áreas de IHC e Sistemas Colaborativos.

Soraia Raupp Musse possui graduação em Bacharelado em Informática pela Pontifícia Universidade Católica do Rio Grande do Sul (1990), mestrado em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (1994), mestrado em Cours Postgrade En Informatique Realité Virtuelle - Ecole Polytechnique Federale de Lausanne na Suíça (1997), doutorado em Doctorat En Science - Ecole Polytechnique Federale de Lausanne (2000) sob supervisão do Prof. Daniel Thalmann e pós doutorado pela University of Pennsylvania em 2016 (USA), onde trabalhou com o Prof. Norman Badler, um dos pioneiros na área de humanos virtuais. Atualmente é professora adjunto da PUCRS, atuando na pós-graduação em Ciência da Computação, orientando alunos de graduação, mestrado, doutorado e bolsistas posdoc. É coordenadora do VHLAB (www.inf.pucrs.br/vhlab) e bolsista de produtividade do CNPq (PQ1D). Sua pesquisa tem ênfase em Processamento Gráfico (Graphics), principalmente nos seguintes temas: computação gráfica, agentes sintéticos virtuais, multidões de agentes virtuais e visão computacional. Já publicou mais de 25 artigos em periódicos, sendo vários deles de grande impacto, como IEEE Signal Processing Magazine, ACM Presence, IEEE TVCG, The Visual Computer, IEEE Transactions on Circuits and Video Technology, Computers & Graphics e etc. No ano de 2007 publicou um livro em coautoria com seu orientador de doutorado Daniel Thalmann, na área de Simulação de multidões (título: Crowd Simulation), editado pela Springer-Verlag, re-editado em 2013. Possui h-index = 24 segundo o Google Scholar e ocupa a posição 1417 na lista dos 6.000 pesquisadores mais referenciados do Brasil (de acordo com o Google Scholar) divulgada pela Webometrics com base nos dados coletados em abril de 2015. É presidente da Comissão Especial de Jogos da SBC desde agosto de 2016.

Vinícius Jurinic Cassol é doutor em Ciência da Computação pela PUCRS - 2016. Durante o doutorado foi supervisionado pela Profa. Dra. Soraia Raupp Musse, tendo sua pesquisa focada em simulação de multidões. Em 2013, realizou doutorado sanduiche na Universidade da Pennsylvania, sob supervisão do Prof. Dr. Norman Badler. Atualmente é professor da Universidade do Vale do Rio dos Sinos - UNISINOS onde coordena o Curso de Desenvolvimento de Jogos Digitais da universidade em São Leopoldo e Porto Alegre/RS além do Estúdio de Jogos Atomic Rocket Entertainment. É membro da

Comissão Especial de Jogos da SBC desde agosto de 2016. Entre as suas principais áreas de interesse encontram-se simulação de multidões, computação gráfica, gamificação e o processo de produção e desenvolvimento de jogos.

Capítulo 1

Simulação de Multidões - Aplicações em Segurança e Conforto de Pessoas

Soraia Raupp Musse e Vinicius Cassol

Abstract

Crowd Simulation is nowadays an important area in applications such as entertainment and security. In particular in security, evacuation planning is an important and difficult task in building design. In this work we present some topics about general crowd simulation aspects and focus specifically on evacuation systems. We present some tools to simulate crowds and also discuss the crowd analysis aspects.

Resumo

Simulação de multidões é hoje em dia uma área importante em aplicações como entretenimento e segurança. Particularmente, em segurança, o planejamento de evacuação de ambientes públicos é uma tarefa importante e difícil no projeto de construção de um novo empreendimento. Neste trabalho apresenta-se alguns tópicos importantes sobre a área de pesquisa em simulação de multidões e foca-se especificamente em sistemas de evacuação de ambientes. São abordadas algumas ferramentas para simular multidões e também discutidos aspectos relacionados à análise do comportamento de uma multidão.

1.1. Introdução

Entusiastas de diferentes áreas observaram o comportamento das pessoas por muitos anos, décadas ou mesmo séculos. Tal observação pôde produzir dados valiosos para serem considerados como objeto de estudo em diferentes campos, da engenharia à psicologia, por exemplo.

A compreensão do comportamento das pessoas é um enorme campo de pesquisa na área de psicologia desde séculos atrás. No início do século XX, Freud já desenvolveu estudos que envolvem a observação do comportamento das pessoas por décadas [1]. Apoiado por estudos de LeBon [2] e McDougall [3], Freud discorre sobre o comportamento dos seres humanos quando fazem parte de um grupo e define a multidão como uma entidade temporária, constituída por elementos heterogêneos que se uniram por um momento específico.

Muitos outros aspectos foram observados no mesmo campo em séculos passados. Um deles é muito importante mencionar e foi observado primeiramente por LeBon [2] que diz que quando parte de uma multidão, os indivíduos podem executar comportamentos incomuns; comportamentos que o indivíduo não realizaria sozinho. Neste tipo de situação os indivíduos podem agir de forma coletiva e um pensamento de multidão surge como uma nova entidade. Esta nova entidade pode fazer com que as pessoas se sintam, pensem e ajam de diferentes maneiras, podendo também realizar comportamentos perigosos que podem ser responsáveis por momentos fatais. Além disso, o indivíduo pode ter seu julgamento afetado. De acordo com o autor, os indivíduos da multidão geralmente podem descartar seus próprios valores e também suas inibições e apresentar comportamentos que não seriam realizados se estivessem sozinhos. Esse comportamento incomum pode desenvolver sentimentos de emoções diferentes nas pessoas. Esses sentimentos, por exemplo ansiedade, nervosismo ou pânico, tornam o indivíduo mais emotivo e às vezes irracional. Da mesma forma, Sighele [4] destaca situações em que as pessoas perdem a razão quando estão em multidões e agem contra diferentes alvos, incluindo o próprio Estado. Ambos os autores discutem o poder de uma multidão, que é capaz de construir uma força incontrolável e imprevisível.

Comportamentos coletivos emergentes, muitas vezes imprevisíveis, podem ocorrer quando as pessoas são parte de uma multidão, e podem compartilhar idéias, sentimentos e ter o mesmo objetivo ou um objetivo semelhante. Além disso, estudos científicos recentes consideram as multidões como uma entidade capaz de pensar [5]. Sighele [4] considera a multidão como uma estrutura ***heterogênea*** **e** ***inorgânica***. Ele considera *heterogênea* porque, geralmente, uma multidão é composta por indivíduos de todas as idades, gêneros e diferentes realidades sociais e culturais. Além disso, uma multidão é considerada *inorgânica* por sua capacidade de emergir de uma maneira repentina, sem um controle formal ou organização.

Sabe-se que alguns lugares podem ser propícios à formação de multidões. Esses locais podem incluir, não exclusivamente, aeroportos e áreas públicas, por exemplo. Conhecendo a existência de lugares propícios à formação de multidões, governos, gestores, pesquisadores, designers e outros profissionais estão interessados no desenvolvimento de diferentes tecnologias para melhorar a evolução desses locais, de forma mais inteligente e, principalmente, mais segura.

A área de *Crowd Simulation* pode ser relacionada com entretenimento (jogos e filmes) e indústrias de segurança. Pensando em entretenimento, pode-se facilmente aplicar a simulação de multidões para povoar cenas de um jogo ou filme com multidões maiores, mais realistas e dinâmicas. Por outro lado, na engenharia de segurança, observa-se alguns problemas abertos de pesquisa, principalmente no que tange à simulação de emergência e às rotas de evacuação. A possibilidade de avaliar a segurança das rotas/refúgios das populações em situações de emergência é certamente relevante no planejamento de eventos que possam conter aglomerações de pessoas (por exemplo Jogos Olímpicos em estádios, estações de trem, etc.). Essa compreensão permite aos engenheiros projetar melhores lugares e também descobrir a melhor maneira de orientar as pessoas ao escolher uma rota de evacuação.

Diferentes abordagens têm sido propostas na literatura motivando o desenvolvimento de diferentes modelos científicos ao longo das últimas décadas. Tais abordagens dizem respeito a simular computacionalmente o comportamento de movimento de pessoas, grupos e até mesmo multidões. Elas foram projetadas com base em diferentes objetivos e níveis de complexidade. O primeiro modelo conhecido é um sistema baseado em regras locais capaz de simular o comportamento de rebanhos e cardumes. Hoje em dia, técnicas complexas têm sido propostas que vão desde *navigation fields* [6] a variados algoritmos de *steering behaviors* [7, 8], aplicados quando se simulam multidões para obter resultados coerentes com a realidade.

Conforme observado por Thalmann e Musse [9], o movimento agregado é belo e complexo de se contemplar. Belo devido à sincronização, homogeneidade e unidade descritas neste tipo de movimento, e complexo porque há muitos parâmetros a serem manejados para fornecer essas características. Existem muitas características que têm sido utilizadas na literatura científica. De acordo com Fruin [10], o comportamento da multidão é afetado pela percepção espacial de cada indivíduo considerando seu próprio conhecimento e inteligência. Ao conhecer o ambiente, o indivíduo pode tomar uma decisão baseada também em padrões sociais e culturais. Quando uma pessoa específica decide se mudar, ela também afetará o quanto as pessoas próximas podem permanecer umas com as outras e influenciar o espaço pessoal dos outros. De acordo com o antropólogo americano Edward Hall, cada pessoa tem um espaço pessoal ao redor do corpo [11]. Hall denomina esse espaço como *proxemics* e seu tamanho pode ser variável com base no tipo de interação e relacionamento das pessoas envolvidas.

Além dessas características, a distância e o tipo de relacionamento entre pessoas e multidões também podem ser afetados pelas características individuais. A individualidade pode ser representada por muitos fatores como sexo, idade de cada indivíduo e seu estado físico. Um aspecto importante nos comportamentos de multidões é como tais fatores de heterogeneidade podem influenciar a evolução/simulação de multidões.

Um tipo específico de literatura de multidões é focado em processos de evacuação em situações de emergência. Ao simular multidões, um conjunto de parâmetros pode ser considerado para reproduzir comportamentos coerentes. Tais parâmetros visam representar:

- *Estrutura física do ambiente*: devem fornecer informações sobre as características

do edifício como dimensões, número de pisos, número de peças, número e localização das saídas, localização das escadas e etc;

- *Contexto/Funcionalidade do local*: as pessoas podem agir de maneiras diferentes de acordo com a funcionalidade do local, por exemplo escritórios, hospitais, escolas, aeroportos, estádios e arenas;
- *Dados populacionais*: número de pessoas no ambiente, idade, sexo, relação entre elas, conhecimento que a população tem sobre o meio ambiente; e
- *Condição do ambiente durante o evento*: Muitos fatores podem ocorrer em um ambiente específico que afeta suas condições. Tais fatores podem incluir a hora do dia (dia ou noite), fumaça, fogo, calor e etc.

Os fatores anteriormente apresentados são apenas um pequeno conjunto de aspectos que podem impactar em um processo de evacuação. A variedade de comportamentos de pessoas torna complexa e desafiadora a reprodução e a simulação virtual de um processo de evacuação.

Em alguns países, os departamentos de segurança costumam especificar a necessidade das empresas desenvolverem uma política de segurança para lidar com situações de emergência [12]. Entre as exigências, tais políticas devem incluir procedimentos de fuga de emergência e atribuições de rota, tais como planos de fuga, mapas de locais de trabalho e áreas seguras ou de refúgio. Estes fatores são importantes, pois uma evacuação desorganizada pode resultar em confusão, ferimentos e danos à propriedade. Em qualquer cenário de emergência, a determinação de um plano de evacuação ótimo ou quase ideal é atualmente um problema aberto. Esta questão está relacionada com a identificação das melhores rotas (em diferentes aspectos, como: conforto, tempo percorrido, tempo total, etc) para uma população específica ao sair de um edifício. Nesse contexto, acredita-se que o emprego de simulação de multidões é uma ferramenta poderosa para determinar diferentes maneiras de um grupo específico de pessoas deixar um certo ambiente.

Este capítulo tem por objetivo discutir questões de simulação de multidões, tanto sob a perspectiva de comportamentos humanos, como de simulação computacional. A organização é assim definida: na Seção 1.2 são apresentados alguns conceitos e classificações de multidões, utilizadas na literatura. Na Seção 1.3 discute-se conceitos relevantes na movimentação de multidões, enquanto na Seção 1.4 considera-se especificamente os casos de evacuação e emergência. Ainda, a Seção 1.5 descreve alguns detalhes sobre os regulamentos nesta área. As Seções 1.6, 1.7 e 1.8 descrevem o estado-da-arte em simulação de multidões, exemplos de tecnologias usadas para este fim e apresentam o CrowdSim, desenvolvido pelos autores, respectivamente. Finalmente, as Seções 1.9 e 1.10 apresentam estudos de caso onde o CrowdSim foi aplicado e considerações finais deste capítulo.

1.2. Conceitos e Classificações

Esta seção apresenta os principais conceitos relacionados à simulação de multidões. Em primeiro lugar, é interessante definir alguns aspectos relacionados com as multidões. Sabemos, como definido anteriormente, que a multidão representa um grande grupo de in-

divíduos no mesmo ambiente. Apesar disso, sua formação pode ocorrer de forma voluntária ou não-voluntária, tanto em situações diárias como em casos específicos (pânico ou emergência). Este é um conceito simples, mas muito importante; por exemplo, as pessoas voluntariamente aceitam ser parte de uma multidão formada pelo público de um festival de música. Por outro lado, uma multidão emerge durante o processo de evacuação necessário em eventos de emergência, como acidentes ou incêndios.

A fim de identificar um grande grupo de pessoas como uma multidão, alguns critérios devem ser observados. Challenger [13] destaca alguns deles:

- *Tamanho*: deve haver uma reunião mensurável de pessoas;
- *Densidade*: os membros da multidão devem ser co-localizados em uma área particular, com uma distribuição de densidade suficiente (não pequena);
- *Tempo*: Indivíduos normalmente devem se reunir em um local específico para uma finalidade específica durante uma certa quantidade de tempo (que não pode ser instantânea);
- *Coletividade*: os membros da multidão devem compartilhar uma identidade social, objetivos e interesses comuns e agir de maneira coerente;
- *Coerência de comportamento*: os indivíduos devem ser capazes de agir de uma maneira socialmente coerente, apesar de se reunirem em uma situação ambígua ou desconhecida.

Como pode ser observado, não existem densidades, tamanho, tempo especificados. Nem mesmo a coerência de comportamentos é especificada em detalhes, pois no que tange às multidões, os aspectos podem ser variados. Além desta falta de definições, diferentes eventos e circunstâncias podem ser o palco para a formação de multidões. Alguns poucos pesquisadores têm trabalhado para categorizar os diferentes tipos de multidões. Não há um conceito de multidão típica, mas uma variedade de tipos de multidão já foram observados, cada um com suas próprias características e comportamentos. Berlonghi [14], em 1995, identificou cinco tipos diferentes de multidões.

- *Espectador*: Uma multidão que assiste a um evento;
- *Demonstrador*: Uma multidão, muitas vezes com um líder reconhecido, organizada por um motivo ou evento específico;
- *Densa*: Uma multidão densa (pessoas/m^2) em que o movimento de pessoas diminui rapidamente e às vezes pode parar. Devido à alta densidade de multidões, as pessoas podem ser arrastadas e comprimidas, resultando em lesões graves e mortes por sufocamento;
- *Violenta*: Uma multidão atacando, aterrorizando ou provocando, sem consideração pela lei ou pelos direitos de outras pessoas;

- *Escape*: Uma multidão que tenta escapar do perigo real ou percebido ou de situações que ameaçam a vida, incluindo pessoas envolvidas em evacuações organizadas, ou empurrões por parte de uma multidão em pânico .

Uma vez que são conhecidos os atributos que podem ser observados para caracterizar e também para entender diferentes tipos de multidões, a próxima seção detalha aspectos relativos à dinâmica de multidões. O objetivo é estudar conceitos de multidões, a fim de compreender principalmente como eles são susceptíveis de se mover e se comportar.

1.3. Dinâmica de Multidões

A observação da evolução da multidão em um lugar específico nos permite observar diferentes aspectos. Entre eles, é interessante destacar o fato de que uma multidão é composta por indivíduos independentes e cada um tem suas próprias necessidades e desejos, mas todos eles compartilham o mesmo objetivo. Esse sentimento é ressaltado por Osório [15], um psicólogo que define um grupo de pessoas como um sistema humano composto pelo conjunto de pessoas capazes de se conhecerem na própria singularidade. Além disso, essas pessoas estão compartilhando objetivos e realizando uma ação coletiva.

Sabemos que uma multidão é uma congregação de pessoas no mesmo ambiente, apesar de que é importante ter em mente que cada membro individual de uma multidão é dono de seu próprio espaço pessoal no ambiente. O antropólogo americano Edward Hall, baseado na idéia do espaço pessoal, apresentou em 1966 o conceito de *proxemics* [11] para representar o espaço pessoal de cada pessoa. O autor também explica que a distância entre as pessoas, quando interagindo umas com as outras, varia de acordo com seus níveis de intimidade. Estes níveis são divididos em quatro faixas possíveis (ver Fig. 1.1): *íntimo* $[0.00, 0.45]m$, *pessoal* $(0.45, 1.20]m$, *social* $(1.20, 3.60]m$ e *público* $(3.60, 7.60]m$.

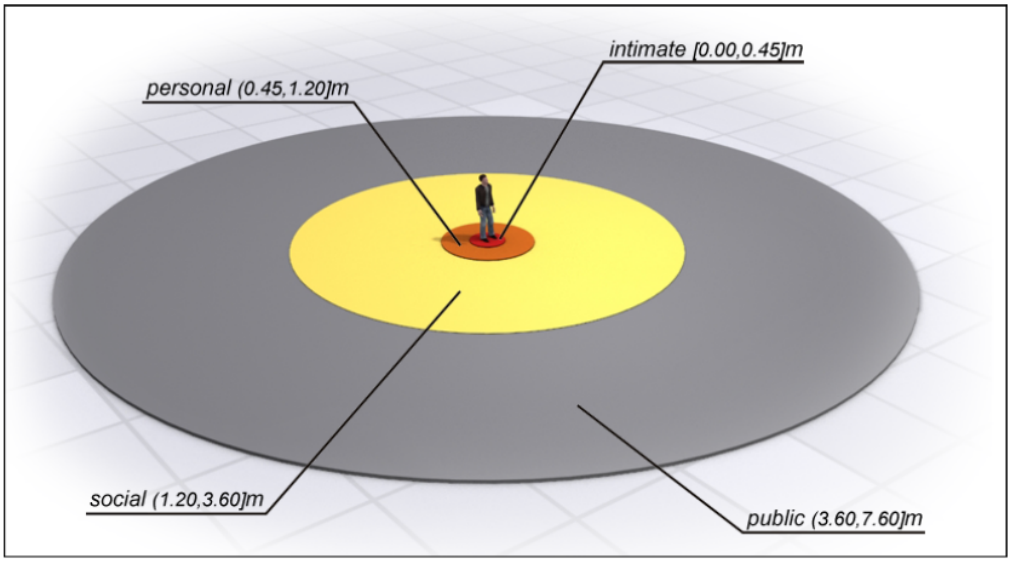

Figura 1.1. Ilustração do *proxemics* de um pessoa baseada nas definições de distâncias de Hall [11]: íntimo, pessoal, social e público.

Still e seus colaboradores [16] estudaram o comportamento das pessoas ao longo

das últimas décadas. Durante seu Ph.D., ele desenvolveu um conjunto de programas de computador para estudar, e também reproduzir, o comportamento de uma multidão. Seu framework chamado *Legion* foi desenvolvido com base em regras para determinar as funções para o fluxo de tráfego humano. Essas regras interagem à medida que os personagens se aproximam do espaço de cada um dos outros associados aos objetos estáticos e dinâmicos do ambiente. Apesar dos resultados da simulação, a pesquisa de Still também apontou comportamentos importantes que podem surgir de multidões reais e também foram observados por outros autores, bem como reconhecidos por instituições internacionais. A seguir resume-se alguns desses comportamentos, que são interessantes para serem reproduzidos quando se trabalha com simulação de multidões:

- *Formação de arcos (Arch formation)*: Acontece quando uma multidão grande e densa empurra para a frente em direção a uma saída estreita. Em situações como esta, obstrução e arqueamento são observados; *i.e.* a saída torna-se obstruída e a multidão forma um arco na frente da saída;

- *Formação de vias (Lanes formation)*: Quando as pessoas se movem na mesma direção ou em direções opostas, elas podem se auto-organizar para criar pistas distintas: uma para cada direção de movimento ou levando em consideração as diferentes velocidades. Este fenômeno de auto-organização ajuda a reduzir as colisões e aumentar a velocidade de movimento. No entanto, em multidões de alta densidade ou nervosas, quaisquer pistas formadas podem quebrar devido a manobras de ultrapassagem contínuas;

- *Efeito dos cantos (Corner effect)*: Quando os membros da multidão localizam-se nos cantos, eles tendem a desacelerar, tornando estes ambientes mais densos, e por consequência perigosos;

- *Efeito do anel (Ring effect)*: Este fenômeno emerge quando uma multidão está observando um evento particular em torno de um ponto particular do interesse, tal como um artista da rua. Em casos como este, uma estrutura de anel emerge, irradiando para fora do ponto de interesse;

- *Efeito de redução de velocidade (Speed reduction)*: esse efeito surge quando um ambiente específico já encontra-se bastante denso e ainda chegam mais pessoas. Neste caso, a velocidade pode ser diminuída, até a ausência de movimento; e

- *Princípio do Menor Esforço (Principle of Least Effort)*: Quando possível, os membros da multidão tentarão adotar o caminho mais rápido. O objetivo é minimizar o tempo, evitar o congestionamento e maximizar suas velocidades.

1.4. Multidões em Situações de Emergência

Como muito bem observado por Lebon [2], quando em multidões, as pessoas podem realizar um comportamento incomum que pode ser responsável por ações irracionais. Na verdade, pode-se dizer que tais ações podem até ser aumentadas em casos de pânico. É fácil olhar para a história, a fim de apontar alguns fatos importantes que exigiram das

pessoas comportamentos de evacuação. Por exemplo: Bombardeamento atômico do Japão durante a Segunda Guerra Mundial, ação dos terroristas em 2001 em Nova Iorque e o Furacão Katrina em 2005. Estes são apenas três exemplos de muitos que infelizmente já aconteceram. Em tais evacuações as pessoas provavelmente foram dirigidas por suas emoções. Elas podem ter sido treinadas ou não, mas inevitavelmente também responderam a comportamentos irracionais.

O sucesso de um evento de evacuação pode estar relacionado com o bom entendimento do processo. Para alcançar o sucesso pode-se apontar três fatores importantes: interpretação, preparação e ação [17]. Podemos entender *interpretação* como o momento em que as pessoas observam e chegam a conclusão da necessidade real de uma evacuação; *preparação* é sobre planejar a melhor rota a seguir, enquanto a *ação* significa a movimentação pela rota escolhida, para alcançar uma área segura. Além disso, o entendimento de outros aspectos específicos é amplamente necessário. Tais pontos são relativos a:

- Tipo de imóvel: escritório, aeroporto, estação ferroviária;
- A compreensão do comportamento dos ocupantes em situação de pânico;
- Distribuição dos ocupantes (inclui idade, sexo e deficiência); e
- Localização de áreas perigosas, bem como locais seguros e saídas de emergência.

O conhecimento dos aspectos anteriores facilita a execução de um processo de saída segura. Hoje em dia, a evolução de uma multidão durante um processo de evacuação geralmente mostra comportamentos que podem evidenciar a boa organização e estrutura do grupo. Isto é devido à cooperação e coordenação realizada pelas pessoas [18].

Geralmente, o grupo tem o poder de influenciar o movimento dos seus membros. Em outras palavras, a escolha da rota de escape, *feita individualmente*, geralmente também é influenciada pelas ações dos outros membros da multidão. Este aspecto pode justificar o fato de que em situações de emergência, as pessoas tendem a se mover na mesma direção que os outros. Desta forma, para Cocking & Drury [18], quando unida por uma situação de emergência, uma multidão física (um grupo de Indivíduos no mesmo local, cada um com sua própria identidade pessoal) pode ser transformada em uma multidão psicológica (ou seja, um grupo de pessoas unidas por uma identidade social comum como membros de uma categoria particular). Além disso, um conjunto de fatores extras (mapeados por [13]) podem influenciar um processo de evacuação:

- *Mobilidade*: um indivíduo que é menos móvel provavelmente precisará de mais tempo para evacuar de um ambiente em uma emergência;
- *Posição Física*: um indivíduo deitado provavelmente terá uma taxa mais lenta de reação e movimento do que um indivíduo em pé;
- *Densidade*: o movimento da multidão será mais lento em um ambiente mais denso;
- *Alerta*: um indivíduo que está menos alerta, por exemplo como resultado de cansaço ou intoxicação, provavelmente reajirá mais lentamente no caso de uma emergência;

- *Visibilidade*: quanto mais visíveis forem as rotas de sinalização e de saída de emergência, mais atraentes estas serão para os membros da multidão;
- *Complexidade do ambiente*: quanto mais complexo for o ambiente, mais indeciso poderá ser o indivíduo, e mais tempo levará para sair.

Estes pontos são importantes e geralmente são considerados a fim de regular o processo de saída de pessoas de ambientes. Grupos governamentais e diferentes organizações em alguns países têm trabalhado para regular o processo de evacuação. A próxima seção apresenta e detalha alguns desses regulamentos.

1.5. Regulamentos para Processo de Evacuação

Atualmente, governos de alguns países e profissionais de diferentes áreas trabalham para especificar medidas efetivas com o objetivo de definir planos de evacuação ótimos. A definição precisa de um plano de evacuação é visada para garantir a segurança das pessoas quando saem de um edifício, especialmente quando ocorre um pânico ou uma situação de atenção. A avaliação dos detalhes de evacuação já é solicitada por organizações internacionais como a UEFA (União da Associação Européia de Futebol) ou a FIFA (Federação Internacional de Futebol) em locais específicos como locais desportivos, estádios e Arenas, além de regulamentos regionais específicos.

A UEFA expressa algumas preocupações relacionadas ao processo de evacuação e que devem ser observadas desde o início do *design* de um novo estádio. A capacidade de segurança é um requisito obrigatório que se concentra, como o nome sugere, em garantir a máxima segurança para os espectadores. É amplamente aceito que todos os espectadores devem ser capazes de sair do estádio para um ponto de segurança dentro de no máximo oito minutos. Este valor, baseado no fluxo máximo observado em estádios, indica 660 pessoas por hora. No entanto, pode haver algum espaço para variações específicas com base no tamanho e design do local.

De acordo com a FIFA, o tempo de evacuação de emergência é em parte baseado no nível de risco e nas vias de evacuação de emergência disponíveis para refúgios/locais de segurança. A organização publicou uma diretriz [1] onde define um conjunto de regulamentos de segurança e segurança do estádio. De acordo com esse guia, fatores como o tipo de construção e materiais utilizados no estádio terão impacto no cálculo do tempo sugerido para a evacuação. Além disso, o incêndio é um dos principais riscos a serem considerados ao calcular o tempo de saída aceitável. Por exemplo, se o risco de incêndio for alto devido à construção do estádio, o tempo de evacuação esperado deve ser reduzido.

O tempo de evacuação de emergência não é um valor fixo. Trata-se de um cálculo que, juntamente com a taxa de passagem apropriada, é utilizado para determinar a capacidade do sistema de saída de emergência para um local seguro.

Os Estados Unidos da América utilizam o *Código de Segurança de Vida* [19], uma diretriz desenvolvida pela NFPA (Associação Nacional de Proteção Contra Incêndios), que fornece detalhes a serem seguidos durante um possível processo de evacuação

[1] `fifa.com/mm/document/tournament/competition/51/53/98/safetyregulations_e.pdf`

de edifícios. Juntamente com o Manual de Design fornecido pela SFPE (Sociedade de Proteção contra Incêndio e Engenharia), os designers de edifícios podem observar, durante a fase de projeto de um edifício, aspectos de saída em relação a características como luzes de saída de emergência e alarmes.

De acordo com o Código de Construção Internacional, todos os edifícios, novos ou antigos, concebidos para ocupação humana, devem dispor de saídas suficientes para permitir a pronta saída dos ocupantes em caso de incêndio ou outra emergência. De acordo com a diretriz, todas as saídas devem descarregar diretamente para a rua ou outro espaço aberto que dê acesso seguro a uma via pública. As ruas, às quais as saídas descarregam, deverão ter largura adequada para acomodar todas as pessoas que saem do edifício. Um EAP (Plano de Ação de Emergência), de acordo com a diretriz, deve cobrir as ações designadas que garantem a segurança dos funcionários de incêndio e outras emergências:

- Procedimentos de fuga de emergência e atribuições de rota de fuga;
- Os procedimentos para contabilizar todos os indivíduos após a evacuação de emergência;
- A forma preferida de reportar incêndios e outras emergências; e
- Nomes de pessoas ou departamentos que podem ser contatados para obter mais informações ou explicações sobre as funções do plano.

Independentemente da diretriz, é importante mencionar que o comportamento das pessoas pode contribuir para congestionamentos que não são previstos durante o desenvolvimento do plano de saída. E esses congestionamentos são a principal causa de acidentes que resultam em feridos e/ou mortos.

A legislação sobre gestão de multidões no Brasil está em desenvolvimento. A atenção na área tem aumentado ao longo dos últimos anos. Tal crescimento é explicado devido a alguns eventos específicos. Em primeiro lugar, um triste desastre que ocorreu em uma boate: Kiss Night Club [2]. Além disso, grandes eventos esportivos que aconteceram no país (Copa do Mundo e Olimpíadas) contribuiram para determinar nova legislação.

A ABNT (Associação Brasileira de Normas Técnicas) definiu, em 2001, a diretriz NBR 9077: 2001 com o objetivo de especificar a regulamentação concernentes às saídas de emergência. De acordo com esta diretriz, a concentração de pessoas, ao usar a saída de emergência, deve ser de até 2 pessoas $/m^2$. A Diretriz Técnica do Governo de São Paulo - Brasil, define pontos específicos relativos à segurança das pessoas quando em saída. Resumimos alguns dos mais importantes:

- Os indivíduos devem atingir um ponto seguro sem caminhar mais de 20 metros em áreas ao ar livre e 10 metros em áreas internas. As saídas de emergência podem ser consideradas como o ponto seguro;
- O tempo para um grupo de pessoas deixar uma área interna pública, como um teatro, não deve exceder 6 minutos; e

[2] `edition.cnn.com/2013/01/28/world/americas/brazil-nightclub-fire.`

- A concentração de pessoas não deve ser superior a 4 pessoas$/m2$.

Optamos por apresentar dados de São Paulo, por esta ser a cidade mais populosa do Brasil. É importante mencionar que cada estado ou cidade pode determinar seu próprio regulamento de construção, se não houver regulamentos federais.

1.6. Estado-da-arte em Simulação de Multidões

A simulação de multidões tem sido estudada no contexto de ciência da computação, nas últimas décadas [9]. Hoje em dia pode-se observar um grande número de pesquisadores trabalhando para fornecer simulações coerentes e realistas aplicadas em áreas como engenharia, sistemas de segurança e animação digital. A fim de prever o comportamento das pessoas, os modelos computacionais são solicitados a produzir resultados de forma realista, quando comparados com o comportamento das pessoas no mundo real.

Os trabalhos mencionados nesta seção visam dar uma visão geral dos métodos recentes centrados na simulação de multidões e de evacuação, planos de evacuação e a possibilidade de lidar com agentes heterogêneos. Primeiramente apresenta-se uma breve classificação dos modelos de simulação em macroscópicos e microscópicos, de acordo com Hamacher & Stevanus [20]:

- *Modelos macroscópicos* são usados principalmente para produzir bons limites inferiores para o tempo global de evacuação e não consideram qualquer comportamento individual durante a situação de emergência. Os resultados da simulação podem ser utilizados para a fase de concepção do planejamento de uma construção; e

- *modelos microscópicos* são capazes de modelar as características individuais e a interação entre as pessoas, que influenciam seu movimento. Algumas leis probabilísticas para o movimento de indivíduos são às vezes empregadas neste conjunto de modelos.

A simulação de multidões é um grande campo de pesquisa na atualidade e motivou o desenvolvimento de várias outras abordagens. Diversas abordagens macroscópicas e microscópicas têm sido propostas para alcançar a reprodução do comportamento das pessoas com algum realismo. Além disso, também pode-se observar algumas abordagens que combinam a eficiência da Modelagem Macro (modelagem baseada em equações) e as vantagens da Micro Modelagem (modelagem baseada em agentes) [21]. Hoje em dia, existem ferramentas capazes de simular um único edifício, bem como uma comunidade inteira povoada por grandes multidões [22, 23]. Além da área de segurança, simulações de multidão podem ser objetos de estudo em outras áreas como o entretenimento. Podem ser aplicados algoritmos robustos para animar, controlar e criar agentes humanos com seu próprio conjunto de capacidades, personalidades e desejos únicos [24].

Especificamente em simulação de situações de emergência, um trabalho pioneiro com relação à reprodução do comportamento da multidão foi proposto por Braun *et al.* [25]. Os autores exploraram as características pessoais dos agentes para simular diferentes reações e comportamentos durante um processo de evacuação. Inspirados em uma

abordagem baseada em Física [26] os autores agregaram um conjunto de recursos para simular agentes e também grupos a fim de reproduzir uma multidão heterogênea. Tais características incluem, entre outros, aspectos como representação de famílias, nível de dependência de outros indivíduos, nível de altruísmo de agentes e também velocidades desejadas. Os autores propuseram um método baseado em forças, compostas do nível de altruísmo e dependência dos agentes de uma mesma família, para manter a coesão dos grupos (grupos de agentes que se movem juntos). Além disso, com base no nível de altruísmo, um agente pode se desagrupar de sua família, a fim de ajudar outros agentes no processo.

O trabalho de Zhu *et al.* [27] foi desenvolvido observando os Jogos Olímpicos de 2008, na China. Nessa época, os autores desenvolveram uma abordagem capaz de reproduzir o tráfego pessoal criado a partir de delegações de atletas de diferentes países e também o público. Um estudo de caso foi realizado considerando o *National Stadium* e considerou aspectos como o número de pedestres e a sua distribuição em situações específicas.

Simular o processo usual de evacuação foi o principal objetivo de Fu *et al.* [28]. A motivação dos autores foi reproduzir o comportamento dos pedestres para representar a seleção da melhor saída levando em consideração um algoritmo de autômato celular de menor esforço. É representado por um conjunto de células 2D onde podemos encontrar pedestres ou obstáculos. Os movimentos e objetivos utilizados para guiar os movimentos são definidos considerando uma abordagem probabilística. O uso de algoritmos de autômatos celulares também está presente nos trabalhos de Ji *et al.* [29], a fim de simular a dinâmica dos pedestres, e de Aik e Choon [30] com o objetivo principal de reproduzir o processo de evacuação. Chu *et al.* [31] desenvolveram a plataforma SAFEgress (Social Agent For Egress) em que os ocupantes dos edifícios são modelados como agentes que conhecem o meio ambiente e suas interações com os grupos sociais na multidão. De acordo com os autores, os resultados mostram que ambas as entidades, agentes e ambiente, podem impactar significativamente o desempenho da evacuação.

Pelechano e colaboradores [32] exploraram diferentes aspectos relativos ao comportamento de multidões virtuais. Um aspecto estudado pelos autores visa melhorar os sistemas de simulação de multidões pela adição de um modelo psicológico [33]. Desta forma, os autores apresentaram um modelo (MACES) [34] que combina modelos de fisiologia, estresse, percepção e emoção com a comunicação de agentes em situações de evacuação. A integração permitiu que o modelo de simulação de multidões gerasse eventos que agentes podem perceber, resultando em comportamentos responsivos, reativos e contextualizados. O objetivo do trabalho de Pelechano & Malkawi [35] foi estudar a importância de incorporar fatores psicológicos e fisiológicos nos modelos de simulação de multidões. Os autores apresentaram uma visão geral dos fundamentos que devem ser usados ao simular o movimento humano mais próximo dos movimentos reais de pessoas, onde a interação entre os humanos é emergente e as taxas de fluxo, densidades e velocidades se tornam o resultado dessas interações, em vez de terem algum valor pré-definido.

1.7. Simuladores de Multidão

Os modelos de evacuação podem ser usados para extrair outros dados além do simples tempo total de evacuação. Como discutido anteriormente, o modelo deve fornecer informações capazes de extrair dados a serem analisados para avaliar a segurança, mas também em situações de conforto. A fim de produzir resultados realistas e também prever comportamentos de multidões com o objetivo de alertar os gerentes de eventos, alguns softwares comerciais são atualmente usados por equipes de gerenciamento de multidões em diferentes países. Nesta seção, apresentamos 3 importantes softwares nesta área: MassMotion, VStep e Legion.

1.7.1. MassMotion

MassMotion, desenvolvido pela Oasys Software [3], é projetado para a criação e execução de simulações com grande quantidade de personagens. O software começou como um simulador de movimento de pedestres e evoluiu para um sistema específico de evacuação em ambientes. O simulador opera em um ambiente 3D, onde cada agente individual é informado no ambiente, representado através de mapa de bits de espaço livre e obstruído em toda a área que pode ser percorrida. Cada agente determina seu melhor local de destino disponível para o próximo quadro da simulação e ajusta sua velocidade e orientação para alcançar essa posição. Este cálculo é executado a uma taxa de cinco quadros por segundo de tempo simulado que é normalmente suficiente para permitir que os agentes se adaptem às condições de mudança dinâmica dentro do ambiente.

1.7.2. *Crowd Control Trainer*

Crowd Control Trainer é um simulador que visa treinar pessoas que trabalham com incidentes relacionados com a multidões e eventos de massa. O software foi desenvolvido pela VSTEP [4], uma empresa certificada pelo ISO9001:2008 [5], em cooperação com o departamento de Polícia de Rotterdam e do Governo Holandês. Esta parceria foi firmada para apoiar seus comandantes de polícia e gerentes de treinamento de controle de multidões. Além disso, o software foi selecionado pelo governo dos Países Baixos como um dos melhores no campo de segurança. A simulação ocorre em uma réplica 3D virtual do ambiente urbano real permitindo o reconhecimento instantâneo e planejamento realista de estratégias de gestão e respostas reais para manifestações e motins. Além disso, o software inclui algoritmos de movimento e inteligência artificial, a fim de calcular o movimento de multidões de qualquer tamanho através do ambiente de treinamento virtual.

1.7.3. Legion

Os produtos de simulação de pedestres da *Legion*[6], desenvolvidos no Reino Unido, incluem um conjunto de ferramentas capazes de lidar com a simulação de comportamentos de pedestres. *Legion-Evac* é uma ferramenta de simulação baseada em agentes, onde cada agente pode se movimentar pelo ambiente, de uma origem para um destino. Nesta trajetória, os agentes interagem com outros agentes e realizam atividades. É importante

[3] oasyssoftware.com
[4] vstepsimulation.com.
[5] iso.org/iso/catalogue_detail?csnumber=46486
[6] legion.com

mencionar que os agentes movem-se através do ambiente de acordo com o princípio de menor esforço. Além disso, elementos aleatórios de comportamento podem ser introduzidos para fazer a simulação mais realista, por exemplo, tamanho da entidade, velocidade, idade e bagagem carregada.

O Instituto Nacional de Padrões e Tecnologia, *Technical Note 1680* [36, 37], provê uma lista padronizada de recursos para alguns dos modelos de evacuação mais proeminentes no mercado. A tabela 1.1 reproduz parte desta revisão para os softwares de simulação de multidão descritos anteriormente: *MassMotion, VStep e Legion*.

	MassMotion	**VStep**	**Legion**
Metodologia Empregada	Comportamental	Comportamental	Comportamental
Construção	Sem restrição	Sem restrição	Sem restrição
Perspectiva do ocupante /	Individual and /Global	Individual	Individual
Modelo de comportamento /	Inteligência Artificial / Probabilístico	Condicional / Probabilístico	Intelligência Artificial / Probabilístico
Movimento	Condicional	Distâncias Inter-pessoais Busca por células vazias	Distâncias Inter-pessoais Condicional /
Definição de rotas	Condicional	Condicional	Condicional
Validação	Simulações / Literature / Outros Modelos	Simulações / Literature / Outros Modelos	Simulações / Literature / Outros Modelos / Avaliação com experts

Tab. 1.1. Análise de características de sistemas de simulação de multidões, adaptado de [36].

1.8. CrowdSim - Desenvolvido no VHLAB@PUCRS

Nas próximas seções descrevemos *CrowdSim*, um software de simulação de multidões desenvolvido para reproduzir o comportamento das pessoas em diferentes situações (desde o movimento normal até grandes casos de evacuação). Ao projetar *CrowdSim*, definimos alguns objetivos principais a serem alcançados no software: *i)* Deve prover uma estrutura fácil de ser usada e controlada; *ii)* os planos de evacuação devem ser parametrizados; *iii)* deve permitir a exportação de vários tipos de dados resultantes, que podem ser usados posteriormente em análises/avaliações; e *iv)* deve lidar com apenas um agente bem como grandes multidões.

Primeiramente, a fim de esclarecer a terminologia usada neste trabalho, seguem as definições de alguns conceitos:

- grafo de navegação/grafo do ambiente: está relacionado ao ambiente específico a ser simulado. Considera salas como nós e portas e passagens como arestas de um grafo. Quando o ambiente simulado é configurado (explicado mais adiante neste capítulo) este grafo é gerado automaticamente;

- plano de evacuação: é um gráfico de evacuação que contém a distribuição da população (número de pessoas) nas regiões onde os agentes devem ser criados e a porcentagem de distribuição em cada bifurcação (pontos de decisão no grafo) até chegarem nas saídas.

CrowdSim é um software de simulação de multidões baseado em regras desenvolvido para simular movimentos coerentes e comportamentos de humanos virtuais em um processo de evacuação [38, 39]. Ele também apresenta dados para análise, que são usados para estimar conforto e segurança dos humanos em simulações em um ambiente específico. Durante a fase de projeto do CrowdSim, procurou-se desenvolver um protótipo capaz de:

- Representar a geometria física de uma construção em um ambiente 3D. Essa representação permite que os engenheiros de segurança usem o software para simular praticamente um plano de ocupação ou evacuação atendendo as restrições físicas reais do ambiente (portas, saídas de emergência, tamanho dos corredores, etc);
- Definir a ocupação espacial da população no ambiente para reproduzir as condições iniciais de um evento de abandono;
- Modelar um plano de saída no contexto de situações de emergência;
- Produzir uma visualização da simulação e gerar os dados de saída, a serem considerados nas análises estatísticas.

Existem dois componentes-chave na estrutura do CrowdSim, organizados em módulos distintos: *Configuration* e *Simulation*. A Figura 1.2 ilustra a arquitetura do software incluindo sub-módulos, as entradas necessárias bem como as saídas produzidas.

Nas seções a seguir descreve-se os módulos de CrowdSim detalhando suas entradas, dependências e fluxo de dados.

1.8.1. Módulo de Configuração

O módulo de configuração solicita, como primeira entrada, a representação 3D do ambiente que será simulado. Esse modelo 3D será utilizado pelo *Environment Manager* para permitir que o usuário defina as regiões passíveis de movimento de acordo com a estrutura do prédio, bem como restrições físicas e obstáculos. Mais especificamente, os componentes de geometria podem ser detalhados como se segue:

- Contextos: São regiões (quadriláteros convexos ou não convexos) nos quais os agentes podem ser criados (início da simulação), mover-se (durante a simulação) ou serem removidos (ao chegarem ao objetivo);
- Portas: Bordas que conectam dois contextos e permitem que os agentes se movam entre eles;

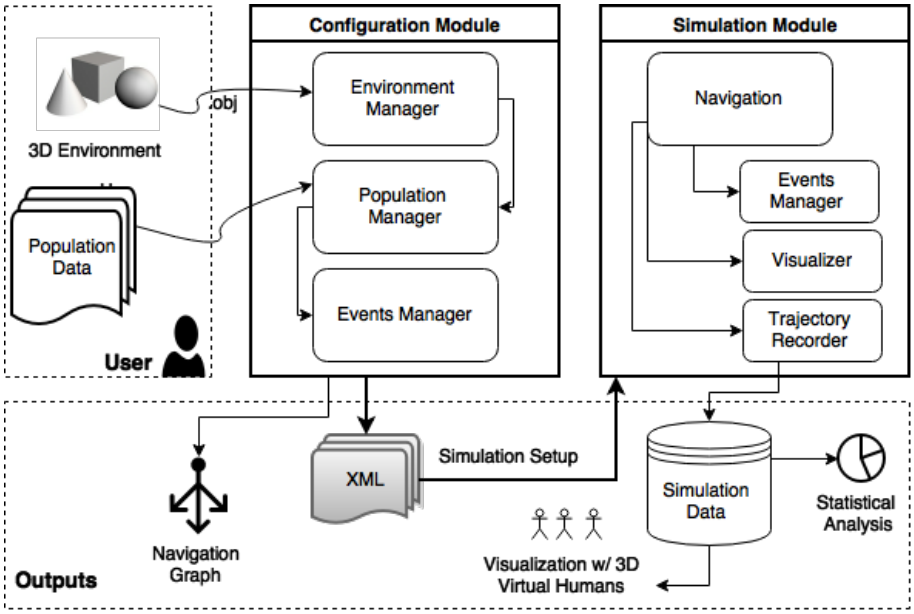

Figura 1.2. Arquitetura do protótipo *CrowdSim*.

- Escadas e rampas: Regiões (quadriláteros convexos ou não convexos) que podem conectar portas de salas diferentes. Os agentes não podem ser criados ou removidos nessas regiões; e
- Obstáculos: Os obstáculos são definidos em salas para restringir o movimento de agentes dentro do ambiente físico.

Para definir o cenário que será simulado, o Gerenciador de Ambiente classifica as regiões transitáveis com diferentes propósitos. Tais áreas transitáveis são chamadas de contextos. Definimos três tipos diferentes de contextos ao especificar um ambiente de simulação: contextos de *birth*, *motion* e *goal*.

Birth Contexts são usados para representar áreas da construção onde os agentes devem ser criados durante a simulação. Nessas áreas, o usuário é solicitado a fornecer o número de agentes a serem simulados que devem ser criados em tal contexto. Além disso, o usuário define as seguintes informações com base no número total de agentes a serem criados:

- *Tamanho dos grupos*: Os agentes podem ser criados em grupos diferentes até que alcancem o número total que deve ser criado no contexto;
- *Tempo de Criação*: Tempo em que grupos de agentes começam a ser criados após o início da simulação;
- *Tempo entre grupos*: Intervalo de tempo a ser considerado ao criar grupos diferentes; e
- *Objetivo*: O contexto (ou conjunto de contextos possíveis) a ser considerado como objetivos a serem alcançados por um agente ao se mover.

Goal Contexts são regiões de interesse a serem consideradas durante o movimento do agente (objetivos). Ao criar um contexto deste tipo, o usuário é solicitado a definir a porcentagem de agentes que devem ser removidos da simulação ao atingir o contexto, a porcentagem de agentes que devem permanecer em movimento nesse contexto e a porcentagem de agentes que devem encontrar outro objetivo e mover-se nessa nova direção.

Os *Motion Contexts* são considerados pelo algoritmo de simulação como regiões de conexão entre contextos de nascimento e objetivo. Eles são importantes nos cálculos das rotas de movimento dos agentes. Além disso, um grafo de conexão é construído como uma saída do módulo de configuração de acordo com conexões entre contextos e suas especificações populacionais. Tais contextos nos permitem reproduzir um ambiente virtual simples, como ilustrado à esquerda, na Fig. 1.3. Este é apenas um exemplo simples para ilustrar uma configuração de ambiente no CrowdSim. Além disso, este ambiente simples nos permite representar facilmente um grafo ambiental calculado por *CrowdSim* como ilustrado à direita, na Figura 1.3.

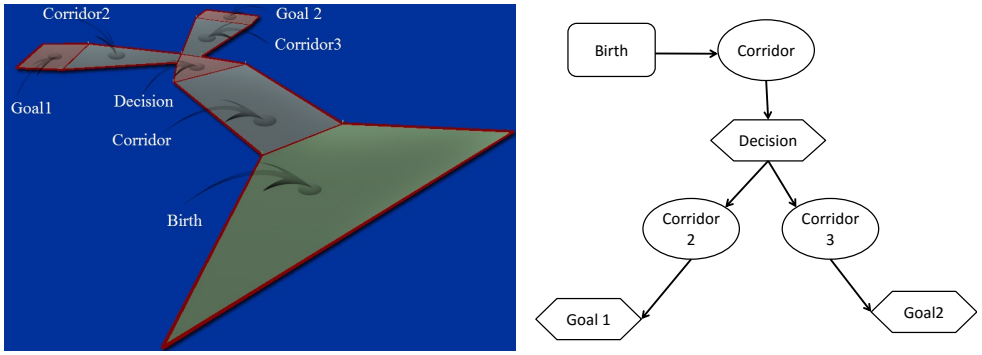

Figura 1.3. Ambiente de simulação simples composto por diferentes tipos de contextos (à esquerda) e respectivo grafo de navegação calculado por CrowdSim (à direita). Diferentes formas no gráfico significam contextos com finalidades diferentes.

Uma vez que o ambiente está mapeado de forma coerente e o usuário definiu todos os dados de população, é possível especificar no sub-módulo *Population Manager* como os agentes devem se comportar ao se mover. Os comportamentos possíveis dos agentes podem ser:

- *Goal Seeking*: Os agentes devem procurar seus objetivos imediatamente ou vagamente, por realização de movimento aleatório;
- *Keep waiting*: Os agentes, quando alcançam alguma região específica do ambiente, podem passar algum tempo nele antes de procurar outro objetivo;
- *Random movement*: os agentes podem escolher destinos aleatórios durante um tempo específico, antes de tentar identificar o melhor caminho para atingir o objetivo principal.

A definição correta dos cenários é crítica neste trabalho, pois a combinação e análise de informações é responsável pela produção de resultados aceitáveis e válidos.

Quando o ambiente é corretamente definido, com todas as regiões definidas, todos os parâmetros configurados e comportamentos desejados especificados, o usuário é capaz de executar o segundo módulo do CrowdSim: Simulação. A transferência de dados entre os módulos de configuração e simulação é realizada atualmente por um arquivo de cenário (XML), capaz de armazenar todas as configurações a serem observadas ao calcular uma nova simulação. Além do arquivo XML, o Módulo de Configuração também gera um gráfico de navegação com a população distribuída inicial nos nós do grafo. Além disso, foi desenvolvido um planejador, executado off-line, para ler o grafo e gerar planos de evacuação diferentes. A principal diferença entre um plano de navegação e um plano de evacuação é que no primeiro define-se onde as pessoas estão no início da simulação e no último define-se a distribuição de pessoas em qualquer bifurcação de gráfico, ou seja, define-se as saídas para cada pessoa/grupo. É claro que adotamos a importante hipótese de que o caminho mais curto nem sempre é o melhor para a simulação de multidões.

1.8.2. Módulo de simulação

O módulo de simulação do CrowdSim é responsável pela navegação de agentes virtuais em um ambiente específico. Essa navegação deve levar em conta o movimento do agente, o controle de colisão, a variação de velocidade e outros comportamentos dos pedestres. Uma configuração de simulação, previamente definida no módulo de configuração, é solicitada como entrada para a simulação. A simulação calcula as rotas de cada agente para atingir uma meta específica, com base nos possíveis caminhos, como mencionado na última seção. As rotas podem ser calculadas com base na especificação do usuário (isto é, um gráfico determinado pelo usuário - um grafo de evacuação, por exemplo) ou calculada para executar o melhor caminho, considerando apenas critérios de distância. O CrowdSim utiliza o algoritmo A* [40] para calcular caminhos mais curtos sem colisões, se forem declarados obstáculos. Durante a simulação do movimento, CrowdSim executa algoritmos para que agentes evitem colisões com outros agentes, usando um método geométrico local simples.

O método para evitar colisões é baseado em regras e definido localmente (baseado na proximidade de distância). Agentes próximos e suas velocidades são usados no teste de colisão para detectar uma possível situação de colisão em um próximo quadro. Se essa situação vai acontecer, um dos agentes envolvidos (definidos aleatoriamente) deve tomar uma decisão baseado nas seguintes possibilidades: *i)* mudar seu vetor de direção (mudanças de 40 deg -positivo ou negativo- são permitidas), ou *ii)* reduzir sua velocidade (mudanças até velocidaden se tornar zero são permitidas). As informações sobre o par de agentes e a decisão tomada são salvas em uma lista de ações passadas, que é armazenada a cada segundo. Se uma nova situação de colisão for detectada para o mesmo par de agentes e ainda houver uma ação na lista de ações passadas, o agente tomará uma decisão diferente, ou seja, se a mudança de direção for salva em ações passadas, então uma mudança de velocidade deve ser realizada. Conseqüentemente, os agentes tentam alcançar seus objetivos, evitando colisões com outros. Este método não é livre de colisão, mas o erro máximo de 10% foi observado em todas as experiências realizadas com CrowdSim.

Cada simulação realizada gera um conjunto de dados de saída para posterior análise:

- Trajetórias do agente durante a simulação;
- variação de velocidade para cada agente;
- tempo de cada agente da simulação;
- tempo total de simulação, e
- densidade local por frame - calcula-se a densidade local contando o número de agentes por metro quadrado em cada contexto, em vez da densidade global (ou seja, número de pessoas divididas pela área de construção).

Os dados de saída são armazenados e podem ser usados para produzir diferentes análises estatísticas. As trajetórias dos agentes podem ser facilmente exportadas para serem visualizadas com humanos virtuais articulados em uma *engine* que fornece visualização realista.

1.9. Estudos de Caso

Nesta seção apresentaremos alguns casos de estudo, onde o CrowdSim foi aplicado.

1.9.1. Simulação no Estádio Olímpico João Havelange (Engenhão) - RJ

Neste estudo de caso, aplicamos o CrowdSim para estudar o processo de evacuação de um Estádio Olímpico. O "Estádio Olímpico João Havelange", denominado Estadio Nilton Santos desde 2010, foi construído para os Jogos Pan-americanos de 2007 e, de acordo com os gerentes do estádio, pode ser considerado o estádio mais moderno da América Latina América e número cinco no mundo [7]. O estádio, sede da equipe de futebol brasileira Botafogo [8], tem capacidade para 46.000 pessoas e foi sede dos Jogos Olímpicos em 2016. O projeto foi realizado como uma parceria entre a PUCRS e o Botafogo com o objetivo principal de fornecer um estudo sobre o processo de saída de público no estádio. Os fatores considerados nas análises representam informações de conforto das pessoas (densidade de pessoas) e também o tempo total de evacuação.

Uma representação do estádio é ilustrada na Figura 1.4. O estádio pode ser acessado por quatro áreas distintas, como apresentado na Fig. 1.4, à esquerda. A compreensão detalhada da estrutura do estádio, ou seja, portões e corredores, é muito importante quando se realiza um projeto de simulação de multidões. Isso ocorre porque a primeira fase do projeto está relacionada com a modelagem do ambiente 3D que oferece as restrições do ambiente para a movimentação das pessoas (uma representação 3D do estádio é ilustrada na Fig. 1.4), à direita.

Ao construir o modelo 3D do estádio é importante ter em conta todas as restrições físicas existentes no estádio real. O tamanho das portas, as dimensões dos corredores e a existência de obstáculos devem existir no modelo 3D. A Figura 1.5 apresenta áreas comuns do estádio que foram consideradas ao desenvolver o modelo 3D.

[7] `http://bfr.com.br/estadioniltonsantos.php`
[8] `http://bfr.com.br`

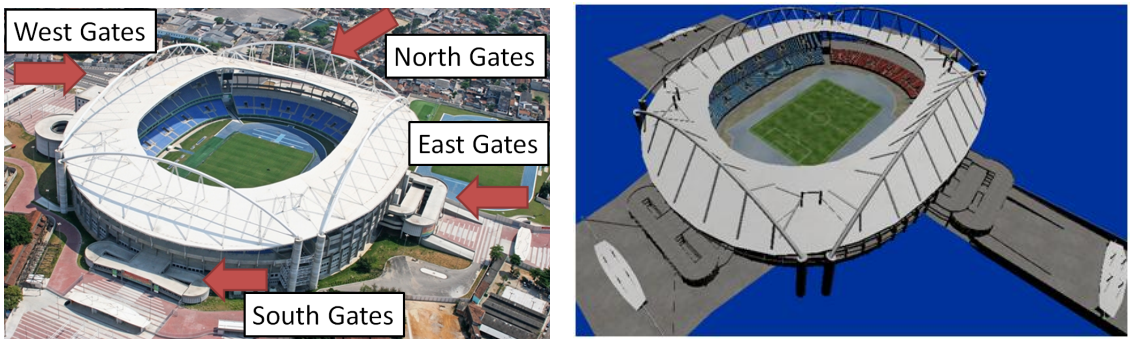

Figura 1.4. À esquerda é apresentada uma visão real do Estádio Olímpico onde se destacam as quatro áreas de acesso ao público; Enquanto que à direita ilustramos uma representação 3D do estádio.

Figura 1.5. Três representações de áreas comuns no estádio, incluindo as arquibancadas e corredores, onde as pessoas podem caminhar.

Outro aspecto importante, considerado neste projeto, foi a especificação correta das áreas das arquibancadas. Na Figura 1.6 é possível observar essas áreas (piso vermelho e cadeiras verdes) e as regiões definidas para permitir o movimento dos pedestres (azul). Considerando essa definição, quando ocorre um evento, por exemplo, o fim de uma partida, os indivíduos são capazes de deixar seus assentos e encontrar a melhor maneira de deixar o estádio.

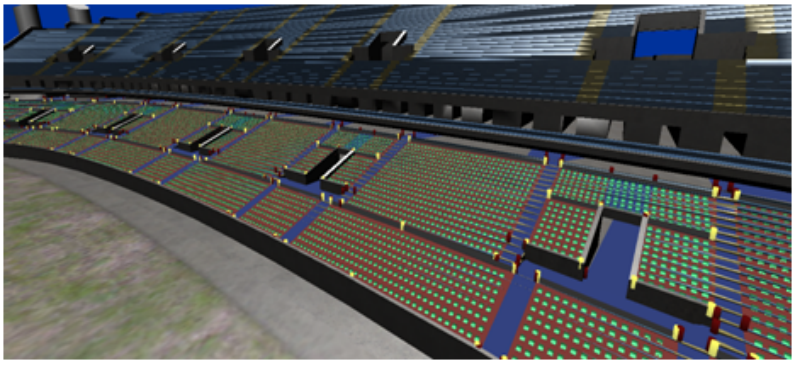

Figura 1.6. Exemplo de configuração de simulação, considerando as áreas de arquibancada.

Neste projeto, simulamos três situações diferentes considerando as informações sobre o histórico dos jogos fornecidos pelos gerentes do estádio:

1. A média da população durante alguns jogos observados: $\tilde{1}$7.000 pessoas;

2. capacidade total: $\tilde{4}$6.000 pessoas; e

3. um caso de teste onde consideramos a capacidade total do estádio juntamente com a indisponibilidade de uma saída (ver Figura 1.7).

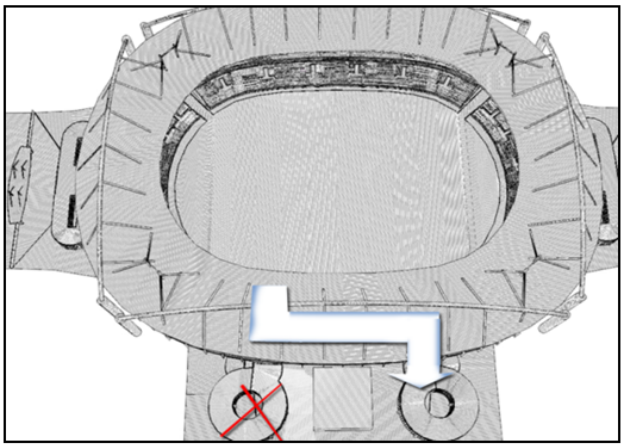

Figura 1.7. Ilustração de uma saída indisponível no estádio.

É importante ressaltar que todos os agentes conhecem sua melhor rota. Mesmo quando uma saída estava indisponível, os agentes sabiam disso e foram direto para a segunda melhor saída. Após as simulações, realizamos algumas análises estatísticas que nos permitiram destacar alguns aspectos:

- Em todas as situações simuladas os agentes foram capazes de caminhar de acordo com a velocidade desejada (1.2m/s);

- O tempo médio de todos os cenários foi de cerca de 7 minutos.

Figura 1.8 ilustra dois momentos das simulações no estádio.

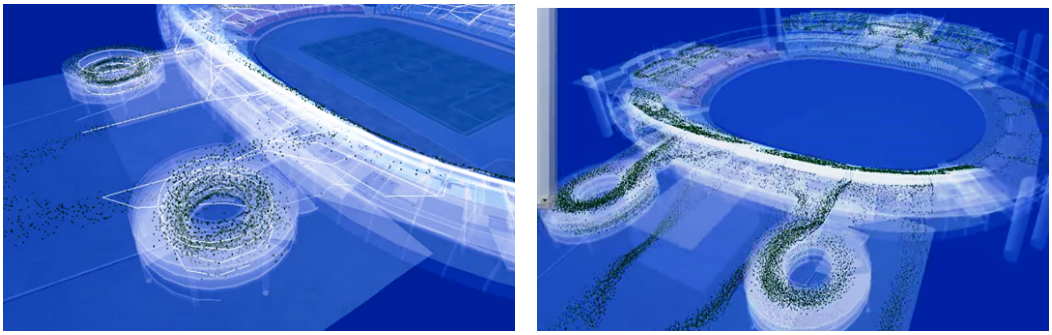

Figura 1.8. Representação visual da simulação de saída no estádio.

1.9.2. Simulação no Colégio Pastor Dohms - Porto Alegre

Neste estudo de caso, aplicamos CrowdSim em uma escola que recebe educação infantil, alunos do ensino fundamental e médio. O projeto foi realizado na Escola Pastor Dohms [9], em Porto Alegre - Brasil. O primeiro passo foi reproduzir a escola em um ambiente 3D como entrada solicitada. Posteriormente, definimos as restrições de ambiente, bem como os comportamentos desejados de acordo com as etapas enumeradas da seguinte forma:

1. Especificou-se as regiões onde o movimento é permitido (por exemplo corredores e escadas) no ambiente 3D. Além disso, definiu-se as regiões onde os agentes devem ser criados (isto é, salas de aula) e as regiões a serem consideradas como objetivos (ou seja, portas de saída). Essas informações são levadas em conta pelo CrowdSim ao executar o planejamento do caminho para calcular rotas de saída. A Figura 1.9 ilustra o modelo 3D da escola (à esquerda) e o ambiente mapeado no simulador, à direita.

2. Os dados populacionais foram definidos de acordo com as especificações da direção da escola. Definiu-se o número de agentes a serem simulados, bem como seus objetivos durante a simulação. A fim de especificar tais informações, considerou-se os valores de acordo com a ocupação real da escola para cada sala de aula, em cada edifício (como ilustrado na Figura1.9). A escola possui duas saídas que são consideradas pelos alunos ao sair do edifício, em dias normais. Além disso, observou-se a existência de portas extras (não utilizadas pelos alunos) que podem ser consideradas como rotas adicionais.

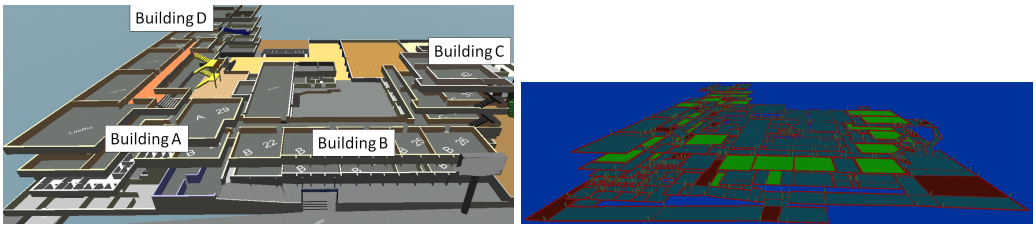

Figura 1.9. O colégio modelado em 3D (à esquerda); e a especificação do ambiente de simulação no CrowdSim (à direita).

Após as especificações do ambiente e restrições populacionais, realizaram-se simulações de acordo com quatro cenários diferentes. Tais cenários foram definidos considerando a população escolar pela manhã (1067 alunos) e pela tarde (729 pessoas), assim como as saídas disponíveis. A (Figura 1.10 ilustra possíveis configurações de saída.

Os quatro cenários simulados são:

1. População da manhã com apenas as principais saídas disponíveis;

2. população da tarde com apenas as saídas principais disponíveis;

3. população da manhã com todas as saídas disponíveis;

[9] http://dohms.org.br/

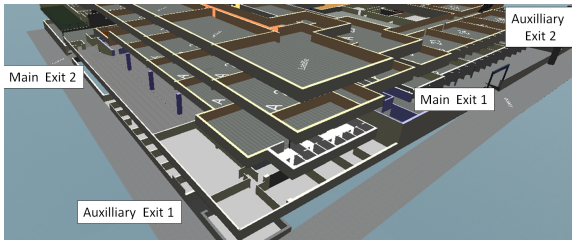

Figura 1.10. Saídas disponíveis da escola.

4. população da tarde com todas as saídas disponíveis.

Para testar os quatro cenários, foi necessário definir rotas capazes de guiar os agentes até a saída mais próxima, de acordo com as localizações das salas de aula. As rotas representam, neste ponto, um plano de evacuação a ser realizado. Neste projeto, quatro planos de evacuação foram desenvolvidos de acordo com a especificação de cada cenário de evacuação. A Figura 1.11 ilustra as rotas a serem seguidas pelos agentes que devem sair do Edifício D da escola. As linhas indicam o caminho a ser seguido enquanto as setas brancas representam a direção do movimento. Como ilustrado anteriormente na Figura 1.9 à esquerda, a escola é composta por quatro edifícios de salas de aula.

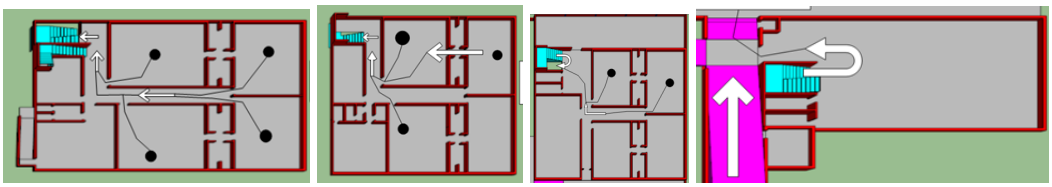

Figura 1.11. Rotas que representam o plano de evacuação do Edifício D.

Além de definir os planos de evacuação (ambiente e dados de pessoas, comportamentos e rotas), é importante enfatizar alguns pontos a serem observados durante a simulação de todos os cenários:

- A distribuição de pessoas por sala de aula foi calculada de acordo com os dados fornecidos pela direção da escola.
- Consideramos um tempo de reação para todos os cenários. Este tempo representa o tempo de resposta de cada agente até que ele começa a se mover, após receber uma orientação para a saída. Consideramos o tempo de resposta como $5s$ para todas as experiências.
- Os agentes não são criados exatamente na mesma hora. Nos experimentos, criamos grupos de 1 a 10 agentes (em cada sala de aula) observando um intervalo de 10 segundos. Este procedimento foi adotado para evitar que todos os agentes comecem a se mover ao mesmo tempo.
- Todos os agentes pretendem mover-se a uma velocidade de $0.8m/s$.

As análises dos resultados da simulação de multidões nos permitiram observar diferentes aspectos importantes durante o processo de saída. Um deles diz respeito à variação da densidade nos edifícios da escola durante a evacuação. Os dados estatísticos mostraram o tempo de simulação quando a maior densidade foi detectada. A estimação do tempo para maior densidade permitiu analisar a simulação e o ambiente para identificar o lugar de alta densidade como região de atenção. Nos quatro cenários simulados, a maior densidade ocorreu na escada dos edifícios C (cenários 1 e 3) e D (cenários 2 e 4). A Figura 1.12 apresenta dois quadros de simulação quando a maior densidade foi detectada no Edifício C (esquerda) e Edifício D (direita).

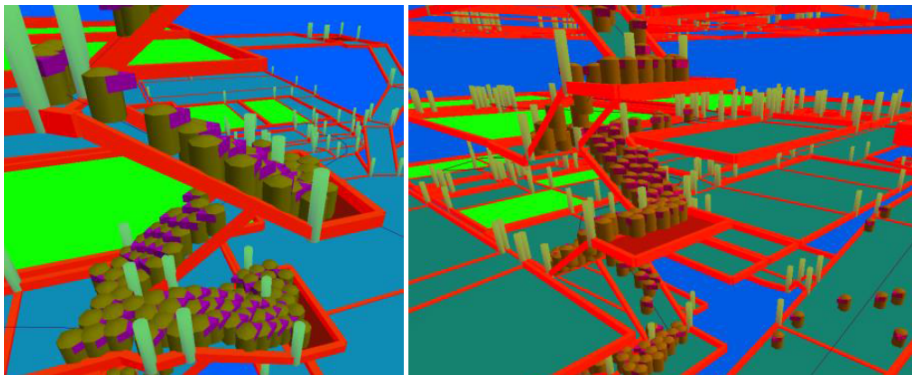

Figura 1.12. Maior densidade detectada por simulações no Edifício C (esquerda) e Edifício D (direita).

1.9.3. Simulação na Boate Santa Mônica - Porto Alegre

Nesta seção detalhamos a aplicação de CrowdSim em uma casa noturna, em Porto Alegre, Brasil. Nosso objetivo foi estudar como as pessoas realizam um processo de evacuação na vida real e assim obter dados para permitir comparações quantitativas. O experimento foi uma experiência compartilhada desenvolvida em parceria com os donos da casa noturna e uma empresa de segurança. No dia em que o experimento foi conduzido, o público concordou em deixar o clube exatamente às 2 da manhã. Alguns dias antes do exercício de saída no clube, CrowdSim foi testado para fornecer diferentes planos de evacuação que poderiam ser usados para estimar o comportamento dos ocupantes. O primeiro passo do processo foi reproduzir o ambiente do clube em 3D. O ambiente tem uma área total de $1010m^2$ e tem 4 andares (veja Figura 1.13 para ver os locais de saída). Uma representação 3D é ilustrada na Figura 1.13.

O ambiente 3D foi necessário para permitir a definição dos possíveis planos de evacuação. Neste modelo, são definidas as regiões onde os agentes devem ser inseridos e removidos da simulação e também especifica-se as regiões onde o movimento é permitido. A especificação do ambiente, definida no módulo de configuração do CrowdSim, de acordo com o modelo 3D do ambiente, possibilita a construção de um gráfico do ambiente. Esse gráfico é ilustrado na Figura 1.14 e é composto por três tipos diferentes de nós, que são representados por cores diferentes:

1. nós alaranjados representam áreas de decisão. Em tais áreas do clube, onde os agentes podem ser criados, eles precisam escolher rotas diferentes a partir desse

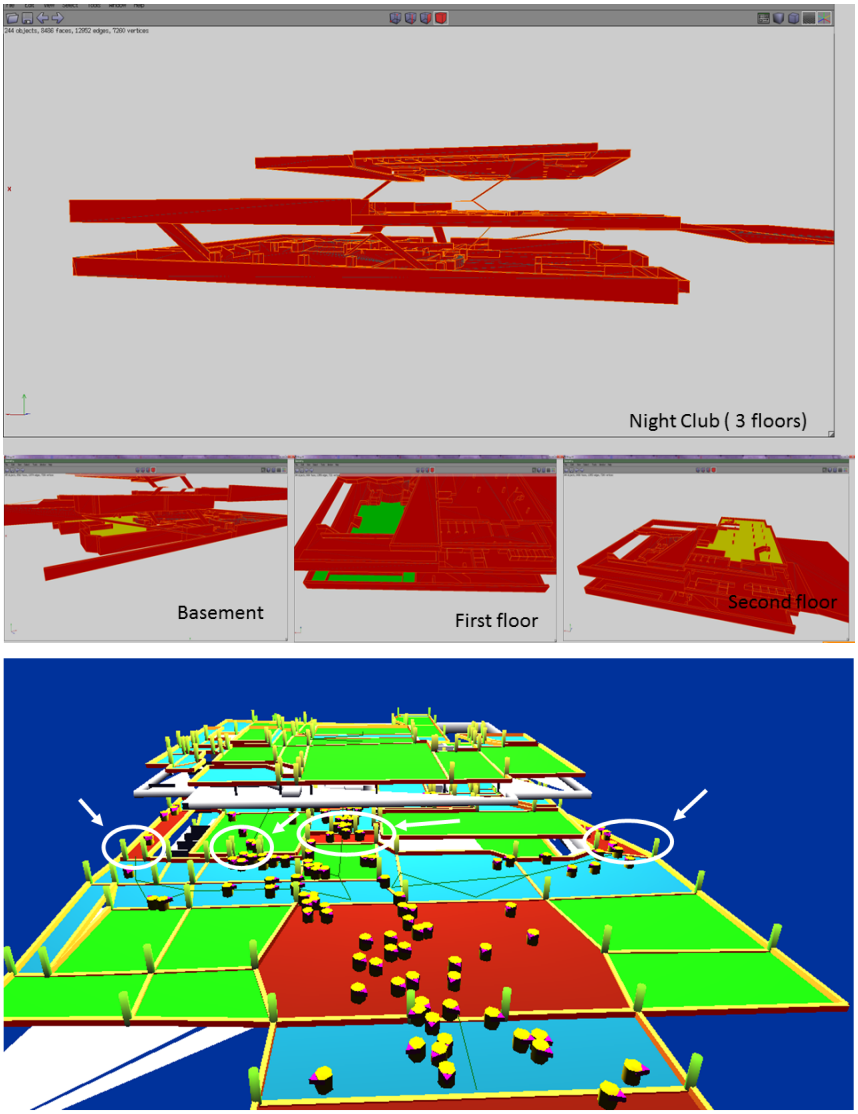

Figura 1.13. Ilustração do modelo 3D da boate e da localização das portas de saída na interface do CrowdSim.

ponto. Esses nós representam as bifurcações no gráfico;

2. nós verdes representam as escadas responsáveis pela conexão dos diferentes andares do clube. Nenhum agente é criado em tais regiões que são consideradas apenas áreas de movimento; e

3. Os nós brancos são regiões onde os agentes podem ser criados na simulação e também, áreas passíveis de movimento. A única exceção é o nodo chamado *Street* que é responsável pela remoção dos agentes na simulação (contexto de saída).

Após a realização de várias simulações, foi possível identificar um conjunto de planos de evacuação plausíveis a serem realizados no exercício da vida real. Na Tabela 1.2 apresentamos os resultados computados de três planos de evacuação que foram projetados e testados no CrowdSim pelos engenheiros de segurança.

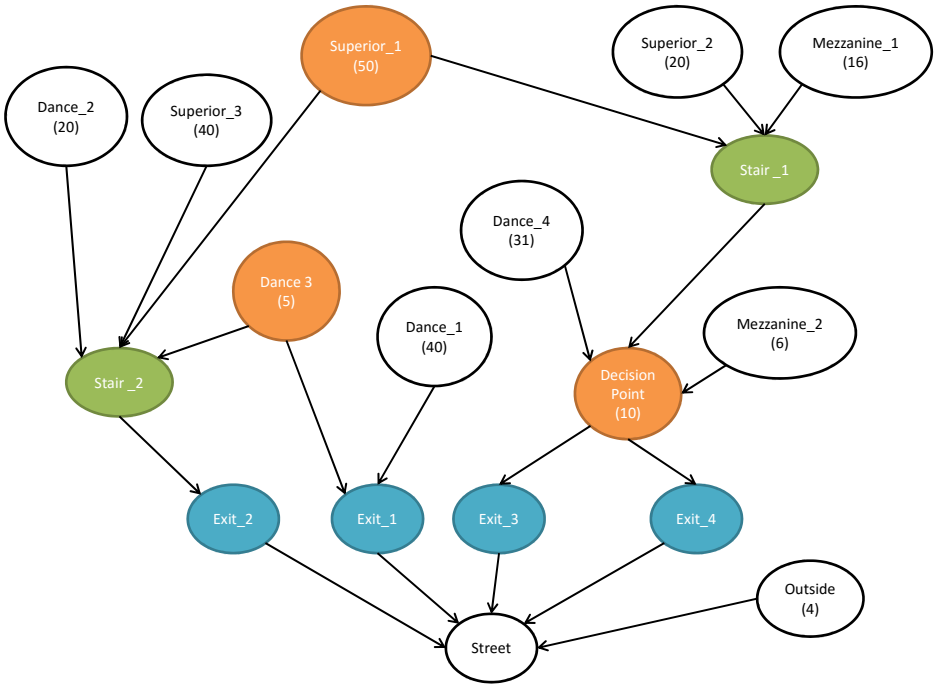

Figura 1.14. Gráfico do clube noturno especificando a estrutura ambiental a ser considerada na geração de possíveis rotas de evacuação. Todos os nós significam regiões caminháveis, enquanto as bordas representam as portas que conectam as salas do lugar. Além disso, nós laranja significma áreas de decisão (onde os agentes podem escolher caminhos diferentes), enquanto os nós verdes representam as escadas que conectam os diferentes andares do clube noturno.

	Simulação 1	Simulação 2	Simulação 3
tempo máximo (sec)	142	142	146
tempo médio (sec)	61	62	64
densidade média $(pessoas/m^2)$	0.1123	0.1138	0.1162
velocidade média m/s	0.80	0.80	0.80
Local com maior densidade	escadas do 2^o andar	escadas do 2^o andar	escadas do 2^o andar
Tempo em que máxima densidade foi observada	Segundo 40	Segundo 39	Segundo 50
Maior velocidade (m/s)	1.3	1.2	1.3
Menor velocidade (m/s)	0.01	0.01	0.005
Maior densidade	5.4	5.4	5.0
Número de pessoas na Door1	54	18	21
Número de pessoas na Door2	12	41	50
Número de pessoas na Door3	80	126	75
Número de pessoas na Door4	100	61	100

Tab. 1.2. Dados quantitativos comparando cenários simulados contendo 240 pessoas.

Avaliando-se a tabela, optou-se por utilizar o plano da Simulação 1 como estratégia para treinamento de evacuação das pessoas reais, na Boate. Assim, a empresa de

segurança começou a treinar indivíduos que trabalham no local. A verdadeira evacuação foi realizada com 240 pessoas que concordaram em participar da experiência. Durante o exercício de saída real, pode-se coletar dados diferentes para avaliar os resultados dessa experiência. Os dados dos ocupantes foram obtidos a partir de vídeos de câmeras de segurança. Esta informação foi muito importante para avaliar este trabalho. A Tabela 1.3 resume a comparação entre os cenários de evacuação real e virtual.

	Simulação	Dados do Mundo Real
Tempo total de evacuação (segundos)	119	175
Densidade máxima (pessoas / m^2)	5.4	4.5
Lugar da maior densidade	Escadas (2^o andar)	Escadas (2^o andar)
Tempo em que a maior densidade foi observada	Segundo 40	Segundo 50
Velocidade máxima (m / s)	1.3	1.5
Velocidade menor (m / s)	0,1	0,2

Tab. 1.3. Dados quantitativos comparando mundos reais e simulados considerando exatamente o mesmo plano de evacuação.

A Figura 1.15 fornece uma imagem capturada durante a evacuação que mostra as pessoas nas escadas (2^o andar) aos 40 segundos, após a simulação começar, e outra imagem no mesmo local e tempo na simulação virtual.

Figura 1.15. Imagens ilustrando as escadas no segundo andar, 40 segundos após o início da simulação, em ambiente real e virtual.

Ao analisar a Tabela 1.3 há claras diferenças no tempo de evacuação. Isso pode ser explicado pelo fato de que pessoas reais não se comportam voluntariamente da mesma forma que em uma verdadeira emergência. Ou seja, pessoas reais, não em pânico, respeitam o espaço de outros, e portanto não atingem as maiores densidades aparentes nos dados de simulação.

1.10. Considerações Finais

As pessoas, quando fazem parte de uma multidão, são capazes de realizar um comportamento incomum,que não realizariam se estivessem sozinhas [2]. Uma multidão é uma entidade importante e sua compreensão é relevante, sobretudo em questões de avaliação de segurança e conforto de populações.

Ao simular multidões, os engenheiros podem validar seus comportamentos e evolução em um ambiente específico de acordo com diferentes circunstâncias e restrições. Além disso, as empresas podem economizar tempo e dinheiro ao simular e analisar o comportamento das multidões durante a fase de projeto de edifícios.

Em particular, neste texto focou-se na simulação de evacuação, para investigar a melhor maneira de uma multidão deixar um ambiente. A fim de exemplificar este problema e suas soluções, apresentou-se CrowdSim, um protótipo desenvolvido no VHLAB [10] para simular multidões em diferentes situações durante um processo de evacuação. Além do protótipo, alguns dos estudos de caso desenvolvidos usando CrowdSim foram apresentados.

Referências

[1] S. Freud, *Group psychology and the analysis of the ego*. New York: New York, Boni and Liveright, 1922.

[2] G. LeBon, *Psychologie des Foules*. Paris: Alcan, 1895.

[3] W. Mc Dougall, *The Group Mind (1920)*. La Vergne, US: Lightning Source, 2009.

[4] S. Sighele, *A multidão Criminosa - Ensaio de Psicologia Coletiva*. Tradução Adolfo Lima., 1954.

[5] C. McPhail, *The Myth of the Madding Crowd*. New York, USA: Walter de Gruyter, 1991.

[6] S. Patil, J. Van Den Berg, S. Curtis, M. Lin, and D. Manocha, "Directing crowd simulations using navigation fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 244–254, 2011.

[7] G. Berseth, M. Kapadia, and P. Faloutsos, "Robust space-time footsteps for agent-based steering," *Computer Animation and Virtual Worlds*, 2015.

[8] C. D. Boatright, M. Kapadia, J. M. Shapira, and N. I. Badler, "Generating a multiplicity of policies for agent steering in crowd simulation," *Computer Animation and Virtual Worlds*, pp. n/a–n/a, 2014. [Online]. Available: http://dx.doi.org/10.1002/cav.1572

[9] D. Thalmann and S. R. Musse, *Crowd Simulation - Second Edition*. Springer-Verlag London Ltd, 2013.

[10] J. Fruin, *Pedestrian Planning and Design*. Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[11] E. T. Hall, *The hidden dimension / Edward T. Hall*, [1st ed.] ed. Doubleday, Garden City, N.Y. :, 1966.

[10] www.inf.pucrs.br/vhlab

[12] U.S. Department of Labor , "How to plan for workplace emergencies and evacuations," online: https://www.osha.gov/Publications/osha3088.pdf, The address of the publisher, 2001.

[13] C. R. M. Challenger, Rose; Clegg, *Understanding Crowd Behaviours: Guidance and Lessons Identified*. York, UK: The Cabinet Office Emergency Planning College, 2009.

[14] A. Berlonghi, "Understanding and planning for different spectator crowds," vol. 18, no. 4, pp. 239–247, 1995-02-01T00:00:00. [Online]. Available: http://www.ingentaconnect.com/content/els/09257535/1995/00000018/00000004/art00033

[15] L. C. Osorio, *Psicologia Grupal: Uma nova disciplina para o advento de uma era*. Porto Alegre: Artmed, 2003.

[16] G. K. Still, "Crowd dynamics," Ph.D. dissertation, University of Warwick, Coventry, UK, 2000.

[17] H. Mu, J. Wang, Z. Mao, J. Sun, S. Lo, and Q. Wang, "Pre-evacuation human reactions in fires: An attribution analysis considering psychological process," *Procedia Engineering*, vol. 52, pp. 290 – 296, 2013, 2012 International Conference on Performance-based Fire and Fire Protection Engineering. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877705813002610

[18] C. Cocking and J. Drury, "The mass psychology of disasters and emergency evacuations: a research report and implications for the fire and rescue service," *Fire Safety, Technology and Management*, vol. 10, no. 2, pp. 13–19, 2008. [Online]. Available: http://eprints.brighton.ac.uk/11416/

[19] *NFPA 101 Life Safety Code*. Quincy, MA: National Fire Protection Association, 2015.

[20] H. W. Hamacher and S. A. Tjandra, "Mathematical Modelling of Evacuation Problems – A State of the Art," in *Pedestrian and Evacuation Dynamics*, M. Schreckenberg and S. D. Sharma, Eds. Berlin: Springer, 2002, pp. 227–266.

[21] N. T. N. Anh, Z. J. Daniel, N. H. Du, A. Drogoul, and V. D. An, "A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks," in *Proceedings of the 10th International Conference on Advanced Agent Technology*, ser. AAMAS'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 371–383. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-27216-5_28

[22] X. Wei, M. Xiong, X. Zhang, and D. Chen, "A hybrid simulation of large crowd evacuation," in *Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on*, Dec 2011, pp. 971–975.

[23] L. B. Zomer, W. Daamen, S. Meijer, and S. P. Hoogendoorn, "Managing crowds: The possibilities and limitations of crowd information during urban mass events," *Planning Support Systems and Smart Cities. Part of the series Lecture*

Notes in Geoinformation and Cartography, pp. 77–97, 2015. [Online]. Available: http://link.springer.com/chapter/10.1007%2F978-3-319-18368-8_5

[24] M. Kapadia, N. Pelechano, J. Allbeck, and N. Badler, *Virtual Crowds: Steps Toward Behavioral Realism*. Morgan & Claypool Publishers, 2015.

[25] A. Braun, S. R. Musse, L. P. L. d. Oliveira, and B. E. J. Bodmann, "Modeling individual behaviors in crowd simulation," in *CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*. Washington, DC, USA: IEEE Computer Society, 2003, p. 143.

[26] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, pp. 487–490, Sep 2000. [Online]. Available: http://arxiv.org/abs/cond-mat/0009448

[27] N. ZHU, J. WANG, and J. SHI, "Application of pedestrian simulation in olympic games," *Journal of Transportation Systems Engineering and Information Technology*, vol. 8, no. 6, pp. 85 – 90, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/B8H0W-4VDRVV0-7/2/3ff342f75f2deb5c1acdb0484b75e002

[28] L. Fu, W. Song, W. Lv, and S. Lo, "Simulation of exit selection behavior using least effort algorithm," *Transportation Research Procedia*, vol. 2, no. 0, pp. 533 – 540, 2014, the Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S235214651400129X

[29] L. Ji, Y. Qian, J. Zeng, M. Wang, D. Xu, Y. Yan, and S. Feng, "Simulation of evacuation characteristics using a 2-dimensional cellular automata model for pedestrian dynamics," *Journal of Applied Mathematics*, 2013.

[30] L. E. Aik and T. W. Choon, "Simulating evacuations with obstacles using a modified dynamic cellular automata mode," *Journal of Applied Mathematics*, vol. 2012, 2012.

[31] M. L. Chu, P. Parigi, K. Law, and J.-C. Latombe, "Modeling social behaviors in an evacuation simulator," *Computer Animation and Virtual Worlds*, vol. 25, no. 3-4, pp. 373–382, 2014. [Online]. Available: http://dx.doi.org/10.1002/cav.1595

[32] N. Pelechano, J. Allbeck, and N. Badler, *Virtual Crowds: Methods, Simulation, and Control (Synthesis Lectures on Computer Graphics and Animation)*. Morgan and Claypool Publishers, 2008.

[33] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler, "Crowd simulation incorporating agent psychological models, roles and communication," DTIC Document, Tech. Rep., 2005.

[34] N. Pelechano and N. I. Badler, "Modeling crowd and trained leader behavior during building evacuation," *IEEE Computer Graphics and Applications*, vol. 26, no. 6, pp. 80–86, Nov 2006.

[35] N. Pelechano and A. Malkawi, "Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches," *Automation in Construction*, vol. 17, no. 4, pp. 377 – 385, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580507000908

[36] E. Kuligowski and S. Gwynne, "What a user should know when selecting an evacuation model," *Fire Protection Engeneering*, pp. 600–611, Oct. 2005. [Online]. Available: http://magazine.sfpe.org/occupants-and-egress/what-user-should-know-when-selecting-evacuation-model

[37] E. D. Kuligowski, R. D. Peacock, and B. L. Hoskins, "A review of building evacuation models, 2nd edition," *Technical Note (NIST TN) - 1680*, December 2010. [Online]. Available: http://www.nist.gov/manuscript-publication-search.cfm?pub_id=906951

[38] V. J. Cassol, R. A. Rodrigues, L. C. C. Carneiro, A. Silva, and S. R. Musse, "Crowdsim: Uma ferramenta desenvolvida para simulação de multidões," in *I Workshop de Simulação Militar - SBGames2012*, 2012.

[39] V. Cassol, C. M. Dal Bianco, A. Carvalho, J. Brasil, M. Monteiro, and S. R. Musse, "An experience-based approach to simulate virtual crowd behaviors under the influence of alcohol," in *IVA'15: Proceedings of the 15th International Conference on Intelligent Virtual Agents*. Berlin, Heidelberg: Springer-Verlag, 2015.

[40] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *SIGART Bull.*, no. 37, pp. 28–29, 1972.

Capítulo 2

A Aplicação de Métodos Qualitativos em Computação

Carla Faria Leitão e Raquel Oliveira Prates

Abstract

Technology is ever more ingrained into people's lives, and thus, has increased computer scientists' need and interest in learning about user related aspects that cannot be measured related to habits, attitudes, behavior, values, emotions, among other psychological, and social-cultural factors. In order to do so, researchers and professionals in the computer field have adopted qualitative methods, some of them are traditionally used in human and social sciences, whereas others are proposed within the computer science field. The goal of this chapter is to present an overview of qualitative methods, their characteristics and their differences compared to quantitative methods. The methods that are more frequently adopted in the computer field are presented, and examples of their application in different areas of computing are presented and discussed.

Resumo

A ampla adoção da tecnologia na vida das pessoas tem gerado uma necessidade e interesse maior em profissionais e pesquisadores da área de computação em conhecer aspectos não mensuráveis dos usuários ligados a hábitos, atitudes, comportamentos, valores e emoções, entre outros fatores psicológicos e socioculturais. Para isso, tem-se percebido uma crescente adoção de métodos de pesquisa qualitativos em computação, sejam aqueles tradicionalmente usados em pesquisas nas ciências humanas e sociais, sejam métodos de base qualitativa criados no interior da área. O objetivo deste capítulo é apresentar uma visão geral de métodos qualitativos e suas principais características, discutindo suas diferenças em relação a métodos quantitativos. São apresentados os métodos mais comumente utilizados em computação e alguns exemplos de sua aplicação.

2.1. Introdução

Os métodos qualitativos apresentam uma longa e consolidada história nas ciências humanas e sociais, resultado, desde a década de 1920, do desenvolvimento de uma

metodologia alternativa aos procedimentos experimentais e quantitativos. Disciplinas como a Antropologia Social, a Sociologia e a Psicologia reagiram à metodologia quantitativo-experimental por considerarem esse modelo de investigação inadequado para a identificação e compreensão da natureza cultural, social e humana de seus objetos de investigação. As diferenças entre fenômenos físico-químicos objetivos e replicáveis e a complexidade e imprevisibilidade de comportamentos, relações humanas e fenômenos culturais, por exemplo, geram não apenas uma distinção nas ferramentas de investigação, mas, sobretudo, uma diferença de concepção a respeito do que será investigado, para qual objetivo e qual tipo de resultado poderá ser obtido ao fim do processo. Há, portanto, por trás dos métodos quantitativos e qualitativos, diferenças muito mais profundas do que o fato de um tipo de método trabalhar com números e o outro não. Trata-se da existência de diferenças de paradigma, ou seja, de modelos de geração de conhecimento, seja este conhecimento de natureza científica, seja de natureza prática, referido a contextos profissionais.

Historicamente, de modo análogo aos de outras áreas das ciências exatas e de tecnologia, a área de Computação volta-se fortemente para o ensino e uso de métodos quantitativo-experimentais. Notadamente em decorrência da natureza algorítmica de seu objeto de estudo e de aplicação prática, a construção de hipóteses, a manipulação de variáveis e a reprodutibilidade e estabilidade da ocorrência de fenômenos mostra-se perfeitamente adequada para a geração de conhecimento científico e para o desenvolvimento de sistemas computacionais.

Mais recentemente, contudo, a difusão maciça de tecnologias da informação (TI) nas mais variadas esferas da vida humana e social, bem como nas mais diferentes culturas e domínios, vem gerando a necessidade de identificar e compreender aspectos não mensuráveis das experiências humanas com TI. Por exemplo, o contraste entre o comportamento estável de sistemas computacionais e a criatividade e a imprevisibilidade com que os usuários os utilizam, e as resistências ou o fracasso na introdução e adoção de uma tecnologia computacional tida como promissora revelam o quanto a computação se defronta com dimensões qualitativas em diferentes ciclos de pesquisa e desenvolvimento. Consequentemente, diversos profissionais de diferentes áreas já vêm buscando conhecer e aplicar métodos qualitativos de investigação em suas atividades.

Ocorre, porém, que a formação típica desses profissionais os prepara para métodos matemáticos e experimentais, mas mostra-se ainda um tanto insuficiente para o uso de métodos qualitativos. A rara oferta de disciplinas ou tópicos de disciplinas sobre o tema, bem como a escassez de bibliografia contextualizada para a área de TI, faz com que os interessados tenham que transpor as dificuldades inerentes à aquisição de conhecimentos em outros campos disciplinares. Este capítulo, parte de um minicurso para alunos e profissionais da área de computação, busca permitir o entendimento introdutório da metodologia qualitativa e de alguns de seus principais métodos e técnicas, os tipos de resultados que se pode alcançar com cada um deles, e como aplicá-los na prática de pesquisa ou desenvolvimento de tecnologias computacionais.

Na primeira seção, em sequência à Introdução, as diferenças entre paradigma quantitativo-experimental e o paradigma qualitativo são sucintamente expostas, e as características comuns aos métodos qualitativos e seus tipos principais são definidas. As seções seguintes apresentam alguns dos métodos segundo o tipo de coleta de dados que realizam: métodos de coleta de dados com usuários e especialistas, métodos de coleta em

contextos reais de uso e métodos de inspeção. Após essa visão geral, são discutidos aspectos relacionados ao tipo de aplicação propiciada pelos métodos: desenvolvimento e pesquisa e, em seguida, comentários finais acerca dos limites e desafios envolvidos na aplicação de métodos qualitativos em computação são brevemente abordados.

2.2. O que São Métodos Qualitativos

Nesta seção, apresentamos as principais diferenças entre os paradigmas qualitativo e quantitativo-experimental e como estas diferenças impactam os tipos de problemas a serem investigados pelos métodos associados a cada paradigma, o tipo de raciocínio e análise aplicados na sua condução, assim como o tipo de resultados que geram. Em seguida, apresentamos as principais características de métodos qualitativos, e o framework PRET A Rapporter (PRETAR) [Blandford, 2013] que descreve os passos gerais a serem seguidos na aplicação de métodos qualitativos.

2.2.1. Diferenças entre os Paradigmas Qualitativo e Quantitativo-Experimental

Por trás da ideia mais imediata de que a principal oposição entre os métodos quantitativos e qualitativos reside na diferença entre quantificar ou não quantificar resultados, há diferenças muito mais profundas. Para além de um tipo de método trabalhar com números e o outro não, existem diferenças de paradigma, ou seja, de modelos de geração de conhecimento, seja este conhecimento de natureza científica, seja de natureza prática em contextos profissionais. Quando existe clareza sobre as diferenças básicas entre o paradigma quantitativo-experimental e o paradigma qualitativo [Creswell, 2009; Denzin e Lincoln, 2006; Leitão, 2009], percebe-se que a escolha de um método reflete muito mais do que a escolha de um tipo de procedimento entre vários para a condução da investigação. Na verdade, a opção pelo uso de métodos qualitativos envolve uma maneira diferente de investigar um fenômeno, desde o tipo de problema e de pergunta/questão a ser construída, até o tipo de respostas e resultados obtidos, passando pelo modo de observar o problema e os instrumentos utilizados para este fim.

Em relação às **características do fenômeno investigado**, o paradigma qualitativo pressupõe que os fenômenos a serem examinados são irreplicáveis, complexos, imprevisíveis e sempre relativos a um contexto de ocorrência, sendo impossível elencar e isolar todas as suas variáveis para conhecê-los através do controle experimental. De um lado, no paradigma quantitativo, enfatiza-se a possibilidade de prever o comportamento dos fenômenos na medida em que se identifica, conhece e controla suas variáveis e espera-se, consequentemente, a replicação dos fenômenos estudados. Já no paradigma qualitativo, reage-se à ideia de que é possível identificar, prever e controlar as formas humanas de agir, sentir, pensar, se relacionar e se organizar em grupos. Os fenômenos humanos são múltiplos, heterogêneos e complexos. Não se prestam à identificação e controle de todos os seus componentes, sendo, portanto, irreplicáveis. Contextualizados, não permitem a abstração do contexto e o universalismo de conclusões.

Destas características, decorre a forma de abordagem do fenômeno, ou seja, qual **o tipo de problema** que cada paradigma se mostra adequado para tratar e como o faz. No paradigma quantitativo, o problema é hipotetizável, ou seja, pressupondo que as condições de ocorrência de um fenômeno podem ser previstas e replicadas, a investigação relaciona-se à formulação e teste de uma hipótese a respeito dessa ocorrência. Já o paradigma qualitativo, considerando o pressuposto da complexidade e imprevisibilidade, não se mostra interessante para testes de hipóteses prévias, mas sim para a exploração de

problemas e contextos a respeito dos quais efetivamente se busca respostas novas e imprevisíveis. Os métodos desse paradigma são inerentemente exploratórios e guiados sempre por uma questão de estudo aberta, que excluem perguntas que visam confirmar hipóteses, expectativas e conhecimentos prévios. Essa distinção mostra-se muito importante na escolha do paradigma sobre o qual repousará uma determinada pesquisa. Não há um paradigma certo ou errado em termos genéricos, mas há tipos de problemas certos para determinado paradigma. Se o pesquisador ou profissional deseja confirmar ou testar uma hipótese ou alguma expectativa prévia, ele não se beneficiará da metodologia qualitativa, ao passo que, se desejar explorar um cenário desconhecido para obter informações em profundidade sobre determinado contexto, encontrará fortes subsídios nesse tipo de metodologia.

A condução da investigação exige diferentes **tipos de raciocínio e de ação** por parte de seus envolvidos. Nos métodos quantitativos, o raciocínio é voltado para a dedução, raciocínio que permite tirar uma conclusão a partir de uma ou mais proposições. Baseado no modelo matemático, o paradigma quantitativo-experimental parte do raciocínio hipotético-dedutivo [Japiassú e Marcondes, 1996], ou seja, define um conjunto de hipóteses que são submetidas à verificação e demonstração por meio da manipulação das variáveis operacionalmente definidas e da análise estatística. Já o paradigma qualitativo tem por base o raciocínio indutivo e interpretativo [Japiassú e Marcondes, 1996], de influência empirista, que vai do particular para o geral a partir da observação e análise dos fenômenos. Esta análise tem como objetivo explorar e construir significados sobre o fenômeno em estudo. Para isso, o pesquisador faz uma análise segmentada dos dados, em geral linguísticos, interpretando-os de modo a atribuir significados a eles e, assim, construir suas categorias de análise. Observação, análise segmentada do conteúdo, atribuição de significados e categorização são atividades realizadas interativamente. Juntas constituem um processo rigoroso e sistemático de interpretação, indutivamente baseado em dados objetiváveis, rastreáveis e identificáveis que podem, sempre que necessário, ser recuperados e reexaminados. Difere, portanto, de nossos mecanismos de interpretação cotidiana, em geral opinativa, informal e subjetiva, temidas características que muitos temem ver associadas às pesquisas qualitativas que realizam [Leitão, 2009].

A interpretação sistemática é reveladora de outra importante diferença entre os dois paradigmas. Esta diz respeito à **posição adotada pelo investigador** ao longo do processo. Dentro do paradigma quantitativo-experimental, vigora a crença na neutralidade do pesquisador que, despido de valores e posições éticas, desvela a realidade dos fenômenos. Já sob a ótica da pesquisa qualitativa, a interpretação, embora sistemática, objetivável e rastreável, é fortemente dependente do conhecimento prévio do pesquisador, de sua bagagem teórica, bem como de seus valores e contornos sociais, culturais e históricos.

O **tipo de resultados** obtido em cada método revela-se tão distinto quanto são os processos de coleta e análise dos dados. No paradigma quantitativo-experimental, os resultados almejados evocam generalização, abrangência e universalidade. Busca-se definir padrões, diretrizes, regras e leis gerais de funcionamento que permitam a replicação do conhecimento obtido. No paradigma qualitativo, a contextualização do objeto de estudo e a interpretação sistemática pressupõem que o tipo de resultado gerado é sempre parcial, situado e relativo, e, portanto, não replicável ou generalizável. Enquanto os métodos quantitativos enfatizam o produto obtido (o quê, por quê), os qualitativos valorizam o processo (como) e a descrição dos achados, e oferecem uma rede de

significados articulados que dão uma perspectiva em profundidade e contextualizada do fenômeno explorado. Essa rede de conhecimentos serve como framework interpretativo a ser reaplicado em outros contextos de investigação. Em outras palavras, na perspectiva qualitativa, o conhecimento produzido não é um produto replicável, mas um processo de compreensão e interpretação reutilizável.

A Tabela 2. 1 resume as diferenças abordadas entre os dois paradigmas. Em seguida, apresentamos as características comuns aos métodos que integram o paradigma qualitativo e alguns de seus principais tipos, escolhidos em função de já virem sendo utilizados na área de Computação.

Tabela 2. 1: Paradigmas quantitativo-experimental e qualitativo

	Paradigma Quantitativo-Experimental	**Paradigma Qualitativo**
Pressuposto sobre os fenômenos em exame	Estabilidade e previsibilidade dos fenômenos Fenômenos abstraídos de seu contexto de ocorrência	Ocorrência não previsível dos fenômenos Fenômenos vinculados ao seu contexto de ocorrência
Tipo de problema / questão	Elaboração e teste de hipóteses a partir de algum conhecimento / expectativa sobre o problema	Exploração contextualizada por meio de questões abertas, sem hipóteses prévias
Raciocínio	Hipotético-dedutivo	Interpretativo-indutivo
Ação do investigador	Manipulação de variáveis Análise Estatísticas	Exploração de significados Análise de conteúdo/discurso
Postura do investigador	Concepção de neutralidade	Envolvimento interpretativo; análise de impactos éticos
Tipo de resultados	Abrangentes. Padrões, generalizações replicáveis	Em profundidade. Framework interpretativo baseado em rede de significados relacionados ao contexto de investigação

2.2.2. Definição e Tipos Principais

Métodos qualitativos são conjuntos diversos, porém sistemáticos, de procedimentos criados e adotados para investigar um determinado fenômeno ou grupo de fenômenos que (ainda) não se prestam à elaboração prévia de hipóteses. Comumente, esses fenômenos referem-se a opiniões, hábitos, valores, atitudes, demandas, desejos, emoções e comportamentos de seres humanos e grupos sociais em diferentes contextos histórico-culturais. O uso da metodologia qualitativa implica no entendimento de que as ações, comportamentos e interações sob exame, por serem desconhecidas e imprevisíveis, não podem ser previstas e antecipadamente traduzidas em hipóteses. Os métodos qualitativos devem ser escolhidos quando se buscam ferramentas exploratórias de investigação, em vez de métodos de checagem, refutação ou confirmação de achados ou impressões anteriores. Além disso, os métodos qualitativos mostram-se especialmente interessantes para o estudo de objetos complexos e multifacetados, cujas ações e comportamentos dificilmente podem ser capturados pela definição e manipulação de um conjunto limitado

de variáveis operacionais. Com este objetivo, os métodos reúnem um conjunto de características comuns expostas em seguida.

Investigação naturalística: os métodos buscam coletar o dado o mais próximo possível do contexto natural de ocorrência do fenômeno, de modo a capturar com minúcias as informações de contexto e não deturpar dados em função da artificialidade da ocorrência do fenômeno. Em vez das condições de controle dos laboratórios e da manipulação de variáveis, que retira o objeto de seu contexto, na metodologia qualitativa busca-se prioritariamente a observação naturalística dos fenômenos ou o discurso direto daqueles que vivem e sentem os fenômenos estudados. Os métodos qualitativos visam a emergência espontânea de significados, ações e comportamentos, evitando a criação de situação artificiais de coleta de dados. Há variações sobre o quanto cada método consegue se aproximar mais ou menos da ocorrência espontânea dos fenômenos. A observação participante do uso de tecnologias no ambiente de trabalho e os estudos etnográficos, por exemplo, estão bem próximos da ocorrência natural. Há, no entanto, estudos qualitativos que não podem ser realizados em situação natural, como é o caso do teste de protótipos, por exemplo. Nesse caso, o estudo deve se afastar tanto quanto possível das condições artificiais de controle experimental, se aproximando, tanto quanto possível, de uma situação real de uso da tecnologia em foco. Outro caso importante diz respeito a dados não observáveis, posto que são fruto de reflexão e de processos internos dos usuários. É possível observar a utilização de um sistema computacional no contexto real de uso, mas não é possível, pela observação, conhecer a opinião, as críticas e dificuldades que o grupo de usuários encontrou. Nestas condições, é necessário um instrumento que possibilite a externalização de processos internos, como é o caso de entrevistas, questionários, diários de uso e grupos de discussão. Estas técnicas, ainda que se afastem (em maior ou menor grau) da ocorrência naturalística, devem se aproximar de uma conversa informal e natural que favoreça a exposição espontânea e não tendenciosa de significados através de elementos verbais e não verbais.

Investigação em profundidade com pequenas amostras: Os métodos qualitativos privilegiam o estudo em profundidade de sua questão de estudo em detrimento do conhecimento amplo, genérico, mas, por consequência, necessariamente mais superficial. Para isto, precisam adotar um foco restrito, preciso e nítido de investigação, investindo na máxima '*menos é mais*' e trabalhando com pequenas amostras. Comumente, para aqueles formados dentro do paradigma quantitativo, esta característica é uma limitação dos métodos qualitativos. Sob outra perspectiva, no entanto, é importante perceber que os métodos qualitativos têm foco e amostra restritos não por problemas na concepção ou condução, mas em função da diferença de objetivo. Conforme metáfora descrita por Leitão [2009], na pesquisa quantitativa, o pesquisador assemelha-se ao fotógrafo que usa uma lente grande angular, pois deseja fazer o registro de uma multidão por meio de uma foto panorâmica. Por outro lado, na pesquisa qualitativa, o fotógrafo mune-se de uma lente *zoom*, para permitir que ele capture o detalhe sutil e invisível ao olhar panorâmico. Para capturar minúcias e detalhes em profundidade, as pesquisas qualitativas são trabalhosas (*labor-intensive*) e de lenta execução (não-automatizáveis). Seus passos envolvem frequentes tomadas de decisão, denso trabalho intelectual e atividade analítica interativa e artesanal. Em resumo, o pesquisador trabalha intensivamente com pequenas amostras ao invés de extensivamente com grandes amostras [Nicolaci-da-Costa et al., 2004].

Investigação de significados a partir da perspectiva do participante: Os métodos qualitativos buscam sempre capturar a perspectiva dos usuários, isto é, como os participantes vivem, percebem, enfrentam ou interagem com as questões que estão sendo investigadas. Logo, os diferentes significados que os usuários dão à questão de pesquisa são o foco do pesquisador. Para isto, explora-se a linguagem falada ou escrita dos participantes, sua linguagem gestual e sua interação com os elementos codificados na tecnologia que esteja sob exame. O modo como interagem com a tecnologia, os caminhos interativos que escolhem, os caminhos interativos preferenciais que declinam, as expressões faciais que acompanham a interação, as falas em situações de uso ou em entrevistas e grupos de discussão e o discurso de usuários em fóruns online são exemplos da riqueza de material discursivo que se presta à construção de significados sobre o fenômeno em estudo da perspectiva dos usuários.

Análise sistemática, iterativa e interpretativa do material coletado: Uma vez coletado o material, os métodos qualitativos propõem processos diversos de análise de dados, de modo a identificar categorias de significação principais que possam se articular em uma rede consistente de conhecimentos sobre o objeto em estudo. Todos eles têm em comum o fato de serem procedimentos sistemáticos, com passos bem definidos, que visam fornecer rigor e objetivação à análise. Enquanto a observação cotidiana é opinativa, informal e subjetiva, o uso de métodos qualitativos é um processo que pode ser acompanhado e rastreado, por meio da relação explicita entre os elementos coletados (e.g. trechos de discurso oral, erros de interação com um sistema) e os significados a eles atribuídos. A atribuição de significados é sempre uma atividade intelectual interpretativa, que envolve um processo indutivo (‘*bottom-up*’) e iterativo de categorização do material. De posse do material, o pesquisador segmenta-o em categorias de análise principais agrupadas em torno de significados comuns. A categorização é feita iterativamente e, em geral, as categorias vão ganhando gradativamente um grau de abstração maior em termos de significados atribuídos. Essas categorias serão sempre rastreáveis, correspondendo, em termos descritivos, a, por exemplo, depoimentos de usuários em entrevistas ou a trechos de interação durante uma seção de uso. Ao fim do processo, as diferentes perspectivas dos usuários sobre o fenômeno estudado mostram-se refletidas em um conjunto articulado de categorias.

Um conjunto expressivo e heterogêneo de métodos já foram desenvolvidos segundo as características e finalidades acima apresentadas. Neste capítulo, para dar uma visão geral dos principais tipos, organizamos os métodos mais utilizados em computação em torno de 3 categorias: estudos que coletam dados diretamente junto a usuários, potenciais usuários ou especialistas no domínio; métodos de coleta em contextos reais de uso e métodos de inspeção que não envolvem usuários.

Os **métodos de coleta de dados diretamente junto a usuários ou especialistas** são tradicionalmente usados nas ciências humanas e sociais para a investigação de estados e processos internos que necessitam ser verbalizados pelos participantes aos pesquisadores posto que não são identificáveis (ao menos com facilidade) por meio de observação. São estudos que focalizam preponderantemente (mas não exclusivamente) a linguagem natural, mais comumente por meio do uso de entrevistas e questionários, presencialmente ou à distância [Seidman, 1998; Weiss, 1995; Nicolaci-da-Costa et al., 2004], e da realização os grupos de foco [Lazar et al., 2010]. Este conjunto de métodos vêm sendo intensamente utilizados e adaptados para estudos e pesquisas na área de computação. Dentre eles, destacam-se aqueles que coletam significados de potenciais ou

reais usuários de tecnologias computacionais desenvolvidas ou em projeto e, ainda, os estudos que buscam informações para um projeto junto a pessoas especialistas no domínio da tecnologia. Além disso, a coleta direta junto a usuários também pode ser feita por meio de métodos de avaliação criados na própria área da computação e que envolvem a participação dos usuários (e.g. Método de Avaliação de Comunicabilidade [Prates et al., 2000]; pensar alto [Wright e Monk, 1991]; ou teste de usabilidade [Preece et al., 2015]). Neste capítulo, privilegiamos a apresentação de métodos que envolvem entrevistas e questionários e os grupos de foco, pelo fato de serem menos familiares para alunos e profissionais de computação do que os métodos criados dentro da própria área para coleta de dados junto a usuários.

O segundo tipo de métodos se refere aos **estudos que privilegiam a coleta de dados em contextos reais,** sejam eles ambientes físicos ou virtuais. Por meio de diferentes técnicas (observação, entrevistas, dinâmicas de grupo, análise documental, etc.), os pesquisadores observam usuários em contextos reais, mas também podem interferir na observação por meio de perguntas e breves entrevistas, por exemplo. Este tipo de método inclui os estudos etnográficos [Lazar et al., 2010; Randall e Rouncefield, 2012], os diários de uso [Lazar et al., 2010; Carter e Mankoff, 2005] e os estudos de caso [Lazar et al. 2010; Yin, 2009].

Finalmente, os **métodos de inspeção** referem-se à avaliação de tecnologias computacionais (ou de seus protótipos) por especialistas em passos sistematicamente definidos [Mack e Nielsen, 1994; de Souza et al., 2006].

2.2.3. Procedimentos e Etapas Gerais

Apesar de muito diversos e numerosos, com diferentes técnicas e agrupamentos de técnicas, os métodos qualitativos têm em comum um conjunto de procedimentos e etapas gerais que são apresentados neste capítulo a partir do framework **PRET A Rapporter (PRETAR),** proposto por [Blandford, 2013] como uma estrutura básica para o projeto, a execução e a apresentação dos resultados de estudos qualitativos. As seguintes categorias integram o framework: Objetivos (***P**urpose)*; **R**ecursos disponíveis (***R**esources and constraints)*; Considerações **É**ticas (***E**thical considerations)* e **T**écnicas de coleta de dados (***T**echniques for data gathering)*; Técnica de **A**nálise ***(A**nalysis technique)* e Apresentação dos **R**esultados ***(R**eporting needs)*. A Figura 1 apresenta uma visão geral das categorias do framework e atividades envolvidas em cada uma delas. Em seguida detalhamos cada uma das categorias.

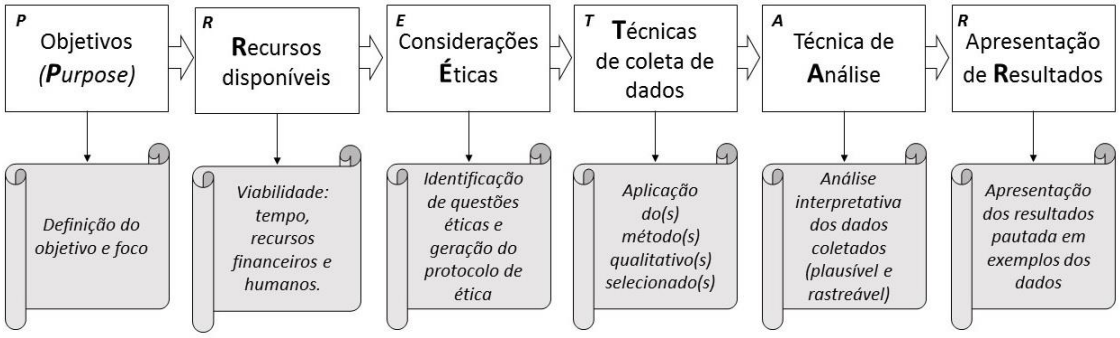

Figura 1 - Visão geral das categorias do Framework PRETAR

Objetivos (*Purpose*): A escolha do método e das técnicas a serem usadas deve estar consistente com o problema a ser investigado. Como já discutido, no caso dos métodos qualitativos, sua adequação volta-se a estudos que têm um objetivo exploratório e focado, que visa a identificação e compreensão em profundidade dos significados e perspectivas sobre uma questão de estudo. Livre de hipóteses sobre o problema e sua solução, o investigador deve ter uma postura de abertura e desconhecimento do objeto de estudo, e deve poder traduzir seus objetivos em uma pergunta aberta (por oposição a perguntas que levam a respostas 'sim' ou 'não') sobre a qual nutra um real desconhecimento sobre a resposta (trata-se de explorar por oposição a confirmar ou refutar suspeitas e preconcepções). Para obtenção de profundidade, este objetivo deve ser nítido e compacto [Nicolaci-da-Costa et al., 2004]. Com isto, evita-se a tentação de saber superficialmente sobre muito, em vez de profundamente sobre pouco.

A questão de estudo deve estar estritamente vinculada a um contexto específico, privilegiando a minúcia do conhecimento contextualizado em detrimento do conhecimento amplo e genérico. Na área de computação, os objetivos dos estudos qualitativos são mais comumente voltados para a identificação de demandas para o desenvolvimento de novas tecnologias e a compreensão a respeito do uso de tecnologias em desenvolvimento ou em estudo.

Recursos disponíveis (*Resources and constraints*): As etapas preliminares de preparação de um estudo envolvem a análise da viabilidade de sua realização considerando o tempo e os recursos financeiros e humanos disponíveis. Quanto aos recursos humanos, é fundamental verificar se há a necessidade da colaboração de especialistas no domínio (e.g. estudos na área de saúde, educação, engenharia aeroespacial, entre outros), bem como se há viabilidade de inserção dos pesquisadores no campo (e.g. autorização para observação etnográfica ou para entrevistas em empresas e para publicação dos resultados coletados). Em termos de tempo, é importante considerar que a pesquisa envolvendo participantes sempre exigirá tempo para imprevistos. Enquanto para o pesquisador, o estudo é o foco prioritário, para os participantes, ele é uma colaboração secundária e, frequentemente, suas reais prioridades sempre poderão surgir para adiar algum compromisso de pesquisa e retardar o cronograma. Esse tempo deve estar previsto no cronograma para evitar pressões desnecessárias sobre os usuários, prejudicando o caráter espontâneo dos dados coletados.

O recrutamento dos participantes é de importância vital para os estudos e, diferentemente dos estudos estatísticos, não segue os critérios de aleatoriedade e de representatividade. As amostras são chamadas intencionais ou propositais, pois estão referidas ao foco nítido e restrito da pesquisa e ao contexto em que ela ocorrerá. São também reduzidas, de modo a permitir o exame aprofundado dos problemas e fatores envolvidos. Comumente, o recrutamento é realizado pela técnica de amostragem em bola de neve, na qual o pesquisador convida os próximos participantes por meio de indicação dos participantes anteriores [Weiss, 1995].

O tamanho da amostra varia. Estudos de caso e estudos etnográficos podem, por exemplo, focalizar a imersão em um único contexto, ou até mesmo em um único participante/local de estudo (e.g. um estudo para observar o processo de desenvolvimento de uma tecnologia em uma determinada empresa, ou um processo de design participativo para o desenvolvimento de uma tecnologia assistiva para um único participante). Diferentemente, um estudo pode considerar a observação de atendimentos de saúde em

diferentes locais (duas ou mais clínicas). De modo análogo, o número de entrevistas qualitativas necessário a uma pesquisa também é definido caso a caso. Algumas vezes, em face da especificidade do tema, apenas poucas entrevistas são possíveis, considerando a *expertise* dos entrevistados (e.g. estudos que envolvem o desenvolvimento de tecnologias de alto grau de especialidade). É mais frequente, contudo, o uso da técnica de saturação [Weiss, 1995; Seidman, 1998; Blandford, 2013], na qual as entrevistas são realizadas até que o conteúdo das mesmas seja recorrente e não ofereça novos *insights* e avanços aos pesquisadores.

O perfil da amostra dos estudos qualitativos também varia quanto à homogeneidade. Alguns estudos adotam a alta-definição [Nicolaci-da-Costa et al., 2004], ou seja, selecionam participantes com características semelhantes, de maneira a facilitar o aprofundamento do conhecimento a respeito daquele perfil. O estudo de uso de uma determinada rede social, por exemplo, certamente será mais minucioso se voltado para um tipo específico de usuários, considerando a enorme diferença e possibilidades de uso da ferramenta. A variação, nesse caso, levaria à superficialidade de conhecimento, enquanto a homogeneidade permitirá o aprofundamento. Há casos, no entanto, em que a amostra de alta variação [Weiss, 1995] é desejável ou necessária. Estudos etnográficos [Lazar et al., 2010; Randall e Roucefield, 2012], por exemplo, costumam levar o pesquisador a travar contato com um grupo heterogêneo de pessoas voltados para um objetivo comum, inexistindo, inclusive, um número razoável de pessoas de perfil semelhante no mesmo grupo. Em um fórum de discussão online sobre antitabagismo, por exemplo, em geral, é formado por pessoas com um perfil variado que interagem em torno do objetivo de parar de fumar. Nesse caso, é o objetivo da interação mais do que o perfil dos participantes o critério do recrutamento. É importante chamar atenção, no entanto, que, diferentemente dos estudos quantitativos comparativos, é bastante arriscada a realização de estudos qualitativos que comparem dois ou mais perfis de usuários. Isto por que a profundidade da análise qualitativa requer tempo e saturação. Somente estudos com muitos recursos humanos e de tempo conseguem aliar profundidade e comparabilidade entre grupos sem incorrer na superficialidade de resultados.

Considerações Éticas (*Ethical Considerations*): Qualquer estudo envolvendo seres humanos (qualitativo ou não) exige a reflexão sobre os riscos éticos envolvidos no estudo e sobre as ações envolvidas para minimizar ou tratar esses aspectos. As questões éticas são mais trabalhadas no contexto dos métodos qualitativos pelo fato de a inserção no campo e o contato com os participantes serem mais intensos e duradouros e a investigação, mais profunda. Em todos os casos, no entanto, a análise ética deve ser feita desde as etapas iniciais de planejamento do estudo, seja ele um estudo científico ou de finalidade prática e profissional. No caso de pesquisas científicas, as considerações éticas são uma exigência legal e, no Brasil, seguem as Resoluções CNS nº 466, de 12 de dezembro de 2012 [Conselho Nacional de Saúde, 2012] e CNS nº 510, de 07 de abril de 2016 [Conselho Nacional de Saúde, 2016], que, embora elaboradas pelo Conselho Nacional de Saúde, se aplicam às pesquisas em computação que envolvem seres humanos ou dados dos mesmos. À exceção de pesquisas com informações de acesso público ou com bancos de dados cujas informações são agregadas e sem possibilidade de identificação individual, todas as pesquisas em IHC devem seguir as Resoluções. A primeira, mais detalhada e mais rigorosa, regulamenta as pesquisas de IHC cujo domínio está relacionado à área de saúde (e.g. pesquisas sobre tecnologias para apoio a tratamento de autistas, pacientes diversos em hospitais, etc.). Já a segunda resolução complementa a

anterior e cria algumas exceções para as pesquisas em outros domínios, se baseando nos critérios relacionados às especificidades metodológicas da investigação no contexto humano, cultural e social. Ambas as resoluções exigem a preparação de um Protocolo de Ética detalhado para apresentação e aprovação da instituição de pesquisa responsável. Além da exigência legal, elas são um guia passo-a-passo do que um pesquisador deve fazer para que seu estudo trate com o devido cuidado dos impactos de todas as fases de execução em seus participantes. Devem, portanto, ser de conhecimento de todos que desejem fazer pesquisa qualitativa.

Alguns aspectos merecem ser destacados. O primeiro refere-se ao fato de que a pesquisa envolve seres humanos não apenas quando lida diretamente com participantes. A análise de material gerado por pessoas também gera impactos sobre as mesmas quando de sua divulgação. É o caso, por exemplo, da análise de material discursivo de fóruns virtuais, com alguma proteção de acesso, ou da instrumentação de sistemas de uma determinada empresa. O segundo aspecto refere-se aos cuidados com a preservação do anonimato e da livre-participação, o que envolve explicações detalhadas sobre os objetivos e etapas da pesquisa desde o primeiro convite e coleta de consentimento em cada etapa de estudo, considerando o contexto de sua realização (presencial ou a distância). Finalmente, destaca-se a importância dos cuidados adicionais em pesquisas que trabalham com grupos chamados vulneráveis, ou seja, crianças, pessoas com problemas psicológicos ou psiquiátricos, alunos em pesquisas durante suas atividades de aprendizado, entre outras. Pesquisas com esses grupos, posto que mais sensíveis, devem ser conduzidas ou supervisionadas por profissionais experientes na prevenção e também na correção de problemas emergentes na fase de coleta de dados, e não raro devem contar com a parceria de profissionais especialistas. Além disso, devem ter o consentimento dos responsáveis dos participantes e, no caso de alunos, não devem interferir no andamento de aulas e cursos, nem fazer parte das atividades do mesmo (o aluno deve poder se recusar a participar da pesquisa e, ainda assim, desenvolver as atividades das aulas ou do curso).

Técnicas de coleta de dados (*Techniques for data gathering*): A definição das técnicas e instrumentos para a coleta de dados qualitativos variam quanto aos objetivos e às limitações do contexto. Quando os objetivos do estudo são mais voltados para aspectos observáveis do comportamento humano, tais como atividades e processos de trabalho e uso de tecnologias, as técnicas mais próximas da observação naturalística mostram-se bastante adequadas (estudos etnográficos e observação participante, seções de design participativo, etc.). Por outro lado, para a investigação de significados sobre processos internos que não são identificáveis pela observação ou o uso de tecnologias ainda não disponíveis em cenários reais, outras técnicas se mostram mais eficazes. Diários de uso, entrevistas e grupos de foco são algumas dessas modalidades. As entrevistas são especialmente interessantes para a investigação de significados quando a perspectiva individual é o foco. Já os grupos de foco revelam-se ferramenta interessante para a construção de significados coletivos, para os quais a influência mútua de opiniões é bem-vinda. Diários de uso trazem vantagens para a construção de significados em um intervalo mais longo de tempo, enquanto entrevistas e grupos de foco são mais comumente registro de um único episódio de coleta de dados.

Independentemente da técnica escolhida, a preservação do foco na questão investigada deve ser um cuidado constante. Isto porque a riqueza e a espontaneidade dos dados e significados coletados em estudos qualitativos são grandes, tornando-se extremamente comum que o pesquisador trave contato com dados que, apesar de

interessantes, não constituem exatamente o foco da pesquisa. A disciplina para manter a nitidez do olhar no foco da observação é importante para escapar da coleta abrangente, porém superficial, e para atingir a profundidade de significados bem recortados em torno da questão de estudo.

Técnicas de Análise de dados (*Analysis techniques*): Mais usualmente, os estudos qualitativos trabalham com significados coletados através de discurso oral ou escrito, de interações presenciais (que mesclam discurso e comportamento não textual – gestos, movimentos de grupo), de interações mediadas por tecnologia ou de interações com tecnologia. Em todos os casos, o material deve ser registrado visando sua manipulação iterativa, dado que é o caráter sistemático e repetido de análise dos significados que faz emergir novos conteúdos e achados, muito distintos das primeiras e superficiais observações que o pesquisador tem enquanto coleta o material.

Em uma primeira etapa, o material coletado deve ser disponibilizado de forma a ser manipulado repetidamente. Vídeos passíveis de notação, transcrições de áudios e anotações de campo e outros materiais devem ser agrupados, e o pesquisador deve ganhar bastante familiaridade com os mesmos. Procede-se, então, à categorização ou codificação do material. Esta análise pode ser *top-down* ou *bottom-up.* Análise *top-down* é aquela que parte de categorias pré-definidas para analisar o material coletado. Em geral, essas categorias são oriundas da teoria utilizada pelo pesquisador para construção da pesquisa. Já a análise *bottom-up* segue o raciocínio indutivo e constrói as categorias de análise a partir do próprio material coletado. Independentemente da abordagem, o processo busca que o pesquisador categorize e interprete o material de modo que sua reorganização faça emergir novos e surpreendentes significados.

O pesquisador começa o processo de análise segmentada por meio da construção de categorias iniciais que organizem os dados. Cada categoria tem a ela associado alguns trechos do material (trechos do discurso de entrevista, trechos da interação com a tecnologia, etc.), fornecendo plausabilidade e rastreabilidade à interpretação do pesquisador. A identificação das categorias e a definição de suas características correspondem ao que Corbin e Strauss [2008] definem como **codificação aberta**. Esse processo envolve as atividades de analisar, segmentar, extrair, examinar, comparar, conceituar e categorizar os dados. Já o segundo processo refere-se à **codificação axial [**Corbin e Strauss, 2008], que aprimora e articula as categorias resultantes da codificação aberta. As categorias iniciais devem se agrupar, então, em torno de temas comuns. Categorias com mais recorrências são priorizadas sem, no entanto, descartar o significado único e singular, por vezes, importante para a pesquisa. Assim, o pesquisador seleciona as categorias mais relevantes e as coloca como referência central para estabelecer relações entre as demais categorias e subcategorias. Os dados são, portanto, agrupados através das conexões entre as categorias. Novas etapas de análise segmentada se sucedem, cada uma reorganizando a precedente e aumentando seu nível de abstração. Análises verticais do material de cada participante se mesclam a análises horizontais que comparam os dados entre participantes [Nicolaci-da-Costa et al., 2004]. Relações e conexões entre temas e categorias devem ser estabelecidas e uma síntese interpretativa do material realizada. Análise segmentada e iterativa do material, seguida de síntese e articulação dos temas e categorias compõem, portanto, um processo de interpretação sistemática que é fortemente mediado pelo entrevistador (por sua bagagem teórica e profissional), mas ancorado em

Tabela 2. 2: PRET A Rapporter - Conteúdo sintético [Blandford, 2013]

	Procedimentos / Etapas	**Observações**
Objetivo (***Purpo***se)	Definição de objetivos e questão de estudo	Pergunta aberta e contextualizada Foco nítido e compacto
Recursos disponíveis	Recursos financeiros, humanos e de tempo Recrutamento de participantes	Trabalho intensivo com pequenas amostras intencionais Amostra homogênea ou de alta-variação, com tamanho definido por saturação
Considerações **É**ticas	Análise dos impactos éticos Elaboração e Aprovação de Protocolo de Ética	Populações vulneráveis Anonimato e consentimento livre e esclarecido Resoluções do Conselho Nacional de Saúde
Técnicas de coleta de dados	Técnicas vinculadas ao contexto natural de ocorrência	Abordagens etnográficas para comportamentos observáveis Entrevistas e Grupos de Foco para processos internos não-observáveis
Técnicas de **A**nálise de dados	Análise de dados indutiva (*bottom-up*) ou guiada por teoria (*top-down*)	Análise iterativa e segmentada (categorias e temas em codificação aberta Síntese das categorias e temas (codificação axial) Rastreabilidade da interpretação (articulação categorias – material empírico)
Apresentação dos **R**esultados	Processo descritivo e com exemplos de dados	Equilíbrio entre nível de abstração e rastreabilidade da interpretação

material empírico, com articulações objetiváveis e explicitas entre dado e categoria interpretativa.

Apresentação dos Resultados (*Reporting needs*): Em termos mais gerais, a apresentação dos resultados dos estudos qualitativos segue as recomendações de cientificidade vigentes, embora enfatize mais a descrição do processo do que como ocorre em estudos quantitativo-experimentais [Creswell, 2009].O uso extensivo de exemplos retirados do material empírico é fundamental para tornar clara a relação objetiva entre códigos/categorias interpretativas e dado (discurso ou elementos codificados na tecnologia analisada, por exemplo). Esta rastreabilidade, essencial para a validação da pesquisa, torna o relato mais extenso e minucioso. Por outro lado, excesso de descrições e exemplos, podem tirar a visão de conjunto e desviar o leitor da mensagem central, ou seja, da contribuição essencial do trabalho. Por isso, um dos desafios do relato de estudos qualitativos é o de deixar claras a articulação dos temas e a contribuição inovadora da pesquisa, equilibrando o nível de abstração do relato das categorias e os exemplos que fornecem rastreabilidade às mesmas.

Na Tabela 2.2, é exposta uma síntese das principais questões e preocupações comuns aos métodos qualitativos e abordadas a partir do framework PRETAR [Blandford, 2013] descrito. Na seção seguinte, os procedimentos específicos de alguns métodos selecionados são abordados, destacando-se as vantagens e limites envolvidos na aplicação de cada um deles. Esses métodos estarão divididos em torno das 3 categorias anteriormente mencionadas: estudos que coletam dados diretamente junto a usuários, potenciais usuários ou especialistas no domínio; métodos de coleta em contextos reais de uso e métodos de inspeção que não envolvem usuários. Para cada uma das 3 categorias, daremos pelo menos um exemplo de uma aplicação (pesquisa).

2.3. Métodos de Coleta de Dados Diretamente Junto a Usuários ou Especialistas

Métodos de coleta de dados diretamente junto a usuários ou especialistas no domínio são métodos que permitem que os participantes dos estudos expressem suas opiniões, perspectivas, atitudes e experiências. No entanto, em todos estes métodos **o que** perguntar, **a quem** e **como** são fundamentais para garantir que se tenha dados relevantes para o seu objetivo. Se as decisões sobre o instrumento de coleta, a seleção de participantes ou o método selecionado não forem bem fundamentadas pode-se acabar com uma grande quantidade de dados que não conseguem atender ao objetivo que motivou a coleta. Em outras palavras, pode-se terminar em uma situação de '*garbage in, garbage out'* que terá pouca utilidade, e tipicamente envolve um custo de preparação, aplicação e análise não desprezível.

Nesta seção apresentamos os métodos de entrevista, questionários e grupos de foco. Cada uma dessas técnicas apresenta suas especificidades e são apresentadas isoladamente. Vale ressaltar, no entanto, que elas podem ser usadas de modo combinado. Questionários podem ser distribuídos antes da realização de entrevistas, para coletar dados mais objetivos e informações, ou antes de grupos de foco, para a coleta mais individualizadas de dados dos participantes do grupo. Além disso, em qualquer um desses métodos, o pesquisador pode considerar o uso de **materiais de estímulo**, isto é, materiais como vídeos, textos ou quadrinhos dentre outros que possam estimular a participação. Normalmente, eles podem ser usados para quebrar o gelo, estimular discussões ou reflexões, como meio de levantar questões difíceis ou mesmo para oferecer um referencial comparativo para os participantes.

2.3.1. Entrevistas

Entrevistas permitem que se tenha uma comunicação direta com pessoas que sejam de interesse para pesquisas ou projeto e avaliação de sistemas. Mostram-se especialmente úteis para coletar significados relacionados a aspectos não capturáveis pela observação direta, ou seja, ligados a processos internos [Seidman, 1998] tais como motivação, desejos, receios, desagrados ou pontos de vista a respeito de sistemas interativos. Se por um lado a coleta direta junto ao usuário traz como vantagem a captura de aspectos não-observáveis, as entrevistas têm como limites a sinceridade e o grau de consciência do usuário sobre um determinado assunto (e.g. o usuário pode omitir determinada opinião para agradar o entrevistador ou pode não ter consciência de uma limitação sua ao avaliar uma tecnologia).

As entrevistas podem ser realizadas presencialmente ou à distância. Neste último caso, pode-se adotar a conversa por áudio ou bate-papo digital. Embora a entrevista por

bate-papo facilite o registro do discurso dos participantes (posto que gera registro escrito do material), o discurso oral tem por características o menor controle e censura e o maior detalhamento e minúcia, gerando, em geral, um material mais rico. A possibilidade de encontrar presencialmente os participantes é fator decisivo para a definição do ambiente para a entrevista (virtual ou real) e sobre a forma de registro (áudio ou texto).

Entrevistas normalmente são classificadas em livres (ou não estruturadas), semiestruturadas ou estruturadas, de acordo com a liberdade que o entrevistador tem de explorar (aprofundando ou definindo novas direções) o roteiro pré-definido. As **entrevistas livres** não seguem nenhum roteiro pré-estabelecido, sendo guiadas apenas pela questão de pesquisa. São especialmente adequadas para estudos exploratórios preliminares, onde sabe-se muito pouco sobre o tema, em particular entrevistas de sondagem com especialistas no domínio de investigação ou com potenciais usuários. Este tipo de entrevista costuma ser usado também como instrumento de pesquisa-piloto para informar a construção de um roteiro definitivo de entrevista. No outro extremo, temos as **entrevistas estruturadas** [Blandford, 2013], cujos tópicos obedecem uma definição e uma sequência rígida de formulação, similar a um questionário. A vantagem desse tipo de roteiro é seu alto potencial de comparabilidade entre as respostas dos participantes, mas sua desvantagem é a restrição à coleta de significados espontâneos e não previstos pelo pesquisador, mas considerados relevantes pelo entrevistado em um fluxo de conversa mais natural. **Entrevistas semiestruturadas** [Seidman, 1998; Nicolaci-da-Costa et al., 2004; Blandford, 2013] são mais comumente utilizadas nas pesquisas da área de computação por buscarem conciliar um certo grau de comparabilidade entre o discurso dos participantes e um espaço para a espontaneidade na emergência de significados e categorias não previstas. Servem-se de um roteiro prévio mas obedecem um fluxo espontâneo de conversa. No que se segue, serão enfatizadas as características e etapas de estudos qualitativos que fazem uso de entrevistas semiestruturadas.

Tal como nos demais estudos qualitativos, após a definição dos objetivos e questão de estudo (aberta e exploratória), o **perfil dos participantes e recrutamento da amostra** é organizado. É possível construir uma **amostra homogênea**, como nos estudos que usam o Método de Explicitação do Discurso Subjacente – MEDS [Nicolaci-da-Costa et al., 2004], de modo a maximizar as características comuns dos participantes (e.g. o uso de um site de compras por jovens rapazes, entre 18 a 21 anos, atletas de basquete). Pode-se, também, dependendo dos objetivos, construir uma **amostra de alta-variação** [Seidman, 1998], maximizando as características dos usuários, mas unindo-os por um fio condutor específico (e.g. uso de uma ferramenta de agenda eletrônica por uma equipe de trabalho com funcionários de diferentes níveis hierárquicos). Como exposto na seção 2.2.3, o tamanho da amostra é usualmente definido pela saturação dos temas levantados. Quando as entrevistas param de trazer novas perspectivas, o número de recrutados é considerado suficiente.

Tendo concluída a definição do perfil de participantes, passa-se **à elaboração do roteiro de entrevista**. Como dito anteriormente, entrevistas livres com usuário de perfil semelhante ao escolhido ou com especialistas podem ajudar na construção de um roteiro de boa qualidade. Após sua elaboração, esse roteiro deve ser testado e aperfeiçoado através de uma aplicação–piloto. O roteiro apresenta itens abertos a serem usados como lembretes dos tópicos a serem cobertos pela entrevista. Evita-se, com este roteiro. a leitura artificial de perguntas prontas diante do entrevistado, priorizando-se que o entrevistador formule a pergunta em tempo real. Todos os itens previstos no roteiro devem integrar a

entrevista, mas a ordem e a forma em que isso é feito deve seguir o fluxo natural da conversa com cada entrevistado. Cada roteiro tem sua lógica própria, mas algumas orientações gerais podem apoiar a realização de uma entrevista com a qualidade desejada:

- O roteiro deve prever uma introdução que reassegura os cuidados éticos com o entrevistado, informa os objetivos da pesquisa e como os resultados serão divulgados.
- Perguntas de abertura devem ser feitas a título de quebra-gelo, visando deixar o entrevistado à vontade.
- Os itens-chave da entrevista devem ser estar definidos em tópicos com base em uma lógica que ajude a organizar o raciocínio e o discurso do participante, permitindo que ele compreenda e mantenha o foco da conversação. Perguntas do geral para o específico, ou que sigam uma sequência cronológica, por exemplo, motivam um fluxo natural de conversação. Esta lógica deve, no entanto, deixar espaço para quebras de sequência, caso o entrevistado espontaneamente encaminhe seu fluxo discursivo em outra direção.
- Todos os tópicos do roteiro devem ser abordados em prol da comparabilidade.
- Perguntas de fechamento, que levem o entrevistado à desaceleração e ao bem-estar devem ser inseridas na conversação.
- Um espaço para comentários livres e para dúvidas deve encerrar a entrevista.

A **condução das entrevistas** deve se dar em local e tempo de preferência do usuário, com privacidade e sem observadores, visando a configuração de um bate-papo espontâneo e informal segundo a perspectiva dos participantes. O pesquisador deverá estar atento também ao código de etiqueta referente a cada entrevistado. Jovens, por exemplo, costumam ser mais informais, tanto na linguagem e forma de relacionamento, quanto nos locais escolhidos para a entrevista. Há, contudo, entrevistas mais formais em locais de trabalho que exigem tratamento mais formal e, inclusive, cuidados especiais com vestimenta.

Antes de iniciar a entrevista, deve-se proceder à coletado consentimento previsto na legislação brasileira [Conselho Nacional de Saúde, 2012, 2016]. Embora seja mais usual a condução de uma única entrevista por participante, como nos estudos com o MEDS [Nicolaci-da-Costa et al., 2004], é também possível realizar mais de uma entrevista por participante [Seidman, 1998], quando o objetivo incluir um maior aprofundamento de experiências ou uma coleta de dados em etapas (e.g. a percepção da evolução de uso de uma determinada tecnologia). É importante lembrar que, seguindo a metodologia qualitativa, as entrevistas buscam coletar dados em profundidade, levando, em média 1 hora para coletar os significados do entrevistado, com cautela e minúcia.

Após a **transcrição e organização do material registrado** com o apoio de diferentes ferramentas de apoio e mídias (gravações em áudio ou em vídeo, material registrado em ferramentas de bate-papo a distância, etc.), recomenda-se que a **leitura livre da íntegra dos conteúdos das entrevistas**, de modo a obter visão de conjunto e familiaridade com os dados.

Em seguida, inicia-se o trabalho de análise sobre o material discursivo registrado, seguindo a modalidade de **análise segmentada e iterativa do material discursivo** sucintamente descrita na seção 2.2.3. A etapa de identificação das primeiras categorias em geral se apoia nos próprios itens do roteiro. Cada item do roteiro de uma entrevista representa uma categoria de análise e subcategorias em torno das mesmas são

identificadas dentro de cada tópico da entrevista. Após essa categorização inicial, com baixo nível de abstração, inicia-se a construção de categorias/códigos com base em todas as respostas de todos os participantes para cada tópico do roteiro. Ao trabalhar com categorias, o pesquisador classifica trechos de entrevistas a partir da fala dos participantes ou a partir de categorias teóricas pré-definidas. Busca-se, então, a identificação das categorias recorrentes em todo o material, agrupando-as em temas de maior grau de abstração [Seidman, 1998, Nicolaci-da-Costa et al., 2004, Blandford, 2013]. Gradativamente novos trechos podem ser encaixados nas categorias/temas que já foram criados ou pode ser necessário criar mais temas. Através desta técnica, o pesquisador pode perceber o que há de comum nas experiências dos participantes entre si, bem como o que há de comum na fala deles com a literatura especializada no tema da pesquisa. Quando uma categoria nova surge, o pesquisador deve ter em mente que ela pode se tornar irrelevante ou importante ao final do estudo, pois somente depois de todas as entrevistas serem categorizadas, será possível distinguir as categorias importantes das irrelevantes. Essa análise interparticipantes é, portanto, iterativa, e tem como objetivo alcançar gradativamente maiores graus de abstração e a articulação das categorias em torno de temas mais amplos [Corbin e Strauss, 2008].

Complementarmente, como propõe o MEDS [Nicolaci-da-Costa et al., 2004], é interessante realizar a análise intraparticipante, que busca verificar conflitos e contradições no interior de uma mesma entrevista (e.g. quando facilidades e obstáculos na adoção de uma tecnologia são identificados pelo mesmo participante). Essas contradições mostram impasses entre discurso e prática e podem ser excelentes indicadores para a pesquisa.

As categorias de análise podem estar relacionadas a significados expressos diretamente pelos conteúdos discursivos (e.g. depoimento sobre os pontos negativos de uma determinada tecnologia que está sendo testada) ou ligadas à expressão de significados pela forma com que o depoimento foi emitido (e.g. silêncios e longas pausas entre os participantes quando indagados sobre as vantagens de uma determinada tecnologia podem indicar desaprovação implícita quanto ao seu uso).

O Quadro 2.1 apresenta brevemente um exemplo de pesquisa qualitativa que faz uso de entrevistas para a coleta de dados.

Quadro 2.1: Exemplo de Entrevista Semiestruturada [Jamison-Powell et al., 2016]

Objetivo: Explorar atitudes em relação à comunicação póstuma online sob a perspectiva das pessoas em luto.

Motivação: Atualmente existem sistemas que oferecem serviços de envio de mensagens póstumas, ou seja, a pessoa deixa mensagens para serem enviadas depois que elas falecerem. No entanto, pouco se sabe sobre como este serviço é visto por quem recebe as mensagens.

Métodos: Entrevistas semiestruturadas

Desafios e limitações: A entrevista não aborda uma experiência real vivenciada pelo participante, mas pede que ele preveja como se sentiria em determinados contextos. Embora esta decisão seja uma limitação para a pesquisa, ela foi tomada devido à dificuldade ética de saber quanto tempo depois da morte de um ente querido seria apropriado entrevistar uma pessoa em luto sobre aspectos de comunicação póstuma.

Critério de recrutamento: Seleção de usuários do Facebook que não tivessem vivenciado de fato a recepção de uma mensagem póstuma. Este segundo critério foi por uma decisão tanto prática, quanto ética.

Número de participantes:14 (7 mulheres e 7 homens)

Perfil dos participantes: Participantes eram funcionários e alunos de uma universidade na Grã-Bretanha, usuários de Facebook e não conheciam sistemas de mensagem póstuma (apenas 2 conheciam).

Material de estímulo: Vídeo apresentando o sistema DeadSoci.al que permite que se programe o envio de mensagens póstumas em sistemas de redes sociais como Facebook (vídeo disponível em: http://youtu.be/mnwX3O902xQ)

Duração das entrevistas: 20 a 40 minutos

Estrutura do roteiro de entrevista: Perguntou aos participantes o que entendiam de sobre os serviços póstumos (como o mostrado no vídeo), e em seguida exploraram reações hipotéticas ao recebimento de mensagens póstumas, e questões associadas os usuários (que vão deixar as mensagens), receptores (pessoas em luto) e os responsáveis pelo processo.

Registro: As entrevistas foram gravadas em áudio e transcritas.

Análise: Foi conduzida uma análise temática a partir das transcrições, através da codificação aberta e sua organização temática através de codificação axial. Foram produzidos 28 códigos, que foram então reduzidos a 19 categorias.

Resultados: Foram identificados 5 temas: *Preparo, Controle, Conectando, Artificialidade e Mortalidade.* Preparo e Controle estavam fortemente relacionados com o processo de luto e endereçam a questão de como mensagens póstumas podem ajudar e influenciar o processo de aceitação da perda de um ente querido. Os outros 3 temas estão associados com novas formas da sociedade lidar com a morte. Com base nos resultados da entrevista, os autores delineiam 5 conflitos que devem ser tratados no projeto de sistemas que pretendam facilitar o envio de mensagens póstumas.

2.3.2. Questionários

O questionário é um instrumento de coleta de dados por escrito, que pode ser usado em pesquisas qualitativas e quantitativas, para coletar a opinião, medir atitudes e levantar dados e significados sobre um determinado fenômeno por meio de um conjunto coerente de perguntas ou afirmativas [Lazar et al., 2010]. Diferencia-se da entrevista por envolver uma **relação não-interativa** entre participante com o pesquisador. O questionário pode ter distribuição **síncrona ou assíncrona** (à distância). Quando síncrona, ainda que o pesquisador tenha contato com os participantes, as informações e significados obtidos são resultado do preenchimento individualizado do instrumento, sem intervenção e mediação do pesquisador (salvo para tirar pequenas dúvidas acerca do instrumento).

Por não necessitar de interação e mediação, os questionários são especialmente úteis quando o contato dos pesquisadores com os participantes é difícil. São também vantajosos pelo baixo investimento de tempo de aplicação e pelo registro já escrito do material discursivo, poupando bastante trabalho por parte dos pesquisadores. Isso acaba

por permitir que um número maior de questionários seja distribuído e analisado, quando em comparação com as entrevistas. Por outro lado, os dados coletados são bem mais superficiais do que aqueles coletados em entrevista. Isto porque os participantes, em geral, escrevem menos do que falam, controlam e censuram mais o discurso escrito do que o falado (perdendo em espontaneidade) e não podem ser estimulados a aprofundar, clarificar e comentar suas respostas ao questionário (ao passo que o bate-papo interativo da entrevista favorece a exploração do discurso em tempo real). Por essa razão é mais frequentemente usado como instrumento complementar de coleta de dados, ao lado de outras técnicas que aprofundam a investigação. Quando o objetivo é fazer um levantamento preliminar do campo de estudos, até mesmo dar insumos ao melhor delineamento do foco e da questão de pesquisa, a maior abrangência e agilidade podem ser critérios decisivos para a adoção do questionário como instrumento complementar na pesquisa qualitativa. De outro modo, a decisão pelo uso do questionário deve ponderar a chance de uma maior participação, porém com resultados mais superficiais. Acrescenta-se a isso o nível de sensibilidade do material a ser coletado. Quando a pesquisa envolve temas mais delicados (e.g. aspectos ligados à privacidade ou conteúdo da vida particular), a entrevista com mediação e presença zelosa do pesquisador pode ser mais recomendada. No entanto, um desafio no uso de questionários é se conseguir uma taxa de retorno satisfatória, já que tipicamente esta taxa costuma ser baixa (segundo Preece et al. [2015], uma taxa de retorno de 40% é considerada boa).

A orientação quantitativa ou qualitativa do questionário é outra questão importante a ser tratada. Quando o questionário é construído com perguntas fechadas ou é uma escala de atitudes (e.g. do tipo Likert) e se busca uma análise estatística dos resultados coletados, a pesquisa deve ser, desde a sua origem, projetada nos moldes quantitativos. Segue, assim, a metodologia dos *surveys* [Lazar et al., 2010], com amostras probabilísticas, representatividade da população estudada e a análise estatística sofisticada dos dados coletados, indo muito além de cálculos simplistas de percentuais e médias das respostas. Um exemplo de questionário de enfoque quantitativo bastante utilizado na área da computação é o *Questionnaire of User Interface Satisfaction* (QUIS)[1], elaborado para mensurar a satisfação dos usuários de sistemas interativos [Chin et al., 1988].

Considerando o foco deste capítulo nas abordagens qualitativas, é fundamental ressaltar que o uso de questionários em pesquisas que seguem esta abordagem deve enfatizar procedimentos de coleta e análise de dados que facilitem a exploração e construção de significados em torno do fenômeno pesquisado (e.g. questões de perguntas abertas). É por vezes desejável o uso de questões com alternativas fechadas, para a coleta de dados mais objetivos (e.g. intensidade de uso de uma tecnologia, nível de formação escolar e acadêmica e outros dados acerca do perfil de participantes). Além disso, perguntas fechadas com alternativas de 'sim' e 'não' funcionam comumente como uma boa introdução a questões abertas para exploração de razões e processos em relação a um fenômeno (e.g. por que o participante não utiliza uma funcionalidade qualquer da tecnologia sob exame). Porém, o recrutamento dos participantes segue os procedimentos de amostra proposital (de alta variação ou homogênea) e a análise dos dados deve ser baseada nos procedimentos descritos na seção 2.2.3 [Corbin e Strauss, 2008; Nicolaci-da-Costa et al., 2004]. Abordagens mistas (quantitativas e qualitativas) são interessantes, porém sofisticadas, exigindo que o pesquisador tenha sólida formação dentro dos dois

[1] QUIS – disponível em: http://lap.umd.edu/quis/ (Última visita em maio de 2017).

paradigmas [Creswell, 2009].

A elaboração do questionário propriamente dito é bem similar ao da construção das entrevistas, em particular das estruturadas. O pesquisador deve elaborar perguntas fixas e agrupá-las em uma sequência coerente, de modo a conduzir o participante em um fluxo de raciocínio direto e encadeado. Diferentemente de um roteiro de entrevista, essas perguntas devem ser previamente enunciadas de modo claro e padronizado para todos os participantes, buscando linguagem próxima ao contexto do grupo. O tempo de preenchimento deve ser previamente testado, assim como a adequação das perguntas, por meio de estudo-piloto. Quando da coleta, os mesmos cuidados recomendados para as entrevistas devem ser tomados: explicitação dos objetivos, solicitação de consentimento, suporte aos participantes, etc.

2.3.3. Grupos de foco

Grupos de foco ou grupos focais envolvem comunicações diretas, não com uma pessoa por vez, mas com várias pessoas de cada vez. O grupo de foco permite que se crie um ambiente para que os vários participantes discutam pontos de interesse definidos em um roteiro. A premissa é que indivíduos desenvolvem suas opiniões em um contexto social através da comunicação com outros, assim, a partir desta discussão, obtém-se não apenas a opinião dos participantes sobre o tema, mas também pontos de consenso ou conflitos entre os participantes do grupo. Por este motivo, grupo de foco é um método mais apropriado para quando se deseja investigar questões relevantes para um grupo ou comunidade do que quando o foco é em experiências individuais.

Como nos demais métodos de coleta de dados diretamente junto ao participante, grupos de foco requerem que se defina o perfil desejado dos participantes e o **tamanho do grupo**. Não existe um número mágico ideal de participantes para um grupo de foco. No entanto, alguns autores acreditam que um grupo de 3 a 8 participantes tende a ser mais eficaz [Cairns e Cox, 2008; Barbour, 2009], ainda que outros definam que o grupo possa chegar até 10 ou 12 pessoas. No entanto, ao definir o tamanho do grupo, o pesquisador deve pensar como este número impacta a dinâmica da discussão. Grupos muito grandes podem gerar conversas paralelas em subgrupos. Por outro lado, pode ser difícil manter o fluxo da discussão se o grupo for muito pequeno [Cairns e Cox, 2008].

O pesquisador deve também definir quantos grupos focais realizar. De novo, não há um **número de grupos** correto, e a decisão deve ser tomada no contexto da investigação. O pesquisador deve considerar que um grupo pode ter características particulares, e que a inclusão de mais grupos pode permitir que se faça afirmações sobre padrões que não sejam específicos à dinâmica de um grupo específico. Pode ser boa ideia deixar espaço para adicionar novos grupos, caso se mostre interessante fazer a comparação intergrupos [Barbour, 2009]. Em casos de pesquisa, a técnica de saturação pode ser usada para definir a quantidade de grupos.

Em relação aos participantes, o pesquisador deve definir não apenas o perfil de cada participante, mas também como deve ser a **composição do grupo**. A composição do grupo pode ter um grande impacto na dinâmica da discussão e, logo, nos dados coletados [Blandford, 2013]. O grupo pode ser definido como homogêneo ou heterogêneo [Lazar et al., 2010]. O **grupo homogêneo** permite que se colete dados que representem uma visão ou contexto compartilhado do perfil definido. Além disso, o compartilhamento de experiências comuns pode facilitar e promover a discussão entre participantes. Por sua

vez, o **grupo heterogêneo** pode ser interessante quando se pretende coletar múltiplas perspectivas (por exemplo, quando se vai desenvolver um sistema para um conjunto amplo de perfis de usuários) ou o objetivo envolve identificar conflitos ou consensos entre participantes com perspectivas distintas. Ainda assim, os participantes devem ter algum interesse no tópico em questão e deve-se pensar em pontos em comum entre os participantes que possam facilitar a discussão. Por exemplo, em um ambiente universitário, se o foco for investigar o uso de um sistema acadêmico que apoia o ensino (e.g. Moodle) seria mais interessante ter dois grupos homogêneos, um para os alunos e outro para os professores, já que a visão que cada um destes grupos tem do sistema e o uso que fazem dele são diferentes. No entanto, se o foco for um sistema de caronas restrito para a comunidade acadêmica, um grupo heterogêneo pode ser mais interessante, uma vez que o sistema seria voltado para um grupo diverso de usuários, e o uso previsto do sistema e questões que possam restringir o uso (e.g. preocupação com segurança) não estão diretamente relacionados ao papel do participante na comunidade acadêmica.

Grupos de foco são semiestruturados ou livres, uma vez que fomentam a discussão entre os participantes. Para os grupos de foco semiestruturados deve-se preparar um **roteiro** para discussão [Barbour, 2009]. No entanto, o roteiro tende a ser breve, uma vez que deve-se ter tempo para que os participantes possam discutir as questões de interesse. Ainda assim, o pesquisador deve considerar a ordem das questões, e fazer uma avaliação piloto do roteiro (ainda que com outros pesquisadores). Recomenda-se iniciar a discussão com questões gerais e inofensivas antes de abordar o tópico escolhido e considerar se materiais de estímulo poderiam auxiliar a discussão.

Assim como em entrevistas, o roteiro deve assegurar os **cuidados éticos** com os participantes, informar os objetivos da pesquisa e como os resultados serão divulgados. Além disso, no caso de pesquisas científicas, os participantes devem ser voluntários e antes do início da sessão do grupo, os participantes devem assinar o termo de consentimento.

A condução de um grupo de foco é feita por um **moderador**. Este moderador tem por objetivo fomentar e manter o fluxo da discussão, gerenciar o tempo, seguir o roteiro, evitar que a discussão fuja do roteiro e gerenciar os participantes. A gerência de participantes envolve não deixar que um ou alguns participantes monopolizem a discussão, envolver participantes tímidos ou mais calados, e evitar que situações de confronto ou constrangedoras surjam entre participantes. Para isso, o moderador deve ser diplomático e ter especial atenção com suas intervenções para que não soem bruscas e nem desencorajem a participação. Deve-se ainda considerar um moderador que não seja por algum motivo intimidador para os participantes, seja por alguma relação prévia que exista entre eles, seja por ser muito diferente do perfil dos participantes. Por exemplo, um grupo focal com meninas adolescentes pode-se beneficiar de uma moderadora mais jovem que se vista de uma forma mais próxima das participantes e com quem elas possam sentir alguma conexão. Pode-se considerar também ter dois moderadores no grupo para que um fique focado na condução da discussão, enquanto outro faça anotações sobre aspectos relevantes da discussão.

Deve-se ainda decidir o **local e horário** onde será realizado o grupo de foco. Não existem lugares que sejam 100% neutros, então deve-se considerar o impacto que o local possa ter nos participantes e no grupo como um todo [Barbour, 2009]. Além disso, conciliar as agendas dos diversos participantes para se marcar uma sessão para a qual

todos tenham disponibilidade pode ser também um desafio, principalmente quando os participantes são voluntários. Uma vez que a sessão seja marcada, deve-se definir também como será feito o **registro**. Não há um consenso sobre qual forma de registro é melhor, se áudio ou vídeo. Por um lado, o vídeo facilita a identificação do participante e permite o registro de comunicação não verbais. Por outro, o posicionamento de câmeras pode ser um desafio e pode intimidar participantes. Finalmente, recomenda-se que para cada grupo de foco seja feita a transcrição da discussão e que se aplique técnicas de **análise** de discurso.

Em relação a entrevistas, grupos de foco permitem que se colete uma quantidade maior de dados em uma única sessão. No entanto, eles não devem ser pensados como um "atalho" para coleta de dados, já que permitem que se colete dados de mais gente em menos tempo. Até porque, como vimos nesta seção, há um custo maior em se moderar a discussão e fazer o registro e transcrição da mesma. A decisão em relação ao uso de grupos de foco ou entrevistas deve ser feita considerando-se o objetivo e contexto da investigação. Grupos de foco são mais adequados para se obter uma visão compartilhada, enquanto entrevistas permitem um aprofundamento na experiência individual. Além disso, se o foco da investigação é um tema sensível (e.g. saúde, finanças, relações pessoais), deve-se considerar que a discussão em grupo pode inibir depoimentos individuais de conteúdo divergente ou constrangedor, gerando uma coleta menos espontânea que a de entrevistas individuais [Lazar et al., 2010]. Porém, em algumas situações, perceber que outras pessoas têm experiências similares pode ajudar as pessoas a falarem sobre suas próprias experiências [Barbour, 2009].

O Quadro 2.2 descreve resumidamente um exemplo de uso de grupo de foco.

Quadro 2.2: Exemplo de Grupo de Foco [Norval et al., 2016]

Objetivo: Explorar formas de tornar redes sociais mais inclusivas através da criação de recomendações para projetistas destes sistemas que indiquem como evitar barreiras que possam dificultar o uso de redes sociais por pessoas idosas.

Motivação: Pesquisas sugerem que uso de redes sociais por pessoas idosas melhoram sua satisfação com a vida, redução da solidão e a comunicação com membros da família. Porém, usuários idosos utilizam redes sociais em uma proporção bem menor que a população mais jovem.

Método: 2 grupos de foco com idosos.

Recrutamento de participantes: Através de um sistema de voluntários para pesquisas em acessibilidade. Contém mais de 800 voluntários, em sua maioria acima de 65 anos.

Registro: Áudio, posteriormente transcrito

Benefício para participante: Um vale no valor de £10 (permitido em outros países, mas não no Brasil).

Roteiro: Foi pedido que discutissem pontos positivos e negativos sobre redes sociais, além de melhorias que poderiam ser feitas. O moderador conduziu a discussão para que abordasse como os participantes usavam redes sociais, o que os motivou a começar a usá-las, por que não a usavam e como os sistemas poderiam se tornar mais adequados.

Relação dos Grupos: No Grupo 1, participantes que não eram usuários tinham preocupações de mais alto nível (e.g. privacidade e segurança), enquanto que os usuários apresentaram problemas de desafios que vivenciaram e superaram (e.g. dificuldade com configurações de privacidade). Assim, a partir deste grupo, resolveram fazer o Grupo 2, apenas com idosos usuários de redes sociais.

Uso dos Resultados do Grupo 1: Resultados foram utilizados para a definição do segundo grupo de foco, apenas com usuários de redes sociais.

Grupo de foco 1:

- **Número de participantes**: 8 (4 homens, 4 mulheres)
- **Perfil**: Usuários de internet, de idade variando de 61 a 80 anos, 2 eram usuários de redes sociais, 2 já tinham sido usuários, e 4 não eram usuários;

Grupo de foco 2:

- **Número de participantes**: 7 (2 homens, 5 mulheres)
- **Perfil**: Usuários de redes sociais, de idade variando de 63 a 78 anos;

Análise: Análise temática foi usada para categorizar e identificar temas relacionados a pontos positivos e negativos de redes sociais.

Triangulação: Análise temática de 2 artigos relevantes sobre o tema e contraste com temas identificados na análise dos grupos de foco. Temas comuns sugerem que eles são relevantes para um contexto mais amplo que o dos grupos explorados.

Resultados: A partir dos temas identificados como pontos negativos na análise, foram geradas recomendações de como evitá-los. Para os temas positivos a recomendação tinha o objetivo de apoiar o ponto. As recomendações foram então avaliadas a partir de um estudo com usuários.

2.4. Métodos de coleta em contextos reais de uso

Nem sempre coletar significados e opinião dos usuários é a melhor forma de aprender sobre um contexto, atividade ou mesmo as necessidades das pessoas. Há questões de estudo mais adequadas à observação para a captura dos dados do que outras. Como já dito, processos internos dificilmente são traduzíveis de modo claro em ações observáveis e, por isso, para serem capturados, precisam de instrumentos que trabalhem sobre material discursivo a respeito dos contextos reais. Por outro lado, processos de trabalho, funcionamentos de grupo, uso de tecnologias móveis e acompanhamento da introdução e difusão de uma tecnologia são alguns poucos exemplos de fenômenos que se mostram apropriados para o estudo em contextos reais de uso. Além disso, o exame de contextos reais de uso é adequado para situações em que a dificuldade reside na insuficiência de conhecimentos para a geração de um instrumento da coleta preciso, pois o pesquisador pode não saber o suficiente para identificar as questões relevantes a serem perguntadas, ou mesmo como perguntá-las. Além disso, os participantes muitas vezes podem omitir aspectos que para eles são triviais ou corriqueiros (e assim, não consideram importantes ou se esquecem de mencionar), mas que, no entanto, são relevantes. Podem até mesmo omitir informações que considerem não estar em linha com a imagem que querem passar de si. Assim, em muitas situações, pode ser interessante coletar dados diretamente no contexto de uso. Nesta seção, apresentaremos diários de uso, métodos etnográficos e estudos de caso.

Por vezes, esses diferentes tipos de estudos têm em comum o fato de envolverem instituições formais e dados sensíveis ou confidenciais de maior nível de complexidade que estudos envolvendo entrevistas, questionários individuais ou a formação de grupos de duração momentânea a partir de indivíduos isolados e sem relação entre si. Nesses últimos casos, o consentimento individual é suficiente, ou, no caso de participantes vulneráveis, de seus responsáveis legais. No entanto, pesquisas em empresas, escolas e instituições em geral envolvem relações hierárquicas, propriedade industrial ou intelectual, processos de ensino-aprendizagem que não devem ser interrompidos, foco na produtividade do grupo, entre outras questões. Por isto, o planejamento prévio da pesquisa, a análise de riscos de interrupção e do tempo requerido para sua realização, as questões éticas e a viabilidade de publicação dos resultados são alguns fatores que devem estar envolvidos na escolha dos métodos. Há que se considerar, por exemplo, a possibilidade de dissolução de um projeto real de trabalho em uma empresa durante a investigação do mesmo pelo pesquisador, ou a proibição de coleta de um importante conjunto de dados observados por questões envolvendo propriedade intelectual. Recomenda-se, no caso de instituições, a coleta de autorizações da direção geral e de cada participante individual, em particular em estudos etnográficos e em estudos de caso (Yin, 2009).

Considerando os riscos éticos ligados à intimidação e uso de autoridade, grupos com participantes de diferentes níveis hierárquicos requerem cuidado especial nos convites à participação e na circulação de informações, de modo a preservar o consentimento livre e espontâneo na participação. Seidman [1998], por exemplo, sugere que os convites sempre sejam pessoais e que se evite que as chefias sejam responsáveis por convocar seus funcionários a participarem de estudos. Sugere ainda que o pesquisador faça comunicados diretos e em grupo sobre as diferentes etapas de desenvolvimento das pesquisas, evitando comunicações individuais ou mediadas por terceiros.

Além das questões acima, a imersão do pesquisador em contextos reais e a variabilidade de ferramentas metodológicas envolvidas fazem com que esse tipo de estudos exija mais experiência, flexibilidade e capacidade de improviso do que os estudos anteriormente descritos, mais passíveis de condução guiada por planejamento e roteiro prévios.

2.4.1. Diários de uso

Um diário é um documento criado por uma pessoa na qual ela registra eventos, experiências ou pensamentos que considera importantes organizados pela data em que é feita a entrada. **Diários de uso** são aqueles no qual se solicita que participantes registrem, ao longo de um período de tempo, informações e experiências que possam ser relevantes para a pesquisa, projeto ou avaliação de sistema interativos.

Embora diários de uso sejam registros feitos pelo próprio participante (o que pode envolver os mesmos desafios da coleta de dados diretamente dos usuários), eles requerem que o registro seja feito no contexto real, e tipicamente logo em seguida à atividade/experiência a ser registrada [Lazar et al., 2010]. Assim, diferentemente do que ocorre em entrevistas e grupos de foco, diários de uso permitem que se colete informações sobre como as pessoas usam a tecnologia em ambientes que possam ser difíceis de serem observados (e.g. como as pessoas assistem televisão em casa) ou que variam ao longo de dias ou período mais longo de tempo (e.g. uso de aplicativos que registram dados de dieta ou exercícios). Porém, de modo análogo às entrevistas e grupos de foco, diário é um bom

instrumento para coletar processos humanos internos, tais como as motivações, sentimentos, humor ou percepção do usuário associado ao uso da tecnologia. É também interessante para coletar intenções do usuário que não resultaram em nenhuma ação (e.g. o usuário pretendia não assistir um comercial no YouTube, mas acabou não realizando a ação de pular o comercial).

Normalmente, na área de tecnologia, os diários são classificados em três tipos: feedback, elicitação e mistos [Carter e Mankoff, 2005]. **Diários de feedback** são aqueles em que o usuário relata aspectos do uso a partir de questões (mais ou menos estruturadas) definidas pelos pesquisadores e, de preferência, logo após o evento, através do preenchimento do diário (coleta de dados assíncrona). **Diários de elicitação** por sua vez, requerem que os participantes coletem alguma informação relevante sobre o evento (e.g. uma foto ou texto/áudio curto) que depois possa servir de base (i.e., ativar sua memória) para o participante falar sobre o evento em uma entrevista. Assim, diários de feedback normalmente são assíncronos (a coleta é feita através de um registro do participante), enquanto diários de elicitação são síncronos (a coleta dos dados é feita na entrevista usando as informações registradas pelos usuários). **Diários mistos** são aqueles em que características dos dois tipos de diários são combinadas. Por exemplo, o participante faz um registro e, juntamente com este registro, é coletada alguma outra informação (e.g. localização do participante estava quando fez o registro) que depois é utilizada em uma entrevista com o objetivo de contextualizar ou aprofundar o entendimento dos registros feitos.

A vantagem do diário de feedback sobre o de elicitação é que, como o registro é feito pouco tempo após o evento ou até mesmo *in loco*, diários de feedback têm maior chance de coletarem dados mais precisos, uma vez que não dependem da memória que o participante tem do evento. Por outro lado, o custo do diário de feedback é que o registro das entradas a cada evento pode ser trabalhoso para o participante e, se o número de eventos é alto, o participante pode não estar motivado a participar, ou pode acabar não registrando todos os eventos relevantes. Em contrapartida, o diário de elicitação tem um custo de registro mais baixo para o participante, mas a coleta da informação dependerá da memória do participante sobre a informação coletada e o evento.

Ao se projetar um estudo usando diários de uso, deve-se pensar no **instrumento de coleta** de dados, tanto em relação à sua estrutura (e.g. livre ou estruturado), quanto ao meio em que vai ser feito (e.g. papel, formulário online, áudio, vídeo, foto). Em relação à estrutura, se o diário é de feedback, com frequência os pesquisadores terão um conjunto de questões que querem coletar e podem definir então um instrumento estruturado para facilitar o registro do usuário. Este instrumento pode conter tanto questões de múltipla escolha (e.g. 'Você assistiu o vídeo até o fim? '), quanto abertas (e.g. 'Por que você escolheu este vídeo para assistir? '). Diários de elicitação tendem a não usar um registro estruturado, uma vez que o objetivo é uma coleta rápida que será explorada em maior profundidade na entrevista posteriormente.

Em relação ao **meio de coleta**, o pesquisador deve pensar sobre que meio utilizar para o registro, levando em consideração o objetivo da pesquisa, o custo para o usuário e a facilidade de se fazer o registro no momento do evento ou pouco depois. Nas pesquisas e avaliações na área de tecnologia, muitas vezes o evento de interesse que dispara o registro no diário envolve o uso de tecnologia. Assim, com frequência, usa-se o mesmo meio para fazer o registro no diário. Se o registro é relativo ao uso de uma rede social,

pode-se disponibilizar um formulário online para a coleta (e.g. [Wisniewski et al., 2016]). Por outro lado, se o foco é no uso do smartphone, pode-se oferecer uma forma de registro em que o participante vai ligar e dar seu depoimento oralmente (registro em áudio) (e.g. [Palen e Salzman, 2002]).

Por exemplo, em um estudo sobre como pessoas recuperam seus e-mails, levando em consideração as diversas contas (pessoais e profissionais), locais e dispositivos, Cecchinato e colegas [2016] usaram um instrumento em que o participante deveria informar, a cada vez que fazia a busca por um e-mail, o horário, uma breve descrição do que estava procurando e por quê, o tipo de conta (pessoal, trabalho ou outra), o dispositivo (smartphone, tablet, laptop ou desktop) e o local onde estava no momento. As questões sobre tipo de conta e dispositivo eram de múltipla escolha, enquanto as demais eram abertas. O usuário podia escolher usar um formulário online ou fazer o registro em papel (também usando o formulário).

Ao se planejar um estudo com diários de uso, é necessário determinar qual o **perfil dos participantes** desejados no estudo. Por exemplo, se a intenção é analisar a experiência de idosos com um determinado sistema de rede social, deve-se recrutar pessoas maiores de 65 anos que já utilizem o sistema em questão no seu cotidiano. Independente do perfil dos participantes é importante se certificar de que eles entendam o objetivo do diário e como devem preenchê-lo, que estejam dispostos a participar do estudo preenchendo o diário e que conseguirão utilizar o instrumento de coleta usado para registro no estudo (i.e. tenham competência para tal e acesso ao instrumento no momento desejado de registro, etc). Uma técnica comum para recrutar participantes é o da amostragem bola-de-neve [Weiss, 1995].

Tal como os demais métodos qualitativos, os diários têm o objetivo de permitir a compreensão de algum aspecto de uso (e não comprovar uma hipótese) e costumam envolver um número pequeno de participantes. No entanto, o número de participantes propriamente dito é definido de acordo com a pesquisa e seus objetivos, e até mesmo pelo número de pessoas dispostas a participar do estudo. Por exemplo, uma pesquisa sobre riscos percebidos por adolescentes americanos em uma rede social contou com a participação de 68 adolescentes [Wisniewski et al., 2016], enquanto outra pesquisa que investigou como usuários fazem buscas por e-mails nos diversos dispositivos e contas contou com a participação de 16 pessoas [Cecchinato et al., 2016]. Um dos desafios da aplicação de diários de uso no Brasil[2] é motivar voluntários a participarem do estudo por um período de tempo, uma vez que a participação pode ser trabalhosa.

Uma vez terminada a coleta de dados, deve-se fazer então sua **análise**. Dependendo do método de coleta, pode ser necessário fazer uma etapa intermediária de mudança de formato para a análise. Por exemplo, se o registro foi feito em papel, pode-se querer passá-lo para o formato digital, ou se o registro foi feito em áudio, pode ser desejável transcrevê-lo. De modo análogo ao material gerado em entrevistas, a análise dos dados de diários envolve o material textual gerado pelos participantes através da

[2]Diferente do Brasil, em vários países, como os Estados Unidos, recomenda-se que se recompense através de pagamento em dinheiro ou produtos os participantes dos diários. Neste caso há o cuidado de se garantir que esta recompensa seja feita de forma independente da quantidade de registros, para que não influencie os dados coletados.

análise das categorias e temas recorrentes [Corbin e Strauss, 2008; Nicolaci-da-Costa et al., 2004]. Em algumas situações, pode ser possível fazer também uma análise quantitativa dos dados. Por exemplo, na pesquisa de Wisniewski e colegas [2016] sobre riscos percebidos por adolescentes em redes sociais, para cada tipo de risco, os autores analisam a frequência com que eventos dos diferentes tipos de risco ocorrem no estudo.

O Quadro 2.3 descreve brevemente um estudo em que diários de uso foram aplicados em uma pesquisa exploratória para entender com as pessoas recuperam os e-mails armazenados em diferentes contas e usando dispositivos distintos [Cecchinato et al., 2016]. A pesquisa combina o diário de uso, com entrevistas pré e pós-uso do diário, mas no Quadro 2.3 focamos apenas no diário.

Quadro 2.3: Exemplo de Diários de uso [Cecchinato et al., 2016]

Objetivo: Entender as práticas que usuários adotam para gerenciar e recuperar e-mails considerando que utilizam diversas contas e dispositivos.

Motivação: As pessoas têm recebido e armazenado uma quantidade maior de e-mails. Um entendimento aprofundado de como gerenciam e recuperam estes e-mails pode indicar oportunidades para melhorar os sistemas de e-mail.

Métodos: diário de feedback, combinado com entrevistas antes e após o diário.

Duração do diário: 1 semana

Número de participantes: 16

Perfil dos participantes: Idade variando de 22 a 25 anos, 9 eram mulheres e 7 eram homens, de profissões diversas, e todos moradores de uma área no sudeste da Grã-Bretanha.

Método de recrutamento: Através de panfletos, chamadas em sistemas de mídia social, e no boca-a-boca.

Critérios de seleção: Dos 35 inscritos para participar, selecionaram 16 que representassem uma maior variedade de ocupações para garantir uma maior variedade de experiência com computadores e uso de e-mail.

Motivação para participantes: Receberam um vale da Amazon, incentivo permitido na legislação inglesa, mas não na brasileira.

Formato do diário: Podiam escolher o formato digital (12 pessoas) ou em papel (4 pessoas).

Evento registrado: recuperação de mensagens de e-mail.

Estrutura do diário: Foi gerado um formulário. Para cada entrada (ou seja, a cada recuperação ou busca por um e-mail ou anexo) deveriam registrar o horário e local se encontravam, o dispositivo e conta usados, e uma breve descrição do que estavam procurando e por quê. (Alguns exemplos iniciais de como preencher o diário foram mostrados aos participantes).

Dados coletados: 239 entradas (em média 15 por participante)

Análise: Codificação axial e aberta para definir tópicos, que então foram discutidos entre os autores.

Resultados: Analisaram o número de contas (em média 3 por pessoa) e dispositivos usados (em média 4 por pessoa) e como se dava o uso – por exemplo, algumas contas só eram acessadas por um tipo de dispositivo (desktop) e não por outros (dispositivos móveis). Descrevem as diferentes práticas adotadas em diferentes contas de email (trabalho x pessoal); e que tipo de e-mail os participantes procuraram nestas diferentes contas, discutindo as razões e formas utilizadas, assim como questões e soluções alternativas adotadas; como a recuperação de e-mails é diferente nos diversos dispositivos, com base nas estratégias adotadas, questões e soluções alternativas que surgiram. Terminam apresentando algumas considerações a partir dos resultados obtidos que poderiam ser feitas para o (re-)design de clientes de e-mail e suas funcionalidades.

2.4.2. Métodos Etnográficos

Tradicional na Antropologia Social e na Sociologia, a etnografia é uma abordagem qualitativa que busca descrever pessoas, suas instituições, comportamentos, produções e cultura a partir da observação detalhada de seu cotidiano. Baseia-se na noção de que, para se entender realmente práticas e contextos complexos, é necessário realizar estudos em profundidade onde o pesquisador imerge no grupo a ser estudado, deixando de ser um observador externo, para se envolver com as atividades cotidianas e pessoas do grupo. Mais notadamente desde a década de 1980, os métodos etnográficos vêm sendo bastante utilizados em computação, em particular nas áreas de sistemas colaborativos (*Computer Supported Cooperative Work- CSCW*) e de Interação Humano-Computador (IHC) [Lazar et al., 2010; Randall e Rouncefield, 2013].

A abordagem etnográfica pressupõe a **imersão prolongada do pesquisador** no campo de investigação, priorizando a **observação e a descrição do cotidiano social**, com o objetivo de tornar visível e compreensível a organização de um determinado grupo social, suas atividades cotidianas, papéis e regras, tal como entendidas pelos seus atores sociais. Seus métodos permitem que o pesquisador adote duas **posturas interpretativas**, indispensáveis à execução adequada da abordagem: familiarização com o exótico e estranhamento do familiar [da Matta, 1978]. **Transformar o exótico em familiar** consiste no desafio de olhar grupos e cotidianos muito distintos da realidade do pesquisador despido dos próprios preconceitos, valores e lógicas de funcionamento, classificação e significação. Consiste em olhar o cotidiano do outro despindo-se de nossas próprias lentes e buscando, por meio de diferentes técnicas, coletar os significados que os próprios atores sociais atribuem àquele cotidiano em estudo. Trata-se de identificar e descrever como os próprios membros de um grupo se percebem e dão sentido ao que fazem, em contraposição a usar a nossa própria visão de mundo para descrever os fenômenos estudados. Por exemplo, trata-se de buscar entender como os jovens fazem uso cotidiano das redes sociais, a partir da ótica dos próprios jovens, sem usar categorias mais antigas que valorizam a interação presencial e veem a interação digital, *a priori*, como isolamento, alienação e vício. A imersão prolongada junto a um grupo de jovens, bem como acompanhamento e registro de suas atividades diárias (online e cotidianas), pode atuar como facilitador do processo de familiarização por parte do pesquisador, visando permitir a identificação de relações entre as interações ocorridas nas redes sociais e as relações presenciais dos jovens, por exemplo. Neste sentido, a imersão prolongada é um mecanismo metodológico para provocar a **empatia**, a colocação do pesquisador no

lugar do grupo pesquisado, desconstruindo suas preconcepções e abrindo espaço para a construção de novas perspectivas e significados sobre um fenômeno.

Na direção oposta, **transformar o familiar em exótico** consiste no desafio de o pesquisador sair de sua zona de conforto para olhar o que já conhece sob novas perspectivas, de modo a permitir novos *insights* sobre antigos fenômenos e a emergência de novos significados sobre os mesmos. Esta posição pode ser experimentada, por exemplo, em estudos que visam construir novos métodos e técnicas de design de sistemas interativos. Para isto, o pesquisador fica imerso em um grupo de pares, e carrega uma série de abordagens e técnicas em sua bagagem, influenciando seu olhar. A imersão e a observação do funcionamento do grupo devem provocar o olhar analítico, que segmenta as atividades, papéis e processos de trabalho para que novas perspectivas possam ser insumo para o desenvolvimento de novos métodos e técnicas de trabalho.

As duas posições são metas ideais, posto que o pesquisador sempre olhará para o contexto de estudo com algum resíduo de sua própria bagagem pessoal e profissional [da Matta, 1978]. Porém, por meio do longo e intenso processo de imersão, o pesquisador pode sair de suas próprias preconcepções, tornar visíveis significados e processos mais sutis e obscuros do funcionamento de um grupo em sua vida cotidiana, e compreender os porquês de determinado comportamento ou tradição, as razões para uma dada forma de dividir uma tarefa, os significados latentes de um determinado hábito. A intenção da imersão, da observação e do registo é ver, sentir e descrever antes de avaliar e categorizar, para construir uma **perspectiva interna** *('insider's view'*) sobre o grupo de pessoas e organização social que se busca examinar [Randall e Rouncefield, 2013]. Para isto, o pesquisador observa detalhadamente práticas cotidianas do grupo estudado, analisa documentos que o grupo já produziu para descrever ou regular essas práticas, participa de reuniões formais e informais desse grupo, mapeia o espaço em que o grupo vive, os diferentes papéis de seus membros, seus mecanismos de identificação, organização, segregação de seus membros, etc.

As ferramentas metodológicas que o pesquisador lança mão para o trabalho etnográfico são diversas. O **diário de campo** é essencial e define-se como um 'espaço' no qual ele faz anotações em tempo real de sua longa **observação** e de suas interpretações. Além de breves frases (escritas ou gravadas) para rememoração posterior, por vezes, rápidos desenhos de espaços físicos de trabalho e a posição que membros ocupam, organogramas e fluxogramas costumam integrar o diário de campo. A modalidade de observação do pesquisador também varia muito. Em algumas práticas, ele pode tão-somente observar, sem interferir diretamente no andamento do grupo, enquanto em outras, ele pode participar diretamente das atividades executadas. Dadas a duração e intensidade da imersão, dificilmente todo o trabalho de campo pode ser gravado e/ou filmado (daí a importância do diário de campo), mas o pesquisador deve priorizar reuniões ou práticas mais importantes para registro e posterior análise. Sempre que necessário para compreender melhor algum ponto da investigação, o pesquisador pode marcar **entrevistas** e realizar suas modalidades livre, semiestruturada ou estruturada. Além disso, a **análise de documentos** produzidos pelo grupo antes e durante a pesquisa pode ser útil para a elucidação de significados, de normas, de processos e da lógica de funcionamento do grupo e de suas práticas.

A variabilidade de recursos e a duração da imersão geram complexidades para a atividade do pesquisador. Além de serem necessárias habilidades técnicas para coleta,

registro e análise dos dados segundo diferentes procedimentos, há determinadas qualidades do pesquisador importantes à boa condução do estudo etnográfico que, no entanto, extrapolam sua formação técnica. Entre elas se destacam: a capacidade de atenção e concentração distribuídas (permitindo a captura, por vezes sem instrumentos de registro, de dados de ocorrência simultânea e de naturezas diversas); capacidade de escuta genuína; alta habilidade para a seleção e priorização de dados (considerando sua diversidade e sua quantidade) e habilidade para manutenção do foco na questão de pesquisa (uma vez que os grupos estão desenvolvendo atividades cotidianas não necessariamente voltadas diretamente para o tópico sob investigação). Em resumo, o desafio na aplicação deste método é o de capturar dados visando um objetivo de *foco nítido* a partir da observação de *práticas difusas* e espontâneas, não necessariamente focadas nos mesmos objetivos. Para ajudar que o pesquisador mantenha a nitidez do seu foco, Randall e Rouncefield [2013] propõem 10 preceitos, a saber:

1. Assumir que o mundo é socialmente organizado.
2. Observar o contexto e suas atividades para identificar a organização e o sentido, frequentemente implícitos, das atividades.
3. Compreender o contexto e suas atividades nos termos que seus próprios membros compreendem.
4. Examinar em profundidade as atividades.
5. Considerar as atividades como situadas, sempre vinculadas a um contexto especifico de ocorrência, que fornece seu sentido e características.
6. Prestar atenção na divisão do trabalho ou das atividades observadas sempre em função das interações reais (em detrimento das informações formais e institucionais).
7. Identificar a sequência e o processo de desenvolvimento das atividades e tarefas.
8. Focalizar pessoas e não papéis institucionalmente definidos.
9. Não distinguir conhecimento de especialistas de conhecimento prático.
10. Não comparar diferentes contextos de estudo considerando-os equivalentes.

Os métodos etnográficos vêm sendo usados crescentemente na computação, em particular as áreas de IHC e CSCW, e mais recentemente na Engenharia de Software. São utilizados principalmente na sua **aplicação em fases iniciais e exploratórias do processo de desenvolvimento** [Lazar et al., 2010; Randall e Rouncefield, 2013; Sharp et al., 2016]. São aplicados para informar designers e desenvolvedores sobre uma determinada prática de trabalho ou de interação social, sobre como essa prática é organizada e dividida, como ela se relaciona com outras práticas e processos, como ela se regula e quais os papéis sociais e questões contextuais envolvidas. Os métodos têm muito valor em identificar idiossincrasias, exceções e conteúdos obscuros e subjacentes dos processos, não disponíveis em descrições institucionais e formalizadas. De posse desses resultados, outros métodos e técnicas de design ou de pesquisa trabalham sobre o insumo obtido. Em resumo, no contexto da computação, métodos etnográficos mostram-se especialmente úteis quando se deseja responder às seguintes perguntas: Qual o tipo de problema a ser tratado? Como o problema se configura? Como se manifesta? [Randall e Rouncefield, 2013].

Um exemplo de pesquisa que faz uso dos métodos etnográficos é resumido no Quadro 2.4.

Quadro 2. 4: Exemplo de Estudo Etnográfico [Reddy, M. et al., 2006]

Objetivo: Investigar o papel da temporalidade no trabalho cooperativo e distribuído, em termos de sua distribuição das atividades no tempo, por meio de um estudo etnográfico realizado junto a profissionais da área médica de uma unidade de tratamento intensivo pós-cirúrgico.

Motivação: Expandir os conhecimentos na área de trabalho em equipes distribuídas, focalizando menos a dimensão espacial, mais frequentemente investigada, e mais a perspectiva temporal. A **relação entre busca de informação e tempo em uma equipe de trabalho colaborativa** é a motivação central.

Método: Etnografia

Duração do estudo: 7 meses de observação

Grupo social observado: Profissionais de saúde de uma unidade de terapia intensiva pós-cirúrgica de um hospital universitário urbano de grande porte.

Motivação para escolha do grupo: Um hospital é um exemplo paradigmático da necessidade de produção, administração e circulação de informações complexas e rápidas visando o bem-estar dos pacientes, em particular para aqueles que necessitam de cuidados intensivos.

Características do grupo: É uma das 9 unidades de tratamento intensivo do hospital, com 20 leitos e alta proporção enfermeiro/paciente (1:2), favorecendo cuidados mais intensos, próximos e abrangentes para os pacientes e maior colaboração entre a equipe

Participantes da equipe: Enfermeiros, farmacêuticos, médicos, cirurgiões, fisioterapeutas, residentes e professores universitários, entre outros profissionais.

Foco do estudo: Observação nos detalhes das atividades de trabalho dos profissionais de saúde que se envolviam e geravam efeitos sobre a busca e administração de informações colaborativas, enfatizando a organização temporal dessas atividades.

Unidade de observação: 1dia de trabalho entre equipe e pacientes.

Técnicas de coleta de dados: 30 entrevistas formais, várias entrevistas informais, observação dos profissionais de saúde, análise documental de regras do hospital, procedimentos e atas de reuniões de equipe, bem como de informações disponíveis nos exames e nos equipamentos médicos de trabalho.

Formato dos registros de coleta de dados: gravações e transcrições de áudio das entrevistas formais, anotações sobre observação do uso de equipamentos e suportes para a circulação da informação de trabalho.

Análise dos Dados: Análise do discurso do material verbal e identificação de unidades de análise relacionadas ao processo de geração e gestão da informação no material documental e de trabalho.

Resultados: Identificação e descrição *contextualizada* de 3 aspectos temporais envolvidos na administração das informações durante um dia de trabalho na área de saúde: trajetórias do tempo (e.g. em que fase da recuperação o paciente está), ritmos do tempo (e.g. tempo que um tipo de exame demora para ficar pronto) e horizontes do tempo (e.g. definição da sequência de tarefas e complexidade envolvidas em uma jornada de trabalho).

2.4.3. Estudos de caso

Estudo de caso é um **estudo detalhado e aprofundado de uma instância específica ou pequeno conjunto de instâncias em um contexto real**. De modo similar à etnografia, é uma observação em profundidade que visa predominantemente explorar novos problemas e situações, descrevendo-os, explicando-os ou demonstrando como acontecem a partir de um ou de poucos casos de ocorrência [Lazar et al., 2010]. Enquanto métodos etnográficos envolvem a imersão prolongada do pesquisador neste contexto, com enfoque maior na observação, na exploração e na participação do pesquisador no cotidiano do grupo, o estudo de caso é, em geral, mais voltado para a tarefa (*goal-oriented*) e costuma propor uma intervenção específica do pesquisador no contexto. No caso da computação, essa intervenção costuma envolver o desenvolvimento de um sistema interativo ou a avaliação do mesmo junto a grupos de usuários.

Embora sempre almejando a análise no contexto real de ocorrência, em particular no caso do estudo de tecnologias em desenvolvimento, por vezes, faz-se necessário criar uma situação artificial de uso. É o caso, por exemplo, da avaliação de protótipos para testar tecnologias e métodos ainda não difundidos em contextos reais. Nesses casos, em vez de situações de controle laboratorial, busca-se, tanto quanto possível, cenários e configurações próximas às condições reais de uso [Yin, 2009]. O acompanhamento de um paciente com deficiência motora quando da utilização de uma tecnologia vestível de apoio a seu movimento em tarefas cotidianas e a observação de grupo utilizando uma nova ferramenta de design em um cenário de uso podem ser exemplos de questões adequadas para adoção do estudo de caso como método de coleta de dados.

Segundo Yin [2009], um estudo de caso pode ser **simples**, quando examina apenas um caso de ocorrência, como, por exemplo, a análise da evolução do uso de uma tecnologia digital por um usuário ao longo de um ano. Pode ainda ser **múltiplo**, ao analisar diferentes casos de uso, tal como a observação de uso de uma ferramenta de design por dois ou três grupos distintos de profissionais. Nos estudos de caso múltiplos, o objetivo não é prioritariamente a previsão, replicação e generalização das situações de uso, mas a composição de uma descrição aprofundada que reflita diferentes perspectivas sobre um mesmo fenômeno para a construção de um framework teórico de interpretação em uma perspectiva analítica *bottom-up*.

O convite à participação em um estudo de caso se faz com base nos critérios mais gerais propostos pela metodologia qualitativa: amostragem proposital com base na adequação entre os objetivos da pesquisa e o participante/ grupo a recrutado. Se o estudo será simples ou múltiplo é uma decisão certamente dependente do tempo de execução (uma observação extensa em duração não comporta numerosos casos) e também da facilidade de ocorrência dos fenômenos. Há casos únicos e irreplicáveis, ao passo que há casos em que a observação de diferentes ocorrências de um mesmo fenômeno lança luz sobre o problema. O estudo de tecnologias assistivas, por exemplo, requer, por vezes, desenvolvimento individualizado segundo as necessidades de cada usuário e, consequentemente, casos individuais. Já o estudo de processos de trabalho pode tornar desejável a observação de grupos múltiplos.

Como em métodos etnográficos, estudos de caso normalmente envolvem a coleta de dados por **diferentes métodos e técnicas** além da observação. No entanto, em estudos de caso, o pesquisador costuma ter uma postura menos integrada e imersiva ao cotidiano grupo sob estudo, integrando-se ao campo mais com papel e objetivos externos ao grupo

[Lazar et al., 2010; Yin, 2009]. Na área da computação, o papel do pesquisador no estudo de caso é mais comumente o de desenvolver de modo mais participativo ou avaliar um sistema ou ferramenta computacional. Neste contexto, as diferentes técnicas se articulam mais comumente para: descrever detalhadamente fenômenos (técnicas de observação de comportamentos ou de uso de tecnologias, análise documental, análise de artefatos físicos, técnicas de elicitação de requisitos e de design); aprofundar quais significados os participantes atribuem aos fenômenos previamente observados (entrevistas, questionários, grupos de foco); e avaliar os fenômenos observados e vivenciados (métodos de avaliação de sistemas por inspeção e com a participação de usuários). Cada método exigirá uma expertise específica e execução adequada e a habilidade para articular **resultados de diferentes naturezas** em um todo coerente, aspectos que conferem aos estudos de caso maior complexidade de execução, o que pode ser obstáculo a pesquisadores menos experientes. Tal como em outros métodos qualitativos, o relato desses resultados deve descrever as categorias e temas de análise e oferecer exemplos claros que forneçam rastreabilidade aos resultados do estudo.

Um importante fator de reflexão sobre a adoção do estudo de caso como método deve ser a **duração** prevista para o estudo [Lazar et al., 2010]. Um estudo pode ser extenso em duração, exigindo disponibilidade de vários encontros ou seções de trabalho de seus participantes ou muitas horas de trabalho em uma única seção. Muito embora, um estudo possa requerer envolvimento intenso de trabalho por participantes e pesquisadores para sua adequada realização, essa exigência pode dificultar ou mesmo inviabilizar a realização do estudo, tanto por ser exaustivo quanto por afastar o interesse em colaborar de usuários que certamente não priorizam o estudo da mesma maneira que os pesquisadores que o propõem. A motivação dos participantes costuma ser fortemente atrelada ao tempo de colaboração envolvido.

No Quadro 2. **5**, o estudo de caso é exemplificado em resumo descritivo de uma pesquisa que faz uso desse método.

Quadro 2. 5: Exemplo de Estudo de Caso [Chagas, 2015]

Objetivo: Identificar e entender as necessidades para desenvolvimento de um protótipo funcional de tecnologia assistiva (TA) configurável para o controle de dispositivos por gesto e interação de voz em casa inteligente. A partir desse caso, propor um framework conceitual de apoio ao desenvolvimento de TAs.

Motivação: Lidar com a variabilidade de tipos e graus de deficiência e das características individuais de usuários de TAs por meio do melhor entendimento sobre o que significa configuração neste domínio.

Método: Estudo de Caso Simples

Duração do estudo: 1 mês para planejamento e elaboração do Protocolo de Ética e 12 meses de execução do estudo

Número de participantes: 1 participante com deficiência

Perfil do participante: 1 participante de 33 anos, tetraplégico há 10 anos em função de acidente, mestre em Administração de Empresas.

Método de recrutamento: Amostra proposital

Critérios de seleção: O próprio participante procurou o laboratório propondo o desenvolvimento da tecnologia.

Motivação para o participante: Protótipo funcional de TA configurável para suas necessidades em sua "casa inteligente".

Imprevistos de Execução: Problemas de saúde do participante inviabilizaram testes por um período e geraram a necessidade de mudanças de direção no desenvolvimento da tecnologia.

Técnicas de coleta de dados: Entrevistas exploratórias para entendimento das limitações do participante, observação de tarefas cotidianas com ou sem apoio de dispositivos computacionais, na casa do participante, 5 ciclos de desenvolvimento e avaliação do protótipo (pesquisa-ação).

Formato dos registros de coleta de dados: anotações textuais em diários de campo do pesquisador, gravações e transcrições de áudio das entrevistas, vídeos da interação do participante e de seus cuidadores.

Análise dos Dados: Análise do discurso do material verbal e identificação de unidades de análise relacionadas ao ambiente físico e computacional para levantamento dos requisitos do protótipo,

Resultados: Proposta de 3 dimensões para identificação e análise de necessidades e oportunidades para configuração de TA: dimensão psico-social; dimensões físicas e dimensões de persistência (temporalidade e volatilidade). Protótipo funcional de uma TA para o participante.

2.5. Métodos de Inspeção

Diferentemente dos métodos vistos nas seções anteriores, que têm sua origem ou base em métodos qualitativos utilizados nas ciências humanas e sociais, métodos de inspeção foram desenvolvidos especificamente para a análise de sistemas computacionais, mais notadamente nas áreas de IHC e de Engenharia de Software. Podem ter base qualitativa ou quantitativa.

Métodos de inspeção em Engenharia de Software seguem mais comumente a abordagem quantitativo-experimental e visam a verificação e validação do software em termos de confiabilidade, performance de execução e da correspondência entre o software e sua especificação, entre outros fatores. Podem ser realizados manualmente ou de forma automatizada e não têm como foco questões relacionadas ao uso real do software [Sommerville, 2011].

Já os métodos de inspeção em IHC são métodos em que um avaliador ou analista examina a interface e interação de um sistema interativo seguindo diretrizes ou passos definidos pelo método a ser aplicado. O avaliador tem por objetivo identificar possíveis interpretações e/ou potenciais problemas que o usuário de um sistema poderia vivenciar ao interagir com o sistema.

Inicialmente estes métodos foram desenvolvidos com foco na análise de usabilidade de interfaces com usuário [Mack e Nielsen, 1994]. No entanto, ao longo dos anos, outras qualidades de uso, como por exemplo acessibilidade, sociabilidade ou

comunicabilidade se tornaram relevantes [Barbosa e Silva, 2010] e métodos voltados para analisar estas qualidades nos sistemas interativos foram propostos. Atualmente pode-se encontrar na literatura um grande número de métodos de inspeção propostos, mas poucos são de fato consolidados.

Outras características podem distinguir os métodos existentes. Em relação à sua **base**, métodos de avaliação podem ser de base empírica ou teórica. Métodos de base empírica são aqueles cuja base é resultado da análise de dados empíricos obtidos indutivamente, enquanto os de base teórica são fundamentados em alguma teoria. Os métodos também podem variar de acordo com o **número de avaliadores** recomendados ou o **conhecimento necessário** para que apliquem o método.

Os métodos podem ser propostos ou ser mais adequados para diferentes **momentos do ciclo de design**, isto é, podem ser aplicáveis para avaliação formativa (ao longo do processo de design) ou somativa (ao final do processo de design).

Os métodos de inspeção qualitativos mais populares e, possivelmente consolidados, para avaliação de usabilidade de interfaces são os métodos de Avaliação Heurística e o de Percurso Cognitivo. O método de **Avaliação Heurística** [Nielsen, 1994], de base empírica, propõe que o avaliador especialista em IHC examine a interface e analise se a mesma está em conformidade com um conjunto de heurísticas ou princípios de usabilidade. Caso o avaliador identifique alguma violação a uma das heurísticas, ele deve registrar o problema, identificando a heurística violada, em que local(is) da interface o problema ocorre, e qual seu grau de severidade. A recomendação é que a avaliação seja feita individualmente por um conjunto de 3 a 5 avaliadores, e posteriormente seja feita uma reunião para consolidação dos relatórios individuais gerados. Segundo Nielsen [1994], o método pode ser aplicado a protótipos funcionais ou mesmo em papel, e, por isso, poderia ser aplicado ao longo do ciclo de design, i.e. avaliação formativa ou somativa.

O método da Avaliação Heurística original apresenta um conjunto de 10 heurísticas[3] que representam princípios gerais de usabilidade que foram derivados empiricamente da análise de 249 problemas de usabilidade. Como o conjunto original de heurísticas é genérico, ao longo dos anos diversos pesquisadores propuseram novos conjuntos para permitirem a avaliação de sistemas desenvolvidos para tecnologias (e.g. interação humano-robô [Clarkson e Arkin, 2007] ou interfaces periféricas para ambientes [Mankoff et al., 2003]) ou domínios específicos (e.g. sistemas colaborativos [Baker et al., 2002] ou jogos [Pinelle et al., 2008]).

O método de **Percurso Cognitivo** [Polson et al., 1992; Wharton et al., 1994], de base teórica, avalia a usabilidade de uma interface com foco na facilidade de aprendizado desta interface pelo usuário. O método é voltado, principalmente, para desenvolvedores de software com o objetivo de que o apliquem na avaliação formativa dos sistemas. O Percurso Cognitivo é fundamentado na teoria das ações proposta por Norman [1986] e

[3] Resumidamente as heurísticas são de Nielsen são: visibilidade do estado do sistema; correspondência entre o sistema e o mundo real; controle e liberdade do usuário; consistência e padronização; reconhecimento em vez de memorização; flexibilidade e eficiência de uso; projeto estético e minimalista; prevenção de erros; ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros; ajuda e documentação. Para uma visão aprofundada sobre o método ou conjunto de heurísticas ver [Nielsen, 1994] ou livros didáticos de IHC (e.g. [Barbosa e Silva, 2010; Preece et al., 2015]) ou mesmo artigos disponíveis na página do NormanNielsenGroup [Nielsen, 1995a,b].

analisa a interação sob esta ótica. No entanto, os passos do método permitem que ele seja aplicado, por um ou mais avaliadores, sem que se tenha um conhecimento específico da teoria subjacente. Para aplicar o Percurso Cognitivo o avaliador deve receber a descrição dos usuários do sistema, que tarefas devem ser analisadas e para cada tarefa o conjunto correto de ações para executá-la e um protótipo (em papel ou funcional) da interface. O avaliador então analisa cada tarefa, executando as ações necessárias e analisando se os usuários saberão que devem executar a ação, se perceberão na interface que a ação está disponível, se associarão a representação da interface à ação correta e, uma vez que tenham executado a ação, se perceberão que estão progredindo para completar a tarefa. A partir desta análise o avaliador identifica os potenciais problemas da interface a serem corrigidos.

Nesta seção, vamos apresentar em mais detalhe o Método de Inspeção Semiótica (MIS) que tem por objetivo permitir a avaliação da comunicabilidade de um sistema. A seleção do MIS se deve ao fato de que ele pode ser aplicado tanto tecnicamente, com foco na qualidade do sistema desenvolvido, quanto cientificamente, com foco na geração de novos conhecimentos sobre IHC ou outras áreas de computação a partir da análise de sistemas interativos [de Souza et al., 2010; de Souza e Leitão, 2009].

2.3.1. MIS - Método de Inspeção Semiótica

O MIS é um método fundamentado na teoria da Engenharia Semiótica [de Souza, 2005], uma teoria da área de IHC que entende a interface de um sistema como sendo uma comunicação do projetista do sistema aos seus usuários. Através desta interface, o projetista transmite aos usuários a quem o sistema se destina, que objetivos o projetista entende que os usuários podem ou querem atingir, e como interagir com o sistema para alcançá-los. Esta mensagem pode ser representada como:

> **Template da metacomunicação:** *"Esta é a minha interpretação sobre quem você é, o que eu entendi que você quer ou precisa fazer, de que formas prefere fazê-lo e por quê. Este é, portanto, o sistema que eu projetei para você, e esta é a forma que você pode ou deve usá-lo para atingir objetivos alinhados com a minha visão."*

Como esta comunicação projetista-usuário se dá através da comunicação sistema-usuário, ela é de fato uma metacomunicação (comunicação feita através de outra comunicação), e é indireta (através do sistema) e unidirecional (o usuário não tem chance de responder projetista durante a interação – momento em que recebe a sua mensagem).

A mensagem enviada pelo projetista é formada por signos, isto é, qualquer coisa que represente algo para alguém [Peirce, 1992-1998]. A teoria da Engenharia Semiótica classifica os signos de uma interface como sendo signos estáticos, dinâmicos ou metalinguísticos [de Souza et al., 2006; de Souza e Leitão, 2009]. **Signos estáticos** são aqueles que podem ser interpretados independente de relações causais ou temporais e representam o **estado** do sistema. Em outras palavras, o seu contexto de interpretação é limitado aos elementos presentes na interface em um dado momento. O botão Salvar que pode ser visto na barra superior da janela do Word é um exemplo de um signo estático. Os **signos dinâmicos**, por sua vez, só podem ser percebidos através da **interação** com o sistema e representam o comportamento do sistema, ou seja, estão relacionados aos aspectos temporais e causais da interface. Assim, a ação de apertar o botão de salvar gera um comportamento de salvar a versão atual do arquivo – este comportamento é dinâmico e representado por diferentes signos (e.g. uma barra de evolução do processo em curso).

Finalmente, os **signos metalinguísticos** são aqueles que se referem a outros signos da interface. Normalmente, eles são usados pelos designers para explicitamente comunicar aos usuários os significados codificados no sistema e como podem utilizá-los. Assim, a explicação associada ao botão salvar que o explica ou a explicação do sistema de ajuda sobre o botão salvar são exemplos de signos metalinguísticos.

Com base na teoria da Engenharia Semiótica, a qualidade da interface é definida pela sua comunicabilidade, isto é a propriedade do sistema transmitir ao usuário de forma organizada e consistente (eficiência) a lógica, a intenção e os princípios de design, realizando assim sua finalidade junto ao usuário (eficácia) [Prates et al., 2000; de Souza e Leitão, 2009]. Para avaliar a comunicabilidade de um sistema os dois principais métodos são: Método de Avaliação de Comunicabilidade (MAC) [Prates et al., 2000; de Souza, 2005] e o Método de Inspeção Semiótica (MIS) [de Souza et al, 2006; Prates e Barbosa, 2007; de Souza e Leitão, 2009; Leitão et al., 2013]. O MIS é um método de inspeção que analisa a emissão da comunicação projetista-usuário (i.e. interface) pelo projetista, enquanto que o MAC[4] analisa como esta comunicação está sendo recebida pelos usuários, através da observação da interação usuário-sistema em cenários de atividades pré-definidos.

O MIS é então um método de avaliação por inspeção que tem como base a teoria da Engenharia Semiótica e analisa a comunicabilidade de uma interface. O avaliador do MIS deve ter conhecimento não apenas de IHC, mas também, ainda que em níveis básicos, de Engenharia Semiótica. Um avaliador é suficiente para gerar uma análise de caminhos interpretativos possíveis e identificar potenciais problemas de um sistema interativo. No entanto, a aplicação por mais avaliadores pode ser interessante para consolidar a análise interpretativa do avaliador [Prates e Barbosa, 2007]. Finalmente, a avaliação é feita através da análise de um sistema, mas pode ser aplicada a protótipos, desde que se tenha acesso não apenas aos signos estáticos da interface, mas a signos dinâmicos e metalinguísticos.

A etapa de preparação do MIS requer que o avaliador defina qual o escopo da avaliação, de acordo com seu objetivo, por exemplo, analisar a parte crítica para o sucesso do sistema (i.e., a parte considerada mais relevante, ou que deve ser melhor que a de um sistema competidor). Em seguida, o avaliador deve fazer uma inspeção informal do sistema, identificando a quem o sistema se destina e os principais objetivos e tarefas que o sistema pretende atender. Finalmente, deve ser criado um ou mais cenários[5] de inspeção que permitam a definição de um foco para análise da intenção comunicativa e caminhos interpretativos preferenciais.

[4]Vale ressaltar que o MAC é também um método qualitativo voltado para avaliação da qualidade da interação em sistemas interativos com participação de usuários, assim como outros métodos de avaliação. No entanto, como estes métodos normalmente são vistos em disciplinas de IHC, neste capítulo não os apresentaremos. O leitor interessado no MAC pode consultar: [Prates et al., 2000; de Souza, 2005; Prates e Barbosa, 2007; de Souza e Leitão, 2009; Barbosa e Silva, 2010; Leitão et al., 2013].

[5]Cenários são uma narrativa, normalmente em linguagem natural, concreta rica em detalhes contextuais de uma situação de uso do sistema envolvendo pessoas, processos e dados reais ou potenciais [Barbosa e Silva, 2010].

Finalmente, o avaliador passa então para a etapa de análise, que envolve 5 etapas:

Passo 1 - Análise dos signos metalinguísticos: O avaliador examina apenas os signos metalinguísticos da interface (i.e. sistema de ajuda, instruções, mensagens de erros, etc.) e faz uma reconstrução da metamensagem que está sendo transmitida apenas por estes signos. Faz-se também registros de potenciais problemas que possam ser identificados nesta etapa.

Passo 2 - Análise dos signos estáticos: O avaliador executa o mesmo processo descrito no passo1, mas, no entanto, leva em consideração na sua análise apenas os signos estáticos. Desta forma, a metamensagem é reconstruída levando-se em consideração apenas estes signos.

Passo 3 - Análise dos signos dinâmicos: Novamente, o avaliador executa o mesmo processo descrito nos passos anteriores, mas agora levando em consideração apenas os signos dinâmicos.

Ao fim do passo 3, o avaliador tem uma visão segmentada da interface, pois tem como resultado a reconstrução da metamensagem sob três perspectivas de comunicação distintas – cada uma considerando apenas os signos de um determinado tipo. Nos passos 4 e 5 do método, o objetivo é de reconstruir e avaliar a metamensagem através da comparação, integração e interpretação das metamensagens geradas.

Passo 4 - Comparação e contraste: Nesta etapa o avaliador compara e contrasta a reconstrução das 3 metamensagens geradas nos passos anteriores. O objetivo do avaliador é analisar a consistência entre elas, identificar pontos em que uma possa complementar ou aprofundar pontos de outras, e se há inconsistências entre elas. Vale ressaltar, como os tipos de signo têm capacidade expressiva diferente (i.e. o que se pode transmitir através de texto em linguagem natural é diferente do que se consegue fazer com um desenho de tela) é natural que as metamensagens geradas não sejam idênticas. É esperado, contudo, que sejam consistentes.

Passo 5 - Apreciação da metacomunicação: Finalmente, no último passo do método, o avaliador gera a metamensagem unificada do sistema, identificando potenciais problemas na comunicação que persistam na unificação (i.e. um potencial problema identificado nos signos estáticos, pode ser resolvido pela comunicação nos signos dinâmicos ou metalinguísticos) e analisando o custo e benefícios das estratégias de comunicação adotadas pelos projetistas. Conclui-se, então, com um relatório sobre a qualidade da comunicabilidade do sistema. A Figura 2 apresenta uma visão geral dos passos do MIS.

Uma das vantagens do MIS é que ele pode ser utilizado para uma **análise técnica ou científica** de um sistema [de Souza e Leitão, 2009; de Souza et al., 2010]. A análise técnica é aquela que tem o foco na qualidade do sistema gerado e cujo objetivo é identificar potenciais problemas que permitam a melhora do sistema em si. A **aplicação científica**[6], por sua vez, tem por objetivo gerar novos conhecimentos principalmente (mas não apenas) para a área de IHC, através da identificação, descrição, exploração ou explicação de fenômenos de interação humano-computador ou levantamento de novas

[6]Neste texto apresentamos superficialmente a aplicação científica do MIS. Para uma explicação aprofundada sobre o uso do MIS científico recomendamos a leitura de [de Souza et al., 2010; de Souza e Leitão, 2009].

questões ou desafios na área. Para a aplicação científica do MIS, embora os 5 passos sejam os mesmos, durante a etapa de preparação, o pesquisador deve também definir a questão de pesquisa que guiará a análise do sistema. Os passos de 1 a 4 não mudam, mas, no passo 5 de uma aplicação científica, a análise tem por objetivo obter as respostas para a questão de pesquisa colocada com base na metamensagem observada. Finalmente, de posse dos resultados científicos, a consolidação deste resultado deve ser feita através de uma triangulação (a etapa de triangulação será explicada na seção 2.6.1).

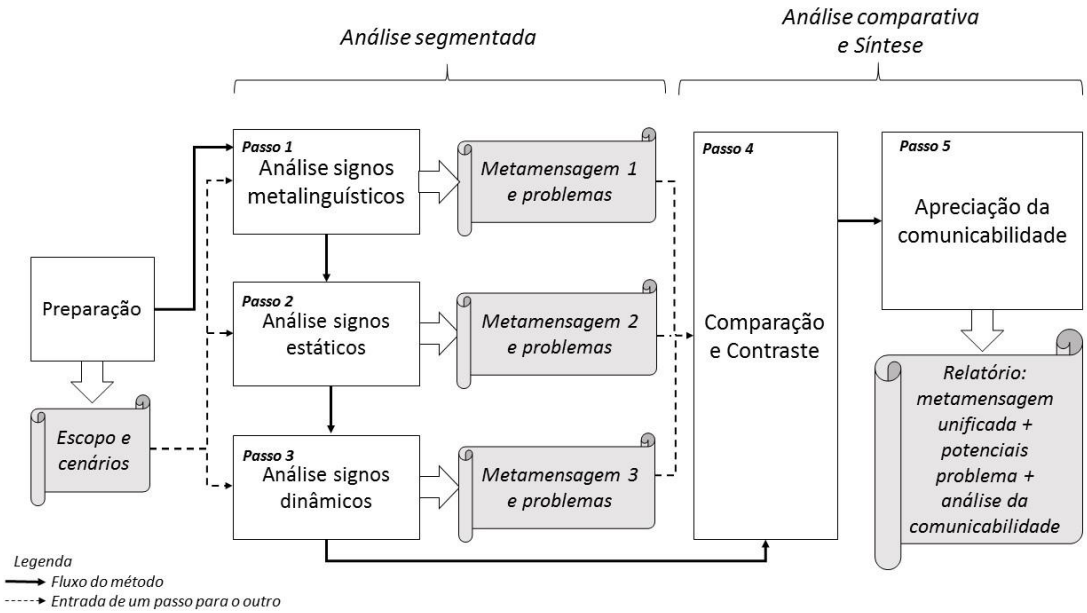

Figura 2 - Método de Inspeção Semiótica

* * *

Os três métodos apresentados nesta seção – Avaliação Heurística, Percurso Cognitivo e Método de Inspeção Semiótica – são métodos qualitativos e interpretativos que têm por objeto a avaliação da qualidade de uso de um sistema interativo. Enquanto a Avaliação Heurística e Percurso Cognitivo requerem um baixo conhecimento prévio do avaliador para que sejam aplicados, o MIS requer que o avaliador tenha conhecimento da teoria subjacente, i.e. Engenharia Semiótica (ainda que não aprofundado). Por outro lado, a Avaliação Heurística e o Percurso Cognitivo podem requerer adaptações para novos domínios e tecnologias ([Pinelle et al., 2008] e [Blackmon et al., 2002], respectivamente), enquanto o MIS, por focar na comunicação projetista-usuário, pode ser aplicado sem adaptações a um amplo contexto de domínios e tecnologias [Reis e Prates, 2011]. Finalmente, uma análise comparativa dos três métodos indica que enquanto o tempo de aplicação da Avaliação Heurística e Percurso Cognitivo tendem a ser mais próximos, o do MIS já é bem maior [Bim et al., 2016].

Um exemplo de aplicação do MIS com finalidade científica é brevemente exposto no Quadro 2.6.

Quadro 2. 6: Exemplo de Aplicação do MIS [Santos e Prates, 2010]

Sistema inspecionado: Wikipédia

Objetivo: Analisar as estratégias comunicadas na interface da Wikipédia para indicar a qualidade dos artigos publicados.

Motivação: A Wikipédia enfrenta riscos associados à qualidade do conteúdo de seus artigos. Assim, é fundamental que o usuário seja capaz de avaliar para um dado artigo publicado qual a qualidade daquela informação e que credibilidade pode atribuir ao artigo.

Tipo de aplicação: Científica

Questão de pesquisa: Que estratégias são usadas na interface de enciclopédias colaborativas referentes à qualidade um artigo?

Escopo: Páginas da Wikipédia (versão em inglês) que dão acesso ao conteúdo e aquelas que relatam algo sobre a qualidade dos artigos.

Cenário: O cenário[7] prevê o uso da Wikipédia por um leitor que quer saber qual a qualidade do artigo de interesse, e também por uma usuária que vai escrever um artigo sobre um novo tema na Wikipédia.

Número de avaliadores: 1

Experiência do avaliador: Experiência prévia em avaliações de interfaces, e aplicação do MIS, inclusive em sistemas colaborativos.

Resultados: Foram identificadas 9 estratégias associadas ao controle de qualidade do conteúdo na Wikipédia, que foram então classificadas em dois grupos de acordo com quem era o responsável por aplicar a estratégia – o usuário ou o sistema. Além da identificação das estratégias, a aplicação do MIS gerou resultados técnicos relativos a problemas de comunicabilidade da interface relacionados à apresentação das estratégias.

2.6. Aplicação em Prática e Pesquisa

Os métodos apresentados neste capítulo podem ser utilizados em uma ampla variedade de situações, contextos e domínios – ambientes e processos de trabalho, interações sociais, atividades de lazer e entretenimento, atividades colaborativas, etc. Têm, portanto, **aplicação em diferentes áreas da computação**. Além disso, com base na discussão realizada dentro do contexto da área de IHC [de Souza e Leitão, 2009], é interessante destacar também que, segundo suas finalidades, os métodos qualitativos podem ter **aplicação científica ou técnica,** sempre tendo como meta a exploração de novas perspectivas e significados, bem como, por consequência, a inovação. De modo bem simples e introdutório, essas modalidades de aplicação e seus usos pelas áreas da Computação são discutidas.

[7] O cenário não é apresentado no artigo, apenas no Relatório Técnico citado que descreve a aplicação do MIS mais detalhadamente.

2.6.1. Uso na Pesquisa Científica

Muito embora a pesquisa científica em computação seja majoritariamente quantitativa, mais e mais estudos da área buscam investigar questões e problemas que melhor se coadunam com o paradigma das pesquisas qualitativas. Como abordado ao longo do capítulo, a investigação de aspectos não quantificáveis do uso de tecnologias computacionais tais como a identificação, a descrição e a análise de processos humanos e sociais e a exploração de diferentes contextos socio-técnicos são exemplos de pesquisas que buscam construir uma rede de significados em torno do desenvolvimento, avaliação e uso de tecnologias computacionais de modo a avançar, não a qualidade do processo de desenvolvimento de um sistema ou aplicativo específico, mas visando avançar o conhecimento científico a respeito de *o que*, *como* e *por que* produzir tecnologia computacional.

No contexto científico, o uso de métodos qualitativos é motivado pelo potencial que estes apresentam para apoiar a inovação, posto que são fundamentalmente ferramentas para a exploração de novos fenômenos e de significados a ele relacionados. Para serem usados com finalidades científicas esses métodos ganham especificidades em termos do tipo de resultados obtidos, em termos do processo de definição da questão sob exame e em função dos mecanismos adotados para validar suas conclusões.

Em relação à **definição do problema sob estudo**, o uso dos métodos deve partir de uma questão de certo nível de abstração, que vá além da exploração de alternativas de requisitos, de design e de avaliação de uma instância específica de tecnologia. Embora a investigação de um problema, na abordagem qualitativa, esteja sempre relacionada a um fenômeno concreto em uma manifestação concreta, seus objetivos vão além do exame situacional, para propor *frameworks* de interpretação para novos casos a serem investigados.

Em relação ao **tipo de resultados obtidos**, a partir da experiência dentro do contexto de IHC [de Souza e Leitão, 2009], é possível considerar que, se usado para fins científicos, a aplicação de métodos qualitativos deve visar e ser capaz de:

- Enunciar **novos problemas** no campo da área da computação (em contraposição a explorar, reproduzir e replicar antigos conhecimentos em novos exemplos e casos);
- Identificar **novas condições de contorno ou desafios** para antigos problemas**;**
- Propor **novas soluções para antigos problemas** por meio da exploração de novos exemplos e ocorrências do fenômeno; ou
- Contribuir para a **formulação de novos conceitos, teorias ou métodos** para a área de estudo.

Para que seus resultados científicos sejam validados entre a comunidade de pesquisadores, tal como ocorre com os estudos quantitativos, seus resultados devem ser submetidos à validação. O processo de validação científica de estudos qualitativos é, no entanto, distinto daqueles que compõem a validação de estudos estatísticos e experimentais [Denzin e Lincoln, 2006; Creswell, 2009]. Em pesquisas quantitativo-experimentais, testa-se, sobretudo, a replicabilidade ou a probabilidade de previsão para garantir a validade do processo de coleta e análise dos dados. Já em estudos qualitativos, já está consolidada uma série de critérios de validação que respeitam as características dos métodos qualitativos, e atestam a inadequação do uso de critérios quantitativos para a validação de pesquisas qualitativas. A validação se dá da perspectiva interna e externa, com novos significados atrelados a essas perspectivas.

Dado que os resultados da pesquisa qualitativa são frutos do processo de interpretação única, rigorosa e sistemática do pesquisador, da **perspectiva interna**, o processo de validação se dá a partir da **plausabilidade e rastreabilidade** das interpretações, ou seja, da possibilidade de se estabelecer e rastrear relações claras e objetivas entre o dado concreto e seu processo de interpretação [Denzin e Lincoln, 2006; Leitão, 2009]. Esse processo de validação se faz pelo uso extensivo de exemplos dos dados coletados. Com isto, a validade dos processos internos de realização da pesquisa científica de base qualitativa pode ser assegurada.

Já da **perspectiva externa**, para validar os resultados de uma pesquisa qualitativa, é necessário proceder à triangulação dos resultados**.** A **triangulação** visa gerar diferentes perspectivas sobre a questão de estudo, checando a consistência entre elas, e não sua homogeneidade e replicabilidade [Denzin e Lincoln, 2006; Creswell, 2009]. O produto da triangulação, quando consistente, é um conjunto de significados e categorias interpretativas articuladas, capazes de gerar uma compreensão profunda do contexto pesquisado e, além disso, um framework interpretativo que pode ser (re)aplicado em outros contextos de investigação.

A **triangulação** pode feita quando se compara processos interpretativos realizados **por diferentes pesquisadores** que se debruçam separadamente sobre o mesmo conjunto de dados gerados pela aplicação de um único método. Este é o caso, por exemplo, de um estudo no qual dois pesquisadores analisam isoladamente o material discursivo de um conjunto de entrevistas e consolidam seus resultados. A triangulação pode também ser realizada pela comparação dos resultados gerados pela aplicação, por um mesmo pesquisador, **de diferentes métodos** para analisar o mesmo problema de estudo. Um exemplo deste tipo de triangulação é a validação dos resultados coletados pela aplicação do MIS, comparando-os com os resultados de um conjunto de entrevistas com usuários. Define-se estas duas formas de triangulação como **endógenas** [de Souza e Leitão, 2009], uma vez que o domínio e a questão de estudo são estáveis, havendo a variação do método ou do sujeito que o executa.

A triangulação pode ser ainda **exógena** [de Souza e Leitão, 2009], quando o mesmo método é aplicado em domínios distintos para a análise comparativa de seus resultados. Neste caso, o problema e o método de investigação são estáveis, variando o domínio no qual é aplicado. É o caso, por exemplo, de uma pesquisa sobre sistemas de recomendação, composta da aplicação do MIS em um sistema de compras e em um site de encontros interpessoais.

Em todos os casos, a triangulação visa gerar diferentes perspectivas sobre a questão de estudo [Denzin e Lincoln, 2006], checando a consistência entre elas, e não sua homogeneidade. O produto da triangulação é um conjunto de significados e categorias interpretativas capazes de gerar uma compreensão profunda de um problema cientifico e, além disso, um *framework* interpretativo que pode ser (re)aplicado em outros contextos de investigação [Leitão, 2009].

2.6.2. Aplicação no Ciclo de Desenvolvimento

Diferentemente do uso científico dos métodos qualitativos, no contexto técnico, a investigação qualitativa pode usada para inovação de um produto tecnológico ou mesmo na introdução de inovações no processo de desenvolvimento da tecnologia. Em relação à geração de **produtos tecnológicos**, em termos gerais, podemos caracterizar o processo de

design de um produto tecnológico em 3 atividades iterativas: análise da situação atual, síntese de uma intervenção e avaliação de uma nova situação [Barbosa e Silva, 2010].

A **análise da situação atual** envolve estudar e interpretar o contexto para o qual se pretende desenvolver um sistema. Assim, normalmente busca-se conhecimento sobre as pessoas envolvidas (seu perfil, necessidades, desejos, experiências, etc.), o ambiente em questão e formas de trabalho, dentre outros fatores que possam ser relevantes para o desenvolvimento do sistema. Assim, podemos ver que os métodos apresentados neste capítulo podem ser utilizados para diferentes fins nesta etapa. Por exemplo, estudos de caso, grupos de foco ou entrevistas e questionários podem ser utilizados para análise das pessoas envolvidas e para elicitação de requisitos; os estudos etnográficos podem ser utilizados para conhecimento do contexto ou formas de trabalho das pessoas, assim como para o mapeamento de demandas reais ou potenciais de novas ferramentas computacionais; e mesmo os métodos de inspeção podem ser utilizados para se obter conhecimento sobre sistemas (competidores ou não) que estejam sendo usados no contexto atual.

A etapa de **síntese de uma intervenção** envolve justamente a geração de um sistema que representa uma solução em que se articula o conhecimento adquirido na situação atual com o conhecimento sobre possibilidades e limitações que a tecnologia pode oferecer. Esta etapa foca principalmente em métodos de modelagem, prototipação e desenvolvimento do sistema. Nesta etapa também pode-se envolver métodos qualitativos etnográficos, como design participativo ou de coleta de dados do usuário com foco na geração de novas ideias, como por exemplo *braindraw* (processo em que se organiza um *brainstorming* visual para que um grupo de participantes gere novas ideias de design ou representações gráficas [Dray, 1992 apud Muller, 2003])[8].

Finalmente, a etapa de **avaliação de uma intervenção** envolve fazer uma apreciação do sistema gerado como solução e do seu impacto na situação atual. Métodos de inspeção foram gerados especificamente para apoiar a atividade de avaliação. Além disso, métodos de coleta de dados diretamente do usuário são frequentemente usados para se avaliar a experiência dos usuários com a tecnologia ou sua percepção e atitude em relação ao sistema ou ao impacto da sua introdução. Finalmente, métodos etnográficos que permitem a coleta em contexto real são úteis para se analisar aspectos relacionados à introdução da tecnologia, como a mesma está sendo utilizada ou seu impacto nas atividades ou relacionamentos em uma organização.

Vale ressaltar que as etapas são iterativas (e não sequenciais), então, os métodos descritos para avaliação de uma intervenção podem (e seria até recomendado que fossem) utilizados para a avaliação formativa, ou seja, durante o processo de design definido pela etapa de síntese de uma intervenção.

A **inovação de processos**, por sua vez, se refere aos estudos sobre o próprio processo de desenvolvimento das tecnologias, seja pela observação etnográfica, por entrevistas ou estudos de caso, em busca de alternativas criativas para o desenvolvimento de tecnologias digitais. Nesta direção, Sharp e colegas [2016] discutem a relevância de estudos etnográficos na pesquisa empírica de Engenharia de Software e propõem um

[8] Neste capítulo, não apresentamos design participativo ou métodos que podem ser úteis para se envolver usuários no processo de design, como por exemplo o *braindraw* citado ou mesmo *brainstorming*. Ao leitor interessado em design participativo recomendamos a leitura de [Muller, 2003].

conjunto de dimensões para auxiliar no planejamento de estudos etnográficos neste contexto.

2.7. Comentários Finais

Este capítulo buscou apresentar uma visão introdutória e abrangente da metodologia qualitativa de investigação, contextualizando-a para a área da computação. Além de uma visão instrumental dos métodos, ou seja, de *para que*, *como* e *quando* é mais interessante aplicá-los, buscou-se também chamar atenção para os pressupostos que envolvem a investigação qualitativa. Quisemos com isso destacar que não se trata tão-somente de uma escolha de ferramentas, mas principalmente de uma postura em relação ao tipo de conhecimento a ser obtido. Isso é muito importante e desafiador para aqueles que se iniciam neste tipo de abordagem, considerando que, em computação, a natureza algorítmica do objeto de estudo traz a previsibilidade como um conceito consequente. No contexto das pesquisas qualitativas dentro da computação, mais localizadas na fronteira da computação com as questões humanas e sociais, corre-se o risco de almejar uma previsibilidade comparável aos resultados lógico-matemáticos. E é justamente nesse contexto que um conflito entre concepções pode acontecer, uma vez que, na raiz das abordagens qualitativas está o pressuposto de que o conhecimento sobre questões humanas e sociais não se dá em bases preditivas. Neste capítulo esperamos ter transmitido um pouco dessa ideia e, juntamente com ela, o valor dos conhecimentos obtidos com métodos qualitativos: um conhecimento contextualizado, exploratório e em profundidade que se oferece como um framework conceitual para aplicação em novos contextos e para insumo e apoio à tomada de decisões complexas no desenvolvimento de tecnologias computacionais.

Referências

Baker, K., Greenberg, S., & Gutwin, C. (2002, November). Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work* (pp. 96-105). ACM.

Barbosa, S.D.J. e Silva, B.S. (2010) *Interação Humano-Computador.* Rio de Janeiro: Editora Campus-Elsevier.

Barbour, R. (2009). *Grupos focais: coleção pesquisa qualitativa.* Bookman Editora.

Bim, S.A., Salgado, L.C.C. e Leitão, C. F. (2016) Avaliação por Inspeção: Comparação entre Métodos de Base Prática, Cognitiva e Semiótica. In *Anais do XV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*.

Blackmon, M. H., Polson, P. G., Kitajima, M., & Lewis, C. (2002, April). Cognitive walkthrough for the web. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 463-470). ACM.

Blandford, A. E. (2013). Semi-structured qualitative studies. Soegard. M.., & Dam, R. F. (Eds). *The Encyclopedia of Human-Computer Interaction.* 2nd edition. The Interaction Design Foundation. Aarhus, Denmark.

Cairns, P., e Cox, A. L. (Eds.). (2008). *Research methods for human-computer interaction* (Vol. 12). New York (NY): Cambridge University Press.

Carter, S. e Mankoff, J. When participants do the capturing: the role of media in diary studies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2005 Apr 2 (pp. 899-908). ACM.

Cecchinato, M. E., Sellen, A., Shokouhi, M., & Smyth, G. (2016, May). Finding email in a multi-account, multi-device world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 1200-1210). ACM.

Chagas, B. A. (2015) *End-User Configuration in Assistive Technologies. A case study with a severely physically impaired user.* Dissertação de Mestrado. PUC-Rio. Disponível em http://www.dbd.puc-rio.br/pergamum/tesesabertas/1321831_2015_completo.pdf

Chin, J.P.; Diehl, V. & Norman, K. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '88), J. J. O'Hare (Ed.). ACM, New York, NY, USA, 213-218. DOI=http://dx.doi.org/10.1145/57167.57203.

Clarkson, E., e Arkin, R. C. (2007). Applying Heuristic Evaluation to Human-Robot Interaction Systems. In *Flairs Conference* (pp. 44-49).

Conselho Nacional de Saúde. (2016) *Resolução número 510/16: Especificidades Éticas das Pesquisas nas Ciências Sociais e Humanas e de outras que se utilizam de metodologias próprias dessas áreas.* Disponível em: http://conselho.saude.gov.br/resolucoes/2016/Reso510.pdf

Conselho Nacional de Saúde. (2012) *Resolução número 466/12 sobre pesquisas envolvendo seres humanos.* Disponível em: http://conselho.saude.gov.br/resolucoes/2012/Reso466.pdf

Corbin, J., e Strauss, A. (2008). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3rd ed.). Thousand Oaks, CA: Sage

Creswell, J. W. (2009) Research Design: *Qualitative, Quantitative, and Mixed Methods Approaches.* 3rd Edition. Los Angeles: Sage Publications.

da Matta, R. (1978) O Ofício do Etnólogo ou como ter anthropological blues. *Boletim do Museu Nacional*, Rio de janeiro. Disponível em: http://www.ppgasmn-ufrj.com/uploads/2/7/2/8/27281669/boletim_do_museu_nacional_27.pdf *(Última visita em maio, 2017).*

de Souza, C. S. (2005). *The semiotic engineering of human-computer interaction*. MIT press.

de Souza, C.S. e Leitão, C.F. (2009) *Semiotic engineering methods for scientific research in HCI.* San Francisco, Calif.: Morgan & Claypool.

de Souza, C. S., Leitão, C. F., Prates, R. O., & da Silva, E. J. (2006). The semiotic inspection method. In *Proceedings of VII Brazilian symposium on Human factors in computing systems* (pp. 148-157). ACM.

de Souza, C. S.; Leitão, C.F.; Prates, R. O.; Bim, S.A. e da Silva, E.J. (2010) Can inspection methods generate valid new knowledge in HCI? The case of semiotic inspection. *International Journal of Human-Computer Studies*, v. 68, p. 22-40.

Denzin, N. K. e Lincoln, Y. S. (2006) *O Planejamento da Pesquisa Qualitativa: Teorias e abordagens.* Porto Alegre: ARTMED, 2006.

Jamison-Powell, S., Briggs, P., Lawson, S., Linehan, C., Windle, K., & Gross, H. (2016). PS. I Love You: Understanding the Impact of Posthumous Digital Messages. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 2920-2932). ACM.

Japiassú, H. e Marcondes, D. (1996) *Dicionário básico de filosofia.* Rio de Janeiro, Jorge Zahar Editor.

Lazar, J., Feng, J. H., e Hochheiser, H. (2010). *Research methods in human-computer interaction*. John Wiley& Sons.

Leitão, C. (2009). Métodos Qualitativos de Pesquisa Científica. *Computação Brasil: Interação Humano-Computador no Brasil*, p. 22-23.

Leitão, C. F., Silveira, M. S., & de Souza, C. S. (2013). Uma introdução à engenharia semiótica: conceitos e métodos. In *Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems* (pp. 356-358). Brazilian Computer Society.

Mack, R. L., e Nielsen, J. (Eds.). (1994). *Usability inspection methods* (pp. 1-414). New York, NY: Wiley & Sons.

Mankoff, J.; Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., & Ames, M. (2003). Heuristic evaluation of ambient displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 169-176). ACM.

Muller, M. J. (2003). Participatory design: the third space in HCI. *Human-computer interaction: Development process*, 4235, 165-185.

Nicolaci-da-Costa, A. M.; Leitão, C. F.; Dias, D. R. (2004) Como conhecer usuários através do Método de Explicitação do Discurso Subjacente (MEDS). In *VI Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*, IHC, 47-56.

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods.* John Wiley& Sons, New York, NY.

Nielsen, J. (1995a) "How to conduct a heuristic evaluation." *Disponível em:* https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/ *(Última visita em abril, 2017).*

Nielsen, J. (1995b) "10 Usability Heuristics for User Interface Design." *Disponível em: https://www.nngroup.com/articles/ten-usability-heuristics/ (Última visita em abril, 2017).*

Norman, D. A. (1986). Cognitive engineering. *User centered system design: New perspectives on human-computer interaction*, *3161*.

Norval, C., Arnott, J. L., & Hanson, V. L. (2014). What's on your mind?: investigating recommendations for inclusive social networking and older adults. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 3923-3932). ACM.

Palen, L., and Salzman, M.. "Voice-mail diary studies for naturalistic data capture under mobile conditions." In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. ACM, 2002.

Peirce, C. (1992–1998). The essential Peirce (Vols. 1 & 2). In N. Houser e C. Kloesel (eds.) The Peirce Edition Project. Bloomington: Indiana University Press.

Pinelle, D., Wong, N., and Stach, T. (2008). Heuristic evaluation for games: usability principles for video game design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1453-1462). ACM.

Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies*, *36*(5), 741-773.

Prates, R. O., e Barbosa, S. D. J. (2007). Introdução à teoria e prática da interação humano computador fundamentada na engenharia semiótica. Atualizações em informática, 263-326.

Prates, R. O., de Souza, C. S., & Barbosa, S. D. (2000). Methods and tools: a method for evaluating the communicability of user interfaces. *interactions*, *7*(1), 31-38.

Preece, J., Sharp, H., e Rogers, Y. (2015). Interaction Design: Beyond Human-Computer *Interaction*, 486. John Wiley and Sons.

Randall, D. e Rouncefield, M. (2013) Ethnography. In: Soegaard, Mads and Dam, Rikke Friis. (Org.). *Encyclopedia of Human-Computer Interaction*. 2ed. Aarhus: Interaction Design Foundation.

Reddy, M., Dourish, P., and Pratt, W. (2006). Temporality in Medical Work: Time also matters. *Computer-Supported Cooperative Work*, 15(1), 29-53.

Reis, D. S., & Prates, R. O. (2011, October). Applicability of the semiotic inspection method: a systematic literature review. In *Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction* (pp. 177-186). Brazilian Computer Society.

Santos, R. L., e Prates, R. O. (2010). Estratégias para comunicar qualidade na Wikipedia. In *Proceedings of the IX Symposium on Human Factors in Computing Systems* (pp. 71-80). Brazilian Computer Society.

Seidman, I. (1998) *Interviewing as Qualitative Research: a guide for researchers in education and social sciences*. New York, Teachers College Press, 1998.

Sharp, H., Dittrich, Y., & de Souza, C. R. (2016). The role of ethnographic studies in empirical software engineering. *IEEE Transactions on Software Engineering*, 42(8), 786-804.

Sommerville, I. (2011) *Engenharia de software*. São Paulo, SP, Pearson Prentice Hall, 9ª Edição.

Weiss, R.S. (1995) *Learning from Strangers: the art and method of qualitative interview studies.* New York, The Free Press.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In *Usability inspection methods* (pp. 105-140). John Wiley & Sons, Inc..

Wisniewski, P., Xu, H., Rosson, M. B., Perkins, D. F. and Carroll, J. M. 2016. Dear Diary: Teens Reflect on Their Weekly Online Risk Experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). ACM, New York, NY, USA, 3919-3930. DOI: https://doi.org/10.1145/2858036.2858317

Wright, P. C., e Monk, A. F. (1991). The use of think-aloud evaluation methods in design. *ACM SIGCHI Bulletin*, 23(1), 55-57.

Yin, R.K. (2009) *Case Study Research, Design and Methods*. Newbury Park, Sage Publications, 4th Edition.

Capítulo 3

Preservação de Privacidade de Dados: Fundamentos, Técnicas e Aplicações

Felipe T. Brito, Javam C. Machado

Abstract

Many organizations have perform data analysis over published data to find hidden patterns and foresee future tendencies. The data mining process is only possible because data can be reached through query services or open access to published data. However, published data may carry out information that uniquelly identify individuals, which may lead to privacy violation. Keeping the utility of the data for mining and still maintaining individual's privacy is a scientific problem that has been studied over the past few years. The objective of this chapter is to present the main concepts of data privacy preserving and to describe the techniques capable of assuring that no one can be reidentified from their publish data. Additionally it describes real world applications that apply these techniques as a way of preserving individual's privacy, while keeping data utility for further data analysis when requested.

Resumo

Muitas organizações realizam análises importantes sobre dados a fim de descobrir padrões ocultos e prever tendências futuras. Para que muitas dessas análises sejam realizadas, é necessário que os dados estejam disponíveis para acesso, seja por meio de publicações ou de serviços de consulta. Entretanto, dados acessíveis pelo público podem conter informações que identificam unicamente indivíduos, causando assim uma violação de privacidade. Manter a utilidade dos dados para que análises sejam realizadas e, simultaneamente, garantir a privacidade dos indivíduos é um problema que tem recebido bastante atenção nos últimos anos. Este capítulo tem por objetivo apresentar os principais conceitos em torno da preservação de privacidade de dados, além das técnicas para assegurar que indivíduos não possam ser reidentificados a partir do compartilhamento de suas informações. Adicionalmente, são demonstradas aplicações em cenários reais que utilizam as técnicas apresentadas como forma de preservação de privacidade, enquanto buscam reter a maior quantidade de informação possível para eventuais análises.

3.1. Introdução

Atualmente, grandes volumes de dados têm sido coletados por governos, corporações ou mesmo instituições ao redor do mundo. Dados são bens muito valiosos para as organizações dos mais diversos tipos, sejam elas bancárias, de seguros, de varejo, de saúde, etc. Por exemplo, uma base de dados de serviços de saúde pode conter informações sobre as reações de seus pacientes a um determinado medicamento ou tratamento. Essas informações podem ser úteis para companhias farmacêuticas fabricarem e distribuírem medicamentos. Bancos também gostariam de entender como seus clientes estão utilizando cartões de crédito, para assim oferecerem outros tipos de serviços com confiança de crédito. Seria ideal que comércios e supermercados pudessem entender o comportamento de seus clientes, e assim ofertarem produtos e serviços de maneira mais assertiva. Outro exemplo seria um conjunto de dados coletado por iniciativas privadas que contém informações sobre mobilidade urbana. Essas informações podem ser muito bem aproveitadas para identificar e prever concentrações de fluxos de automóveis, melhorar o transporte público, planejar obras de infraestrutura, entre outros benefícios. Muitos desses padrões estão escondidos na grande quantidade de dados coletados.

A mineração de dados é o processo de extração de informações, não conhecidas a priori, a partir de grandes conjuntos de dados [28]. O sucesso da mineração ocorre devido à disponibilidade dos dados com qualidade e ao compartilhamento efetivo das informações. Dessa forma, é possível realizar análises para descobrir padrões ocultos e/ou para prever tendências futuras, permitindo tomadas de decisão baseadas no conhecimento e não apenas em opiniões próprias ou intenções. Muitas informações coletadas podem servir para que empresas forneçam serviços de valor agregado para seus clientes, o que, por sua vez, resulta em maior receita e consequentemente em maior desenvolvimento para a sociedade. Para realizar um processo de análise eficiente é necessário que os dados estejam disponíveis de alguma forma. Se não há dados confiáveis em quantidade suficiente para a realização de pesquisas e análises, torna-se obscuro o uso da informação em prol do desenvolvimento da sociedade.

Uma forma de disponibilização e utilização de dados que tem ganhado bastante visibilidade nos últimos anos é o modelo de dados abertos [27]. Nesse modelo, dados são geralmente disponibilizados por instituições governamentais, em um formato computacional aberto e processável por máquina, isto é, que não possui restrição tecnológica para ser utilizado, a fim de permitir cidadãos comuns, empresas, instituições de ensino e organizações não-governamentais utilizá-los de maneira inovadora, gerando valor para a sociedade. Dessa forma, dados abertos são considerados fundamentais para acelerar o progresso da ciência e possibilitar investimentos em educação, saúde, segurança pública, transporte, uso racional de recursos e desenvolvimento sustentável. Dentre os benefícios dessa prática destacam-se: maior transparência sobre as atividades governamentais; acompanhamento das políticas públicas; melhoria da troca de informações entre órgãos e esferas de governo; incentivo à sociedade em desenvolver soluções para o bem comum; fomento a novos mercados de tecnologia da informação; estímulo a inovação e pesquisa; entre muitos outros.

Sob outra perspectiva, ao mesmo tempo que a disponibilização de dados para análises e descoberta de padrões é fundamental para o desenvolvimento da sociedade contem-

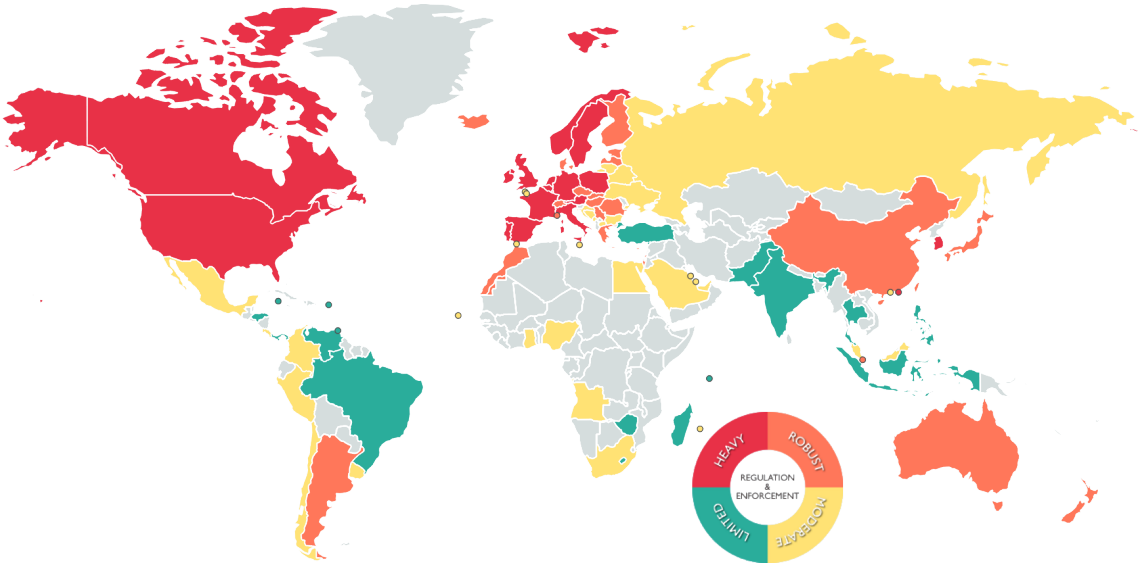

Figura 3.1. Nível de regulamentação das leis de proteção de dados ao redor do mundo (Fonte: [2]).

porânea, essa liberação pode colocar em risco a privacidade dos indivíduos. Se os dados coletados forem mantidos em seu formato original, ou se esses dados caírem em mãos erradas por meio de vazamento de informações, indivíduos podem ser facilmente identificados, e assim, ter sua privacidade violada. Quando dados são publicados, seja para qualquer tipo de análise, um usuário malicioso pode ser capaz de descobrir informações sensíveis sobre indivíduos por meio de seus semi-identificadores. Semi-identificadores são dados que podem ser combinados com informações externas e assim utilizados para reidentificar indivíduos. Logo, um adversário poderá descobrir que o registro no conjunto de dados publicado pertence a um indivíduo com uma probabilidade alta. Dessa forma, caso uma publicação aconteça de maneira ingênua, tal fato pode levar a sérios riscos de violação de privacidade, uma vez que esses dados fornecem informações sensíveis, tais como costumes sociais, doenças, preferências religiosas, sexuais, entre outras.

Para lidar com a privacidade de dados dos cidadãos, governos ao redor do mundo definem regulamentos obrigatórios os quais as organizações devem seguir, por exemplo, HIPAA (*Health Insurance Portability and Accountability Act*) [5] nos Estados Unidos, FIPPA (*Freedom of Information and Protection of Privacy Act*) [4] no Canadá, Diretivas de Proteção de Dados da União Europeia [3], entre outros. No Brasil, ainda não há uma consolidação de leis sobre privacidade nos moldes de países como Estados Unidos, Canadá, ou mesmo da União Europeia, todavia, o direito à privacidade no país é regido pela Constituição Federal de 1988, pelo Novo Código Civil (lei 10.406/02, em especial o artigo 21) e também pelo Marco Civil da Internet (lei número 12.965, sancionada em 23 de abril de 2014). A Figura 3.1 apresenta o nível de regulamentação das leis de proteção de dados ao redor do mundo. O objetivo dessas leis é controlar o acesso às informações que as organizações detêm o controle. Tais organizações também podem ter suas próprias políticas de privacidade. Mesmo assim, elas necessitam tomar medidas concretas para proteger os dados de seus usuários, investigando métodos e ferramentas para preservar dados confidenciais e não permitir que adversários violem sua privacidade.

Garantir a privacidade dos indivíduos tem impacto direto na qualidade dos dados publicados, visto que a privacidade e a utilidade dos dados são princípios inversamente proporcionais. Quanto maior a privacidade, menor será a utilidade dos dados para análise, e vice-versa. Manter a utilidade dos dados e simultaneamente garantir a privacidade dos indivíduos é um problema extremamente complexo. Por isso, vários modelos de privacidade de dados têm sido propostos por pesquisadores com o objetivo de resolver esta questão. Um modelo pode ser classificado em sintático ou diferencial. O primeiro refere-se a uma condição na qual os dados devem obedecer antes de serem disponibilizados. Ou seja, os dados só serão publicados se obedecerem a um determinado critério. Já o segundo propõe-se a disponibilizar resultados de consultas, tendo como base um modelo matemático onde são compartilhadas apenas informações estatísticas sobre o conjunto de dados original.

Este capítulo tem por objetivo aprofundar conhecimentos em torno do tema privacidade, particularmente sobre fundamentos e técnicas para assegurar que indivíduos não possam ser reidentificados, além de descrever aplicações da privacidade de dados no mundo real, que visam balancear utilidade e privacidade. A Seção 3.2 apresenta os princípios básicos sobre o tema, bem como a definição de privacidade e como ela pode ser garantida. A Seção 3.3 esclarece o problema da utilidade dos dados em oposição à garantia de privacidade. Já a Seção 3.4 apresenta uma visão geral das técnicas mais utilizadas para anonimizar dados, destacando a generalização, a supressão e a perturbação de dados. Métricas para determinar a qualidade dos dados anonimizados são apresentadas na Seção 3.5. Os modelos sintáticos de privacidade mais comuns são descritos na Seção 3.6. O modelo de privacidade diferencial, proposto nos últimos anos como novo paradigma de privacidade, é detalhado na Seção 3.7. A Seção 3.8 exemplifica a utilização das técnicas vistas neste capítulo por meio de aplicações reais. E por fim, a Seção 3.9 apresenta as considerações finais do capítulo.

3.2. Fundamentos da Privacidade de Dados

O estudo da privacidade abrange disciplinas desde a filosofia à ciência política, teoria política e legal, ciência da informação e, de forma crescente, engenharia e ciência da computação. Um consenso entre os pesquisadores é que privacidade é um assunto complexo, com muitas questões envolvidas. O conceito de privacidade está relacionado a pessoas, mais precisamente ao direito que as pessoas têm em manter um espaço pessoal, sem interferências de outras pessoas ou organizações. Dessa forma, compete a elas decidir manter suas informações sob seu exclusivo controle, ou informar, decidindo a quem, quando e onde suas informações estarão disponíveis. Contudo, a privacidade tem sido utilizada como moeda de troca em serviços "gratuitos", nos quais o usuário provê informações pessoais para fazer uso desses serviços. Nesse contexto, quais as informações que, de fato, não devem ser divulgadas? A que tipos de ataque minhas informações estão sujeitas? Quais as maneiras de proteger meus dados? Nesta seção iremos discutir essas e outras questões relativas aos fundamentos da privacidade de dados.

3.2.1. O que é Privacidade e por que ela é importante

Muitas definições de privacidade foram elaboradas e defendidas ao longo do tempo. O reconhecimento do direito do indivíduo à privacidade está enraizado na história. De acordo

com Laurant [29] a mais antiga referência à privacidade remonta ao Alcorão e às declarações de Maomé. Mais tarde, no século XIX, o conceito de privacidade foi estendido à aparência pessoal dos indivíduos, provérbios, atos, crenças, pensamentos, emoções, sensações, etc. Já na década de 1980, Gavison [23] define privacidade como uma condição medida em termos de grau de acesso que outros têm a você, com base na informação, atenção e proximidade. Dessa forma existem, em termos jurídicos, três princípios fundamentais e independentes que compõem a privacidade: o sigilo (ou segredo); o anonimato; o isolamento (ou solidão). Já na última década, Daniel Solove [40] propôs uma taxonomia de privacidade que consiste em quatro grupos de atividades: (i) coleta de informações; (ii) processamento de informações; (iii) divulgação de informações; (iv) invasão. Cada grupo contém uma variedade de atividades que podem criar problemas relacionados à privacidade.

Independentemente do tempo e do meio onde os indivíduos estão inseridos, existe a necessidade de privacidade, mesmo que alguém a desconheça ou não se importe com ela. A privacidade encontra uma barreira para a sua existência no mundo virtual, devido à facilidade da transmissão da informação. Quando conectado a uma rede, um dispositivo (computador, notebook, tablet, celular, etc.) pode estar vulnerável a todo tipo de ataque externo. As informações contidas nestes dispositivos podem ser acessadas e divulgadas para o resto da rede.

É bastante comum os usuários confundirem os conceitos de privacidade e de segurança. Apesar de privacidade e segurança serem temas relacionados, suas definições remetem a polos opostos, como duas pontas de uma gangorra. É inevitável que um concorra com o outro. Por exemplo, recentemente, mesmo diante do terrorismo que acomete os Estados Unidos, grande parte dos americanos afirmaram que não estão dispostos a compartilhar seus e-mails pessoais, mensagens de texto, telefonemas e registros de atividade na Web com investigadores anti-terrorismo [6]. Em outras palavras, eles não estão dispostos a abrir mão de sua privacidade em troca de segurança.

Quando se trata de dados, a segurança visa regular o acesso durante todo o ciclo de vida do dado, enquanto a privacidade define como será realizado esse acesso, na maioria das vezes com base em leis e políticas de privacidade. Neste ponto, também surge o conceito de controle de acesso como forma de fornecer segurança a um conjunto de dados. O controle de acesso se refere a regras específicas de quem está autorizado a acessar (ou não) determinados recursos, isto é, quando um conjunto de usuários está apto a acessar um conjunto de dados. A privacidade aqui está associada a regras de controle de acesso efetivas, que permitem a revelação da informação apenas por usuários autorizados. Contudo, a privacidade dos indivíduos não está garantida apenas com o controle de acesso eficiente, visto que os usuários com acesso àquelas informações podem ser maliciosos, e assim capazes de divulgar informações sensíveis acerca daqueles indivíduos.

Incidentes envolvendo violação de privacidade têm ocorrido em diversos lugares ao redor do planeta. Em 2014, milhares de funcionários do serviço de ambulância e requerentes de benefícios de programas de habitação do Reino Unido tiveram seus dados pessoais violados acidentalmente [7]. Informações como idade, gênero e religião de mais de 2.800 funcionários foram publicados na Web por engano. Outro fato foi divulgado pela *Community Health Systems* (CHS), nos Estados Unidos, a qual afirmou que cerca

de 4,5 milhões de dados de identificação de pacientes haviam sido roubados [1]. Neste caso, nenhum dado médico/clínico ou número de cartão de crédito foi violado, mas os dados incluíam informações que seriam úteis para descoberta de identidade, como nomes de pacientes, endereços, datas de nascimento, números de telefone e números de seguro social de milhões de indivíduos. Quando estes incidentes desfavoráveis acontecem, as organizações enfrentam processos legais, prejuízos financeiros, perda de imagem e, acima de tudo, a perda de seus clientes.

Cada vez mais organizações têm desembolsado milhões de dólares para proteger as informações de seus clientes. E por que gastar tanto recurso assim? É fato que algumas informações são tão importantes que não podem ser reveladas. Geralmente as informações que necessitam de maior privacidade e que não podem sofrer ataques são as informações que identificam unicamente indivíduos ou que são sensíveis a eles, como por exemplo condição financeira, doenças, etc. As características de cada uma dessas informações são detalhadas a seguir.

3.2.2. Microdados e Privacidade

Dados podem ser representados por uma tabela, onde cada coluna corresponde a um atributo e cada linha a um registro. Esses dados são denominados microdados (ou dados tabulados). Em geral, microdados têm um conjunto fixo de atributos, que são comuns em uma coleção de registros. Quatro tipos de atributos podem existir em conjuntos de dados desse tipo [20]:

- **Identificadores explícitos**: são atributos que identificam unicamente indivíduos, tais como "nome", "CPF", "e-mail", etc., e são sempre removidos antes de serem publicados;
- **Semi-identificadores**: são todos aqueles atributos que não são identificadores explícitos mas podem potencialmente identificar um indivíduo, especialmente quando agrupados. São exemplos de semi-identificadores em dados relacionais "data de nascimento" e "CEP";
- **Atributos sensíveis**: contém informações sensíveis sobre indivíduos, tais como "doença", "salário", etc.;
- **Atributos não sensíveis**: é qualquer tipo de atributo que não se enquadra em nenhuma das categorias anteriores.

Dados sensíveis armazenados em sistemas de banco de dados sofrem riscos de divulgação não autorizada. Por esse motivo, tais dados precisam ser protegidos. Para exemplificar os tipos de atributos, tendo em vista os riscos de divulgação não autorizada, considere o exemplo de microdados de duas tabelas, uma de cliente bancário e outra de conta. A Tabela 3.1 mostra dados de clientes contendo identificadores explícitos e semi-identificadores. A tabela considerada como tal, isto é, sem nenhuma modificação, não possui nada de tão confidencial, pois a maioria das informações nela contidas também estão disponíveis em bases de dados públicas ou em redes sociais, como por exemplo o Facebook.

Identificadores Explícitos		Semi-identificadores			
ID	**Nome**	**Idade**	**Gênero**	**Endereço**	**Telefone**
1	David	22	Masculino	Av. L	98533 1234
2	John	23	Masculino	Av. K	98772 2531
3	Helton	25	Masculino	Av. K	98156 0092
4	Maria	32	Feminino	Rua J	99913 9026

Tabela 3.1. Exemplos de identificadores explícitos e semi-identificadores em dados tabulados de clientes.

Diferente da Tabela 3.1, a Tabela 3.2 apresenta dados de contas bancárias de seus usuários contendo algumas informações consideradas sensíveis e que não devem ser divulgadas. Todavia, quando acessada de forma isolada, possui informações sobre seus clientes que são úteis para análises e descoberta de padrões.

Identificadores Explícitos	Atributos Sensíveis		
ID	**Conta**	**Tipo**	**Saldo (R$)**
1	2234-0	Corrente	1.033,25
2	7749-2	Corrente	814,92
3	8491-7	Corrente	515,09
4	5723-1	Poupança	2.194,79

Tabela 3.2. Exemplos de identificadores explícitos e atributos sensíveis em dados tabulados de conta bancária.

A partir da divulgação das Tabelas 3.1 e 3.2, adversários podem obter informações de uma vítima e serem capazes de explorar conhecimento sobre ela por meio de ligação de informações. Para isso, eles associam registros públicos a um indivíduo alvo, cujas informações estão contidas no conjunto de dados publicado, violando assim sua privacidade. Além disso, adversários também são capazes de inferir valores de atributos sensíveis a partir desse conhecimento.

3.2.3. Conhecimento Adversário e Ataques à Privacidade

Uma violação de privacidade ocorre por meio de um ataque, i.e., quando um adversário é capaz de associar o proprietário de um dado a um registro em um conjunto de dados, utilizando um conhecimento previamente adquirido de fontes externas. Por exemplo, o adversário pode saber que a vítima mora ao lado de sua residência, assim ele pode inferir informações como endereço, CEP, gênero da vítima, etc. O adversário pode também utilizar dados de serviços baseados em localização, como um *checkin* em uma rede social realizado por uma vítima em uma determinada localização. O adversário pode ainda ter acesso a dados abertos de uma vítima, caso ela seja funcionária de órgãos públicos, por exemplo. Dessa forma, o conhecimento adversário é tido muitas vezes como imprevisível e deve ser considerado em soluções de preservação de privacidade, mesmo diante da

incerteza de como ele foi obtido.

Um adversário é capaz de violar a privacidade de indivíduos através dos seguintes ataques:

- **Ataque de Ligação ao Registro**: o objetivo do adversário é reidentificar o registro de um indivíduo específico ou de um indivíduo qualquer, cujas informações aparecem no conjunto de dados publicado;
- **Ataque de Ligação ao Atributo**: nesse tipo de ataque o adversário pode ser capaz de inferir atributos sensíveis de uma vítima mesmo sem reidentificar seus registros, baseando-se no conjunto de valores sensíveis associados ao grupo no qual a vítima pertence;
- **Ataque de Ligação à Tabela**: tanto os ataques de ligação ao registro quanto de ligação ao atributo assumem que o atacante sabe que o registro da vítima foi publicado. Nesse tipo de ataque o adversário está interessado em inferir com convicção a presença ou a ausência da vítima nos dados publicados;
- **Ataque Probabilístico**: esse tipo de ataque não foca em quais registros, atributos ou tabelas o atacante pode associar informações sensíveis a indivíduos, mas sim destaca como o atacante mudaria seu pensamento probabilístico acerca de um indivíduo após ter acessado o conjunto de dados publicado.

3.2.4. Maneiras de Proteger Dados Sensíveis

Uma das tarefas mais árduas no tema de privacidade e segurança da informação é proteger dados sensíveis de possíveis ataques. Quando um conjunto de dados é disponibilizado para fins estatísticos, de pesquisa, ou de testes, técnicas de preservação de privacidade são necessárias para evitar a descoberta de informações sensíveis por usuários maliciosos. Por exemplo, um hospital disponibiliza publicamente dados sobre seus pacientes para auxiliar pesquisadores da área médica a descobrirem causas de doenças, ou para estatísticos afirmarem a frequência da ocorrência de um determinado vírus. Uma vez que esses dados contêm informações sensíveis sobre pacientes, tal hospital não deve liberá-los de uma maneira ingênua, devido ao alto risco de violação da privacidade. Como forma de proteger efetivamente a privacidade dos indivíduos, o detentor dos dados precisa garantir que eventuais descobertas de informações não ocorram no conjunto de dados disponibilizado.

Existem três grandes maneiras de contornar esse problema: criptografia; tokenização; anonimização. A criptografia é considerada uma das técnicas mais antigas para se proteger dados e, quando bem executada, se torna uma técnica bastante robusta no quesito privacidade. Essa técnica utiliza um algoritmo capaz de embaralhar matematicamente dados sensíveis, gerando substitutos ilegíveis. Esses substitutos podem ser transformados de volta, para seus valores originais, através da utilização de uma chave de acesso. Nesse caso, dados criptografados não são legíveis e consequentemente não são úteis para análises. Outro quesito acerca da técnica de criptografia é o gerenciamento da chave de acesso. Caso ela não seja bem controlada e caia em mãos erradas, há uma total perda de privacidade. A criptografia não é o foco deste capítulo, visto que não é amplamente utilizada no campo da privacidade.

Tokenização é uma técnica de proteção de dados utilizada principalmente quando empresas buscam proteger dados confidenciais já armazenados ou em movimentação para a nuvem. Essa técnica gera aleatoriamente um valor de *token* sem formatação específica a partir de um registro original e armazena o mapeamento desse *token* com seu respectivo valor original em uma base de dados. Dessa forma, *tokens* não podem ser revertidos aos seus valores originais sem o devido acesso à tabela de mapeamento. Por exemplo, o nome "Francisco" pode ser mapeado para o *token* "F+YCO" e o número de conta corrente "3256-6" pode se tornar "ES*X-2" após um processo de tokenização. A principal diferença entre essa técnica e a criptografia é que, na tokenização, os dados originais são completamente substituídos por caracteres que não tem nenhuma conexão com os dados originais. Outro fato relevante sobre essa técnica é que, embora o *token* seja utilizável dentro de seu ambiente de aplicação nativo, é completamente inútil em outro contexto. Dessa forma, a tokenização é ideal para proteger números de cartões de crédito, números de seguro social, além de outras informações que identificam indivíduos de forma única.

A anonimização é constituída por um conjunto de técnicas que modificam dados originais, de tal forma que os dados anonimizados não se assemelham aos dados originais, mas ambos possuem semântica e sintaxe bastante semelhantes. O termo anonimato representa o fato do sujeito não ser unicamente caracterizado dentro de um conjunto de sujeitos. O intuito da anonimização é poder compartilhar informações com outras entidades, as quais poderão utilizá-las para diversas finalidades, sem que haja violação de privacidade. Modificações implicam em perda de informação e, consequentemente, em diminuição da utilidade dos dados. Logo, o desafio da anonimização é transformar os dados de tal forma que a privacidade dos indivíduos é protegida, enquanto a utilidade dos dados é mantida. Esse desafio é discutido com mais detalhes a seguir.

3.3. O Desafio de Disponibilizar Dados

Não se deve apenas pensar na privacidade dos indivíduos e esquecer a utilidade dos mesmos. Dados disponibilizados precisam ser úteis para que outros tipos de usuários possam realizar atividades importantes. Isso torna o processo de disponibilização de dados uma tarefa nada trivial. Além disso, existem diversas partes envolvidas em um cenário de disponibilização de dados [44], conforme mostra a Figura 3.2. As características dessas partes são detalhadas a seguir:

- **Cliente / Fornecedor dos Dados**: é representado por um indivíduo ou por uma organização que compartilha seus próprios dados. Por exemplo, um indivíduo que fornece seus dados pessoais como nome, endereço, gênero, data de nascimento, telefone, e-mail para cadastro em uma agência bancária;

- **Organização**: é evidenciada por qualquer tipo de entidade que serve à realização de ações de interesse social, político, etc. Por exemplo organizações bancárias, de saúde, de comércio eletrônico, ou mesmo de redes sociais, que detém uma certa quantidade de informações sobre seus clientes. Elas são responsáveis por proteger os dados de seus usuários a qualquer custo. Em caso de vazamento de informações, as organizações enfrentam prejuízos financeiros, processos legais, perda de reputação e perda de seus clientes;

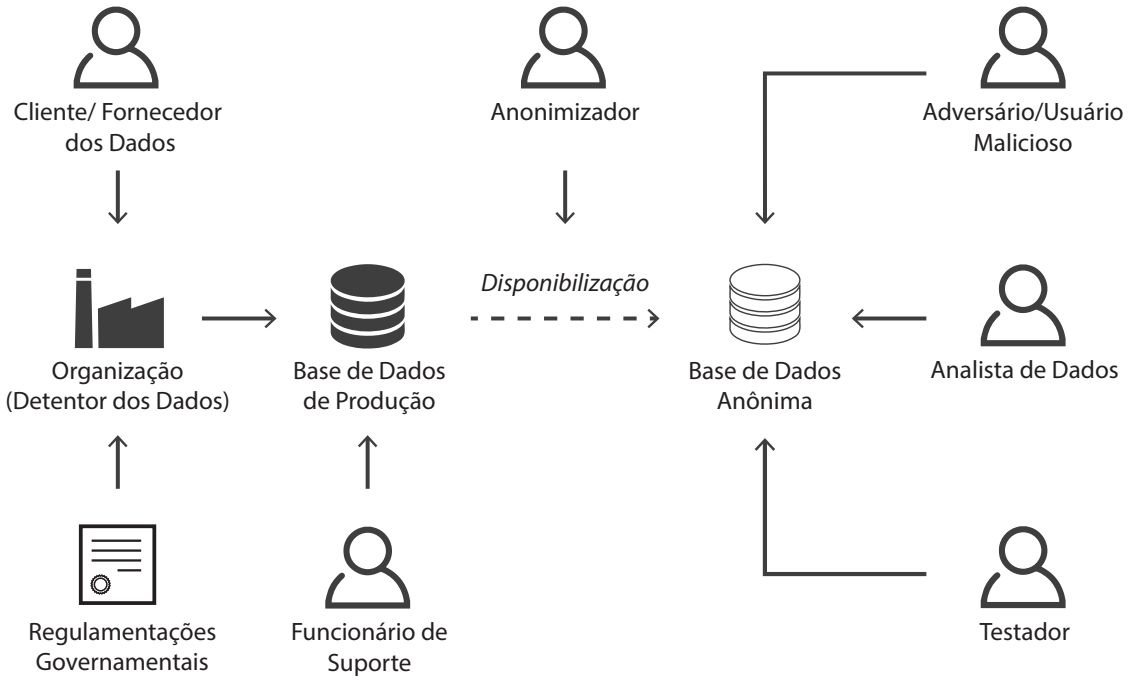

Figura 3.2. Partes envolvidas em um cenário de disponibilização de dados.

- **Regulamentações Governamentais**: define quais regras de proteção de dados as organizações devem seguir. HIPAA nos Estados Unidos e FIPPA no Canadá são exemplos de regulamentações. Vale ressaltar que as próprias organizações podem definir regulamentações internas;

- **Anonimizador**: é representado por um indivíduo ou organização que aplica técnicas de anonimização sobre os dados, para que analistas e testadores possam usufruir dos dados sem comprometer a privacidade dos mesmos. O papel do anonimizador também pode ser representado por um software que realiza a transformação dos dados de maneira automática.

- **Analista de dados**: utiliza os dados anonimizados para realizar minerações e descobrir padrões. Algumas regulamentações determinam que análises só podem ser realizadas sobre dados anonimizados. Por esse motivo, é importante que os dados disponibilizados suportem as funcionalidades de mineração de dados;

- **Testador**: a terceirização de testes de software é comum entre muitas empresas. Testes de alta qualidade exigem dados de teste também com alta qualidade, que estão presentes nos sistemas de produção e contém informações sensíveis de clientes. Para que testes possam ser realizados com eficiência, o testador precisa de dados extraídos dos sistemas de produção, porém anonimizados e provisionados para testes;

- **Funcionário de Suporte**: tem como objetivo auxiliar no funcionamento dos requisitos de negócio dos clientes. Por esse motivo, esse colaborador possui acesso ao conjunto de dados original e a todos os dados sensíveis de clientes;

- **Adversário**: também conhecido como atacante ou usuário malicioso. Ele visa obter dados acerca de um indivíduo específico e assim descobrir informações sensíveis que violem sua privacidade.

Diante de todas essas partes envolvidas no processo de disponibilização de dados, um questionamento pode ser levantado: como proteger a privacidade dos clientes/fornecedores de dados e ao mesmo tempo garantir que os dados sejam úteis para testadores ou analistas de dados?

Pesquisas mostraram que a abordagem mais promissora para proteger a privacidade dos indivíduos é anonimizar os dados antes de disponibilizá-los publicamente ou para terceiros [21, 47]. Para isso, o detentor dos dados deve modificá-los de tal forma que nenhuma informação sensível sobre indivíduos possa ser descoberta a partir de uma publicação. Além disso, ele deve garantir que os dados sejam úteis para que eventuais análises possam ser efetuadas com qualidade. Por esse motivo, o detentor dos dados também deve buscar uma solução que preserve ao máximo a utilidade das informações as quais ele deseja disponibilizar. Em contrapartida, a não disponibilização de dados impede que governos, organizações e instituições possam tirar proveito de análises importantes, padrões e tendências para a sociedade, pesquisas científicas, entre outras atividades, dificultando assim o crescimento de tais entidades. Portanto, quanto maior o grau de privacidade associado aos dados, menos úteis aqueles dados serão para eventuais análises.

É fato que este é um problema desafiador, uma vez que qualquer alteração sobre os dados distorce sua utilidade. Além disso, nem sempre regulamentações são favoráveis no balanceamento entre privacidade e utilidade. Para demonstrar essa questão, considere o seguinte caso no domínio de saúde: a regulamentação governamental HIPAA, nos Estados Unidos, afirma que, qualquer atributo pessoalmente identificável (por exemplo nome, telefone, data de internação, etc.), deverá ser completamente anonimizado no conjunto de dados publicado. Ao mesmo tempo, profissionais de saúde podem compartilhar dados de paciente com parceiros externos para que sejam realizadas análises da eficácia de um determinado tratamento, por exemplo. Contudo, será impossível analisar este acontecimento, visto que a data de registro do paciente é anonimizada de acordo com as leis de privacidade da HIPAA, ou seja, não é possível analisar a eficácia do tratamento sem ter conhecimento sobre o tempo em que o paciente está sendo acompanhado. Dessa forma, há uma enorme demanda entre pesquisadores na área de privacidade em propor modelos, técnicas e algoritmos que atendam aos mais diversos tipos de balanceamento entre privacidade e utilidade em inúmeros contextos. A Figura 3.3 mostra o caso geral de balanceamento entre utilidade e privacidade [44].

A estratégia de criptografia citada na seção anterior, não fornece utilidade (valor 0), porém dispõe de alta privacidade (valor 1) quando os dados estão criptografados. De maneira oposta, ela fornece alta utilidade (valor 1) mas nenhuma privacidade (valor 0) quando os dados estão descriptografados. Ou seja, é uma estratégia praticamente 0 ou 1 em termos de privacidade e utilidade. Já na anonimização de dados, o nível de privacidade e utilidade pode flutuar entre o intervalo $[0, 1]$, idealmente na zona cinza da Figura 3.3. Dessa maneira, pode-se controlar o nível de ambos os indicadores. Assim, a melhor solução de balanceamento entre privacidade e utilidade dependerá da métrica de utilidade e do modelo de privacidade adotados para a disponibilização de dados.

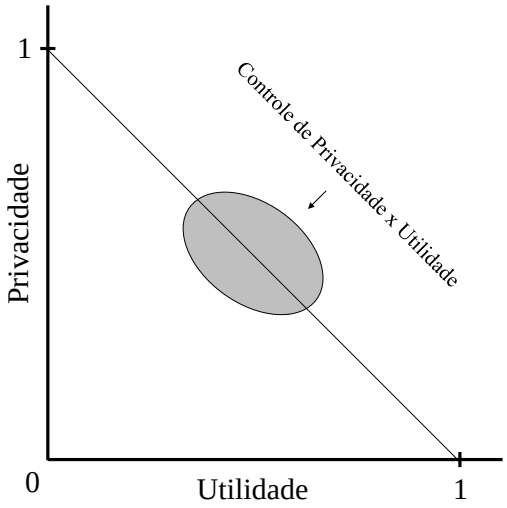

Figura 3.3. Balanceamento entre utilidade e privacidade.

3.4. Técnicas de Anonimização de Dados

A publicação de dados pode levar a sérios riscos de violação de privacidade devido à existência dos semi-identificadores. Isso pode acarretar em consequências graves por causa do uso não autorizado de informações sensíveis pertencentes aos indivíduos. Como forma de solucionar esse problema, uma estratégia ingênua seria a não publicação dos dados, para qualquer finalidade [48]. Contudo, isso evitaria que governos, organizações, etc. pudessem tirar proveito de análises importantes de padrões e tendências para a sociedade, dificultando o possível crescimento dessas entidades. Outra maneira de evitar o problema da publicação de dados seria disponibilizar apenas dados estatísticos para análise, porém essa estratégia é limitada ao conhecimento estatístico que o detentor dos dados possui, uma vez que ele deseja apenas publicar o dado, e não o analisar previamente antes de liberá-lo. A abordagem mais promissora para solucionar o problema da preservação de privacidade em uma publicação é anonimizar os dados antes de qualquer liberação [21]. Uma abordagem convencional para anonimizar dados tem sido praticada com a remoção dos identificadores explícitos de indivíduos, como nome, CPF, e-mail, etc. do conjunto de dados antes de uma publicação. Contudo, o trabalho em [42] demonstra que, simplesmente remover esses identificadores, não é suficiente para proteger a privacidade dos indivíduos, devido à existência dos semi-identificadores.

Esta seção apresenta uma visão geral das técnicas mais utilizadas para anonimização e publicação de dados. Em um processo de anonimização, um conjunto de dados original D é transformado em um novo conjunto D', por meio de modificações. O objetivo é evitar a descoberta de informações sensíveis por usuários maliciosos. As seguintes técnicas de anonimização são detalhadas a seguir: *generalização*; *supressão*; *perturbação*. Todas elas produzem um conjunto de dados D' menos preciso que o conjunto original D, no entanto ambos os conjuntos diferem no quesito perda de informação e também na proteção da privacidade ocasionada por cada técnica.

3.4.1. Generalização

O objetivo da generalização é aumentar a incerteza de um adversário ao tentar associar um indivíduo a seu registro, ou a informações sensíveis, no conjunto de dados publicado. Nesta técnica, os valores dos atributos que são considerados semi-identificadores são substituídos por valores semanticamente semelhantes, porém menos específicos. Dessa forma, a generalização preserva a veracidade dos dados quando aplicada sobre registros. A técnica de generalização pode ser aplicada em atributos tanto categóricos quanto numéricos. Assim, para cada categoria de atributos, pode existir uma hierarquia de generalização que representa a semântica desses atributos. Ao utilizar essa hierarquia, os valores de registros de uma determinada categoria são substituídos por valores menos específicos, abrangendo assim um maior domínio de valores para aquele atributo. Por exemplo, o valor tabulado "25" (domínio numérico) pode ser substituído pelo intervalo "[20, 29]" em um conjunto de dados publicado. A operação contrária à generalização é denominada especialização.

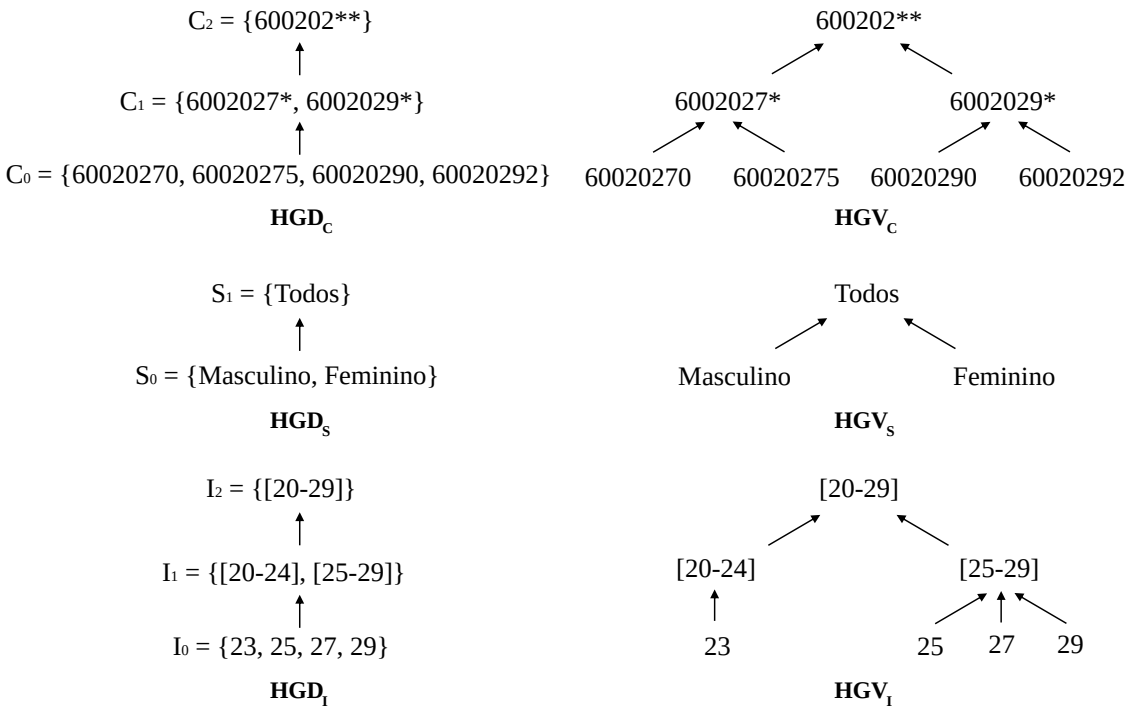

Figura 3.4. Exemplos de HGD e HGV para os atributos CEP (C), gênero (S) e idade (I).

Uma hierarquia de generalização pode ser predefinida por um domínio, sendo assim chamada de *hierarquia de generalização de domínio* (HGD). Uma HGD consiste em um conjunto de domínios totalmente ordenados pelo domínio da generalização de um atributo. Intuitivamente, considere dois domínios Dom_i e Dom_j. A comparação $Dom_i <_D Dom_j$ indica que os valores em Dom_j são generalizações dos valores em Dom_i. Dessa maneira, dado um domínio *Dom* e uma HGD, pode-se definir uma *hierarquia de generalização de valor* (HGV) como sendo o conjunto de valores em um determinado domínio, mas que são parcialmente ordenados pelo domínio da generalização de um atributo. Dessa forma, sendo dois valores v_i e v_j, a comparação $v_i \leq_V v_j$ indica que v_j é

uma generalização de v_i. A Figura 3.4 apresenta exemplos de hierarquias de generalização de domínio e de valor para os atributos CEP, gênero e idade, respectivamente. Neste exemplo, considerando o atributo CEP, a seguinte comparação pode ser considerada: '60020270' $<_V$ '6002027*' $<_V$ '600202**'. Portanto, o valor '600202**' é uma generalização de '6002027*', que por sua vez também é uma generalização do valor '60020270'.

Operações de generalização aplicadas de maneira ingênua sobre atributos semi-identificadores podem não gerar dados úteis para eventuais análises. Por esse motivo, é necessário encontrar uma generalização mínima, isto é, o conjunto mínimo necessário de alterações que devem ser aplicadas a um conjunto de dados, com o objetivo de manter sua utilidade e ao mesmo tempo atender aos requisitos de privacidade estabelecidos. O conjunto de todas as generalizações possíveis para todos os atributos semi-identificadores formam uma estrutura de reticulado, onde cada nó dessa estrutura corresponde a uma possível estratégia de generalização. A solução ótima para o problema da generalização mínima pode ser dada pelo nó do reticulado que satisfaz os requisitos de privacidade e resulta na menor perda de informação. Uma abordagem para encontrar a solução ótima para esse problema seria enumerar todos os nós da estrutura de reticulado e retornar aquele que produz a menor quantidade de distorção nos dados. Contudo, alguns autores já provaram que o problema de encontrar a generalização ótima é NP-difícil [9, 49]. Logo, heurísticas devem ser utilizadas para reduzir o espaço de busca e encontrar uma solução aproximada da solução ótima. A Figura 3.5 mostra o conjunto de todas as generalizações possíveis para os atributos semi-identificadores CEP, gênero e idade, formando o retículo desses atributos. Cada elemento C_i, S_j e I_k representam o grau de generalização baseado em suas respectivas hierarquias de generalização.

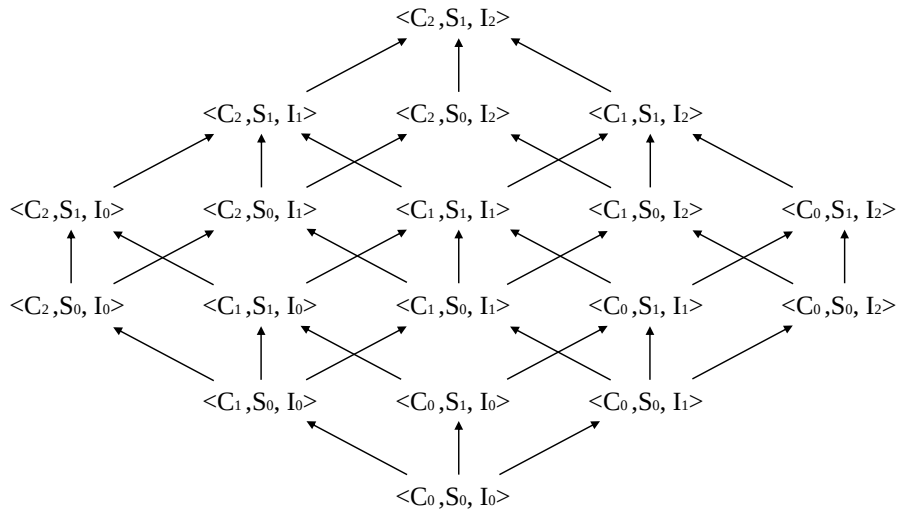

Figura 3.5. Reticulado do conjunto de todas as possíveis generalizações para os atributos CEP (C), gênero (S) e idade (I) baseado em suas respectivas hierarquias.

A operação de generalização pode ser aplicada tanto para todos os registros de um atributo semi-identificador quanto para apenas alguns. Como abordagem mais habitual, todos os registros de um atributo são mapeados para um mesmo valor generalizado (me-

nos específico), obedecendo sua hierarquia de generalização. Este processo é denominado *generalização global* [26, 31]. Por outro lado, diferentes registros do mesmo atributo podem ser generalizados com diferentes valores em uma hierarquia de generalização. Este processo é chamado de *generalização local* [24, 32]. Em resumo, a generalização global anonimiza todos os registros de um atributo da mesma maneira, utilizando sempre os mesmos valores da hierarquia de generalização, enquanto a generalização local anonimiza diferentes registros de um atributo com diferentes valores, obedecendo também a sua hierarquia de generalização.

Em se tratando de generalização a nível de atributo, pode-se aplicar essa técnica de quatro maneiras distintas: (i) generalização de domínio completo; (ii) generalização de sub-árvore; (iii) generalização entre irmãos; (iv) generalização de células. Os exemplos a seguir utilizarão a hierarquia de generalização de valor para o atributo *profissão*, conforme a Figura 3.6.

- **Generalização de domínio completo**: nesta abordagem, todos os valores de um atributo semi-identificador são generalizados para o mesmo nível, obedecendo sua hierarquia de generalização [32]. Esta técnica possui o menor espaço de busca quando comparada às outras técnicas, entretanto é a que produz mais distorção nos dados devido a exigência de todos os valores estarem anonimizados no mesmo nível na hierarquia de generalização. Por exemplo, se as profissões "Dentista" e "Médico" forem generalizadas para profissionais da "Saúde", então "Analista" e "Gerente" também devem ser generalizados para profissionais da "TI".

- **Generalização de sub-árvore**: neste esquema, a exigência de que todos os valores de um atributo estejam no mesmo nível da hierarquia de generalização aplica-se apenas aos valores das sub-árvore [25]. Em outras palavras, quando um valor de um atributo é substituído por um valor generalizado, obedecendo sua hierarquia de generalização, todos os outros valores nas folhas da sub-árvore também serão igualmente generalizados. Por exemplo, se a profissão "Médico" for generalizada para profissional da "Saúde", o valor "Dentista" também deverá ser generalizado, contudo "Analista" e "Gerente" não sofrem distorções, pois pertencem a outra sub-árvore na hierarquia de generalização.

- **Generalização de irmãos**: conhecido na literatura como *sibling generalization*, esta abordagem é semelhante à generalização de sub-árvore mas não exige que todos os valores de irmãos nas folhas de uma sub-árvore sejam generalizados [31]. Este esquema produz menos distorção do que o esquema de generalização de sub-árvore, pois atua apenas sobre os valores de atributos que necessitam ser distorcidos. No exemplo da Figura 3.6, caso a profissão "Analista" seja generalizada para um profissional da "TI", os valores do conjunto de dados que possuem "Gerente" como profissão não necessitam de generalização.

- **Generalização de células**: nas abordagens anteriores, caso o valor de um atributo seja generalizado, então todos os demais registros com o mesmo valor também devem ser generalizados. Dessa forma, tais abordagens são consideradas generalizações globais, anonimizando todos os registros de um atributo da mesma maneira. Já

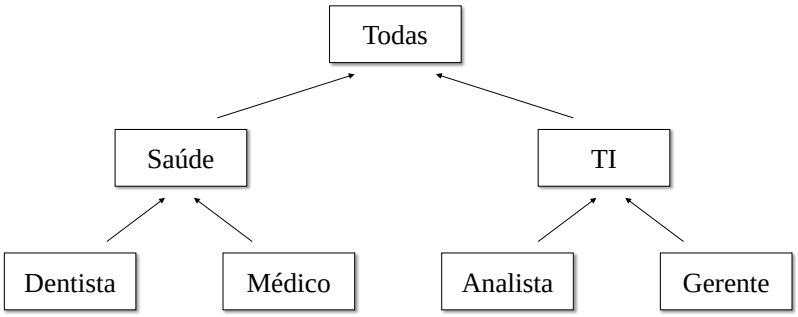

Figura 3.6. Hierarquia de generalização para o atributo semi-identificador Profissão.

no esquema de generalização de células, apenas algumas instâncias de um atributo semi-identificador são generalizadas, permitindo que outras instâncias se mantenham inalteradas [45].

3.4.2. Supressão

A supressão de dados é outra estratégia utilizada para compartilhamento e publicação de dados que ao mesmo tempo preserva a veracidade dos mesmos. Essa é uma técnica na qual um ou mais valores em um conjunto de dados são removidos ou substituídos por algum valor especial, possibilitando a não descoberta de semi-identificadores por adversários. A supressão de dados também pode ser vista como um tipo específico de generalização, no qual os registros são generalizados para o valor menos específico (nó raiz) na hierarquia de generalização de valor, que engloba todos os valores de um determinado atributo [48].

De maneira semelhante à técnica de generalização, a supressão de dados pode ser aplicada de maneira global ou local. A *supressão global* refere-se à remoção de todas as instâncias de um valor de atributo, garantindo que aqueles valores não serão descobertos em um conjunto de dados publicado, uma vez que todos foram removidos. Já a *supressão local* é caracterizada pela remoção de apenas algumas instâncias de um valor de atributo, contudo deve-se garantir que os valores restantes não possam ser descobertos. Os principais tipos de supressão de dados são:

- **Supressão de registro**: nessa abordagem, um registro é removido inteiramente do conjunto de dados. Consequentemente, nenhum valor de atributo é disponibilizado para os usuários [26, 31].
- **Supressão de valor**: refere-se a remoção ou a substituição de todas as instâncias de um valor de um atributo por um valor especial (como "*" ou "todos"). Por exemplo, os valores de atributo salário abaixo de R$ 30.000,00, em uma tabela de empregados, podem ser removidos ou substituídos por "*", enquanto os demais valores não sofrem distorções [46, 48].
- **Supressão de células**: nessa técnica, apenas algumas instâncias de valores de um atributo são removidas ou substituídas por um valor especial, caracterizando uma *supressão local* [36]. Por exemplo, pode-se remover apenas metade dos valores de

atributo salário abaixo de R$ 30.000,00, em uma tabela de empregados. Assim, instâncias de salário podem conter valores abaixo ou acima de R$ 30.000,00, além de valores suprimidos. Essa estratégia pode levar a inconsistências em eventuais análises de dados.

3.4.3. Perturbação

Essa abordagem tem sido comumente utilizada em controle de descoberta estatística [25], devido à sua simplicidade, eficiência e capacidade de preservar informações estatísticas. A ideia geral dessa técnica é substituir os valores dos atributos semi-identificadores originais por valores fictícios, de modo que informações estatísticas calculadas a partir dos dados originais não se diferenciem significativamente de informações estatísticas calculadas anteriormente sobre os dados perturbados. Ao contrário das técnicas de generalização e supressão, que preservam a veracidade dos dados, a perturbação resulta em um conjunto de dados com valores sintéticos. Muitas vezes isso acarreta em informações sem sentido para aqueles que irão utilizá-las. As técnicas mais comuns para a perturbação de dados são:

- **Adição de ruído**: essa técnica é geralmente aplicada sobre atributos numéricos. A ideia geral é substituir um valor original de atributo "v" por "$v + r$", onde "r" é um valor, denominado ruído, escolhido aleatoriamente a partir de uma distribuição. Um valor de atributo "v" também pode ser substituído pelo produto "$v \times r$" Em outras palavras, os valores de atributos são perturbados com um determinado nível de ruído, que pode ser adicionado ou multiplicado pelo valor original de cada atributo [43]. Consequentemente, mesmo que um atacante consiga identificar um valor individual de um atributo confidencial, o valor original não será revelado. A vantagem dessa técnica é que ela preserva algumas propriedades estatísticas, como média e correlação, mas ao mesmo tempo ela pode gerar alguns valores sem significado ou sem expressividade;

- **Permutação de Dados**: nessa abordagem, dois valores do mesmo atributo (de dois registros diferentes) são permutados. Isso mantém algumas características estatísticas dos dados, como contagem e frequência dos atributos [16]. Essa técnica não altera o domínio dos atributos, todavia as possíveis permutações de valores para diferentes atributos podem gerar registros sem sentido, e consequentemente, informações equivocadas. Por exemplo, considere os atributos "gênero" e "profissão". Caso haja uma permutação entre valores que gere um registro contendo "masculino" e "professora", esse registro não faria sentido, visto que o valor "professora" é do gênero feminino;

- **Geração de Dados Sintéticos**: nessa técnica, primeiramente um modelo estatístico é gerado a partir do conjunto de dados e, em seguida, são gerados dados sintéticos que seguem tal modelo [8]. São esses dados sintéticos que devem ser disponibilizados para o usuário final. A vantagem dessa técnica é que todas as propriedades estatísticas dos dados são preservadas. Contudo, pode-se gerar também alguns valores sem sentido e que não existem no mundo real.

O mascaramento de dados também é um método de perturbação utilizado para publicar conjuntos de dados com informações que pareçam reais, mas que não revelam informações sobre nenhum indivíduo. O objetivo principal do mascaramento é disponibilizar bases de dados para testes ou treinamento de usuários. Isso protege a privacidade dos dados pessoais presentes no banco de dados, bem como outras informações sensíveis que não possam ser colocadas à disposição para um time de testes ou para usuários em treinamento. Algumas técnicas de mascaramento de dados são descritas a seguir:

- **Substituição**: corresponde à substituição aleatória de dados por informações similares, mas sem nenhuma correlação com o dado real. Por exemplo, a substituição de sobrenome de família por algum outro proveniente de uma lista aleatória de sobrenomes;
- **Embaralhamento (*Shuffling*)**: é uma estratégia semelhante a substituição mas com a diferença de que o dado é derivado da própria coluna da tabela. Assim, o valor do atributo A em uma determinada tupla c_i é substituído pelo valor do atributo A em uma outra tupla c_k, selecionada randomicamente, onde $i \neq k$;
- ***Blurring***: essa técnica é aplicada a dados numéricos e datas. A técnica altera o valor do dado por alguma percentagem aleatória do seu valor real. Logo, pode-se alterar uma determinada data somando ou diminuindo um determinado número de dias, de forma aleatória;
- **Anulação/Truncagem**: essa técnica substitui os dados sensíveis por valores nulos (NULL) no conjunto de dados publicado. Geralmente esta técnica é utilizada quando os dados existentes na tabela não são necessários à realização de testes ou treinamentos.

3.5. Perda de Informação e Medidas de Utilidade dos Dados

A anonimização de dados causa perda de informação e muitas vezes compromete a utilidade dos dados, ou seja, quanto mais anonimizados forem os dados, menos úteis eles serão para o usuário final. Como forma de preservar a utilidade dos dados publicados, deve-se assegurar que o mínimo de distorção deva ser gerado na anonimização. Essa distorção causada por um processo de anonimização é denominada perda de informação. Há algumas métricas para se medir a perda de informação. Tais métricas podem ser utilizadas tanto para medir a utilidade do conjunto de dados publicado em relação aos dados originais, ou serem utilizadas como métricas de busca, com o objetivo de guiar os passos em busca da melhor solução de anonimização no espaço de todas as possibilidades [20].

A seleção de uma métrica de utilidade apropriada depende principalmente do algoritmo de anonimização adotado e do objetivo da disponibilização de dados. Por exemplo, se um algoritmo de anonimização utiliza uma hierarquia de generalização, então as métricas de perda de informação, que levam em conta o custo das operações de generalização, devem ser consideradas. Existem métricas que são utilizadas para cenários de publicação e disponibilização de dados em geral. Essas métricas, denominadas *métricas de uso geral*, são utilizadas no cenário em que o publicador dos dados não tem conhecimento prévio

sobre a área de aplicação dos dados a serem liberados ou quando não há objetivos específicos pré-definidos para o uso desses dados [20]. Por esse motivo, essas métricas devem levar em consideração muitos fatores e reter a maior quantidade de informação sempre que possível. Nesse caso, os dados publicados se tornam disponíveis para todos, como por exemplo na Internet, de modo que usuários com interesses distintos possam realizar análises de acordo com sua necessidade.

É fato que toda operação de generalização ou supressão causa alguma distorção sobre os dados. Em função disso, uma das métricas de uso geral utilizada para avaliar a utilidade dos dados é a *Precisão* [41]. Essa métrica obtém a distorção de dados por meio da atribuição de uma penalidade a cada instância de um valor de atributo que é generalizado ou suprimido. Se o valor de um atributo em um registro não é generalizado ou suprimido então não há distorção. Uma distorção é calculada para cada célula e é definida como a altura do nó correspondente ao valor generalizado na hierarquia de generalização dividida pela altura máxima da hierarquia. A distorção total do conjunto de dados é calculada pela soma de todas as distorções das células. Ao se obter a soma de todas as distorções das células e normalizar o resultado pelo número total de células, obtêm-se a métrica de precisão. Formalmente, a precisão de um conjunto de dados anonimizado $D(A_1,...,A_{N_a})$ é definida como:

$$Prec(D) = 1 - \frac{\sum_{i=1}^{N_a}\sum_{j=1}^{|D|}\frac{h}{|HGV_{A_i}|}}{|D| * |N_a|}$$

A precisão $Prec(D)$ assume valores entre 0 e 1. $|N_a|$ representa o número de atributos pertencentes ao conjunto de semi-identificadores, $|D|$ equivale ao número de registros da tabela, enquanto que h representa a altura da hierarquia de generalização de valor do atributo A_i após a generalização e por fim $|HGV_{A_i}|$ é a altura máxima da hierarquia. Quanto maior a precisão, maior a utilidade dos dados e consequentemente, os dados anonimizados são mais semelhantes ao conjunto de dados original.

A perda de informação causada por generalizações também pode ser medida utilizando a métrica *ILoss* [50]. Essa métrica captura a fração de nós folha, baseados em uma hierarquia de generalização, que são generalizados. Para um determinado registro, a métrica *ILoss* é calculada encontrando a soma dos valores de *ILoss* para todos os atributos daquele registro. É importante salientar que diferentes pesos podem ser aplicados a diferentes atributos. O *ILoss* global de um conjunto de dados pode ser obtido pela soma de todos os valores *ILoss* obtidos para os registros. Seja V_g um nó na hierarquia de generalização H de um atributo $A \in SI$, onde SI é conjunto de semi-identificadores. $|V_g|$ é definido como o número de folhas na sub-árvore de V_g. Seja $|D_A|$ o número de valores no domínio do atributo A, isto é, o número total de folhas de H. A métrica *ILoss* para um valor específico é calculada como:

$$ILoss(V_g) = \frac{|V_g| - 1}{|D_A|}$$

Considere agora $|W_i|$ como uma constante positiva que especifica a penalidade do atributo A_i sobre V_g. Essa constante pode ser definida pelo usuário a fim de determinar a

importância de cada atributo. A métrica $ILoss(r)$ em termos de registro é dada por:

$$ILoss(r) = \sum_{V_g \in r} (W_i * ILoss(V_g))$$

Finalmente, a perda total de informação no conjunto de dados anonimizado D é dada pela soma de todos os *ILoss* de registros:

$$ILoss(D) = \frac{\sum_{r \in D} ILoss(r)}{|D|}$$

Algumas métricas utilizam o conceito de classe de equivalência para mensurar a qualidade dos dados anonimizados. Formalmente, uma classe de equivalência é definida da seguinte forma:

Definição 1 *Considere uma série de atributos $A = \{A_1, ..., A_n\}$ em um conjunto de dados D. Uma classe de equivalência E é um conjunto de todos os registros em D que contém valores idênticos para os atributos em A.*

Por exemplo, a Tabela 3.3 abrange duas classes de equivalência para o conjunto de atributos $A = \{$Idade, Gênero, CEP$\}$. São elas: $E_1 = \{[20,30]$, Masculino, 60800***$\}$ e $E_2 = \{> 40$, *, 60790***$\}$.

Idade	Gênero	CEP
[20,30]	Masculino	60800***
[20-30]	Masculino	60800***
>40	*	60790***
>40	*	60790***
>40	*	60790***

Tabela 3.3. Conjunto de dados contendo duas classes de equivalência.

Uma das métricas que utiliza o conceito de classe de equivalência é a métrica de *discernibilidade*. Ela foi introduzida por Bayardo [26] e aborda a noção de perda de informação através de uma penalidade para cada registro por ser indistinguível de outros registros no conjunto de semi-identificadores. Nessa métrica, a qualidade é baseada justamente no tamanho da classe de equivalência E e no conjunto de dados D.

A métrica de discernibilidade C_{DM} atribui para cada registro r no conjunto de dados D uma penalidade determinada pelo tamanho da classe de equivalência contendo r. Se um registro pertence a uma classe equivalente de tamanho s, a penalidade para o registro é s. Se uma tupla é suprimida, então é atribuída uma penalidade de valor $|D|$. Essa penalidade refere-se ao fato de que uma tupla suprimida não pode ser distinguida de qualquer outra tupla no conjunto de dados. Formalmente, a discernibilidade é medida como sendo:

$$C_{DM} = \sum_{classesEq} |E|^2$$

No entanto, uma vez que essa métrica é baseada na dimensão das classes de equivalência, a mesma perda de informação é dada para todos os registos nas classes de equivalência de mesma dimensão. Porém, esses registros podem ser generalizados de maneiras diferentes e, portanto, possuem diferentes níveis de distorção que não são levados em conta pelo C_{DM}.

Outra métrica utilizada para mensurar a utilidade dos dados e que também lida com classes de equivalência é denominada *tamanho médio das classe de equivalência* (C_{AVG}) [32]. Esta métrica mede o quão bem uma partição, isto é, uma classe de equivalência, se aproxima do melhor caso. Seu objetivo é reduzir a média normalizada do tamanho das partições. Dado que cada registro é generalizado em classes de equivalência de pelo menos k registros indistinguíveis, essa métrica é definida como:

$$C_{AVG} = (\frac{totalRegistros}{totalClassesEq})/(k)$$

Utilizar a mesma métrica para disponibilizar dados a usuários distintos nem sempre é uma boa ideia, visto que essas métricas podem não se adequar as diferentes demandas dos diferentes usuários. Para contornar essa questão, são propostas as *métricas de finalidade específica*, que tem por objetivo atender as demandas de usuários quando um conjunto de dados é publicado para finalidades exclusivas. Desse modo, tais métricas são utilizadas caso a finalidade dos dados seja conhecida no momento da publicação ou caso os dados estejam sendo publicados para fins de mineração específicos. Nestes casos, as métricas de perda de informação que melhores se adequam aos objetivos específicos são justamente as métricas que devem ser adotadas antes da disponibilização. Por exemplo, se os dados forem liberados com o propósito de construir um classificador para um certo atributo, os valores que são importantes para a classificação não devem ser generalizados ou suprimidos. Ou seja, não se deve anonimizar os valores cujas distinções são essenciais para diferenciar os rótulos das classes do atributo alvo. Para atingir esse objetivo, o erro de classificação nas instâncias futuras deve ser considerado no cálculo da perda de informação.

A métrica mais comum utilizada nesta categoria é a *medida de classificação (CM)* [25]. A ideia dessa métrica é penalizar cada registro r que é suprimido ou generalizado para uma classe de equivalência, em que a classe de r não é a classe majoritária. A classe majoritária é aquela que contém o maior número de registros. Dessa forma, CM pode ser calculado pela soma de todas as penalidades de cada registro, normalizado pelo número total de registros.

$$CM = \frac{\sum_{r \in D} Penalidade(r)}{N}$$

D é o conjunto de dados, N é o número de registros em D e $Penalidade(r)$ é definido como:

$$Penalidade(r) = \begin{cases} 1 & \text{se } r \text{ é suprimido} \\ 1 & \text{se } classe(r) \neq Maj(classeEq(r)) \\ 0 & \text{caso contrário} \end{cases}$$

Também existem algumas métricas que, além de informações, levam em consideração os requisitos de privacidade para se medir a perda de informação proveniente do processo de anonimização. O objetivo dessas métricas é buscar a anonimização que minimiza a perda de informação enquanto maximiza o ganho de privacidade. Tais métricas são denominadas *métricas de trade-off*. Dessa forma, essas métricas calculam tanto o ganho de informações quanto a perda de privacidade a cada iteração de anonimização, de modo que o *trade-off* ideal possa ser encontrado para ambos os requisitos necessários.

Suponha que o conjunto de dados anônimo a ser disponibilizado é construído iterativamente por meio da aplicação de operações de especialização. Cada especialização divide um valor geral em diferentes filhos, de modo que há algum ganho de informação $IG(s)$, e ao mesmo tempo perda de privacidade $PL(s)$. A métrica $IGPL(s)$ [22] busca justamente encontrar a melhor especialização s que maximiza o ganho de informação para cada perda de privacidade. Dessa forma ela pode ser definida como:

$$IGPL(s) = \frac{IG(s)}{PL(s)+1}$$

A escolha de $IG(s)$ e $PL(s)$ depende da métrica de informação e do modelo de privacidade a serem adotados.

3.6. Modelos de Privacidade Sintáticos

Modelos de privacidade sintáticos têm como objetivo estabelecer uma determinada condição, a qual os dados devem satisfazer, após um processo de anonimização. Tais modelos utilizam, na maioria das vezes, generalização e/ou supressão nos dados até uma condição sintática ser atendida, de modo que o conhecimento adversário se torna restrito na descoberta de atributos sensíveis a partir de semi-identificadores.

Nesta seção iremos apresentar alguns desses modelos de privacidade sintáticos e detalhar seus pontos positivos e negativos com o auxílio de exemplos. Vale ressaltar que serão vistos apenas modelos de como preservar a privacidade e não algoritmos específicos de como atingir a propriedade específica de cada um deles.

3.6.1. k-anonimato

O modelo de privacidade k-anonimato é o mais conhecido no campo da anonimização de dados. Foi proposto por Sweene [42] como forma de proteção ao ataque de ligação ao registro. Esse modelo assegura que, para cada combinação de valores de semi-identificadores, existem pelo menos k registros no conjunto de dados publicado, formando uma classe de equivalência. O k-anonimato atua sobre o princípio da indistinguibilidade, isto é, cada registro em um conjunto de dados k-anônimo é indistinguível de pelo menos outros k-1 registros em relação ao conjunto de semi-identificadores. Dessa forma, garante-se que cada registro não pode ser ligado a um indivíduo por um adversário com probabilidade maior que $\frac{1}{k}$.

Neste modelo, o valor de k define o nível de privacidade e, ao mesmo tempo, atua diretamente sobre a perda de informação. Assim, quanto maior o valor de k, maior será a privacidade e menor a utilidade dos dados. Não existem abordagens analíticas para determinar o melhor valor de k [13]. A complexidade na escolha desse valor depende de muitos critérios, como por exemplo dos requisitos de privacidade provenientes do detentor dos dados, dos requisitos de utilidade por parte de analistas, testadores, pesquisadores, etc., do nível de generalização, dentre outros critérios.

Para ilustrar a utilização desse modelo, iremos considerar que se deseja publicar os dados da Tabela 3.4 seguindo o modelo 2-anonimato. Vamos considerar que os identificadores explícitos são Placa, Motorista e CPF e como atributos sensíveis Tipo de Multa e Valor da Multa e, consequentemente, os demais como atributos semi-identificadores: Data de Nascimento e Data da Infração.

	Placa	**Motorista**	**CPF**	**Data de Nascimento**	**Data da Infração**	**Tipo de Multa**	**Valor da Multa (R$)**
1	HXR-1542	José Pereira	258.568.856	14/03/1977	03/01/2013	1	170
2	HTS-5864	Jorge Cury	566.548.584	04/03/1977	03/01/2013	2	250
3	HUI-5846	Paula Maria	384.987.687	24/05/1977	03/01/2013	1	170
4	HTR-5874	Jandira Lima	054.864.576	20/04/1978	04/01/2013	1	170
5	HOI-6845	José Sá	244.684.876	22/05/1978	04/01/2013	2	250
6	HQO-5846	Kilvia Mota	276.684.159	13/05/1978	05/01/2013	2	250
7	HUY-8545	José Pereira	538.687.045	15/05/1978	05/01/2013	1	170

Tabela 3.4. Dados sobre infrações de trânsito (Fonte: [11]).

Após aplicar supressão nos identificadores explícitos e generalização nos atributos sensíveis a Tabela 3.5 é gerada. Nesta tabela podemos perceber quatro classes de equivalência para os semi-identificadores: Classe A = "$03/1977,01/2013$" nas linhas 1 e 2; Classe B = "$05/1977,01/2013$" registro 3; Classe C = "$04/1978,01/2013$" com o registro 4 e Classe D = "$05/1978,01/2013$" nas linhas 5, 6 e 7. Perceba que, mesmo após aplicar algum nível de generalização, a regra 2-anonimato está sendo violada para os registros das linhas 3 e 4. Se algum atacante tiver disponível como conhecimento prévio a data de nascimento e souber que esses indivíduos estão representados nos dados publicados, apesar da generalização, ele consegue inferir o registro referente aos indivíduos. Neste caso, alguma outra operação de anonimização complementar deve ser adotada para esses grupos, como a supressão dos registro conforme a Tabela 3.6 que atende ao critério $k = 2$.

É mostrado em [20] que este modelo é efetivo contra ataques de ligação ao registro, porém não é um modelo adequado para prevenir ataques de ligação ao atributo. Considere, por exemplo, a Tabela 3.6 que atende ao modelo 2-anonimato e cujo adversário tenha conhecimento prévio que José Sá (linha 5) nasceu em 1978 e que foi multado em Janeiro de 2013. O adversário consegue inferir que o valor da multa de José foi de R$ 250 com probabilidade $\frac{2}{3}$, i.e., maior do que $\frac{1}{2}$ exigido pelo modelo. Outro fator que não é considerado pelo modelo é o caso de uma publicação possuir mais de um registro referente ao mesmo indivíduo. Destaca-se, ainda, que o problema de encontrar uma k-anonimização ótima é considerado NP-difícil, conforme demonstrado em [36].

Como forma de contornar algumas das desvantagens do modelo k-anonimato, fo-

	Placa	Motorista	CPF	Data de Nascimento	Data da Infração	Tipo de Multa	Valor da Multa (R$)
1	*	*	*	03/1977	01/2013	1	170
2	*	*	*	03/1977	01/2013	2	250
3	*	*	*	05/1977	01/2013	1	170
4	*	*	*	04/1978	01/2013	1	170
5	*	*	*	05/1978	01/2013	2	250
6	*	*	*	05/1978	01/2013	2	250
7	*	*	*	05/1978	01/2013	1	170

Tabela 3.5. Dados sobre informações de trânsito anonimizados (Fonte: [11]).

	Placa	Motorista	CPF	Data de Nascimento	Data da Infração	Tipo de Multa	Valor da Multa (R$)
1	*	*	*	03/1977	01/2013	1	170
2	*	*	*	03/1977	01/2013	2	250
3	*	*	*	*	*	*	*
4	*	*	*	*	*	*	*
5	*	*	*	05/1978	01/2013	2	250
6	*	*	*	05/1978	01/2013	2	250
7	*	*	*	05/1978	01/2013	1	170

Tabela 3.6. Tabela no modelo *2-anonimato* (Fonte: [11]).

ram propostos outros modelos de privacidade sintáticos, como l-diversidade, t-proximidade, δ-presença, entre outros.

3.6.2. l-diversidade

O modelo l-diversidade [34] busca prover proteção contra ataques de ligação ao atributo, i.e., os casos em que um adversário pode inferir informações sensíveis sobre registros mesmo sem identificá-los. Surgiu como forma de sanar essa limitação do k-anonimato. Para evitar esse tipo de descoberta, o modelo exige que cada classe de equivalência possua, pelo menos, l valores distintos para cada atributo sensível. Isto garante que um atacante, mesmo com conhecimento prévio que lhe permita descobrir a classe de equivalência de um indivíduo, não consiga inferir o atributo sensível do mesmo com probabilidade maior que $\frac{1}{l}$.

Na Tabela 3.7, onde os atributos Idade, CEP e Cidade foram classificados como semi-identificadores e o atributo Doença como sensível, os registros estão anonimizados segundo o modelo 4-anonimato. Porém, perceba que se um adversário possuir conhecimento prévio de que o CEP de um dado indivíduo é 540020, é possível deduzir que este pertence à classe de equivalência "$40,540020$" - registros das linhas 5 a 8. E, com isso, ele consegue inferir que o indivíduo sofre de bronquite.

Ao converter a Tabela 3.7 para o modelo 3-diversidade, não é preciso fazer modificações para a classe de equivalência "$85,560001$", pois esta já possui quatro valores distintos para o atributo sensível, satisfazendo a condição imposta pelo modelo. Já para a

classe "40, 540020" é necessário aplicar alguma técnica para satisfazer o requisito. Uma possibilidade seria suprimir os registros das linhas de 5 a 8. Uma outra solução é demonstrada na Tabela 3.8 onde os atributos sensíveis das linhas 5 e 6 foram alterados de forma que, agora, satisfazem ao modelo.

	Idade	**CEP**	**Cidade**	**Doença**
1	<85	560001	*	Sinusite
2	<85	560001	*	Gripe
3	<85	560001	*	Diabetes
4	<85	560001	*	Hérnia
5	<40	540020	*	Bronquite
6	<40	540020	*	Bronquite
7	<40	540020	*	Bronquite
8	<40	540020	*	Bronquite

Tabela 3.7. Conjunto de dados no modelo 4-anonimato (Fonte: [44]).

	Idade	**CEP**	**Cidade**	**Doença**
1	<85	560001	*	Sinusite
2	<85	560001	*	Gripe
3	<85	560001	*	Diabetes
4	<85	560001	*	Hérnia
5	<40	540020	*	Sinusite
6	<40	540020	*	Diabetes
7	<40	540020	*	Bronquite
8	<40	540020	*	Bronquite

Tabela 3.8. Conjunto de dados no modelo 4-anonimato e 3-diversidade (Fonte: [44]).

Como mencionado em [44] e [20], o l-diversidade apresenta alguns pontos não cobertos pelo modelo:

- Impacto na utilidade dos dados quando o modelo é aplicado em cenários onde há grande número de repetições para o valor de um atributo sensível e pouco/nenhum de outros valores, pois neste caso é necessário introduzir grande distorção ou supressão nos dados.
- *Skewness Attack*: ocorre quando o atacante tem conhecimento prévio e descobre tanto a classe de equivalência de um indivíduo quanto a distribuição dos atributos sensíveis, que pode ser obtida apenas analisando a tabela publicada. De posse dessas duas informações, o atacante pode obter o valor de um atributo sensível com uma chance maior do que a proporcionada pela distribuição global. Seja, por exemplo, para o modelo 2-diversidade uma tabela onde apenas dois por cento dos indivíduos tem diagnóstico positivo para o atributo sensível HIV. Se uma determinada classe de equivalência com quatro registros apresenta dois destes com diagnóstico positivo e dois com negativo, infere-se com uma chance de $\frac{1}{2}$ que o indivíduo é positivo contra a probabilidade global de $\frac{1}{50}$.

- Ataque de similaridade: ocorre quando, mesmo que os atributos sensíveis sejam distintos, qualquer um deles fornece uma informação sensível ao atacante, e.g., no caso em que o atributo sensível é uma doença no modelo 2-diversidade e os valores para a classe de equivalência de um dado indivíduo descoberta pelo atacante a partir de conhecimento prévio são úlcera ou gastrite, por similaridade ele consegue inferir que o indivíduo tem uma doença de estômago.
- Incapacidade de lidar com a semântica de relação dos novos valores ao substituir os originais, e.g., na Tabela 3.8 onde os valores Bronquite (linhas 5 e 6) foram substituídos por Sinusite e Diabetes.

3.6.3. t-proximidade

O modelo t-proximidade [33] propõe-se a corrigir algumas limitações do l-diversidade no que diz respeito à proteção contra *skewness attack*, onde o adversário pode inferir informações sobre atributos sensíveis a partir do conhecimento da frequência de ocorrência dos atributos na tabela (conforme exemplificado na Seção 3.6.2). Para isso, esse modelo visa assegurar que a distribuição dos dados de um atributo sensível em cada classe de equivalência seja próxima à sua distribuição global (i.e. na tabela completa).

A distância máxima entre as classes e a distribuição global é definida através do parâmetro t. Para mensurar a diferença entre a distribuição da classe de equivalência e a global é utilizada a "distância global de retaguarda" (*EMD: Earth Mover Distance*) cujo resultado contém valores reais no intervalo $[0,1]$, observando que, quanto maior o valor da distância, mais fraca é a proteção. No trabalho [33] EMD é adaptada a calcular a distância $D[P,Q]$ onde $P = \{p_1, p_2, ..., p_m\}$ é a distribuição dos atributos de uma dada classe, m o número de atributos sensíveis e $Q = \{q_1, q_2, ..., q_m\}$) a distribuição global.

As principais limitações [20] do t-proximidade são resumidas em: (i) falta de flexibilidade para especificação de diferentes níveis de privacidade para cada atributo sensível, (ii) a função EMD não é adequada para ataques de ligação ao atributo quando estes são numéricos e (iii) garantir o t-proximidade poderá degradar bastante a utilidade para garantir a mesma distribuição em todos as classes de equivalência.

3.6.4. δ-presença

Este modelo foi proposto em [38] e objetiva evitar a vinculação de um indivíduo a uma tabela publicada, ou seja, busca proteger a privacidade de indivíduos contra o ataque de ligação à tabela. O modelo δ-presença define o limite $\delta = (\delta_{max}, \delta_{min})$ para a probabilidade de um adversário inferir a presença de um indivíduo em um conjunto de dados tabulados. Este modelo pode prevenir indiretamente ataques de ligações ao registro e ao atributo, visto que um atacante possui no máximo $\delta\%$ de confiança de que o registro da vítima está presente na tabela publicada, então a probabilidade de uma ligação ser bem-sucedida ao registro e ao atributo sensível é no máximo $\delta\%$.

Para exemplificar um ataque de ligação à tabela, vamos utilizar um exemplo de [20]. Suponha que o detentor dos dados tenha liberado informações anonimizadas de pacientes na Tabela T (Tabela 3.9) e, ainda, que um adversário tenha acesso a dados públicos da Tabela E (Tabela 3.10) e sabe que todos os registros da primeira estão contidos na segunda ($T \subset E$). Com isso consegue-se deduzir que a probabilidade de Alice estar

presente em T é $\frac{4}{5} = 0.8$ pois há 4 registros em T e 5 registros em E contendo a classe de Alice: "$Artista, Feminino, [30-35)$". Pelo mesmo raciocínio, a probabilidade de Bob estar presente é de $\frac{3}{4} = 0.75$.

Profissão	Gênero	Idade	Doença
Profissional	Masculino	[35-40)	Hepatite
Profissional	Masculino	[35-40)	Hepatite
Profissional	Masculino	[35-40)	HIV
Artista	Feminino	[30-35)	Gripe
Artista	Feminino	[30-35)	HIV
Artista	Feminino	[30-35)	HIV
Artista	Feminino	[30-35)	HIV

Tabela 3.9. Tabela de pacientes no formato 3-anonimimato (Fonte: [20]).

Nome	Profissão	Gênero	Idade
Alice	Artista	Feminino	[30-35)
Bob	Professional	Masculino	[35-40)
Cathy	Artista	Feminino	[30-35)
Doug	Professional	Masculino	[35-40)
Emily	Artista	Feminino	[30-35)
Fred	Professional	Masculino	[35-40)
Gladys	Artista	Feminino	[30-35)
Henry	Professional	Masculino	[35-40)
Irene	Artista	Feminino	[30-35)

Tabela 3.10. Tabela externa no formato 4-anonimato (Fonte: [20]).

3.7. Modelo de Privacidade Diferencial

Os modelos de privacidade apresentados até aqui consideram que a violação de privacidade ocorre quando dados são publicados em formatos tabulados e o adversário utiliza informações de fontes externas para reidentificar indivíduos. Sob outra perspectiva, a Privacidade Diferencial investiga a ideia de publicar resultados de consultas, ao invés de dados tabulados, de tal forma que são adicionados ruídos a esses resultados. Em outras palavras, os dados provenientes de consultas são perturbados como forma de garantir a privacidade dos indivíduos. Assim, um atacante não será capaz de concluir algo com 100% de confiança. A sua principal convicção é de que as conclusões obtidas sobre um indivíduo são referentes aos dados de toda a tabela, e não apenas a um registro em particular. Por esse motivo, o modelo de privacidade em questão propõe evitar ataques probabilísticos.

3.7.1. Conceitos Básicos

A Privacidade Diferencial é um modelo matemático proposto em [17] que fornece sólidas garantias de privacidade. O objetivo desse modelo é disponibilizar informações estatísticas sobre um conjunto de dados sem comprometer a privacidade de seus indivíduos.

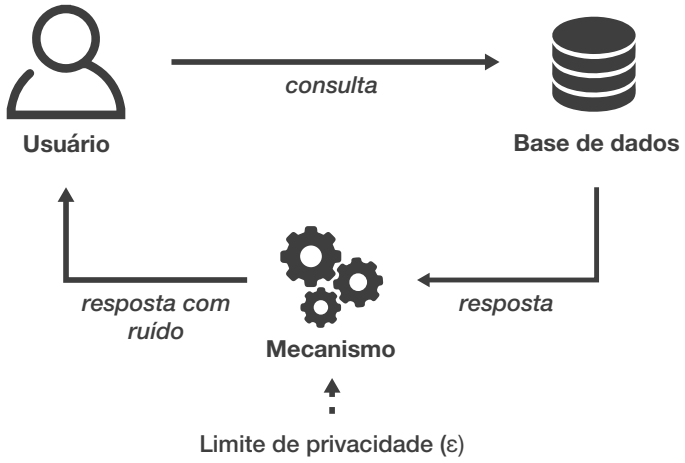

Figura 3.7. Ambiente interativo no modelo de Privacidade Diferencial.

A Privacidade Diferencial é satisfeita por um algoritmo aleatório, geralmente chamado de mecanismo. Este modelo foi projetado em um ambiente interativo, onde os usuários submetem consultas a um conjunto de dados e este, por sua vez, responde por meio de um mecanismo de anonimização. Este ambiente interativo é apresentado na Figura 3.7. Esse mecanismo proporcionará a privacidade, introduzindo "aleatoriedade" e protegendo os resultados das consultas sobre o conjunto de dados original.

A Privacidade Diferencial assegura que qualquer sequência de resultados (isto é, resposta de consultas) é igualmente possível de acontecer independente da presença de qualquer indivíduo no conjunto de dados [19]. Assim, a adição ou remoção de um indivíduo não afetará consideravelmente o resultado de qualquer análise estatística realizada no conjunto de dados [14]. Portanto, um adversário não deve ser capaz de aprender nada sobre um indivíduo específico que ele já não poderia ter aprendido antes sem acesso ao conjunto de dados.

3.7.2. Definição Formal

Dado um algoritmo aleatório (mecanismo) M, este mecanismo garante ε-Privacidade Diferencial se para todos os conjuntos de dados vizinhos D_1 e D_2 no conjunto de dados, estes diferem de no máximo um elemento, e para todo S contido na variação de resultados de M, isto é, para todo S $\subseteq$ Range(M),

$$Pr[M(D_1) \in S] \leq exp(\varepsilon) \times Pr[M(D_2) \in S],$$

onde Pr é a probabilidade dada a partir da "aleatoriedade" de M. Em resumo, a definição formal afirma que a diferença entre as probabilidades de uma consulta retornar o mesmo resultado em dois conjuntos de dados é limitada pelo parâmetro ε. Dessa forma, para qualquer par de entradas que diferem de apenas um registro, para cada saída, um adversário não poderá distinguir entre os conjuntos de dados D_1 e D_2 baseado apenas na resposta fornecida pelo mecanismo.

A Figura 3.8 mostra um exemplo de probabilidades de saída de um algoritmo M, nos conjuntos de dados vizinhos D_1 e D_2, a partir de um valor de ε. O algoritmo M fornece

garantias de privacidade adicionando ruído aleatório no seu retorno, i.e., $M(D) = f(D) + ruido$, onde f é a resposta de uma consulta realizada por um usuário. Nesse exemplo, as probabilidades seguem o mecanismo de Laplace, o que resulta em uma distribuição com pico mais acentuado do que uma distribuição normal. O conceito de mecanismo é definido na Seção 3.7.3.

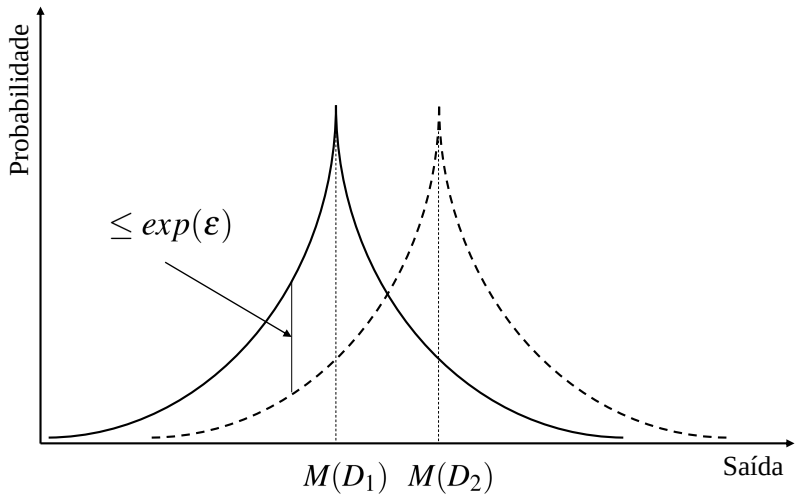

Figura 3.8. Probabilidades de saída de um algoritmo aleatório M sobre os conjuntos de dados vizinhos D_1 e D_2.

Se um indivíduo, portanto, escolhe participar de um conjunto de dados D onde análises estatísticas serão feitas através de um mecanismo que é ε-Diferencialmente Privado, esse mecanismo irá garantir que não haverá um aumento na probabilidade de violação de privacidade se comparado com a probabilidade quando o indivíduo escolhesse não participar do conjunto de dados. Dessa forma, podemos concluir que, como a Privacidade Diferencial é uma propriedade estatística sobre como o mecanismo funciona, as garantias que ela oferece são altas, inclusive essas garantias não dependem de poder computacional ou informações que um atacante possa ter obtido.

A definição formal de Privacidade Diferencial mostrada acima não leva em consideração como podemos escolher o parâmetro ε, principalmente porque esse parâmetro não possui correlação explícita com a privacidade dos indivíduos como em outras técnicas vistas. Esse parâmetro depende da consulta que está sendo feita e dos próprios dados que estão no conjunto de dados. A literatura em geral concorda que o valor de ε deva ser pequeno, como por exemplo 0.01, 0.1 ou até logarítmo natural $ln2$ ou $ln3$ [18]. Quanto menor o valor de ε, maior a privacidade. Dessa forma a escolha do valor de ε deve ser experimental e algumas vezes este valor é encontrado empiricamente, portanto, para cada mecanismo, deve ser feita uma análise para escolher o parâmetro adequado utilizando métricas [39] para avaliar a precisão da resposta do mecanismo com diversos valores de ε [30].

3.7.3. Mecanismo e Sensibilidade

Como foi dito anteriormente, a Privacidade Diferencial foi definida em um modelo interativo, onde o usuário submete consultas a uma base de dados D e, consequentemente, um

determinado mecanismo fornece uma resposta ε-Diferencialmente Privada. Porém, existem diversas formas de se atingir a Privacidade Diferencial através de um mecanismo. O objetivo das técnicas que utilizam esse modelo de privacidade é criar um mecanismo M que irá adicionar um ruído adequado para produzir uma resposta a uma consulta f feita pelo indivíduo, de forma que esse ruído seja independente do conjunto de dados D.

A quantidade de ruído necessária depende do tipo de consulta f aplicada sobre um conjunto de dados. Dessa forma precisamos definir o que é a sensibilidade de um conjunto de dados D. Antes disso, porém, precisamos entender o que são conjuntos de dados vizinhos.

Definição 2 *Dado um conjunto de dados D, todos os conjuntos de dados D_i decorrentes da remoção de um indivíduo i do conjunto de dados original D são definidos como vizinhos.*

Por exemplo, considere o conjunto de dados D na Figura 3.9a. Um de seus vizinhos pode ser obtido pela remoção do registro de ID $= 3$, resultando na Figura 3.9b, uma vez que não houve alteração nos valores dos registros.

ID	Peso (Kg)	Altura (m)
1	87,2	1,70
2	81,2	1,62
3	74,2	1,75
4	60,0	1,61
5	78,5	1,58

(a)

ID	Peso (Kg)	Altura (m)
1	87,2	1,70
2	81,2	1,62
4	60,0	1,61
5	78,5	1,58

(b)

Figura 3.9. Exemplo de conjuntos de dados vizinhos.

Definição 3 *Seja D o domínio de todos os conjuntos de dados. Seja f uma função de consulta que mapeia conjuntos de dados a vetores de números reais. A sensibilidade global da função f é:*

$$\Delta f = \max_{x,y \in D} \| f(x) - f(y) \|_1$$

para todo x, y diferindo de no máximo um elemento, ou seja, vizinhos. [18].

A sensibilidade então vai medir quanta diferença um usuário faz ao ser removido do conjunto de dados na resposta da função de consulta. Isso é fundamental para o cálculo adequado do ruído a ser adicionado pelo mecanismo, uma vez que quanto maior o valor de Δf, mais ruído terá de ser adicionado à resposta do mecanismo para mascarar a remoção de um indivíduo, de forma a assegurar a privacidade do mesmo [15].

O mecanismo de Laplace é o método mais comum e simples para alcançar a Privacidade Diferencial. A adição de ruído é baseada na geração de uma variável aleatória da distribuição de Laplace com média μ e escala b de forma que

$$Laplace_{\mu,b}(x) = \frac{1}{2b}\exp(-\frac{|x|}{b})$$

Podemos então definir formalmente o mecanismo de Laplace.

Definição 4 *Dada uma função de consulta $f : D \to \Re$, o mecanismo de Laplace M:*

$$M_f(D) = f(D) + Laplace(0, \Delta f/\varepsilon)$$

fornece ε-Privacidade Diferencial. Onde $Laplace(0, \Delta f/\varepsilon)$ retorna uma variável aleatória da distribuição de Laplace com média zero e escala $\Delta f/\varepsilon$.

3.7.4. Exemplo Explicativo

Considere o seguinte exemplo explicativo de utilização da Privacidade Diferencial em pequena escala. Suponha um conjunto de dados da Receita Federal, que contém o número de imóveis que um determinado indivíduo declarou, conforme Figura 3.10.

ID	Nome	Nº Imóveis
1	Roney	4
2	André	2
3	Leo	7
4	Bruno	1

Figura 3.10. Exemplo de conjunto de dados original contendo o número de imóveis de cada indivíduo.

Suponha também que a consulta f a ser realizada sobre essa base de dados retorna a soma de todos os imóveis de todos os indivíduos. Para aplicar a Privacidade Diferencial sobre esses dados é necessário calcular f para cada vizinho do conjunto original. A resposta real da consulta é 14. A Figura 3.11 mostra os conjuntos de dados vizinhos gerados a partir da base original e suas respectivas respostas da consulta f.

Para garantir a privacidade acerca de um indivíduo, ele participando ou não do conjunto de dados, é preciso calcular a variação máxima que a ausência do indivíduo provoca no resultado da consulta. Essa variação é proveniente da remoção do registro de $ID = 3$. Em seguida, é necessário calcular a sensibilidade da consulta aplicada ao conjunto de dados. Conforme definida nesta seção, a sensibilidade é calculada pela maior diferença $|f(D) - f(D_i)|$, e ocorre quando $i = 3$, pois gera como resultado $|14 - 7| = 7$. Por fim, o ruído a ser adicionado para atender ao modelo de Privacidade Diferencial, utilizando um mecanismo de Laplace, deve ser igual a $Laplace(0, \frac{7}{\varepsilon})$.

O parâmetro ε é definido pelo detentor dos dados. A Tabela 3.11 apresenta cinco exemplos de ruído, respostas e probabilidade de ocorrência após a aplicação da Privacidade Diferencial sobre o conjunto de dados original da Figura 3.10, considerando $\varepsilon = 1$.

ID	Nome	Nº Imóveis
2	André	2
3	Leo	7
4	Bruno	1

$f(D_1) = 2 + 7 + 1 = 10$

ID	Nome	Nº Imóveis
1	Roney	4
3	Leo	7
4	Bruno	1

$f(D_2) = 4 + 7 + 1 = 12$

ID	Nome	Nº Imóveis
1	Roney	4
2	André	2
4	Bruno	1

$f(D_3) = 4 + 2 + 1 = 7$

ID	Nome	Nº Imóveis
1	Roney	4
2	André	2
3	Leo	7

$f(D_4) = 4 + 2 + 7 = 13$

Figura 3.11. Conjuntos de dados vizinhos gerados a partir da base original e suas respectivas respostas da consulta f (soma).

Ruído	*f(D) + ruído*	*Pr(f(D) + ruído)%*
-4,58	9,42	3,70
-0,15	13,85	6,98
12,15	26,15	1,25
-6,43	7,57	2,85
2,89	16,89	4,72

Tabela 3.11. Cinco possíveis valores de ruído, resposta e probabilidade de ocorrência após a aplicação da Privacidade Diferencial.

Assim, após utilizar o mecanismo de Laplace, o valor de ruído de $-4,58$ possui probabilidade de ocorrência de $3,7\%$ sobre o valor original da consulta f (cuja soma do número de imóveis original é igual a 14), resultando em um valor anonimizado de $9,42$ imóveis. De forma análoga, o valor de ruído de $-0,15$ possui uma probabilidade um pouco maior de ocorrência $(6,98\%)$, caso a mesma consulta seja realizada nesse conjunto de dados, conforme mostra a Tabela 3.11.

3.7.5. Limitações e Desafios

A principal promessa da Privacidade Diferencial, como foi dito anteriormente, é que, se podemos extrair informações sem os dados de um indivíduo, então a privacidade do mesmo não foi violada. Contudo, nada impede que um adversário com conhecimento externo possa descobrir algo sobre um indivíduo sem ferir essa promessa da Privacidade Diferencial, exatamente porque esse indivíduo poderia nem fazer parte do conjunto de dados, mas poderia ter semelhanças com os outros indivíduos e o resultado do mecanismo ser utilizado para descobrir informações. Esse é um ponto de bastante criticidade da abordagem, porém isso não é considerado uma violação, já que o modelo de privacidade é relativo, i.e., participar ou não de um conjunto de dados que será submetido para análises somente aumenta ligeiramente o risco de descoberta [12].

No entanto existem alguns problemas na utilização da Privacidade Diferencial. O principal deles é o cálculo da sensibilidade da consulta que pode ser complexo e gerar muito ruído, o que irá afetar significativamente a utilidade dos dados. Apesar de existirem técnicas para minimizar essa perda, ainda é difícil balancear o ruído adicionado e manter a garantia da privacidade. O valor do parâmetro ε na prática também é difícil de ser estimado, especialmente porque existem poucos trabalhos relacionados e também pelo fato de que o seu valor não é uma medida direta de privacidade e sim um limitante do impacto que um usuário faz em um conjunto de dados. Os valores retornados pelo mecanismo podem não ser adequados para utilização em áreas específicas devido à natureza (probabilística) incerta de como esses dados são gerados. Então muitas vezes, nas aplicações reais, as técnicas de anonimização como *k*-anonimato ainda são bastante utilizadas.

3.8. Aplicações

Existem implementações práticas de aplicações que visam garantir a privacidade dos usuários diante de um mundo cada vez mais conectado e que cada vez mais gera informações através dos navegadores de internet e *smartphones*. Um exemplo desse tipo de aplicação é o *ARX*[1], um software livre que tem por objetivo prover fácil compreensão e suporte a diversos modelos de privacidade. O *ARX* dá suporte tanto para modelos de privacidade sintáticos, isto é, *k*-anonimato, *l*-diversidade, *t*-proximidade e δ-presença, como para modelos de privacidade semânticos, i.e ε-Privacidade Diferencial. O *ARX* também possibilita a anonimização dos dados após aplicadas operações de generalização e supressão, conforme Figuras 3.12 e 3.13. Nesse caso foi utilizado o modelo de privacidade *k*-anonimato.

	sex	age	race	marital-status	education	native-coun...	workclass
74	Male	52	White	Married-civ-spouse	Doctorate	United-States	Local-gov
75	Male	52	White	Married-civ-spouse	Masters	United-States	Local-gov
76	Male	51	White	Married-civ-spouse	Some-college	United-States	State-gov
77	Male	51	White	Married-civ-spouse	Bachelors	United-States	State-gov
78	Male	51	White	Married-civ-spouse	Bachelors	United-States	Local-gov
79	Male	51	White	Married-civ-spouse	Bachelors	United-States	Local-gov

Figura 3.12. Exemplo de dados originais no ARX.

	sex	age	race	marital-status	education	native-coun...	workclass
74	Male	[51, 60]	White	Spouse present	Higher education	North America	Government
75	Male	[51, 60]	White	Spouse present	Higher education	North America	Government
76	Male	[51, 60]	White	Spouse present	Higher education	North America	Government
77	Male	[51, 60]	White	Spouse present	Higher education	North America	Government
78	Male	[51, 60]	White	Spouse present	Higher education	North America	Government
79	Male	[51, 60]	White	Spouse present	Higher education	North America	Government

Figura 3.13. Exemplo de dados anonimizados pelo ARX.

O ARX também dispõe de métodos para a análise da utilidade desses dados e análise dos riscos de identificação (Figuras 3.14 e 3.15). Os respectivos resultados mostram a quantidade de registros em risco de violação de privacidade, o maior risco de violação e a porcentagem de sucesso caso um atacante deseje associar informações contidas na mesma

[1] http://arx.deidentifier.org/

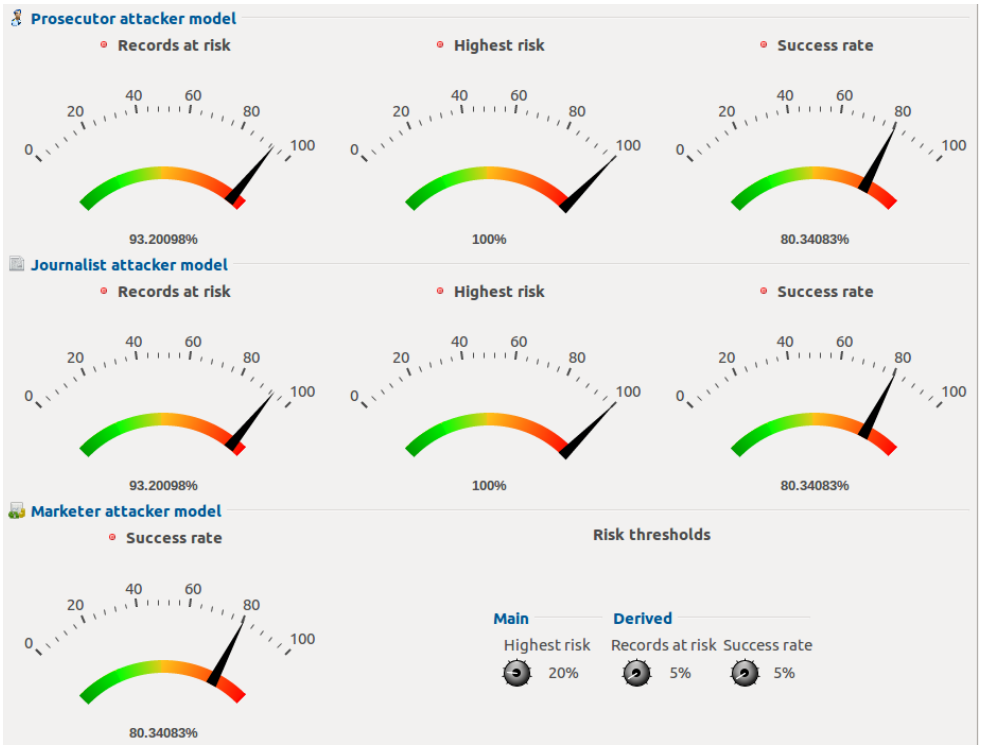

Figura 3.14. Exemplo de análise de riscos nos dados originais feita pelo ARX.

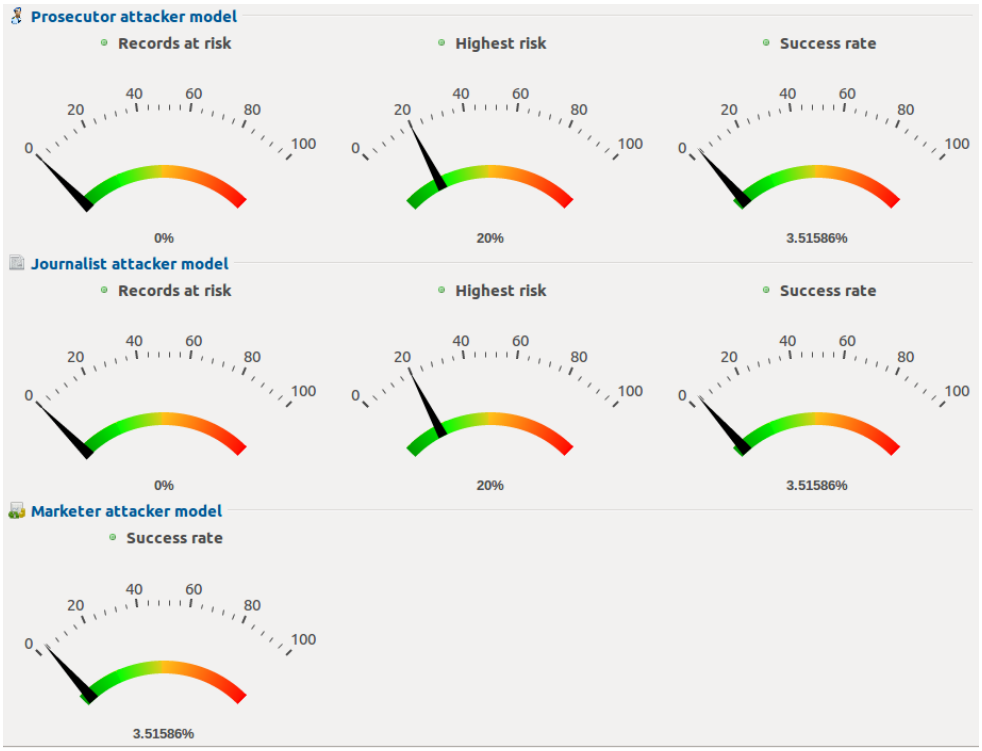

Figura 3.15. Exemplo de análise de riscos nos dados anonimizados feita pelo ARX.

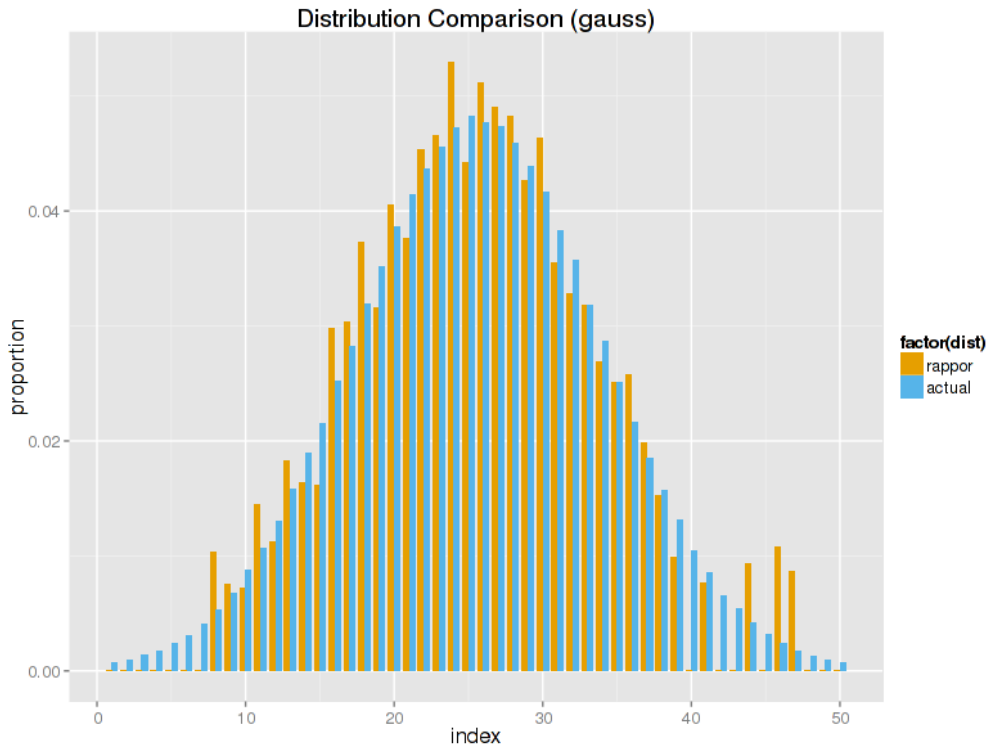

Figura 3.16. Exemplo de análise de distribuição feita pelo RAPPOR.

tabela. Além do Software com interface gráfica, o *ARX* também disponibiliza uma API que é capaz de possibilitar a anonimização de dados para qualquer aplicação Java.

Por outro lado, muitas organizações detentoras de dados têm desenvolvido aplicações para garantir a privacidade de dados coletados de seus clientes utilizando o modelo de Privacidade Diferencial. A Microsoft, por exemplo, desenvolveu o *PINQ* [35], uma plataforma de análise de dados projetada para fornecer garantias de Privacidade Diferencial. O *PINQ* utiliza um sistema de consultas próprio, semelhante ao SQL, denominado LINQ, e age como uma camada intermediária entre uma base de dados e o detentor dos dados, permitindo-o realizar consultas sem comprometer a privacidade dos indivíduos pertencentes a base. Um protótipo do *PINQ* está disponível para download no site oficial da Microsoft para experimentação. O *PINQ* possui vários exemplos de aplicações, além de ter sido projetado para que possa ser utilizado mesmo sem conhecimentos profundos sobre privacidade.

Com a grande necessidade de monitorar estatísticas de usuários, por exemplo configurações do navegador, o Google também lançou uma ferramenta denominada *Rappor* (*Randomized Aggregatable Privacy Preserving Ordinal Responses*), que visa proteger essas informações e facilitar a vida de quem precisa de dados coletados através de aplicações. Essa ferramenta utiliza a perturbação na coleta de dados, mantendo as informações estatísticas necessárias para realizar suas análises e preservando a privacidade dos usuários que utilizam seu navegador. É possível encontrar a implementação do demo no Github, que faz uma simulação e uma análise dos dados utilizando Python e R. Um exem-

plo de uma das análises feita pelo *Rappor* é vista na Figura 3.16, onde pode-se visualizar a distribuição dos dados originais, gerados pela simulação, e a distribuição dos dados alterados pelo *Rappor* para garantir a privacidade diferencial.

Outros dois exemplos recentes de aplicações no mundo real que também utilizam as técnicas vistas neste capítulo são apresentados em [37] e [10]. No primeiro, denominado $DP-WHERE$, os autores buscam preservar a privacidade de dados de mobilidade utilizando a rede de dados de celulares. Eles demonstram que é possível balancear privacidade e utilidade em aplicações práticas utilizando grandes volumes de dados. O segundo trabalho propõe liberar estatísticas acerca de 70 milhões de senhas utilizando Privacidade Diferencial. Os autores provam que o mecanismo proposto introduz mínima distorção nos dados, assegurando que a lista de senhas disponibilizada é muito próxima da lista original.

Por fim, a Apple anunciou em 2016 que vem aplicando Privacidade Diferencial na coleta de dados dos usuários do iOS 10. Ela foi a primeira a adotar o modelo de Privacidade Diferencial em larga escala, apesar da Microsoft já estudar o assunto há um certo tempo. Dessa forma, recursos como *Siri* e até o *QuickType* poderão predizer melhor as palavras que, por exemplo, um determinado conjunto de usuários mais utiliza. Com a utilização de Privacidade Diferencial, a Apple está enviando mais informações dos dispositivos para seus servidores que antes, mas garantindo a privacidade de seus usuários através da Privacidade Diferencial.

3.9. Considerações Finais

Este capítulo conclui que a preservação de privacidade de dados acerca de indivíduos é um problema bastante desafiador. Técnicas de anonimização têm sido utilizadas para a disponibilização de dados sensíveis, buscando um balanceamento perfeito entre privacidade e utilidade, que atenda às diversas partes envolvidas no processo de disponibilização de dados. Diferentes tipos de ataques à privacidade têm sido empregados por usuários maliciosos com a intenção de violar informações sensíveis de bases de dados abertas. Para tal fim, os atacantes utilizam conhecimento que muitas vezes é imensurável, devido aos diversos cenários em que informações podem ser obtidas. Além da anonimização, foram brevemente discutidas a criptografia e a tokenização como soluções alternativas para proteção de dados sensíveis. Em abordagens de privacidade sintáticas, que estabelecem uma determinada condição a qual os dados devem pertencer antes de serem disponibilizados, os modelos de privacidade utilizam, na grande maioria dos casos, técnicas de supressão ou de generalização. Já o modelo de Privacidade Diferencial, outro paradigma apresentado, procurar fornecer soluções de preservação de privacidade de um modo mais interativo, onde uma consulta é realizada sobre conjuntos de dados e então é adicionado ruído aleatório sobre seu resultado. Aplicações no mundo real, desenvolvidas por empresas preocupadas com a privacidade de seus usuários, implementam, via de regra, o modelo de Privacidade Diferencial, principalmente devido a possibilidade de interação entre os usuários e a base de dados, o que preserva as informações estatísticas necessárias para que analistas de dados possam realizar pesquisas de maneira mais precisa sem comprometer a privacidade dos fornecedores de dados.

Finalmente, vale ressaltar que não existe a "bala de prata" que atende a qualquer

requisito de privacidade e ao mesmo tempo fornece dados úteis para qualquer tipo de análise. Também não é a mudança de paradigma que vai solucionar o problema da disponibilização de dados. Tanto o paradigma de anonimização sintática, quanto o modelo de Privacidade Diferencial apresentam questões que devem ser vistas como oportunidades de pesquisas e desenvolvimento. Não se deve abandonar uma abordagem em prol de uma outra. Avanços em ambos os paradigmas são necessários para garantir que o futuro ofereça cada vez mais proteção à privacidade de indivíduos e ao mesmo tempo haja dados úteis e disponíveis para pesquisadores, testadores, analistas de dados, e muitos outros.

Agradecimentos

Este trabalho foi parcialmente financiado com recursos da CAPES e do LSBD/UFC.

Referências

[1] 4.5 Million Records Stolen from Community Health by Chinese Hackers (2014). *Infosecurity Magazine*, https://www.infosecurity-magazine.com/news/45-million-records-stolen-from/, Acessado em: 03.04.2017.

[2] Data Protection Laws of the World. https://www.dlapiperdataprotection.com/index.html, Acessado em: 09.05.2017.

[3] Directive 95/46/EC of the European Parliament. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML, Acessado em: 09.05.2017.

[4] FIPPA Legislative Review. http://www.gov.mb.ca/chc/fippa/fippareview.html, Acessado em: 09.05.2017.

[5] HIPAA for Individuals. https://www.hhs.gov/hipaa/for-individuals/index.html, Acessado em: 09.05.2017.

[6] Most americans unwilling to give up privacy to thwart attacks (2017). *Heuters*, http://www.reuters.com/article/us-usa-cyber-poll-idUSKBN1762TQ, Acessado em: 05.04.2017.

[7] Thousands of personal details exposed in latest uk data breach blunders (2014). *Infosecurity Magazine*, https://www.infosecurity-magazine.com/news/thousands-of-personal-details, Acessado em: 03.04.2017.

[8] Aggarwal, C. C. and Yu, P. S. (2008). A framework for condensation-based anonymization of string data. *Data Min. Knowl. Discov.*, 16(3):251–275.

[9] Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., and Zhu, A. (2005). *Anonymizing Tables*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[10] Blocki, J., Datta, A., and Bonneau, J. (2016). Differentially private password frequency lists. *IACR Cryptology ePrint Archive*, 2016:153.

[11] Branco Jr, E. C., Machado, J. C., and Monteiro, J. M. (2014). Estratégias para proteção da privacidade de dados armazenados na nuvem. In *XXIX SBBD: Tópicos em Gerenciamento de Dados e Informações 2014*. Sociedade Brasileira de Computação (SBC).

[12] Clifton, C. and Tassa, T. (2013). On syntactic anonymity and differential privacy. *Transactions on Data Privacy*, 6(2):161–183.

[13] Dewri, R., Ray, I., Ray, I., and Whitley, D. (2008). On the optimal selection of k in the k-anonymity problem. In *24th ICDE International Conference on Data Engineering*, pages 1364–1366.

[14] Domingo-Ferrer, J., Sánchez, D., and Soria-Comas, J. (2016a). Database anonymization: Privacy models, data utility, and microaggregation-based inter-model connections. *Synthesis Lectures on Information Security, Privacy, & Trust*, 8(1):1–136.

[15] Domingo-Ferrer, J., Sánchez, D., and Soria-Comas, J. (2016b). Database anonymization: Privacy models, data utility, and microaggregation-based inter-model connections. *Synthesis Lectures on Information Security, Privacy, & Trust*, 8(1):1–136.

[16] Domingo-Ferrer, J. and Torra, V. (2001). A quantitative comparison of disclosure control methods for microdata. *Confidentiality, disclosure and data access: theory and practical applications for statistical agencies*, pages 111–134.

[17] Dwork, C. (2006). Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming*, pages 1–12.

[18] Dwork, C. (2008). Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer.

[19] Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

[20] Fung, B. C., Wang, K., Fu, A. W.-C., and Yu, P. S. (2010a). *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Chapman & Hall/CRC, 1st edition. ISBN 978-1-4200-9148-9.

[21] Fung, B. C. M., Wang, K., Chen, R., and Yu, P. S. (2010b). Privacy-preserving data publishing: A survey of recent developments. *ACM Computer Survey*.

[22] Fung, B. C. M., Wang, K., and Yu, P. S. (2005). Top-down specialization for information and privacy preservation. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pages 205–216.

[23] Gavison, R. (1980). Privacy and the limits of law. *The Yale Law Journal*, 89(3):421–471.

[24] He, Y. and Naughton, J. F. (2009). Anonymization of set-valued data via top-down, local generalization. *Proceedings of the VLDB Endowment*, 2(1):934–945.

[25] Iyengar, V. S. (2002). Transforming data to satisfy privacy constraints. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 279–288.

[26] Jr., R. J. B. and Agrawal, R. (2005). Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pages 217–228.

[27] Kitchin, R. (2014). *The data revolution: Big data, open data, data infrastructures and their consequences*. Sage.

[28] Larose, D. T. (2014). *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons.

[29] Laurant, C. (2003). Privacy and Human Rights 2003: an International Survey of Privacy Laws and Developments. *In Electronic Privacy Information Center*.

[30] Lee, J. and Clifton, C. (2011). *How Much Is Enough? Choosing ε for Differential Privacy*, pages 325–340. Springer Berlin Heidelberg.

[31] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. (2005). Incognito: Efficient full-domain k-anonymity. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*, pages 49–60.

[32] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R. (2006). Mondrian multidimensional k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*, page 25.

[33] Li, N., Li, T., and Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *23th ICDE International Conference on Data Engineering (ICDE)*, pages 106–115.

[34] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). *l*-diversity: Privacy beyond *k*-anonymity. *In ACM Transactions on Knowledge Discovery from Data (TKDD)*.

[35] McSherry, F. (2010). Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 53(9):89–97.

[36] Meyerson, A. and Williams, R. (2004). On the complexity of optimal k-anonymity. In *Proceedings of the 23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Paris, France*, pages 223–228.

[37] Mir, D. J., Isaacman, S., Cáceres, R., Martonosi, M., and Wright, R. N. (2013). DP-WHERE: differentially private modeling of human mobility. In *Proceedings of the 2013 IEEE International Conference on Big Data, 2013, Santa Clara, CA, USA*, pages 580–588.

[38] Nergiz, M. E., Atzori, M., and Clifton, C. (2007). Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 665–676, New York, NY, USA. ACM.

[39] Nguyen, H. H., Kim, J., and Kim, Y. (2013). Differential privacy in practice. *Journal of Computing Science and Engineering*, 7(3):177–186.

[40] Solove, D. J. (2008). *Understanding Privacy*. Harvard University Press.

[41] Sweeney, L. (2002a). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588.

[42] Sweeney, L. (2002b). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, pages 557–570.

[43] Tan, V. Y. F. and Ng, S. (2007). Generic probability density function reconstruction for randomization in privacy-preserving data mining. In *Machine Learning and Data Mining in Pattern Recognition, 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007, Proceedings*, pages 76–90.

[44] Venkataramanan, N. and Shriram, A. (2016). *Data Privacy: Principles and Practice*. Chapman and Hall/CRC. ISBN 978-1-4987-2104-2.

[45] Wang, K. and Fung, B. C. M. (2006). Anonymizing sequential releases. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 414–423.

[46] Wang, K., Fung, B. C. M., and Yu, P. S. (2005). Template-based privacy preservation in classification problems. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), 27-30 November 2005, Houston, Texas, USA*, pages 466–473.

[47] Willison, D., Emerson, C., Szala-Meneok., K., Gibson, E., Schwartz, L., and Weisbaum, K. (2008). Access to medical records for research purposes: Varying perceptions across research ethics boards. *Journal of Medical Ethics 34*, pages 308–314.

[48] Wong, R. C. and Fu, A. W. (2010). *Privacy-Preserving Data Publishing: An Overview*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

[49] Wong, R. C., Li, J., Fu, A. W., and Wang, K. (2006). (alpha, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA*.

[50] Xiao, X. and Tao, Y. (2006). Personalized privacy preservation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, USA, June 27-29, 2006*, pages 229–240.

Capítulo 4

Computação aplicada a Cidades Inteligentes: Como dados, serviços e aplicações podem melhorar a qualidade de vida nas cidades

Fabio Kon, Eduardo Felipe Zambom Santana

Abstract

With the growth of the urban population, the infrastructural problems and limited resources of thousands of cities around the world affect negatively the lives of billions of people. Making cities smarter can help improving city services, decreasing social inequalities, and increasing the quality of life of their citizens. Information and communication technologies (ICT) are a fundamental means to move towards smarter city environments. Using a software platform on top of which Smart City applications can be deployed facilitates the development and integration of such applications. However, there are, currently, significant technological and scientific challenges that must be faced by the ICT community before these platforms can be widely used. This chapter presents the state-of-the-art and the state-of-the-practice in Smart Cities environments. We analyze various Smart City initiatives around the world and describe the most used technologies. We also describe a few Smart Cities platforms, which facilitate the development and integration of Smart City applications and services. Finally, we enumerate open research challenges and comment on our vision for the area in the future.

Resumo

Com o crescimento da população urbana, problemas de infraestrutura e de acesso limitado a recursos em diversas cidades ao redor mundo afetam negativamente a vida de bilhões de pessoas. Tornar as cidades mais inteligentes pode ajudar a melhorar os serviços urbanos aumentando a qualidade de vida de seus cidadãos e diminuindo as desigualdades sociais. A Tecnologia da Informação e Comunicação (TIC) são meios fundamentais para esse objetivo. Uma plataforma de software pode ser usada para facilitar enormemente a criação e integração de aplicações robustas para cidades inteligentes. Entretanto, ainda existem desafios técnicos e científicos significativos que necessitam ser

enfrentados antes que essas plataformas possam ser amplamente utilizadas. Este capítulo apresenta o estado da arte e o estado da prática em iniciativas e plataformas de software para cidades inteligentes. Para isso, analisamos diversos projetos de Cidades Inteligente ao redor do mundo e descrevemos as tecnologias mais utilizadas. Descrevemos também algumas plataformas para Cidades Inteligentes que facilitam o desenvolvimento e a integração de aplicações e serviços em Cidades Inteligentes. Finalmente, apresentamos os desafios de pesquisa ainda em aberto e discutimos a nossa visão para a área no futuro.

4.1. Introdução

Desde 2009, a maior parte da população mundial vive em cidades (United Nations 2009) e a infraestrutura e os recursos existentes nessas cidades muitas vezes não são suficientes para comportar o crescimento e a concentração da população. Uma forma de enfrentar esse problema é tornando as cidades mais inteligentes, otimizando o uso dos seus recursos e infraestrutura de uma forma sustentável, diminuindo as diferenças sociais e melhorando a qualidade de vida de sua população. Para atingir esse objetivo, tecnologias de informação e comunicação (TIC) podem ser empregadas para coletar e analisar uma grande quantidade de informações geradas por diversas fontes de dados da cidade como, por exemplo, redes de sensores, sistemas de trânsito e dispositivos dos cidadãos. Esses dados podem ser utilizados de formas inovadoras e criativas para a criação de aplicações integradas que melhorem os serviços da cidade e o uso de seus recursos. Entretanto, usar todos os dados de uma forma efetiva e eficiente é um desafio bastante complexo.

Neste capítulo serão mostradas aplicações e serviços de Cidades Inteligentes desenvolvidas em uma grande variedade de cenários, como por exemplo na melhoria e monitoramento do trânsito (Djahel et al. 2014; Barba et al. 2012), no monitoramento das condições da cidade (Vakali et al. 2014), para o controle de multidões (Franke et al. 2015), para o monitoramento do sistema de coleta de lixo (Perera et al. 2014), em sistemas de saúde (Hussain et al. 2015), de segurança pública (Galache et al. 2014) e o gerenciamento de recursos como água (Pérez-González and Díaz-Díaz 2015) e energia elétrica (Yamamoto et al. 2014).

Um problema na maioria das aplicações de cidades inteligentes, é que normalmente os sistemas são direcionados a um problema específico e são desenvolvidos sempre desde o início com pouco reuso de software e sem a comunicação entre esses sistemas. Essa abordagem leva a um maior trabalho, ao uso não otimizado dos recursos e impede a criação de aplicações que integrem dados e serviços de diversos domínios, o que é uma das principais características de cidades inteligentes.

Para resolver os problemas de integração entre as aplicações, uma das abordagens mais utilizadas tanto em pesquisas acadêmicas como em experimentos já realizados em algumas cidades é a utilização de uma plataforma de software que oferece diversos mecanismos e características não-funcionais para a utilização dos dados e serviços da cidade de uma forma integrada e com o objetivo de facilitar a implementação de aplicações de Cidades Inteligentes para cidadãos e administradores da cidade.

No entanto, muitos desafios técnicos e de pesquisa ainda precisam ser resolvidos antes que ambientes de Cidades Inteligentes eficazes e robustos sejam completamente desenvolvidos. Alguns dos maiores desafios são: permitir a interoperabilidade entre os

diversos componentes da cidade, garantir a privacidade e a segurança dos cidadãos e sistemas da cidade, gerenciar o armazenamento e o processamento de grandes quantidades de dados, oferecer a escalabilidade necessária para o aumento da população da cidade, incentivar a população a usar as aplicações e serviços oferecidos e lidar com a heterogeneidade de dispositivos como sensores e *smartphones*.

Este capítulo tem o objetivo de apresentar o conceito de Cidades Inteligentes, apontando as definições mais aceitas na literatura e discutindo as tecnologias necessárias para o desenvolvimento de cidades inteligentes. Depois serão descritas iniciativas, aplicações e serviços implantados em algumas cidades ao redor do mundo. Posteriormente, serão apresentados projetos de pesquisa para a implementação de plataformas de software para Cidades Inteligentes incluindo os requisitos funcionais e não funcionais que as plataformas devem disponibilizar. Também serão apresentados desafios técnicos e de pesquisa que ainda precisam ser resolvidos para o desenvolvimento de plataformas de Cidades Inteligentes eficientes e eficazes.

O restante deste capítulo está organizado da seguinte maneira, a Seção 4.2 discute as mais citadas definições de cidades inteligentes. A Seção 4.3 apresenta as principais tecnologias utilizadas para a implantação de cidades inteligentes. A Seção 4.4 apresenta diversas iniciativas de cidades inteligentes ao redor do mundo. A Seção 4.5 descreve plataformas de cidades inteligentes desenvolvidas em projetos de pesquisa e comerciais. A Seção 4.6 apresenta os requisitos identificados a partir das plataformas e iniciativas analisadas. A Seção 4.7 apresenta uma arquitetura de referência derivada a partir dos requisitos. A Seção 4.8 lista os principais desafios técnicos e de pesquisa para a criação de cidades inteligentes. A Seção 4.9 discute as implicações de Cidades Inteligentes para os diferentes papeis da cidade. Finalmente, a Seção 4.10 aponta as conclusões deste capítulo.

4.2. Definições de Cidades Inteligentes

Nesta seção, serão apresentadas e discutidas diversas definições de Cidades Inteligentes encontradas na literatura. Essas definições consideram desde as mudanças sociais esperadas com Cidades Inteligentes como o empoderamento e a melhora na qualidade de vida da população até o uso de TICs para a melhora na infraestrutura e nos serviços da cidade e na otimização do uso dos recursos da cidade.

A Tabela 4.1 apresenta diversas definições de cidades inteligentes. A maioria dessas definições citam explicitamente que o objetivo de uma cidade inteligente é a melhoria da qualidade de vida do cidadão. Algumas definições (Giffinger et al. 2007; Guan 2012) não estabelecem por qual meio isso deve ser alcançado, enquanto outras definem que isso será alcançado através da construção de uma infraestrutura tecnológica para melhorar os serviços da cidade (Caragliu et al. 2011; Dameri 2013; Harrison et al. 2010).

A maioria das definições citam a necessidade do uso de tecnologia da informação para otimizar o uso da infraestrutura da cidade, o gerenciamento dos recursos e os serviços da cidade (Harrison et al. 2010; Washburn et al. 2009). Algumas dessas definições ainda adicionam a necessidade do desenvolvimento sustentável da cidade, com a melhoria no uso de recursos como água e energia elétrica (Caragliu et al. 2011; Dameri 2013).

Tabela 4.1. Definições de Cidades Inteligentes

Definição	Autor
"A Smart City is a city well performing built on the 'smart' combination of endowments and activities of self-decisive, independent and aware citizens"	(Giffinger et al. 2007)
"A city to be smart when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance"	(Caragliu et al. 2011)
"A smart city is a well-defined geographical area, in which high technologies such as ICT, logistic, energy production, and so on, cooperate to create benefits for citizens in terms of well-being, inclusion and participation, environmental quality, intelligent development; it is governed by a well-defined pool of subjects, able to state the rules and policy for the city government and development"	(Dameri 2013)
"A city that monitors and integrates conditions of all of its critical infrastructures, including roads, bridges, tunnels, rails, subways, airports, seaports, communications, water, power, even major buildings, can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services to its citizens"	(Hall et al. 2000)
"A city connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city"	(Harrison et al. 2010)
"A smart city, according to ICLEI, is a city that is prepared to provide conditions for a healthy and happy community under the challenging conditions that global, environmental, economic and social trends may bring."	(Guan 2012)
"The use of Smart Computing technologies to make the critical infrastructure components and services of city which include city administration, education, healthcare, public safety, real estate, transportation, and utilities more intelligent, interconnected, and efficient"	(Washburn et al. 2009)

Um aspecto relevante é a necessidade de uma cidade inteligente facilitar também o crescimento econômico da cidade (Dameri 2013) possibilitando a inclusão e participação de toda a população na sociedade. Duas definições (Dameri 2013; Giffinger et al. 2007) citam a participação da sociedade na decisão dos governos através de governos participativos. Outras questões importantes levantadas pelas definições, são o monitoramento da infraestrutura da cidade, como ruas, pontes, linhas de trem (Hall et al. 2000), o monitoramento do uso de recursos como água e energia elétrica (Hall et al. 2000) e a integração entre todos os serviços da cidade (Harrison et al. 2010; Washburn et al. 2009).

Além das definições apresentadas anteriormente, Giffinger et al. (Giffinger et al. 2007) descrevem seis dimensões para verificar o quão inteligente é uma cidade, que são: *Smart Economy*, *Smart People*, *Smart Governance*, *Smart Mobility*, *Smart Environment* e *Smart Living*, os quais traduzimos livremente como economia, população, governança, mobilidade, meio-ambiente e vida inteligentes. Muitos autores aceitam essa classificação (Hernández-Muñoz et al. 2011; Papa et al. 2013) e há ainda um *benchmark* desenvolvido para classificar as cidades mais inteligentes da Europa usando essas dimensões [1]. Essas dimensões são definidas da seguinte forma:

- **Economia Inteligente** mede o quão bem preparada economicamente uma cidade está, utilizando parâmetros como qualidade das empresas instaladas e o seu ambiente para empreendedorismo. Algumas ações desenvolvidas relacionadas a esta dimensão são incentivos a empresas para o desenvolvimento de soluções tecnológicas para a cidade e a melhoria do ambiente de negócios com legislação adequada à inovação e infraestrutura para negócios.

- **População Inteligente** mede o desenvolvimento da população da cidade usando parâmetros como educação, emprego e renda. Algumas ações relacionadas a esta dimensão são projetos para inclusão digital dos cidadãos e programas de educação científica e tecnológica.

- **Governança Inteligente** mede a qualidade e transparência dos órgãos públicos municipais com parâmetros como facilidade no uso dos serviços públicos, investimentos em tecnologia e transparência nos dados e no uso de recursos da cidade. Algumas ações relacionadas a esta dimensão são a criação de governos participativos e a divulgação de informações sobre a cidade em portais de transparência e de dados abertos.

- **Mobilidade Inteligente** mede a facilidade da mobilidade na cidade nos diversos modais de transporte como ônibus, metrô, carro e bicicleta. Usa parâmetros como quilômetros de congestionamento, tamanho da malha metroviária e quantidade de pessoas que usam transporte público ou não-poluente. Algumas ações relacionadas a esta dimensão são o monitoramento em tempo real do fluxo nas vias da cidade, o uso de sensores para indicar vagas de estacionamento livres e aplicações para facilitar e incentivar o uso de transporte público e sustentável, tais como bicicletas.

- **Meio-Ambiente Inteligente** mede a sustentabilidade na cidade usando parâmetros como poluição ambiental, eficiência no uso de recursos como água e energia elétrica

[1] Smarts Cities in Europe - `http://www.smart-cities.eu`

e a quantidade de lixo reciclado. Algumas ações relacionadas a esta dimensão são a medição da qualidade do ar e água da cidade, o uso de fontes renováveis de energia e a medição em tempo real dos recursos utilizados em residências.

- **Vida Inteligente** mede a qualidade de vida da população usando parâmetros como entretenimento, segurança e cultura como quantidade de áreas verdes, número de bibliotecas e taxa de homicídios da cidade. Algumas ações relacionadas a esta dimensão são o uso de aplicações para o acompanhamento da saúde de idosos, o processamento automático de imagens de câmeras de segurança e aplicativos que mostram os eventos culturais programados na cidade.

Atualmente a expressão Cidades Inteligentes *(Smart Cities)* está bem estabelecida, porém existem algumas outras expressões que também indicam características similares à ideia de Cidades Inteligentes. Algumas dessas expressões são: Cidades Digitais *(Digital City)*, Cidades do Conhecimento *(Knowledge City)* e Cidades Conectadas *(Wired City)*.

Neste capítulo, apenas a expressão Cidades Inteligentes *(Smart Cities)* foi considerada; isso porque, atualmente ela é a mais utilizada, como mostra a Figura 4.1 gerada pela ferramenta Google Trends[2], a qual mostra a quantidade de buscas feitas por cada expressão. Além disso, essa expressão é a única que claramente define que a cidade deve disponibilizar serviços integrados aumentando a inteligência da cidade para melhorar a qualidade de vida de todos seus cidadãos.

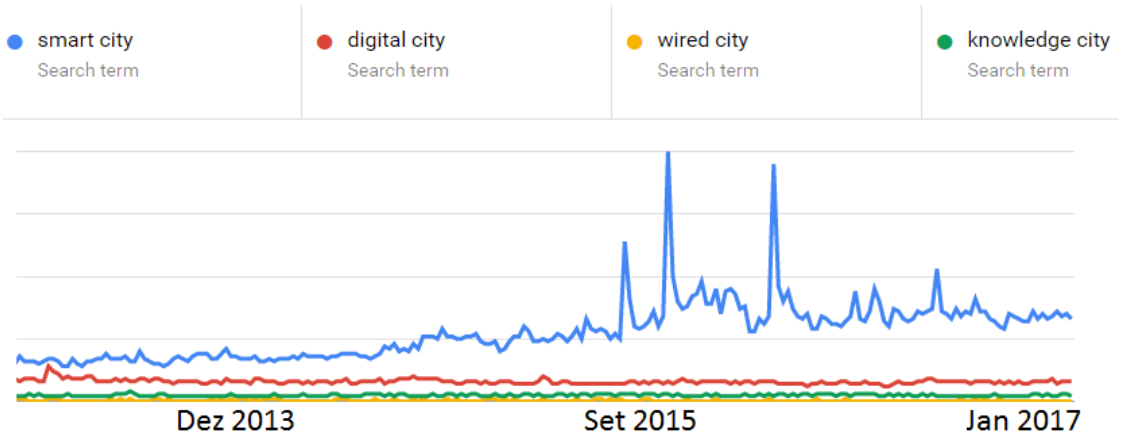

Figura 4.1. Pesquisas relacionados a expressões relacionadas a Cidades Inteligentes

As expressões Cidades Digitais e Cidades Conectadas estão relacionadas ao fornecimento de serviços digitais aos cidadãos utilizando uma infraestrutura de TI, mas sem necessariamente a integração entre os diversos tipos de aplicações e domínios que podem existir em uma cidade. A expressão Cidade do Conhecimento, além de ser pouco usada, está relacionada mais ao domínio da educação, tendo o objetivo de empoderar o cidadão através da educação. Dois trabalhos apresentam uma discussão sobre essas diferentes expressões (Cocchia 2014; Yin et al. 2015).

[2]Google Trends - https://trends.google.com

Existem iniciativas em diversas cidades ao redor do mundo, a maioria na Europa (Caragliu et al. 2011; Manville et al. 2014), diversas nos Estados Unidos [3], Japão e China (Liu and Peng 2013) e alguns projetos em outras partes do mundo como Brasil (Fortes et al. 2014), Emirados Árabes (Janajreh et al. 2013) e Coreia do Sul (Kshetri et al. 2014). Esses dados mostram que a grande maioria dos projetos estão concentrados em países desenvolvidos, existem alguns poucos projetos em países em desenvolvimento. No Brasil já existem diversas iniciativas como em São Paulo, Búzios, Recife e Joinville. Nenhum projeto foi encontrado nos países mais pobres do globo. A Figura 4.2 mostra um mapa com as iniciativas encontradas na literatura ou páginas dos projetos.

Figura 4.2. Algumas iniciativas de Cidades Inteligentes ao redor do mundo.

Alguns exemplos de iniciativas bastante avançadas em Cidades Inteligentes são: Santander, na Espanha, que através do projeto SmartSantander já implantou uma grande redes de sensores na cidade para coletar dados como temperatura, vagas livres de estacionamento e ruído. Amsterdã, na Holanda, que possuí diversos projetos de Cidades Inteligentes como uso de carros elétricos, incentivo ao uso de bicicletas e transporte público e o monitoramento automático das condições da cidade.

4.3. Tecnologias e Conceitos

Apresentamos agora as principais tecnologias usadas na criação da infraestrutura de uma Cidade Inteligente, i.e., (1) Internet das Coisas, para possibilitar a conexão de diversos dispositivos na rede da cidade como sensores, sinais de trânsito e dispositivos de usuários; (2) Big Data, viabilizando o armazenamento e o processamento de grandes quantidades de dados coletados na cidade; (3) Sensoriamento Móvel Participativo, que possibilita que dispositivos de usuários como smartphones coletem uma grande quantidade de dados sobre a cidade; e (3) Computação em Nuvem, fornecendo um ambiente escalável e elástico que suporte a grande demanda de recursos computacionais necessários em uma cidade inteligente.

[3] 10 Smartest Cities in USA - http://www.fastcoexist.com/3021592/the-10-smartest-cities-in-north-america

4.3.1. Internet das Coisas

A Internet das Coisas (*Internet of Things* ou IoT) é a conexão de objetos do cotidiano, tais como lâmpadas para iluminação pública, semáforos de trânsito, sensores de qualidade de água e ar, câmeras de vídeo, etc. à rede Internet. Os objetos devem ser identificados com um nome único, sua posição e estado conhecidos, e devem ser acessíveis por meio de uma rede interoperável (Coetzee and Eksteen 2011).

Podem haver uma grande variedade de “Coisas"conectadas em um sistema de IoT, desde celulares, relógios e computadores até veículos e geladeiras. O *Cluster of European Research Projects on IoT* (Sundmaeker et al. 2010) define “Coisas” como participantes ativos da rede que são capazes de interagir e comunicar-se com outros elementos da rede e com o ambiente. Essa comunicação ocorre para a troca de dados e informações sobre o ambiente. A Internet das Coisas conecta o mundo digital e físico adicionando serviços e inteligência para a internet sem a intervenção direta de seres humanos. Podemos destacar três componentes principais em um sistema de IoT: (1) o hardware, como sensores, atuadores e aparelhos de comunicação; (2) o middleware para o processamento e armazenamento dos dados capturados pelo hardware e (3) uma camada de apresentação na qual usuários ou administradores do sistema podem acessar, manipular e analisar os dados (Gubbi et al. 2013).

A Internet das Coisas é bastante adequada para o gerenciamento dos milhares de dispositivos que estarão conectados em uma cidade inteligente. Assim, os dados coletados na cidade são enviados para as plataformas de software ou para as aplicações para que sejam armazenados e processados possibilitando a criação de serviços inovadores para a cidade.

Diversas iniciativas de Cidades Inteligentes usam IoT para a manutenção e gerenciamento dos dispositivos da cidade, como por exemplo o SmartSantander (Sanchez et al. 2014) que já possui mais de 20 mil sensores instalados na cidade de Santander, o Padova Smart City (Zanella et al. 2014), que possui mais de 300 sensores instalados e o *Array of Things*, que está instalando uma grande rede de sensores na cidade de Chicago.

A Internet das Coisas possui uma enorme quantidade de aplicações potenciais em Cidades Inteligentes. Alguns exemplos são: monitoramento da estrutura de prédios históricos, detecção se latas de lixo estão cheias, monitoramento de barulho perto de áreas críticas como escolas e hospitais, monitoramento das condições de semáforos e lâmpadas de iluminação pública e o monitoramento do uso de energia elétrica e água em Casas Inteligentes (Zanella et al. 2014).

4.3.2. Big Data

A expressão *Big Data* se refere a um conjunto de técnicas e ferramentas para o armazenamento e manipulação de conjuntos de dados muito grandes, onde tecnologias tradicionais, como bancos de dados relacionais e ferramentas de processamento sequencial, não suportam o vasto volume de dados (Chen et al. 2014; Demchenko et al. 2014). Big Data possui quatro características marcantes ilustradas na Figura 4.3:

- Volume: a quantidade de dados gerados e coletados em diversos tipos de aplicações está aumentando exponencialmente e as ferramentas de Big Data devem ser capazes

de lidar apropriadamente com esse desafio.

- Variedade: os dados podem ser coletados de diferentes fontes e com diferentes formatos e estruturas; como dados estruturados como os dados dos cidadãos, dados semi-estruturados como os dados de sensores e os dados não-estruturados como câmaras de vídeo de segurança e de trânsito.

- Velocidade: o processamento de dados deve ser rápido e, em muitos casos, em tempo real, ou esses dados podem se tornar inúteis como dados coletados de sensores de veículos, a análise de redes sociais e informações sobre o trânsito da cidade.

- Veracidade: como os dados serão coletados de múltiplas fontes de dados, é importante garantir a qualidade desses dados, utilizando fontes confiáveis e consistentes. Isso é importante para evitar erros comprometendo a análise dos dados.

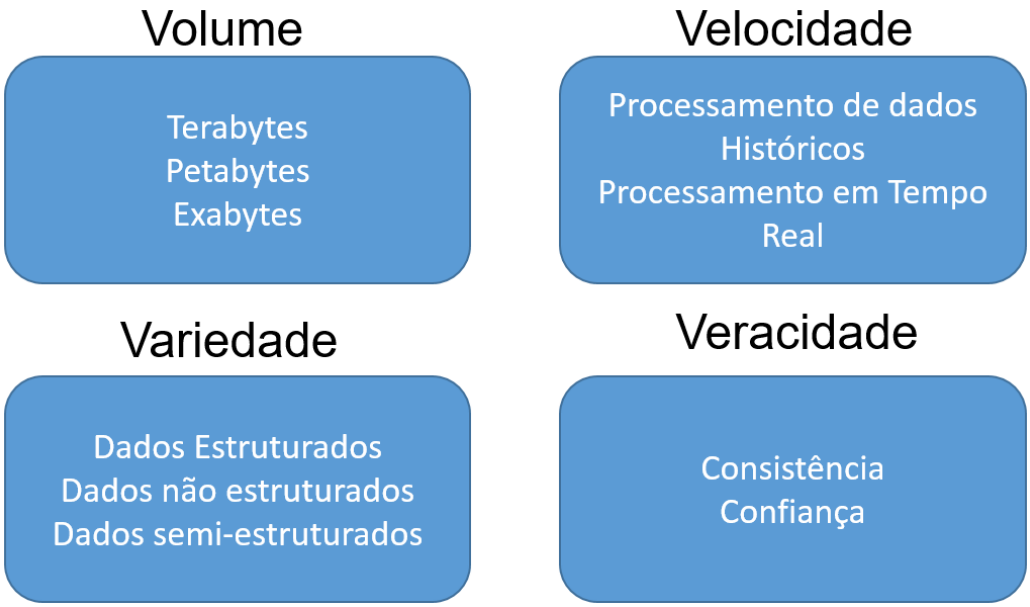

Figura 4.3. 4 Vs de Big Data

Big Data não é apenas uma estrutura de armazenamento moderna e escalável como bancos de dados NoSQL, ou ferramentas de processamento paralelo poderosas como o Hadoop (Polato et al. 2014; Goldman et al. 2012), mas sim a transformação de todo o ciclo de vida dos dados dentro de uma aplicação, para suportar a coleta, armazenamento, processamento, análise e visualização de grandes conjuntos de dados.

No contexto de cidades inteligentes, ferramentas de Big Data estão sendo empregadas para permitir o gerenciamento da grande quantidade de dados gerados nas cidades. Por exemplo, dados que são gerados por sensores periodicamente sobre as condições da cidade como temperatura, qualidade do ar e pluviometria, dados gerados por cidadãos através de telefones celulares e redes sociais e veículos como ônibus que podem enviar periodicamente sua posição e velocidade para aplicações.

Muitas ferramentas de Big Data já estão sendo usadas por iniciativas e plataformas de software para cidades inteligentes. Algumas dessas ferramentas são bancos de dados NoSQL (Khan et al. 2013; Bain 2014) como o MongoDB e o HBase, ferramentas de processamento paralelo (Parkavi and Vetrivelan 2013; Takahashi et al. 2012) como o Apache Hadoop e o Apache Spark, processadores de fluxos de dados em tempo real (Girtelschmid et al. 2013) como o Apache Storm e ferramentas de visualização de dados (Khan et al. 2013) como o RapidMiner.

Os bancos de dados NoSQL são utilizados principalmente para armazenar dados não estruturados da cidade, como por exemplo leituras de sensores e posições de veículos. Ferramentas de processamento paralelo são utilizadas para o processamento de grandes conjuntos de dados, e são utilizados tanto para processamento de dados históricos com o Hadoop ou Spark quanto para processamento em tempo-real de grandes fluxos de dados também com o Spark e o Storm.

Há inúmeras possibilidades de aplicação de tecnologias e ferramentas de Big Data em cidades inteligentes. Podemos citar como exemplos o reconhecimento de padrões em trânsito usando dados históricos para descobrir as causas e evitar congestionamentos, facilitar as decisões de administradores da cidade usando análises sobre grandes conjuntos de dados, prever a quantidade de energia elétrica utilizada em diferentes dias e horários utilizando dados históricos e fluxos de dados em tempo real, prever a demanda do uso de transporte público utilizando dados históricos sobre a venda de passagens e detectar automaticamente problemas de segurança pública utilizando fluxos de dados de sensores e redes sociais (Al Nuaimi et al. 2015).

Além da grande quantidade de dados, também está sendo bastante discutida a ideia de dados abertos (*Open Data*). Muitas cidades ao redor do mundo já disponibilizam grande quantidade de dados para a consulta da população como Dublin (Stephenson et al. 2012), Barcelona[4], Chicago[5] e São Paulo[6]. Além dos portais de dados abertos, também existe a disponibilização de dados sobre equipamentos públicos como estações de empréstimo de bicicletas e os veículos de transporte público da cidade. Dois problemas ainda comuns nessas iniciativas são a demora na publicação dos dados e a publicação de dados incompletos ou que são difíceis de interpretar.

4.3.3. Sensoriamento Móvel Participatório

O Sensoriamento Móvel Participatório (Mobile Crowd Sensing) é um paradigma de sensoriamento que utiliza os dados de dispositivos dos usuários do sistema, como telefones e relógios inteligentes e veículos (Guo et al. 2015). Essa tecnologia permite a utilização de uma grande quantidade de dados como a localização do usuário, o uso de sensores embutidos nos dispositivos e as condições de trânsito na região do usuário.

No contexto de cidades inteligentes, essa tecnologia pode ser utilizada para a implementação de uma grande quantidade de aplicações e serviços. Alguns exemplos citados por Guo et al. (Guo et al. 2015) são:

[4]Open Data Barcelona - http://opendata.bcn.cat/opendata/en
[5]City of Chicago, Data Portal - https://data.cityofchicago.org
[6]São Paulo Dados Abertos - http://dados.prefeitura.sp.gov.br

- **Saúde**: Aplicações para medir a exposição das pessoas a poluição do ar combinando dados de sensores estáticos e da localização das pessoas recuperados de seus dispositivos e para o acompanhamento do tratamento de doenças utilizando smartphones.

- **Transporte**: Aplicações que reportam em tempo real as condições de trânsito utilizando dados de dispositivos ou veículos dos usuário e o uso de smartphones para calcular a lotação de um ônibus.

- **Segurança**: Aplicações que indicam para usuários se o lugar que ele está é seguro ou não utilizando o histórico de crimes da regiões.

- **Planejamento Urbano**: Aplicações que analisam a movimentação das pessoas na cidade para encontrar padrões de mobilidade, facilitando o planejamento de novas vias, itinerários de ônibus e novas linhas de metrô.

A utilização dos dados dos dispositivos dos usuários tem duas grande vantagens sobre a implantação de uma rede de sensores estáticos: Não possui custos para a implantação e manutenção dos dispositivos e tem uma área de cobertura maior, pois as pessoas se movimentam por toda a cidade. Entretanto, o sensoriamento móvel participatório possui grandes desafios ainda como o engajamento da população, problemas de privacidade e a confiabilidade dos dados.

4.3.4. Computação em Nuvem

A Computação em Nuvem oferece uma infraestrutura elástica, robusta e altamente disponível para o armazenamento e processamento de dados, o que é essencial para aplicações de Cidades Inteligentes. Adicionalmente, uma cidade inteligente pode ser altamente dinâmica, requerendo reconfigurações automáticas de sua infraestrutura, o que é também facilitado pela computação em nuvem.

Alguns autores (Distefano et al. 2012; Aazam et al. 2014) descrevem um novo paradigma a partir da combinação da Computação em Nuvem e da Internet das Coisas, chamando-o de "Cloud of Things". A ideia é armazenar e processar todos os dados coletados de uma rede IoT em um ambiente de Computação em Nuvem, o que já é realizado em diversas iniciativas de cidades inteligentes (Mitton et al. 2012; Tei and Gurgen 2014).

Outro conceito relacionado ao uso da Computação em Nuvem em Cidades Inteligente é o Software como Serviço (Software as a Service - SaaS) (Fox et al. 2013). Perera et al. estendem esse conceito, usando a expressão "Sensing as a Service" (Perera et al. 2014). O objetivo é fornecer a aplicações e serviços os dados de sensores em uma infraestrutura de Computação em Nuvem. A plataforma ClouT também usa esse conceito e define as expressões *City Application Software as a Service* (CSaaS) e *City Platform as a Service* (CPaaS) disponibilizando todos as funcionalidades da plataforma como serviços em um ambiente de Computação em Nuvem (Tei and Gurgen 2014).

Resumindo, Computação em Nuvem é ideal para fornecer a infraestrutura para armazenar e executar os serviços de uma cidade. Os dados podem ser colhidos por uma rede implantada com as ideias de Internet das Coisas e enviados para a infraestrutura de

Computação em Nuvem, onde os dados podem ser processados utilizando ferramentas de Big Data. Essa combinação ajuda a oferecer diversos requisitos não-funcionais importantes como escalabilidade, elasticidade e segurança (Chen et al. 2014; Aazam et al. 2014).

4.4. Aplicações e Serviços de Cidades Inteligentes

Nesta seção apresentaremos diversos serviços e aplicações de Cidades Inteligentes desenvolvidos e testados em diversas cidades como Santander, Barcelona, Amsterdã e São Paulo. Os serviços e aplicações serão divididos utilizando as dimensões apresentadas na Seção 4.2. Também discutiremos os dados, as tecnologias e a infraestrutura necessária nas aplicações em cada uma das áreas.

Uma aplicação ou serviço pode ser relacionado a mais de uma dimensão, como por exemplo uma aplicação de mobilidade que tem o objetivo de melhorar o trânsito da cidade e consequentemente também reduz a emissão de poluentes melhorando o meio ambiente. Nesses casos, descreveremos o serviço na dimensão que o projeto foi inicialmente pensado e também indicaremos os outros benefícios alcançados.

4.4.1. Economia Inteligente

Esta subseção apresenta projetos que possuem o objetivo de melhorar a economia da cidade, atraindo investimentos e criando mais e melhores empregos. Alguns serviços e aplicações comuns nessa área são relacionados a turismo, maior eficiência na utilização de recursos e a atração de empresas e startups para a cidade.

Em Santander foi desenvolvida uma aplicação de realidade aumentada [7] para dispositivos móveis que contém informações de mais de 2700 pontos de interesse da cidade como museus, livrarias, pontos de ônibus, oficinas de turismo e estações de aluguel de bicicletas, além de mostrar em tempo real a posição de ônibus e táxis (Sanchez et al. 2014). Essa aplicação tem o objetivo de facilitar a vida dos turistas, informando os pontos mais importantes da cidade e também facilitando a locomoção dentro da cidade. A Figura 4.4 mostra uma imagem dessa aplicação, na qual são mostradas informações sobre uma linha de ônibus e de um ponto de interesse na cidade.

Em Cagliari, na Italia, está em desenvolvimento uma plataforma baseada na Internet das Coisas e na Computação em Nuvem para recolher dados e criar serviços para os turistas da cidade (Nitti et al. 2017). Para testar a aplicação, foi desenvolvido um estudo de caso no qual um turista seleciona uma lista de Pontos de Interesse (PdI), que deseja visitar na cidade. Cada PdI possui diversos dados estáticos, como endereço e horário de funcionamento e dados capturados em tempo real como o tamanho na fila de entrada e a quantidade de visitantes. Com esses dados, o aplicativo calcula qual a melhor sequência de PdIs que o turista deve visitar. O objetivo da aplicação é maximizar o tempo do turista, permitindo que ele consiga visitar o maior número de atrações possíveis no tempo que ele estiver na cidade.

Também relacionado a turismo, em Amsterdã está sendo utilizado o CitySDK Tourism API (Pereira et al. 2015), uma ferramenta que permite o desenvolvimento de

[7] SmartSantanderRA - https://play.google.com/store/apps/details?id=es.unican.tlmat.smartsantanderra

Figura 4.4. Aplicação de realidade aumentada do projeto SmartSantander.

aplicações para ajudar os turistas que visitam a cidade. Essa ferramenta coleta os dados do portal de dados abertos da cidade, que estão em formato CSV, XLS e arquivos texto e os disponibiliza em uma API de fácil acesso e processamento para as aplicações. Alguns dos dados compartilhados são os pontos de interesse da cidade como museus, parques e construções históricas, eventos que estão acontecendo na cidade e itinerários turísticos.

Búzios, no estado do Rio de Janeiro, é uma das primeiras cidades do Brasil (Fortes et al. 2014) a iniciar um projeto para a implantação de uma infraestrutura de Cidade Inteligente. O projeto teve três objetivos principais, tornar a cidade mais sustentável, com uso mais racional dos recursos e com mais eficiência. Entre as principais ações realizadas na cidade estão a implantação de uma rede de energia elétrica inteligente, a criação de prédios inteligentes, onde seja possível monitorar o uso de recursos de casas e edifícios públicos e comerciais e a melhoria dos sistemas de comunicação da cidade utilizando tecnologias de comunicação como Wi-Fi, redes Mesh e pelas linhas de energia (*Power Lines Communication - PLC*).

Em Amsterdã, na Holanda, está sendo implantado o primeiro *Smart Electricity Grid* em uma região da cidade com aproximadamente 10 mil habitações. Nessa rede é possível que os usuários consumam e produzam energia e que possam acompanhar em tempo real o uso de energia em suas casas. Além disso, esse projeto também facilita o monitoramento e manutenção da rede pelas autoridades da cidade.

Os dados utilizados nas aplicações e serviços apresentados nesta seção são informações sobre pontos de interesse na cidade como museus, praças e parques, a posição de veículos como ônibus e táxis e informações sobre o uso de recursos na cidade como água, energia elétrica e água.

4.4.2. População Inteligente

Esta subseção apresenta projetos que visam a melhorar parâmetros sociais relativos à população da cidade como educação, emprego e renda e também o empoderamento dos cidadãos com dados que permitam que eles façam escolhas melhores. Alguns serviços e aplicações comuns nessa área são relacionados a educação como aplicações para melhorar o ensino e para facilitar a inclusão digital dos cidadãos e a melhoria do ambiente de negócios da cidade, aumentando a quantidade e a qualidade dos empregos.

Uma iniciativa na Inglaterra ensina estudantes a trabalhar com conjuntos de dados relacionados à cidade (Wolff et al. 2015). A ideia é permitir que os cidadãos conheçam ferramentas para analisar os dados independentemente da vontade de empresas ou de governantes. A proposta desse trabalho é estender as atividades de classe com atividades como *Hackathons* voltados para o desenvolvimento de aplicações e serviços para a cidade. Alguns testes já foram feitos com conjuntos de dados sobre o uso de energia elétrica.

Em Barcelona, foi criado um laboratório (*Barcelona Urban Innovation Lab & Dev*) que pesquisa diversos problemas urbanos e que fomenta a participação do setor privado no desenvolvimento de produtos e serviços relacionados à melhora do espaço urbano (Bakıcı et al. 2013). O laboratório disponibiliza recursos humanos e ferramentas para apoiar o desenvolvimento das soluções. O objetivo desse laboratório é atrair empresas para o desenvolvimento das ferramentas para a cidade, criando empregos que necessitam de alta qualificação e também criando soluções para melhorar a vida na cidade.

Nesta seção foram apresentados serviços que tem o objetivo de empoderar o cidadão com educação e melhores possibilidade de emprego. Dois exemplos de ações são as de educação de estudantes e a atração de empresas para o desenvolvimento de soluções para a cidade.

4.4.3. Governança Inteligente

Esta subseção apresenta projetos ligados a área de governança da cidade, que tem o objetivo de facilitar a administração da cidade e possibilitar a maior participação da população nas decisões da cidade. Alguns serviços e aplicações comuns nessa área são relacionados a plataformas e portais de monitoramento da cidade, a criação de portais de dados abertos e o incentivo da população a participar das decisões dos gestores da cidade.

Seattle é considerada por alguns rankings a cidade mais inteligente dos Estados Unidos [8]. Nessa cidade, foi realizada uma pesquisa (AlAwadhi and Scholl 2013) com cidadãos e agentes públicos questionando quais os principais serviços, aplicações e iniciativas que estão sendo desenvolvidas para a melhoria da qualidade de vida da população da cidade e quais os seus maiores benefícios. Entre os projetos citados estão o portal de dados abertos da cidade[9], a infraestrutura para apoiar o uso de carros elétricos e a instalação de um CRM (*Customer Relationship Management*) para controlar a comunicação de cidadãos com a prefeitura. A maioria dos benefícios apontados na adoção desses projetos são a melhoria dos serviços da cidade, a diminuição de custos, o aumento na eficiência e a economia de energia elétrica.

[8] http://www.fastcoexist.com/3021592/the-10-smartest-cities-in-north-america

[9] data.seattle.gov - https://data.seattle.gov/

Em Chicago foi desenvolvida a plataforma WindyGrid (Thornton 2013) que tem o objetivo de coletar, armazenar e processar os dados da cidade. Com isso, é possível visualizar as operações da cidade de forma unificada utilizando dados em tempo real e dados históricos. Alguns dos dados coletados da cidade são estatísticas e eventos sobre o trânsito da cidade, ligações de telefones de emergência (911), dados sobre edifícios públicos e publicações sobre a cidade em redes sociais como o Twitter. Na implementação da plataforma, foram utilizadas ferramentas de Big Data como o banco de dados NoSQL MongoDB e ferramentas de processamento paralelo de dados.

A plataforma disponibiliza três funções principais para os administradores da cidade: monitoramento de incidentes utilizando dados das ligações de emergência e de redes sociais; visualização de dados históricos, no qual o usuário pode ver todos os dados relacionados a um mesmo evento; e a análise de dados avançados em tempo real, na qual são mostrados, em um mapa da cidade, eventos que estão ocorrendo na cidade utilizando diversas fontes de dados. Essa ferramenta permite a monitoração em tempo real da cidade facilitando a reação das autoridades a eventos inesperados como crimes, acidentes e atentados terroristas.

Em Amsterdã, existem diversos projetos para aumentar a transparência dos gastos e ações dos administradores da cidade, entre eles o *Budget Monitoring* que permite que cidadãos e entidades acessem e façam sugestões para o orçamento da cidade, o *Smart City SDK*, que permite que desenvolvedores de aplicações utilizem os dados coletados na cidade em tempo real como dados de trânsito, chegadas e partidas de aviões e o clima e o *AmsterdamOpent* que é uma plataforma para que cidadãos façam sugestões para os governantes da cidade.

Diversas cidades ao redor do mundo já disponibilizam portais de dados abertos para permitir que a população possa acompanhar e fiscalizar o poder público. Como por exemplo Barcelona[10] disponibiliza uma grande quantidade de dados da administração pública como orçamento e despesas, dos serviços oferecidos pela cidade e dados sobre a população da cidade. Em Dublin, na Irlanda, com a plataforma de dados abertos Dublinked (Stephenson et al. 2012) que possibilita a cidadãos, empresas e pesquisadores o acesso a mais de 200 conjuntos de dados, entre eles dados em tempo real das posições dos ônibus, monitoramento da cidade e de estações de aluguel de bicicleta.

São Paulo possui diversos projetos relacionados a disponibilização de dados da cidade para os cidadãos, entre eles o portal de dados abertos da cidade[11], o GeoSampa[12], que disponibiliza diversas dados cartográficos da cidade como a localização dos equipamentos públicos, pontos de ônibus, árvores, feiras livres, pontos de alagamento, etc. e a API Olho Vivo[13], que permite a descoberta em tempo real da posição de todos os ônibus da cidade, informação esta que permitiu o desenvolvimento de vários aplicativos móveis oferecendo informações sobre o transporte público na cidade.

Outra ferramenta interessante para o compartilhamento dos dados da cidade com os cidadãos são os *Dashboards*. Essas ferramentas normalmente apresentam dados em

[10]OpenData BCN - http://opendata.bcn.cat/opendata/ca

[11]Dados Abertos São Paulo - http://saopauloaberta.prefeitura.sp.gov.br

[12]GeoSampa - http://geosampa.prefeitura.sp.gov.br

[13]API Olho Vivo - http://www.sptrans.com.br/desenvolvedores/APIOlhoVivo.aspx

tempo real em um mapa da cidade sobre condições climáticas, qualidade do ar, condições de trânsito e estados de equipamentos públicos. Um bom exemplo de cidade que disponiliza *dashboards* é Dublim[14], que disponibiliza diversos dados em tempo real como temperatura, qualidade do ar níveis de ruído e nível dos rios em diversas partes da cidade. A Figura 4.5 apresenta um exemplo de um *dashboard* que apresenta dados do trânsito da cidade.

Figura 4.5. *Dashboard* da cidade de Dublin.

Os dados utilizados em aplicações e serviços nesta seção são diversas informações sobre a cidade, podendo ser dados históricos ou em tempo-real. Exemplos de dados históricos, normalmente disponibilizados em portais de dados abertos, são os gastos da cidade, pesquisas de mobilidade e a localização de equipamentos públicos. Exemplos de dados em tempo-real, normalmente disponibilizados em *dashboards* ou em APIs, são as condições de trânsito, temperatura em diversos pontos da cidade e a localização dos ônibus.

4.4.4. Mobilidade Inteligente

Esta subseção apresenta projetos ligados à área de mobilidade, que tem como principais objetivos facilitar o fluxo das pessoas e monitorar o trânsito para aumentar a segurança das vias da cidade. Alguns serviços e aplicações comuns nessa área são o monitoramento do trânsito por câmeras de segurança, serviços que encontram as melhores rotas para os usuários tanto para transporte privado quanto para transporte público e o uso de indicações para facilitar a busca de vagas de estacionamento.

Utilizando a plataforma SmartSantander foi desenvolvido um projeto para mostrar para os motoristas lugares livres para estacionamento na cidade e também para prever a utilização desses lugares em eventos na cidade (Vlahogianni et al. 2014). Essa aplicação tem o objetivo de evitar que os motoristas fiquem rodando o centro da cidade procurando vagas, o que faz aumentar o trânsito na cidade e também aumentar as emissões de poluentes. A Figura 4.8 mostra algumas dessas vagas monitoradas no mapa da cidade.

[14]Dublin DashBoards - http://www.dublindashboard.ie

A cidade de Barcelona também está desenvolvendo projetos para incentivar o uso de formas sustentáveis de transporte, como um projeto para estimular o uso de carros elétricos, no qual mais de 300 pontos de recarga de carros foram instalados na cidade e o projeto para o uso de bicicletas compartilhadas que conta com mais de 420 estações para o empréstimo de bicicletas.

Amsterdã também está implantando diversas soluções na área de controle e monitoramento de trânsito, alguns projetos interessantes que estão sendo realizados na cidade são: o incentivo ao uso de carros elétricos, disponibilizando estações de recarga de bateria em várias partes da cidade, o monitoramento das principais vias da cidade para o rápido atendimento de problemas no trânsito, a reserva de vagas de estacionamento na cidade, evitando a busca por uma vaga, diminuindo a emissão de CO_2 e o incentivo ao uso de bicicletas.

Dois projetos dignos de nota foram desenvolvidos na cidade de Madrid, Espanha. O primeiro para encontrar a melhor rota de transporte público para um cidadão usando um smartphones (Foell et al. 2014) e informações contextuais do dispositivo como a localização. Um segundo utiliza celulares dos passageiros do ônibus para estimar a lotação dos coletivos (Handte et al. 2014). Essa informação pode ser visualizada por pessoas que estejam esperando o ônibus no ponto para decidir se esperam um próximo ônibus mais vazio.

Em São Paulo, foi implantado o Painel do Ônibus, desenvolvido pela startup Scipopulis[15] e utilizado pela Secretaria de Transportes de São Paulo, pela São Paulo Transportes (SPTrans) e pela Companhia de Engenharia de Tráfego (CET), que monitora os mais de 14.000 ônibus da cidade e mostra a velocidade dos ônibus em tempo real em todas as ruas. Integrando dados de diversas fontes (posições GPS dos ônibus, segregação do viário, acidentes, entre outros) a informação é contextualizada em relação à hora do dia, ao tipo de via (corredor, faixa ou viário compartilhado), aos acidentes na região e à quantidade de ônibus passando por aquela via. É possível monitorar linhas de ônibus completas ou trechos de uma linha, e visualizar o histórico de velocidades de cada trecho. O painel é utilizado pelas equipes de operação, planejamento e gestão da rede de transportes para identificar gargalos crônicos, problemas eventuais, identificar vias onde implementar faixas exclusivas e corredores, os horários em que elas devem funcionar, o efeito das mudanças no viário e no planejamento das linhas na velocidade dos ônibus, suportar a escolha de projetos de adequação do viário de acordo com o seu impacto na velocidade dos ônibus e na quantidade de pessoas beneficiadas, entre outros processos importantes na gestão de transportes. A Figura 4.6 apresenta uma tela do Painel do Ônibus sendo executado.

Os dados utilizados nas aplicações e serviços apresentados nesta subseção são informações sobre vagas de estacionamento livres coletadas por sensores, fluxos de vídeo que monitoram vias da cidade, dados de dispositivos de cidadãos sobre seus veículos ou sobre o transporte público e dados sobre o uso de meios alternativos de transporte como bicicletas.

[15] Scipopulis - http://www.scipopulis.com/

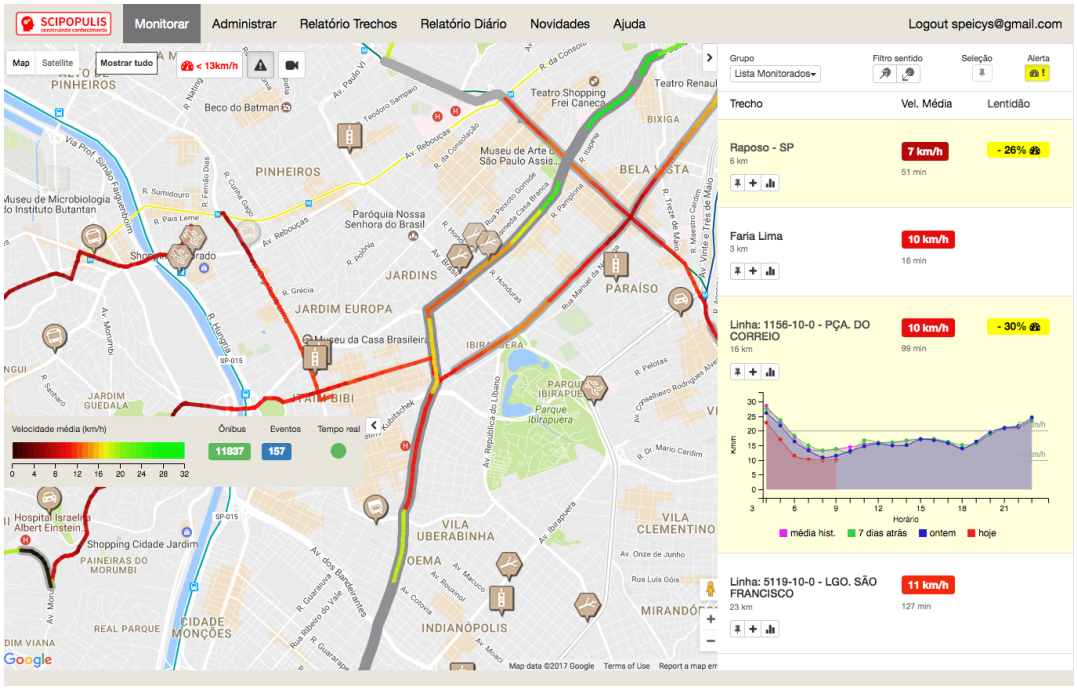

Figura 4.6. Painel do Ônibus da cidade de São Paulo.

4.4.5. Meio Ambiente Inteligente

Esta subseção apresenta projetos ligados à área de meio ambiente, que tem o objetivo de tornar a cidade mais sustentável, melhorando serviços como a coleta e a reciclagem de lixo, a distribuição eficiente de recursos como água e eletricidade e a diminuição da poluição na cidade.

Masdar é um bairro na cidade de Abu Dhabi, nos Emirados Árabes Unidos que está sendo construído com o objetivo de testar diversas iniciativas de Cidades Inteligentes, principalmente no uso de fontes energéticas renováveis, no uso consciente de água e na redução da quantidade de lixo gerado. Além disso, a cidade foi planejada com uma rede de transporte inteligente para reduzir a necessidade do uso de veículos individuais, diminuindo a emissão de poluentes. Nesse bairro, todas as construções são projetadas de forma que economizem os recursos e produzam a própria energia com o uso de painéis solares.

Em Manchester, está sendo desenvolvido um projeto de construção de casas inteligente, na qual os moradores podem verificar em tempo real o uso de recursos utilizados como energia elétrica e água. O objetivo desse projeto é diminuir a quantidade das emissões de carbono na cidade e na economia de recursos naturais como água e energia elétrica (Manville et al. 2014).

Santander implementou, em 2014, um projeto para gerenciar a coleta de lixo na cidade (Díaz-Díaz et al. 2017). Esse projeto utiliza dados de mais de 1000 sensores que monitoram o quão cheias estão as lixeiras da cidade. Além disso, os caminhões de lixo e os depósitos de lixo também são monitorados. Esse sistema permite que os caminhões de lixo visitem apenas lixeiras que estão cheias, diminuindo os quilômetros rodados. Três

grandes benefícios desse serviço são a diminuição dos gastos com os caminhões, a diminuição de emissões de poluentes pelos veículos e o melhor gerenciamento do lixo recolhido na cidade. Barcelona também possui um projeto parecido, no qual é possível monitorar o estado atual das lixeiras na cidade. A Figura 4.7 apresenta o mapa da cidade com os sensores nas lixeiras.

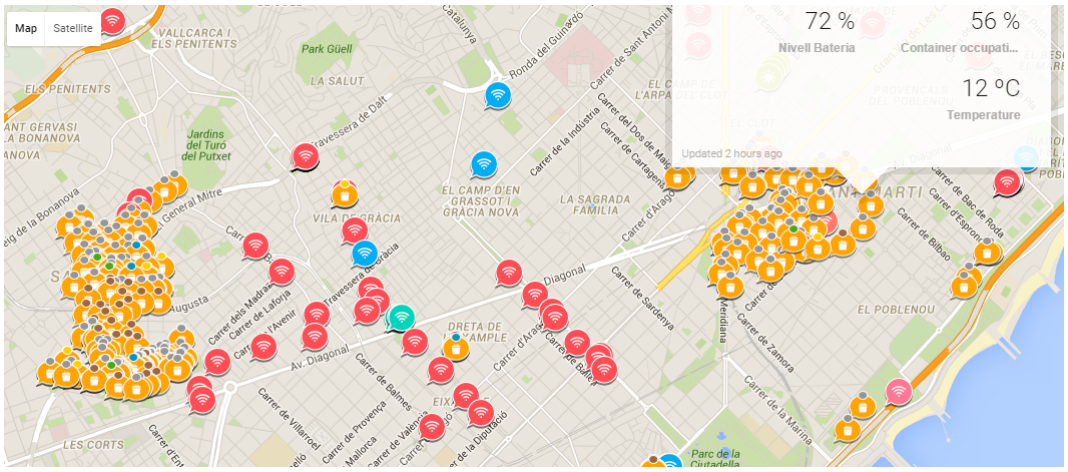

Figura 4.7. Sensores indicando porcentagem de uso nas lixeiras em Barcelona

Também em Santander, está sendo desenvolvido um projeto de iluminação inteligente, no qual são planejados a instalação de mais de 23.000 lâmpadas de LED, um sistema de detecção de movimento para ligar ou desligar as luzes e uma infraestrutura para o monitoramento do funcionamento das luzes na cidade. Com esse sistema a cidade pretende diminuir em 80% o consumo de energia no sistema de iluminação da cidade. Deverá haver economia também na manutenção das lampadas, que hoje é feita através de rondas para encontrar lampadas que estejam com problema.

Em Barcelona já está em operação, em um distrito da cidade, um sistema para aquecimento e resfriamento de edifícios (March and Ribera-Fumaz 2016). O sistema funciona com uma rede de água que passa por diversos edifícios do distrito, principalmente órgãos públicos, e que utilizando a energia gerada pelos incineradores de lixo da cidade esquentam ou resfriam a água na tubulação. Foi estimado que esse sistema utiliza 35% menos energia e diminui em 50% as emissões de poluentes que os sistemas convencionais.

Os dados utilizados nas aplicações e serviços apresentados nesta subseção são informações sobre o uso de recursos em edifícios como água e energia elétrica, a capacidade de uma lixeira para evitar que caminhões de lixo tenham que passar por todas as lixeiras da cidade e a detecção de movimento dos cidadãos para acender ou apagar lâmpadas.

4.4.6. Vida Inteligente

Esta subseção apresenta projetos ligados à melhoria da qualidade de vida do cidadão, que tem o objetivo de melhorar serviços que se relacionam diretamente à rotina dos cidadãos como segurança, atividades culturais e atividades esportivas. Alguns serviços relacionados a essa dimensão são aplicações que avisam sobre eventos que está acontecendo na

cidade, para o monitoramento de lugares muito movimentados e para relatos de problemas em lugares públicos tais como parques e ruas.

Em Santander, foi implantada uma rede de mais de 20 mil sensores e atuadores na cidade que coletam uma grande quantidade de dados em diversas regiões da cidade, como temperatura, espaços livres de estacionamento, identificadores de pontos de interesse e luminosidade para o desenvolvimento de serviços que melhoram a vida dos cidadãos. A Figura 4.8 mostra um mapa no qual cada ponto é um elemento da cidade que envia dados para a plataforma.

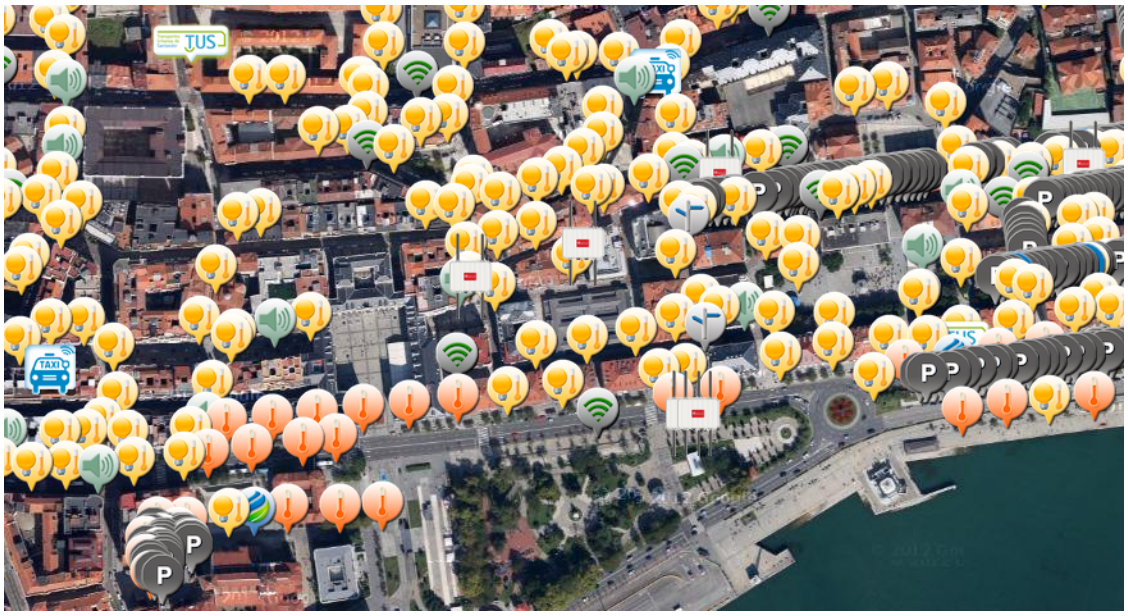

Figura 4.8. Mapa dos elementos da plataforma Smart Santander

Entre os projetos desenvolvidos está uma aplicação que informa os cidadão sobre as atividades programadas na cidade como shows, exposições, cursos e atividades físicas. Utilizando os dados recolhidos na plataforma são também informadas as condições da região onde a atividade será realizada como o nível de ruído e iluminação e a quantidade de vagas de estacionamento disponíveis. Outra aplicação desenvolvida utilizando a plataforma SmartSantander é a que permite que cidadãos reportem problemas encontrados nos parques da cidade facilitando a manutenção das áreas verdes da cidade.

Em Dublin, utilizando o mesmo *dashboard* apresentado na Seção 4.4.3, são disponibilizados vídeos em tempo real de diversos pontos da cidade, o que permite a monitoração de diversos problemas que ocorrem na cidade como crimes e acidentes de trânsito.

Os dados utilizados nas aplicações e serviços apresentados nesta subseção são informações sobre atividades que ocorrerão na cidade e fluxos de vídeos capturados em tempo-real na cidade.

4.5. Plataformas de Cidades Inteligentes

Nesta seção, apresentaremos três projetos de pesquisa para o desenvolvimento de plataformas de software para cidades inteligentes que utilizam as tecnologias descritas na seção

4.4. A partir dessas plataformas, serão levantados os principais requisitos funcionais e não funcionais necessários na implementação de uma plataforma para Cidades Inteligentes.

4.5.1. OpenIoT

O OpenIoT é uma plataforma para suportar a criação de aplicações baseadas na Internet das Coisas, essa plataforma é utilizada no projeto Vital (Petrolo et al. 2014) para a implantação de um ambiente de Cidades Inteligentes em diversas cidades da Europa como Londres, Turim e Madrid. A Figura 4.9 apresenta uma visão geral da arquitetura dessa plataforma, a qual possui três planos: o **Plano Físico**, o *Plano Virtualizado* e o *Plano de Utilidades e Aplicações*.

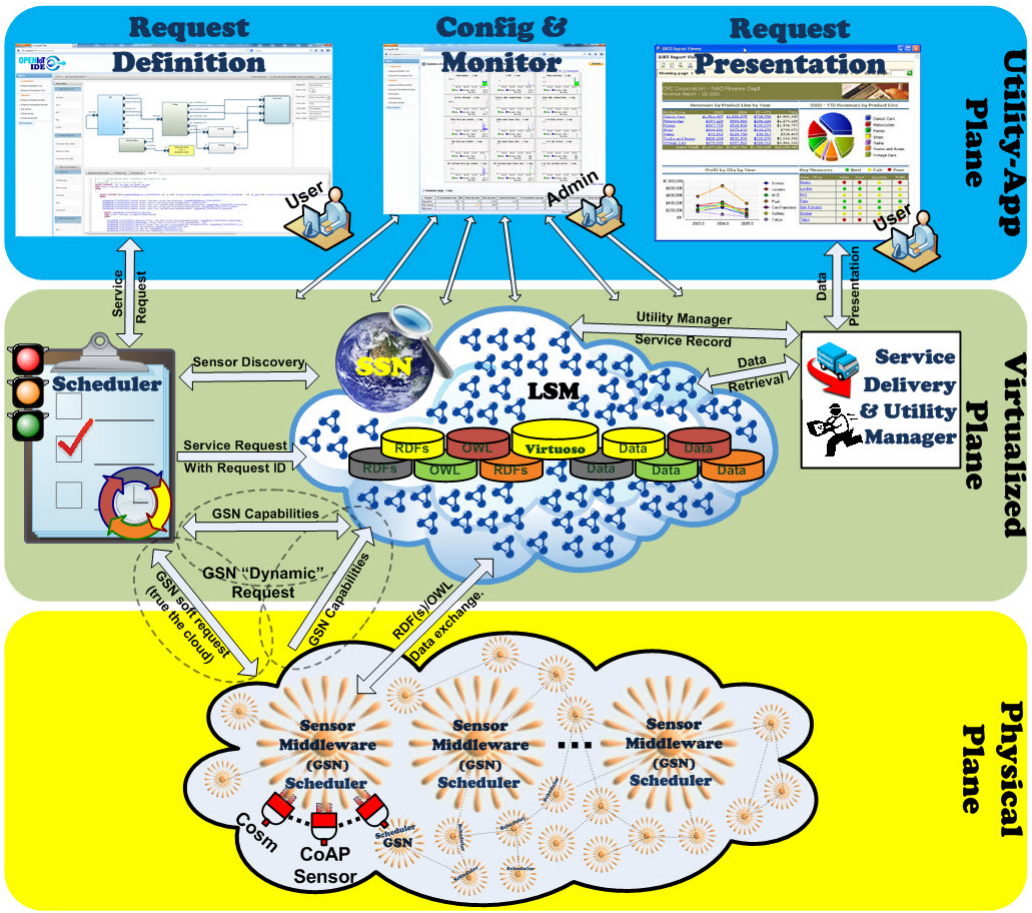

Figura 4.9. Plataforma OpenIoT (Petrolo et al. 2014)

O Plano Físico é um middleware responsável por coletar, filtrar, agregar e limpar os dados de sensores, atuadores e diversos outros tipos de dispositivos. Ele age como uma interface entre o mundo físico e a plataforma OpenIoT. A versão atual do OpenIoT usa o X-GSN (Calbimonte et al. 2014), um middleware de código aberto para gerenciar, controlar e monitorar dispositivos IoT.

O Plano Virtualizado tem o objetivo de armazenar os dados, executar serviços e agendar a execução desses serviços. Os principais componentes dessa camada são os seguintes:

- O **Escalonador (Scheduler)** que recebe requisições por serviços e garante o acesso aos recursos que esses serviços necessitam como fluxos de dados e tempo de processamento. É também responsável por identificar os sensores necessários para a execução dos serviços.

- O **Armazenamento de Dados na Nuvem (Cloud Data Storage)** armazena todos os dados da plataforma, i.e., os dados coletados da rede de sensores da cidades, de configuração da plataforma e das aplicações que são executadas na plataforma. Para o armazenamento e processamento dos dados capturados na cidade é utilizado o *Linked Sensor Middleware*, que possibilita o armazenamento e processamento de dados utilizando ontologias com o formato padrão RDF no banco de dados Virtuoso[16] (Le-Phuoc et al. 2012).

- O **Gerenciador de Serviços e Utilidades (Service Delivery and Utility Manager)** tem quatro funções principais: possibilitar a definição de serviços sobre a plataforma, executar os serviços requisitados por usuários e aplicações, permitir a definição de parâmetros de configuração da plataforma e fazer o monitoramento de toda a infraestrutura que é executada na plataforma. Adicionalmente, esse componente mantém um histórico de todos os serviços utilizados na plataforma para permitir a cobrança pelo seu uso.

A Camada de Utilidades e Aplicações é a interface da plataforma com o usuário e possui os três componentes seguintes:

- A **Definição de Requisições (Request Definition)** permite que usuários definam novas aplicações usando os serviços e os dados que estão disponíveis na plataforma, permitindo inclusive a composição de serviços.

- A **Apresentação de Requisições (Request Presentation)** executa uma aplicação criada no componente Definição de Requisições. Quando uma aplicação é executada, ela se comunica com os componentes Gerenciador de Serviços e Utilidades para recuperar os resultados da execução dos serviços na plataforma.

- A **Configuração e Monitoramento (Configuration and Monitoring)** permite a configuração dos parâmetros da plataforma como, por exemplo, intervalos para a leitura de sensores, prioridade de serviços e aplicações e permissões de usuários e o monitoramento da execução de todos os componentes e dispositivos da plataforma.

Utilizando a plataforma OpenIoT, pesquisadores desenvolveram um sistema para o gerenciamento de lixo de uma cidade (Anagnostopoulos et al. 2015). Nessa aplicação, existem sensores dentro das cestas de lixo indicando se ela está cheia ou vazia. As cestas podem ser priorizadas para o recolhimento do lixo dependendo de sua localização, por exemplo, cestas próximas a escolas ou hospitais. Além disso, o sistema calcula a quantidade de lixo coletado e a quantidade de quilômetros rodados pelos caminhões de lixo para possibilitar a análise dos dados.

[16] Virtuoso - https://github.com/openlink/virtuoso-opensource

O OpenIoT é uma plataforma bastante completa atendendo à maioria dos requisitos necessários para a criação de uma cidade inteligente. Os pontos fortes dessa plataforma são o middleware para o armazenamento dos dados coletados na cidade, suas ferramentas para a definição dos serviços e o fato da plataforma ser de software livre. Entretanto, a plataforma não oferece coleta de dados de outras fontes importantes como redes sociais e não oferece suporte para o pré-processamento dos dados, o que é bastante relevante quando a quantidade de dados é muito grande.

4.5.2. SmartSantander

SmartSantander é uma plataforma experimental para o desenvolvimento de aplicações e serviços para Cidades Inteligentes. Ela é composta por um grande número de dispositivos IoT implantados em diversos cenários urbanos que coletam diferentes tipos de dados, uma rede de computadores, chamados *Gateways* que gerenciam e monitoram esses dispositivos e Servidores, que armazenam e processam os dados coletados na cidade. A arquitetura da plataforma SmartSantander é formada pelas três camadas seguintes:

- **IoT Nodes** são os nós físicos implantados na cidade. A maioria desses dispositivos são de baixa capacidade de processamento e na maioria dos casos apenas coletam algum dado da cidade. Por estarem implantados no ambiente urbano, esses dispositivos estão sujeitos a falhas e vandalismo, por isso são necessárias a monitoração e manutenção dos dispositivos.

- **IoT Gateways** responsáveis por conectar os dispositivos espalhados pela cidade aos servidores nos quais os dados serão armazenados e processados. Essa camada também é responsável pela monitoração dos dispositivos que estão conectados a cada *gateway*. Eles são responsáveis pela gestão desses dispositivos e podem reconfigurar os dispositivos automaticamente e em tempo de execução.

- **Servers** nos quais os dados serão armazenados e processados. Essa camada é composta por hardware de alto poder computacional para garantir a escalabilidade e elasticidade da plataforma. Esses servidores podem servir como repositórios de dados, servidores de aplicações e serviços para a mineração e processamento dos dados.

A Figura 4.10 mostra um exemplo da arquitetura do SmartSantander sendo utilizada por uma aplicação para o sensoriamento participativo na cidade. Na camada do servidor existem componentes para fazer o diretório dos dispositivos que existem na cidade, um subsistema de apoio a aplicações que é responsável pelo armazenamento e acesso aos dados e o componente especifico da aplicação.

A Figura 4.10 mostra também que podem existir diversos *gateways* que se conectam a um conjunto de nós espalhados pela cidade. Os *gateways* recebem os dados dos dispositivos e os enviam para o servidor e também monitoram os dispositivos.

A plataforma SmartSantander tem como pontos fortes (1) a coleta de dados de uma grande rede de sensores da cidade de Santander, que mostra que é possível suportar o grande fluxo de dados em uma plataforma de Cidade Inteligentes, (2) a monitoração

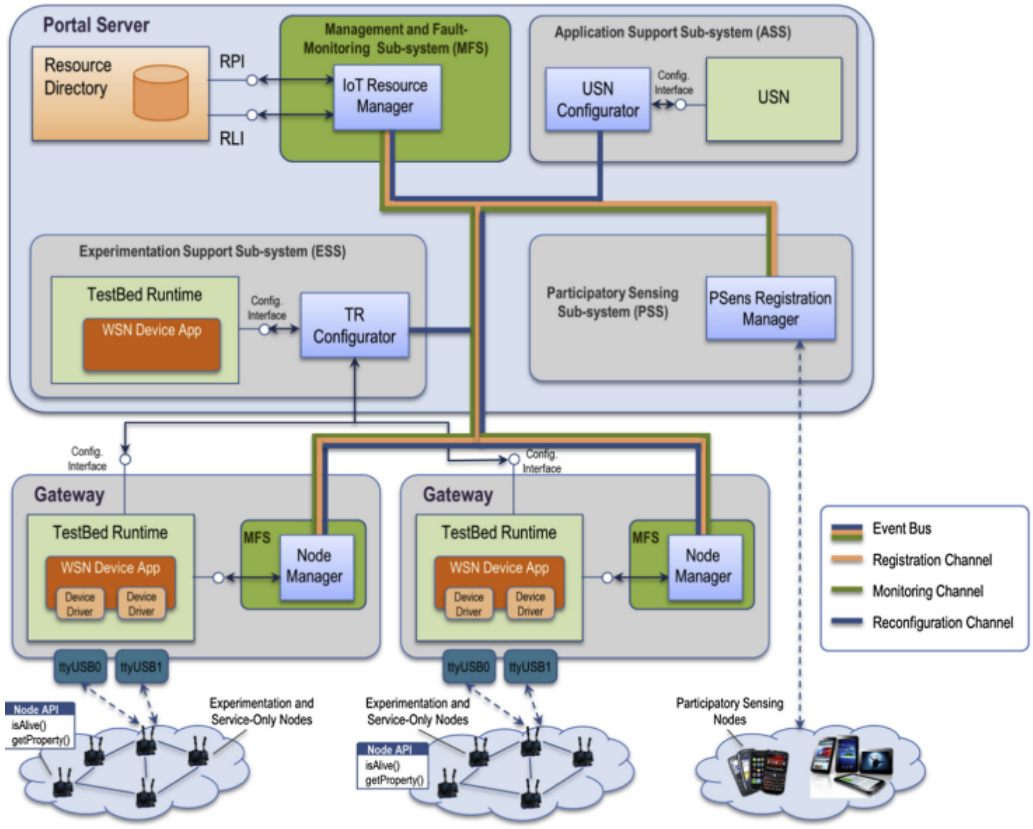

Figura 4.10. Architetura da Plataforma SmartSantander (Sanchez et al. 2014)

dos dispositivos espalhados pela cidade, que possibilita a fácil solução de problemas como dispositivos que perdem comunicação com a plataforma ou quebrados e (3) a possibilidade do uso da plataforma para o desenvolvimento de várias aplicações para a cidade.

Usando a infraestrutura do SmartSantander foram desenvolvidos diversos outros projetos, como por exemplo o SEN2SOC (Vakali et al. 2014) que captura fluxos de dados dos sensores da cidade e de redes sociais dos cidadãos para criar aplicações. Dois exemplos de aplicações são a reação dos cidadãos a algum evento na cidade e a construção de mapas de calor com os dados sobre a poluição do ar na cidade. Outro projeto é o CiDAP (Cheng et al. 2015) que será apresentado a seguir.

4.5.3. CiDAP

A plataforma CiDAP (*City Data and Analytics Platform*) utiliza ferramentas de Big Data com o objetivo de processar o grande volume de dados coletados da cidade para adicionar inteligência e contexto nas aplicações e serviços desenvolvidos para a cidade. Os dados processados pela plataforma são coletados por um Middleware IoT independente. A plataforma foi testada utilizando os dados do SmartSantander (Cheng et al. 2015). A Figura 4.11 apresenta os cinco principais componentes da sua arquitetura:

- Os **IoT-Agents** se conectam com o middleware IoT como um *gateway* para coletar os dados dos dispositivos e armazenar na plataforma. Cada fonte de dados disponí-

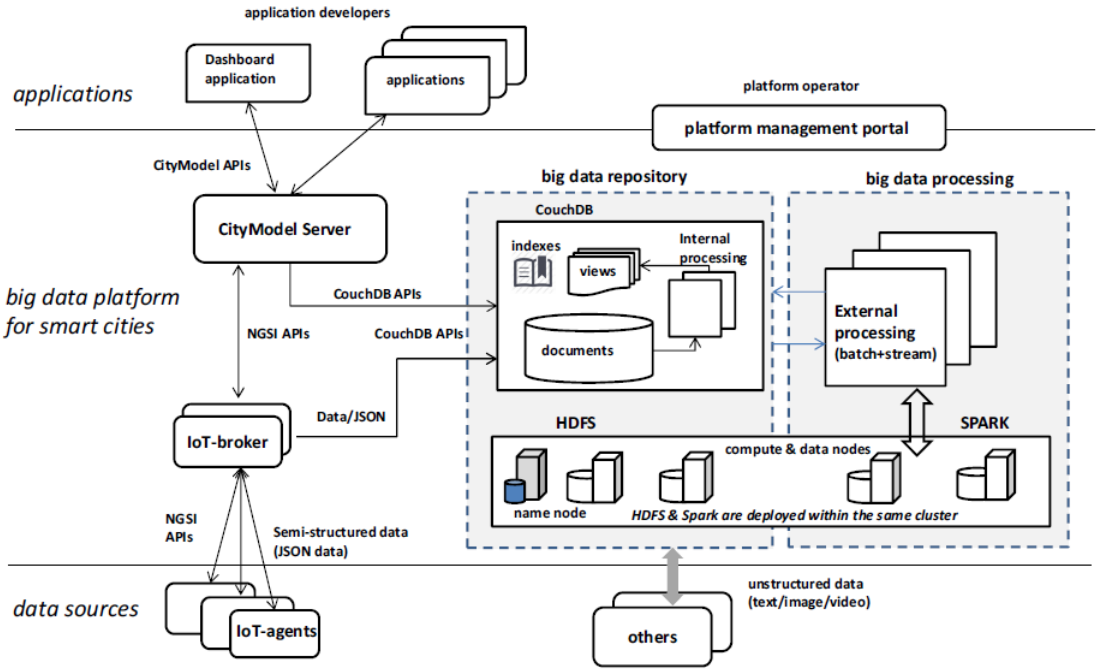

Figura 4.11. Plataforma CiDAP (Cheng et al. 2015)

vel no middleware IoT é mapeada em um IoT Agent.

- Os **IoT-Brokers** agem como uma interface unificada para os *IoT Agents*, facilitando o acesso aos dados coletados pelo middleware. Esse componente se comunica com o Repositório de Big Data para enviar os dados que serão armazenados ou com o CityModel para serem utilizados diretamente nas aplicações.

- O **Big Data Repository** armazena os dados coletados da cidade e também os dados processados utilizando o componente de processamento de Big Data. A plataforma utiliza o banco de dados NoSQL CouchDB [17], o qual armazena os dados em documentos no formato JSON. Esse componente possui também uma ferramenta de processamento interno para fazer processamentos simples e rápidos nos dados como a transformação dos dados em novos formatos ou a criação de novas tabelas e visões.

- O **Big Data Processing** é responsável por processamentos complexos ou demorados como a agregação dos dados ou algoritmos de aprendizado de máquina usando os dados armazenados no repositório de Big Data. Além disso, esse componente também processa dados históricos utilizando processamento em lote ou processa os dados em tempo real utilizando fluxos de dados. Esse componente é implementado utilizando a ferramenta Apache Spark (Zaharia et al. 2010).

- O **City Model Server** é a interface da plataforma para as aplicações externas. Ele é implementado como uma API com uma interface para os principais dados da cidade, como dados de sensores e resultados de análise dos dados. Esse componente

[17] http://couchdb.apache.org

possibilita dois tipos de consultas aos dados da plataforma, uma utilizando uma API REST na qual é possível efetuar consultas aos últimos valores coletados por um sensor ou outro dispositivo ou a resultados de algum processamento. A outra forma de consulta é a assinatura a uma fonte de dados, assim, a aplicação recebe periodicamente dados dessa fonte através de um mecanismo de publicação/assinatura.

A plataforma CiDAP tem como objetivo armazenar e processar um grande volume de dados urbanos. Isso é importante porque a quantidade de dados coletados em uma cidade inteligente será muito grande. Os pontos fortes dessa arquitetura são a preocupação com a escalabilidade e elasticidade das estruturas de armazenamento e processamento. Além de disponibilizar ferramentas para o processamento de dados históricos e em tempo real, outro fator interessante é que a plataforma já foi efetivamente testada utilizando os dados reais do SmartSantander.

4.6. Requisitos de uma Plataforma de Cidades Inteligentes

A partir da análise das iniciativas introduzidas na Seção 4.4 e das plataformas descritas na Seção 4.5, apresentamos agora os principais requisitos que devem ser considerados na implementação de uma plataforma de Cidades Inteligentes.

4.6.1. Requisitos Funcionais

O principal objetivo de uma plataforma para cidades inteligente é facilitar o desenvolvimento de aplicativos para a cidade. Assim, a maioria das plataformas apresentadas oferecem funcionalidades para a coleta, armazenamento e compartilhamento dos dados urbanos e para o desenvolvimento e execução de serviços e aplicações para a cidade. Os principais requisitos funcionais para plataformas de software para cidades inteligentes são os seguintes.

- **Gerenciamento de Dados:** Uma Cidade Inteligente manipula uma quantidade enorme de dados, por isso, é necessário que as plataformas implementem diversas atividades relacionadas ao ciclo de vida dos dados da cidade, como a **coleta**, o **armazenamento**, a **análise** e a **visualização** dos dados. Diversas técnicas e ferramentas podem ser usadas para suprir esse requisito, como por exemplo bancos de dados NoSQL para dados não-estruturados ou semi-estruturados, bancos de dados relacionais para dados estruturados, ferramentas de *Big Data* para a análise e processamento dos dados e geradores de relatórios e imagens para a visualização dos dados (Hernández-Muñoz et al. 2011; Cheng et al. 2015).

- **Ambiente para Execução de Aplicações:** Algumas plataformas oferecem suporte para a **execução** de aplicações da cidade facilitando a **implantação** e a **integração** entre essas aplicações. Algumas plataformas oferecem um ambiente para a implantação de serviços e aplicações (Apolinarski et al. 2014); outras oferecem um serviço para a execução de aplicações desenvolvidas com ferramentas da própria plataforma (Petrolo et al. 2014; Wu et al. 2014).

- **Gerência da Rede de Sensores:** Uma das principais características de Cidades Inteligentes é a necessidade de gerenciar uma grande rede de dispositivos instalados

na cidade como sensores que coletam dados do ambiente, sensores que verificam o funcionamento do mobiliário urbano e sensores que monitoram o trânsito. Essa rede pode ser complexa e grande, como por exemplo a rede do projeto SmartSantander que conta com mais de 20 mil sensores em uma cidade pequena; numa cidade grande como São Paulo ou Rio de Janeiro, idealmente deveríamos ter de centenas de milhares a milhões de sensores. Algumas das atividades necessárias nesse requisito são a **adição**, **remoção**, **monitoramento** e **coleta de dados dos sensores**. Além de Santander, outras cidades que já possuem uma rede de sensores razoavelmente explorada são Barcelona, Padua, Chicago, Dublin e Amsterdã.

- **Processamento de Dados:** O processamento dos dados coletados na cidade é essencial para a criação de diversos serviços e aplicações para Cidades Inteligentes como para o entendimento de algum fenômeno que ocorre na cidade, para pesquisar a melhor opção de transporte para o usuário e para identificar áreas de risco. Diversas ferramentas são utilizadas para o processamento de dados em Cidades Inteligentes, como máquinas de inferência (normalmente associadas ao uso de ontologias), processadores de *workflows* (para executar aplicações nas quais os dados passam por diversos estágios de processamento, como no atendimento de saúde de um paciente, ou em uma solicitação à prefeitura) e ferramentas de Big Data para o processamento de grandes quantidades de dados. Esses componentes processam grandes conjuntos de dados com o propósito de **analisar**, **verificar**, **agregar** e **filtrar** os dados coletados da cidade (Girtelschmid et al. 2013; Cheng et al. 2015).

- **Acesso aos Dados:** Para possibilitar o desenvolvimento de aplicações para a cidade, é essencial que os dados coletados e processados possam ser acessados por aplicações e serviços externos à infraestrutura da cidade. Diversas iniciativas já disponibilizam seus dados através de **portais de dados abertos**, mas é fundamental também, que os dados nesses portais sejam disponibilizados em **formatos padronizados** e com **meta-dados descritivos** associados. Para isso, várias cidades utilizam **APIs bem definidas** que facilitam o desenvolvimento de aplicações automatizadas. Outra forma interessante de disponibilizar os dados, é através de **serviços de publicação/assinatura** (*publish/subscribe*), por meio dos quais clientes que manifestam interesse em tópicos específicos recebem dados sempre que uma fonte gerar novos dados de interesse para cada cliente.

- **Gerenciamento de Serviços:** A maioria das plataformas e iniciativas de Cidades Inteligentes adota a **arquitetura orientada a serviços** para oferecer as funcionalidades da plataforma em um ambiente de Computação em Nuvem. Alguns dos serviços oferecidos são: acesso aos dados coletados pelos sensores e dados processados, componentes para o processamento de dados como máquinas de inferência e algoritmos de aprendizado de máquina, componentes para a execução de *workflows* e serviços de gerencia de usuários da plataforma. Algumas plataformas possibilitam ainda que desenvolvedores implantem serviços na plataforma e os disponibilizem para outras aplicações da cidade. É importante também possibilitar operações sobre serviços, como composição e **coreografias** (Issarny et al. 2011) para a criação de novos serviços e aplicações (Lee and Rho 2010; Piro et al. 2014) de forma **automatizada e escalável** (Leite et al. 2014).

- **Ferramentas para o Desenvolvimento de Software:** Como um dos principais objetivos de uma Cidade Inteligente é o fornecimento de aplicações e serviços que facilitem a vida do cidadão, é necessário que as plataformas facilitem o desenvolvimento desses sistemas. Para isso, devem ser fornecidas ferramentas que facilitem a utilização dos serviços fundamentais e componentes básicos da plataforma. Alguns exemplos de ferramentas utilizadas são: interfaces visuais para a descrição de aplicações utilizando as fontes de dados e serviços disponíveis na plataforma, ferramentas para a descrição de *workflows*, a utilização de ferramentas para a geração de relatórios e análise de dados e até o desenvolvimento de um kit para o desenvolvimento de aplicações (*Software Development Kit* ou SDK) com diversas ferramentas integradas (Elmangoush et al. 2013; Apolinarski et al. 2014).

- **Definição de um Modelo da Cidade:** Para melhorar o entendimento do funcionamento de diversos serviços da cidade e permitir a análise e o processamento automático desse funcionamento, é fundamental o desenvolvimento de modelos da cidade. Tais **modelos podem representar aspectos estáticos da cidade**, como o mapa da cidade com a localização das ruas e equipamentos públicos, ou **aspectos dinâmicos da cidade**, como o fluxo de veículos, as zonas de congestionamento em diferentes horas do dia e dias da semana ou a variação na utilização dos serviços de saúde da cidade ao longo do progresso de epidemias de diferentes doenças infecciosas. Além de melhorar o entendimento da cidade, os modelos também facilitam a análise automática dos dados através de algoritmos de aprendizado de máquina. Algumas plataformas utilizam esses modelos para permitir as consultas aos dados da cidade em uma linguagem de consulta própria e outras utilizam os modelos para facilitar a definição das aplicações e serviços da cidade, utilizando linguagens de descrição de processos como linguagens de definição de *workflows* e BPMN (*Business Process Modeling Notation*) (Cheng et al. 2015; Privat et al. 2014).

Baseado nos requisitos funcionais descritos acima, é possível observar que as principais atividades das plataformas são controlar o ciclo de vida dos dados da cidade:

1. coletar os dados com a rede de sensores e atuadores,

2. gerenciar os dados na plataforma,

3. processar os dados da cidade utilizando seu modelo de dados e

4. compartilhar os dados coletados e processados permitindo acesso externo a esses dados.

Essas atividades são bastante relacionadas com as tecnologias usadas para a implementação de cidades inteligentes como IoT para a implementação da rede de sensores, o gerenciamento e processamento de dados com Big Data e o gerenciamento de serviços com Computação em Nuvem.

4.6.2. Requisitos não-funcionais

A maioria dos requisitos não-funcionais de uma cidade inteligente são relacionados ao fato de que estamos diante de enormes e heterogêneos sistemas distribuídos. Isso exige bons níveis de escalabilidade, adaptabilidade e interoperabilidade. Outros requisitos são relacionados a manipulação de dados críticos dos cidadãos e da cidade como privacidade e segurança. Os principais requisitos não-funcionaise plataformas de software para cidades inteligentes são os seguintes.

- **Interoperabilidade:** Diferentes dispositivos, sistemas, aplicações e plataformas compõem um ambiente de uma cidade inteligente e todos esses componentes devem operar de uma maneira integrada. Por exemplo, sensores de múltiplos fabricantes, sistemas implementados em diferentes linguagens de programação e em diferentes sistemas operacionais, plataformas que compartilham dados e usuários e sistemas legados que devem se comunicar com as novas aplicações da cidade, tudo isso deve trabalhar em conjunto de forma harmoniosa. Existem diversas técnicas adotadas para alcançar esses requisitos: o uso de interfaces genéricas e padrões da indústria, a aplicação de web semântica para integração dos componentes da plataforma e o uso de serviços de nomes e de descoberta de recursos baseados em ontologias para identificar diferentes dispositivos e fontes de dados(Villanueva et al. 2013; Gurgen et al. 2013).

- **Escalabilidade:** A quantidade de usuários, dados, aplicações e serviços em uma Cidade Inteligente será muito grande e tende a aumentar ao longo do tempo, com a integração de mais serviços e o aumento da população. Por exemplo, na cidade de Santander, uma cidade média, já existem mais de 20.000 sensores instalados coletando uma grande quantidade de dados da cidade; a plataforma CiDAP que coletou mais de 50 GBs de dados em três meses. Esse requisito não funcional é importante para o funcionamento de diversos requisitos funcionais como o gerenciamento da rede de sensores e atuadores, o gerenciamento de dados e o gerenciamento de serviços (Bain 2014; Takahashi et al. 2012). Basicamente, o que devemos buscar é que a qualidade do serviço oferecido pela plataforma não seja prejudicada à medida em que a escala do sistema aumenta e que os acordos de nível de serviço (*SLA*) sejam respeitados.

- **Elasticidade:** A quantidade de usuários que acessam os serviços de uma cidade inteligente pode variar bastante durante o dia ou, por exemplo, durante os horários de pico. Os serviços relacionados ao trânsito da cidade terão muito mais requisições na hora do *rush* do que de madrugada, ou durante um grande evento cultural, os serviços relacionados ao atendimento a turistas terão uma grande demanda. Por isso, é importante que a infraestrutura da cidade seja redimensionada dinamicamente de acordo com a necessidade para evitar a falta de recursos quando eles forem necessários e evitar o desperdício quando não mais forem utilizados. A maioria das iniciativas que disponibilizam seus serviços em um ambiente de Computação em Nuvem (Wu et al. 2014; Khan et al. 2015; Petrolo et al. 2014) citam a necessidade desse requisito, uma vez que as tecnologias da Nuvem surgiram tendo elasticidade como um dos objetivos principais.

- **Segurança:** Usuários maliciosos podem fazer uso fraudulento dos serviços e dados disponibilizados pela plataforma. É fundamental que as plataformas implementem mecanismos específicos de segurança que ofereçam controle de acesso, criptografia, autenticação e a proteção dos dados da plataforma, da infraestrutura e das aplicações da cidade (Piro et al. 2014; Hernández-Muñoz et al. 2011; Petrolo et al. 2014).

- **Privacidade:** Um ambiente de cidade inteligente coleta e manipula diversos dados sensíveis a usuários, governos, empresas e ONGs da cidade como a localização de pessoas e suas ações, dados governamentais, registros médicos e dados sigilosos de negócios. É um desafio usar todos esses dados escondendo ou trocando informações para impossibilitar a identificação de quem está transmitindo a informação (Cardoso and Issarny 2007). Algumas das estratégias utilizadas para alcançar esse objetivo são o uso de criptografia, dispositivos para controlar o acesso aos dados da plataforma como certificados digitais e biometria bem como a anonimização dos dados (Apolinarski et al. 2014; Mylonas et al. 2015).

- **Sensibilidade ao Contexto:** Como a situação da cidade e dos cidadãos muda constantemente, muitas aplicações de cidades inteligentes podem apresentar melhores resultados usando informações contextuais. Alguns exemplos de informações sobre usuários que podem ser usadas são a localização, a atividade sendo realizada e a linguagem do dispositivo de acesso. Exemplos das informações sobre a cidade utilizadas são condições de trafego, clima e qualidade do ar (Khan et al. 2013; Cheng et al. 2015). Alguns exemplos de aplicações que usam sensibilidade ao contexto são: mostrar a aplicação em uma língua diferente para turistas, mudar a rota de um motorista para evitar áreas congestionadas ou poluídas e definir a recomendação do uso de modo de transporte dependendo da previsão do tempo.

- **Adaptabilidade:** Ainda relacionado à sensibilidade ao contexto, muitas plataformas adaptam o seu comportamento de diferentes formas, em diferentes dimensões, baseado no contexto dos usuários ou da cidade. Alguns dos objetivos desse requisito são aumentar a tolerância a falhas, utilizar um servidor mais próximo a um usuário para atender sua requisição, decidir se um processamento será em tempo real ou em lote e adaptar dados de diferentes fontes para uma representação comum. Adaptação de dados, por exemplo, é bastante utilizada em plataformas que utilizam os conceitos de Internet das Coisas para adaptar o funcionamento das rede de sensores da cidade (Girtelschmid et al. 2013; Privat et al. 2014).

- **Extensibilidade:** A capacidade de adicionar serviços, componentes e aplicações à plataforma é importante para possibilitar a evolução da plataforma para atender a novos requisitos e funcionalidade que surgem ao longo do tempo. Para tanto é fundamental a utilização de boas práticas de Programação Orientada a Objetos e de Arquitetura de Software tais como os princípios SOLID e uma boa metodologia ágil de desenvolvimento baseada fortemente em testes automatizados (Fox et al. 2015).

- **Configurabilidade:** Uma plataforma de cidade inteligente possui uma grande quantidade de configurações e parâmetros para adaptar o seu funcionamento a diferen-

tes contextos em tempo de execução, por exemplo definindo limiares de poluição e congestionamento e a prioridade de um serviço. Assim, é importante permitir a (re)configuração de diversas variáveis da plataforma. Pode-se utilizar um portal para centralizar as configurações da plataforma, porém, devido ao grande número de configurações necessárias, é desejável que a própria plataforma conheça o seu contexto de execução e consiga alterar dinamicamente suas configurações sem a necessidade da intervenção de um operador humano (Wan et al. 2012; Privat et al. 2014).

Observando os requisitos não-funcionais descritos nesta seção, é possível observar que eles são importantes para diversos requisitos funcionais. Por exemplo, Escalabilidade é importante para o gerenciamento da rede de sensores e dos dados da cidade Segurança e Privacidade são importantes para todos os requisitos funcionais relacionados ao gerenciamento de dados da plataforma e Configurabilidade é importante para todas as funcionalidades da plataforma. Boa parte desses requisitos funcionais são também os principais desafios técnicos e científicos para o desenvolvimento de sistemas para cidades inteligentes que serão descritos na próxima seção.

4.7. Uma Arquitetura de Referência para Plataformas de Cidades Inteligentes

A partir dos requisitos funcionais e não-funcionais levantados na seção anterior, apresentamos agora uma arquitetura de referência destacando os elementos mais importantes para uma plataforma de software para cidades inteligentes e como eles se inter-relacionam. O principal objetivo desta arquitetura é facilitar a compreensão, implementação e a integração de serviços e aplicações para cidades inteligentes. A Figura 4.12 apresenta uma visão geral da arquitetura.

A camada **Nuvem e Infraestrutura de Rede** é responsável pela hospedagem e comunicação entre os dispositivos e os serviços de software implantados na cidade. O objetivo desse componente é possibilitar a integração física de todos os dispositivos que estão conectados à plataforma, incluindo servidores, sensores, atuadores e dispositivos de usuários. A Computação em Nuvem é usada como mecanismo de suporte a diversos requisitos não-funcionais essenciais na plataforma, tais como escalabilidade, elasticidade e extensibilidade.

Em um nível de abstração superior à camada de Nuvem e Rede, a arquitetura de referência prevê o **Middleware de IoT** e o **Middleware de Serviços**. O primeiro administra e faz a interface das "coisas" implantadas na cidade possibilitando uma efetiva comunicação desses dispositivos com a plataforma. O Middleware de Serviços gerencia os serviços que a plataforma irá disponibilizar para as aplicações que serão implementadas utilizando a plataforma, fornecendo funcionalidades como implantação, publicação, descoberta, monitoração e composição de serviços.

Para disponibilizar serviços e aplicações melhores para os cidadãos, é importante que a plataforma armazene alguns dados e preferências dos usuários, isso é papel do componente de **Gestão de Usuários**. Porém, para garantir a privacidade do usuário, esses dados devem ser devidamente protegidos e devem ser coletados apenas com a expressa

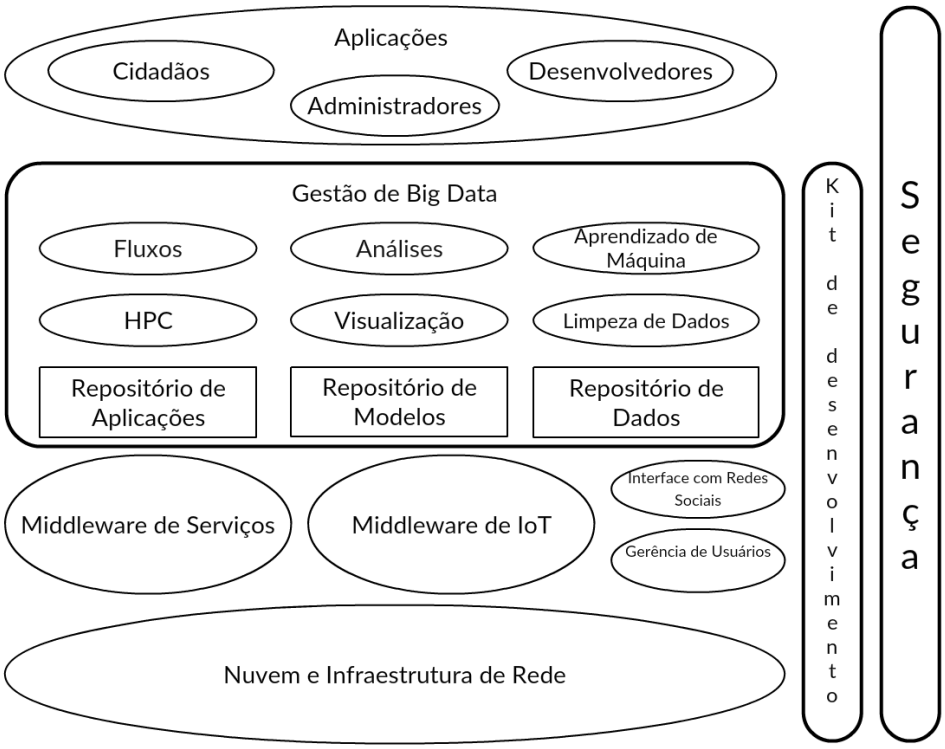

Figura 4.12. Arquitetura de Referência para Plataformas de Cidades Inteligentes

autorização dos usuários. Além disso, como haverá diversas aplicações sendo executadas na plataforma, esses dados serão úteis para oferecer um mecanismo único de autenticação (*single sign-on*).

Redes Sociais serão bastante importantes para cidades inteligentes, podendo ser usadas para coletar dados sobre as condições da cidade a partir de atualizações de cidadãos e também como um canal eficiente de comunicação entre a plataforma, os administradores da cidade e a população. Portanto, é importante facilitar a integração da plataforma com as redes sociais existentes, o que é de responsabilidade do **Gateway de Rede Social**.

A **Gestão de Big Data** é uma camada reunindo vários módulos que cuidam da gestão dos dados na plataforma. Entre suas responsabilidades está o armazenamento dos dados da cidade inteligente e, para isso, são utilizados três repositórios: (1) um **Repositório de Aplicações** para armazenar as aplicações, incluindo seu código fonte, arquivos executáveis e documentação; (2) um **Repositório de Modelos** para armazenar os diversos modelos que descrevem a cidade tais como modelos de trânsito, da rede de sensores, da rede de distribuição de energia e mapas e (3) um **Repositório de Dados** para armazenar dados coletados dos sensores, cidadãos e aplicações. Devido à diversidade de tipos e à quantidade de dados que serão coletados, existirão diversos repositórios de dados espalhados pela cidade que utilizarão tanto bancos de dados relacionais quanto NoSQL.

Além do armazenamento de dados, a camada de Gestão de Big Data também é responsável pelo processamento dos dados da cidade. Existem dois tipos de processamento que são mais adequados para diferentes situações: **Processamento de Fluxos de dados** para realizar o processamento de fluxos de dados contínuos e de processamento em

tempo real e **Processamento em Lote**, para analisar grandes conjuntos de dados e realizar o processamento de dados históricos. Ainda, esse modulo deve ser capaz de realizar diversos tipos de pré-processamento como filtragem, normalização e transformações.

Existe ainda um componente incluindo algoritmos de aprendizado de máquina que tem como objetivo "descobrir" de forma automatizada o comportamento da cidade inferindo modelos que descrevem a dinâmica dos agentes presentes na cidade. Dessa forma, será possível o processamento dos dados históricos a fim de gerar modelos que serão utilizados para prever o comportamento futuro da cidade. Finalmente, como uma cidade inteligente vai produzir uma quantidade enorme de dados, um componente para fazer a **Limpeza de Dados** é necessário, excluindo dados que não são mais necessários e armazenando dados antigos em dispositivos mais lentos e com maior capacidade.

Inúmeras ferramentas de software livre de alta qualidade podem ser utilizadas para a implementação dessa camada. Para o processamento de fluxos de dados alguns exemplos são o Apache Storm e o Apache Spark. Para o processamento em lote de grandes conjuntos de dados o Apache Hadoop (Polato et al. 2014) e o Apache Spark (Zaharia et al. 2010). Para os repositórios de dados, bancos de dados relacionais como o PostgreSQL[18] e o MySQL [19] para armazenar dados sobre a plataforma como usuários e configurações e bancos de dados NoSQL, tais como como o CouchDB (Anderson et al. 2010), MongoDB (Banker 2011) ou Cassandra [20], para o armazenamento de dados de sensores e dispositivos.

Utilizando a plataforma e os componentes descritos acima, desenvolvedores de software poderão desenvolver mais facilmente aplicações para cidades inteligentes. Disponibilizando dados e serviços abertos da cidade, cidadãos comuns podem executar, ou até mesmo desenvolver novas aplicações sobre a infraestrutura da cidade inteligente utilizando os dados de sensores, dispositivos e de serviços da cidade. As aplicações, além de usar os dados da plataforma, também poderão gerar novos dados que serão salvos na plataforma e que poderão ser usados por outras aplicações.

A plataforma deve ainda disponibilizar ferramentas para facilitar o desenvolvimento das aplicações com um ambiente integrado de desenvolvimento *(Integrated Development Environment - IDE)*, bibliotecas de componentes e *frameworks* e um simulador para permitir testes com diferentes cenários e situações.

Todos os componentes da plataforma devem levar em consideração vários dos requisitos não-funcionais apresentados na Seção 4.6.2 como escalabilidade, segurança, privacidade e interoperabilidade. Escalabilidade é fundamental por causa da quantidade de dados, dispositivos, usuários e serviços que uma plataforma de cidades inteligentes terá que gerenciar. Privacidade e Segurança são importantes porque uma plataforma de cidades inteligentes coleta, armazena e processa dados sensíveis dos cidadãos, empresas, ONGs e da administração da cidade. Interoperabilidade permitirá a operação integrada de diferentes tipos de serviços, diapositivos e aplicações.

[18]PostgreSQL - https://www.postgresql.org/

[19]MySQL - https://www.mysql.com/

[20]Apache Cassandra - http://cassandra.apache.org/

4.8. Desafios Científicos e Tecnológicos

Nesta seção serão apresentados os principais desafios de pesquisa para a implantação de plataformas, aplicações e da infraestrutura de cidades inteligentes. Entre esses desafios estão a segurança e a confiabilidade das informações dos cidadãos, o custo e as dificuldades da criação e manutenção de infraestrutura, sistemas e plataformas e os desafios inerentes à implementação e implantação de grandes sistemas distribuídos como escalabilidade e heterogeneidade. Além disso, há ainda desafios sociais e culturais importantes, como o incentivo à colaboração da população e o bom uso dos recursos públicos.

4.8.1. Privacidade

Diversos dados dos cidadãos serão manipulados pela infraestrutura, pelas aplicações e pelas plataformas de uma cidade inteligente. Esses dados devem ser protegidos para evitar que usuários mal-intencionados consigam acesso aos mesmos. Também é indispensável que a forma que os dados serão armazenados e usados sejam notificados aos usuários e que qualquer alteração seja avisada antecipadamente.

4.8.2. Segurança

Além de garantir a privacidade dos dados dos usuários, a infraestrutura da cidade também deve ser segura. Deve ser possível a prevenção e recuperação de ataques à segurança dos sistemas. Isso é importante, pois diversos serviços dependerão do correto funcionamento de todos os elementos da infraestrutura da cidade, como sensores, aplicações e plataformas. Além disso, com a evolução das cidades inteligentes, o cidadão deve ficar dependente desses serviços, assim como hoje a maioria da população de uma grande cidade é dependente do sistema de transporte público.

Alguns exemplos de ataques que uma cidade inteligente pode sofrer são: ataque à infraestrutura de hardware, alterando os valores de leitura de sensores, ataques de negação de serviço, que deixam serviços indisponíveis e vandalismo nos elementos físicos da cidade (Piro et al. 2014; Hernández-Muñoz et al. 2011; Petrolo et al. 2014).

Alguns autores discutem que a cidade deve assegurar que seus sistemas sejam a prova de ciberterrorismo e cibervandalismo destacando que uma cidade com uma rede de sensores e atuadores deve ser especialmente segura, pois um usuário malicioso pode controlar a infraestrutura da cidade causando sérios problemas como acidentes, erros nas leituras de dados e ataques aos serviços públicos (Hancke et al. 2012; Gurgen et al. 2013).

4.8.3. Gestão dos Dados

Uma das principais características de cidades inteligentes é o grande volume de dados gerados. Esses dados podem ser estruturados, como as informações dos cidadãos, semi-estruturados como dados de leituras de sensores e não-estruturados como os fluxos de imagens de câmeras de tráfego e de segurança.

Alguns dos desafios de pesquisa nessa área são:

- **Armazenamento:** A quantidade de dados que deverão ser salvos em uma cidade inteligente é muito grande, por isso são necessárias ferramentas que sejam capazes de lidar com esse volume. Também serão necessários mecanismos que facilitem o

armazenamento e recuperação de dados estruturados e não estruturados.

- **Processamento:** Assim como no item anterior, para comportar o grande volume de dados, serão necessárias ferramentas que consigam fazer o processamento em um tempo aceitável. Algumas ferramentas já estão começando a ser usadas para esse fim como o Spark, o Hadoop e o Storm, mas muita pesquisa ainda é necessária para torná-las eficazes para a enorme quantidade de possibilidades de uso que se abre com as aplicações em cidades inteligentes.

- **Modelos:** Além do desempenho, outro desafio é a dificuldade da construção de modelos de dados completos e eficientes para que seja possível o processamento dos dados, a implementação de aplicações e a utilização de algoritmos de agregação, clusterização e inferência. Pesquisas científicas e tecnológicas nas áreas de Inteligencia Artificial, Mineração de Dados, Reconhecimento de Padrões e Aprendizado de Máquina serão necessárias para solucionar esses problemas.

- **Confiança:** Os diversos dispositivos que compõem uma cidade inteligente estão sujeitos a falhas. Por isso, é necessário o uso de ferramentas e algoritmos de validação dos dados. Por exemplo, em uma região da cidade em que o sinal do GPS não estiver bom, a leitura da posição de um ônibus ou de um sensor pode ser incorreta. Os desafios aqui são tanto diferenciar os dados bons dos dados ruins quanto desenvolver mecanismos para inferir os dados corretos a partir do conjunto total de dados disponíveis.

Em todos os desafios citados acima, é necessário garantir diversos requisitos não-funcionais como elasticidade (permitindo que a quantidade de recursos utilizados seja adequada à demanda), escalabilidade (para que o hardware e os serviços suportem picos de demanda) e privacidade (porque os dados analisados pertencerão aos cidadãos ou organizações da cidade).

4.8.4. Escalabilidade

Todos os componentes de uma cidade inteligente devem ser escaláveis para conseguir atender o crescimento da demanda por serviços e dados da cidade. Isso é necessário devido ao aumento populacional que a cidade pode ter e também para suportar eventos inesperados que podem fazer o acesso às aplicações e plataformas aumentarem ordens de magnitude em um pequeno intervalo de tempo, tais como grandes eventos, engarrafamentos e desastres naturais.

Além disso, é esperado que a quantidade de dados coletados aumente constantemente, devida à implantação de mais dispositivos, à criação de novos serviços e aplicações e ao aumento populacional. Por isso, um dos principais desafios na implantação da infraestrutura e na implementação de plataformas, serviços e aplicações é garantir que eles sejam escaláveis.

4.8.5. Heterogeneidade

A interoperabilidade entre a infraestrutura, aplicações e plataformas é um dos principais desafios para a construção de uma cidade inteligente. Para isso, é necessário lidar com a

grande heterogeneidade dos componentes de hardware e software que compõem o ambiente de uma cidade inteligente.

Alguns exemplos de onde esse desafio ocorre são: na instalação de múltiplos sensores e atuadores de diferentes fabricantes que possuem diferentes protocolos, nos diferentes tipos de semáforos que já estão instalados nas cidades e utilizam diferentes protocolos de comunicação e nas aplicações legadas da cidade que foram implementadas com diferentes linguagens de programação e interfaces quase nunca compatíveis.

Alguns autores citam diferentes módulos de uma cidade inteligente que devem lidar com esse desafio. Naphade et al. (Naphade et al. 2011) discute o problema dos dados que são coletados de diversas fontes e que necessitam de um modelo comum para poderem ser agregados e processados. Outros autores (Wenge et al. 2014) defendem a definição de padrões em vista dos dispositivos, sistemas e domínios heterogêneos.

4.8.6. Manutenção e Implantação da Infraestrutura

A criação de um ambiente de cidade inteligente demandará diversos investimentos para a implantação da infraestrutura necessária, como a implantação de uma rede de sensores e atuadores, a melhoria das redes sem fio, a integração entre os diferentes sistemas da cidade e a coleta de dados da infraestrutura já existente como pontos de ônibus e semáforos.

Após a implantação, será necessário também fazer a manutenção de todos esses componentes, pois todos estão sujeitos a falhas parciais e a quebras por completo. Por exemplo, um sensor pode ser danificado por vandalismo ou por acidente, ou pode fornecer dados errados por uma falha natural do equipamento. Isso será especialmente desafiador devido à grande quantidade de dispositivos instalados.

Este é um desafio técnico importante pois, mesmo a manutenção da infraestrutura já existente como ruas, semáforos, pontos de ônibus, sinalização e praças, já não é a ideal em muitas partes do mundo.

A dificuldade da manutenção de todo o software e hardware necessário para uma cidade inteligente, principalmente por causa da quantidade de dispositivos espalhados pela cidade é um desafio enorme (Perera et al. 2014; Wenge et al. 2014; Hancke et al. 2012). Parece ser fundamental a adoção de estratégias baseadas em monitoramento automático acopladas à solução automatizada de problemas com a infraestrutura.

4.8.7. Custos

Um problema para a implantação de uma infraestrutura de cidades inteligentes são os custos para a construção de todos os componentes necessários. Esse custo inclui a aquisição e instalação de todos os dispositivos necessários na cidade como sensores, atuadores, servidores e equipamentos de comunicação, o desenvolvimento do software, a contratação de equipes para manutenção e gerenciamento e a adaptação às mudanças nos processos da cidade.

Al Nuaimi et al. (Al Nuaimi et al. 2015) discutem a possibilidade de um projeto de cidade inteligente não ser corretamente desenvolvido, acarretando em custos elevados que serão desperdiçados. Alguns exemplos citados são o uso de um novo sistema de semáforos, que se for mal implementado pode piorar o trânsito e até causar acidentes.

Por isso, é citada a necessidade de monitoramento constante de todas as iniciativas de cidades inteligentes e o desenvolvimento de projetos pilotos em escala reduzida antes da implantação em toda a cidade.

Outro problema, também relacionado a custos, é que os benefícios da implantação de muitos dos serviços de cidades inteligentes apenas são percebidos pela população no longo prazo. Isso desincentiva os políticos, que muitas vezes estão interessados apenas na próxima eleição, a investirem em projetos desse tipo que possuem um custo elevado e que não necessariamente trazem benefícios a curto prazo.

4.8.8. Colaboração

Um desafio importante é incentivar a população a utilizar os serviços que serão disponibilizados na cidade e a compartilhar dados e informações sobre esse uso. Muitos dos serviços de cidades inteligentes dependem do engajamento da população. Wu et al. (Wu et al. 2014) citam a necessidade da criação de sistemas que incentivem a colaboração criando um ambiente onde os usuário de serviços e aplicações tenham uma relação de benefício mútuo quando eles contribuem, compartilham e usam dados da cidade.

Um projeto interessante para aumentar o engajamento da população em aplicações de Cidades Inteligentes é o sitema MITOS (*Multi-Input TranspOrt planning System*) (Diamantaki et al. 2013) que incorpora na plataforma SmartSantander elementos e mecanismos de jogos. O sistema permite a distribuição de premiações para usuários que realizam uma ou um conjunto de tarefas, como por exemplo o usuário que utilizou mais o transporte público em um mês, ou que pegou um determinado número de ônibus em um dia.

4.9. Implicações

Para melhorar a qualidade de vida das pessoas e otimizar o uso dos recursos da cidade, diversas iniciativas e intervenções serão necessárias na cidade para que elas fiquem mais "inteligentes". Isso traz diversas implicações para diversos agentes envolvidos nos processos da cidade como cidadãos, prefeitos, vereadores, desenvolvedores, administradores de sistemas, empreendedores e cientistas.

O principal objetivo de cidades inteligentes é a melhora na qualidade de vida dos cidadãos, isso será alcançado através do desenvolvimento de serviços e aplicações inteligentes utilizando os dados coletados na cidade e atuando sobre o seu espaço otimizando o funcionamento dos equipamentos e da infraestrutura urbana. Isso alterará profundamente o dia-a-dia das pessoas, facilitando tomadas de decisões baseadas em informações mais confiáveis e economizando tempo com novos e melhores serviços. Além disso, para melhorar os resultados das novas aplicações e serviços, as pessoas devem participar ativamente desses sistemas, compartilhando seus dados com as aplicações e com outros cidadãos.

As tecnologias citadas neste capítulo indicam a infraestrutura necessária para a implantação de ambientes de Cidades Inteligentes. Isso pode ajudar administradores da cidade a tomar decisões quanto aos investimentos necessários como a compra, instalação e manutenção da rede de sensores, a quantidade de servidores e dispositivos de comu-

nicação necessários e os custos para o desenvolvimento dos sistemas necessários para a cidade. Além disso, é possível analisar as iniciativas de cidades inteligentes existentes e definir quais terão maior impacto na cidade.

Para desenvolvedores de aplicações e *startups* que implementam aplicações relacionadas a cidades, essas iniciativas abrem uma grande oportunidade de negócios, principalmente com a utilização dos dados que as prefeituras disponibilizam em portais de dados abertos ou via APIs com dados em tempo real.

Finalmente, para pesquisadores da área de cidades inteligentes, é necessário entender os desafios técnicos e científicos ainda em aberto para a criação de uma cidade verdadeiramente inteligente. Também será necessário pesquisar quais os impactos (positivos e negativos) que essas tecnologias trarão para o cidadão, administradores, meio ambiente e os serviços da cidade.

Outro impacto importante de Cidades Inteligentes será nas ideias de E-Governo e E-Democracia. E-Governo é relacionado a disponibilização de serviços digitais pelos governos municipais, estaduais e federal. Esses serviços poderão ter maior qualidade e ter um impacto maior na qualidade de vida das pessoas caso possam integrar dados de diversas áreas.

Além da ideia de E-Governo, Cidades Inteligentes também podem facilitar a ideia de E-Democracia. E-Democracia tem três aspectos principais: transparência, abertura e engajamento (Van der Meer et al. 2014). Transparência refere-se com a possibilidade dos cidadãos consultarem documentos referentes as decisões dos administradores das cidades. Abertura é a disponibilidade dos dados dos processos e da infraestrutura da cidade. Finalmente, Engajamento, é a oportunidade para que os cidadãos contribuam nas decisões da cidade.

Alguns exemplos de ferramentas de E-Democracia são portais de dados abertos, já disponíveis em muitas cidades, *dashboards* e APIs abertas para a consulta de dados sobre as condições e a infraestrutura da cidade e aplicações que permitem que cidadãos discutam e proponham alterações em decisões do poder público.

4.10. Conclusões

Com o crescimento da população das grandes cidades ao redor mundo e o grande número de problemas que elas enfrentam, a necessidade de tornar as cidades mais inteligentes é cada vez maior. Os principais benefícios disso são a otimização da infraestrutura e serviços da cidade, o uso mais sustentável dos recursos e, consequentemente, a melhoria da qualidade de vida da população.

Este capítulo mostrou diversas iniciativas de cidades inteligentes ao redor do mundo como nas cidades de Santader, Amsterdã e Barcelona. Esses exemplos mostram que existem diversos projetos interessantes sendo desenvolvidos, mas que, em 2017, nenhuma cidade ainda possui uma infraestrutura completa de hardware e software para a coleta e análise dos dados urbanos e o desenvolvimento de aplicações e serviços para os cidadãos.

Além dessas iniciativas práticas, discutimos diversos projetos de pesquisa que estão desenvolvendo plataformas de software servindo de suporte para a implementação

de aplicações e processamento dos dados coletados na cidade. Entre as funcionalidades oferecidas por essas plataformas estão o gerenciamento de dados e serviços e o fornecimento de ferramentas que facilitam o desenvolvimento de aplicações. Além disso, as plataformas buscam prover um conjunto essencial de requisitos não-funcionais como escalabilidade, adaptação e sensibilidade ao contexto.

Ainda existe uma grande quantidade de desafios técnicos e de pesquisa no desenvolvimento de cidades inteligentes que precisam ser melhor explorada. Entre eles destacamos a necessidade de garantir a privacidade dos dados dos usuários nos sistemas da cidade, a segurança para tornar a infraestrutura e os sistemas da cidade a prova de usuários mal-intencionados, a escalabilidade na comunicação, armazenamento e processamento de dados para que mais cidadãos possam usar os serviços oferecidos, os altos custos ainda proibitivos em muitos casos e a dificuldade de manutenção de toda a infraestrutura de hardware e software da cidade que podem tornar as iniciativas de Cidades Inteligentes inviáveis.

Nas próximas duas décadas, presenciaremos o surgimento de centenas de cidades ao redor do mundo onde a Tecnologia da Informação e da Comunicação estará totalmente permeada no ambiente urbano ajudando a tornar as cidades mais sustentáveis, agradáveis e eficientes e menos estressantes e violentas. No entanto, outro fator absolutamente fundamental a ser pesquisado sãos os direitos humanos e a própria cidadania no contexto das Cidades Inteligentes do futuro que terão um alto grau de automação e monitoramento e uma grande influência no dia-a-dia de todos os habitantes da cidade. A possibilidade de efeitos negativos sérios não é remota e deve passar a entrar na pauta das discussões sobre o tema.

Finalmente, é importante lembrar que os problemas e prioridades das cidades do países desenvolvidos da Europa e da América do Norte não são os mesmos encontrados na América Latina e África, por exemplo. Nos países ricos, normalmente, a preocupação é trazer mais eficiência a cidades já bem estruturadas, com uma boa qualidade de vida. Por outro lado, nos países em desenvolvimento e sub-desenvolvidos, normalmente os problemas estão mais relacionados a enormes diferenças sociais e desigualdades na oferta de recursos e serviços a diferentes camadas da população.

Esperamos que a comunidade científica, trabalhando conjuntamente com nossos empreendedores, governantes e a população, possam desenvolver soluções criativas e eficazes para os desafios elencados acima de forma a atingir o objetivo essencial das cidades inteligentes: contribuir para a melhoria da qualidade de vida de todos os habitantes das cidades.

Referências

Aazam, M., Khan, I., Alsaffar, A. A., and Huh, E.-N. (2014). Cloud of things: Integrating internet of things and cloud computing and the issues involved. In *Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on*, pages 414–419. IEEE.

Al Nuaimi, E., Al Neyadi, H., Mohamed, N., and Al-Jaroodi, J. (2015). Applications of big data to smart cities. *Journal of Internet Services and Applications*, 6(1):25.

AlAwadhi, S. and Scholl, H. J. (2013). Aspirations and realizations: The smart city of seattle. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 1695–1703. IEEE.

Anagnostopoulos, T., Kolomvatsos, K., Anagnostopoulos, C., Zaslavsky, A., and Hadjiefthymiades, S. (2015). Assessing dynamic models for high priority waste collection in smart cities. *Journal of Systems and Software*, 110:178 – 192.

Anderson, J. C., Lehnardt, J., and Slater, N. (2010). *CouchDB: the definitive guide*. "O'Reilly Media, Inc.".

Apolinarski, W., Iqbal, U., and Parreira, J. X. (2014). The gambas middleware and sdk for smart city applications. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pages 117–122.

Bain, M. (2014). Sentilo - sensor and actuator platform for smart cities.

Bakıcı, T., Almirall, E., and Wareham, J. (2013). A smart city initiative: the case of barcelona. *Journal of the Knowledge Economy*, 4(2):135–148.

Banker, K. (2011). *MongoDB in action*. Manning Publications Co.

Barba, C. T., Mateos, M. A., Soto, P. R., Mezher, A. M., and Igartua, M. A. (2012). Smart city for vanets using warning messages, traffic statistics and intelligent traffic lights. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 902–907. IEEE.

Calbimonte, J.-P., Sarni, S., Eberle, J., and Aberer, K. (2014). Xgsn: An open-source semantic sensing middleware for the web of things. In *7th International Workshop on Semantic Sensor Networks*, number EPFL-CONF-200926.

Caragliu, A., Del Bo, C., and Nijkamp, P. (2011). Smart cities in europe. *Journal of urban technology*, 18(2):65–82.

Cardoso, R. S. and Issarny, V. (2007). Architecting pervasive computing systems for privacy: A survey. In *The Working IEEE/IFIP Conference on Software Architecture, 2007. WICSA'07*. IEEE.

Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209.

Cheng, B., Longo, S., Cirillo, F., Bauer, M., and Kovacs, E. (2015). Building a big data platform for smart cities: Experience and lessons from santander. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 592–599.

Cocchia, A. (2014). Smart and digital city: A systematic literature review. In Dameri, R. P. and Rosenthal-Sabroux, C., editors, *Smart City*, Progress in IS, pages 13–43. Springer International Publishing.

Coetzee, L. and Eksteen, J. (2011). The internet of things – promise for the future? an introduction. In *IST-Africa Conference Proceedings, 2011*, pages 1–9.

Dameri, R. P. (2013). Searching for smart city definition: a comprehensive proposal. *International Journal of Computers & Technology*, 11(5):2544–2551.

Demchenko, Y., de Laat, C., and Membrey, P. (2014). Defining architecture components of the big data ecosystem. In *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 104–112. IEEE.

Diamantaki, K., Rizopoulos, C., Tsetsos, V., Theona, I., Charitos, D., and Kaimakamis, N. (2013). Integrating game elements for increasing engagement and enhancing user experience in a smart city context. In *Intelligent Environments (Workshops)*, pages 160–171.

Distefano, S., Merlino, G., and Puliafito, A. (2012). Enabling the cloud of things. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 858–863. IEEE.

Djahel, S., Doolan, R., Muntean, G., and Murphy, J. (2014). A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. *Communications Surveys Tutorials, IEEE*, PP(99):1–1.

Díaz-Díaz, R., Muñoz, L., and Pérez-González, D. (2017). The business model evaluation tool for smart cities: Application to smartsantander use cases. *Energies*, 10(3).

Elmangoush, A., Coskun, H., Wahle, S., and Magedanz, T. (2013). Design aspects for a reference m2m communication platform for smart cities. In *Innovations in Information Technology (IIT), 2013 9th International Conference on*, pages 204–209.

Foell, S., Kortuem, G., Rawassizadeh, R., Handte, M., Iqbal, U., and Marrón, P. (2014). Micro-navigation for urban bus passengers: Using the internet of things to improve the public transport experience. In *Proceedings of the First International Conference on IoT in Urban Space*, URB-IOT '14, pages 1–6, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Fortes, M., Ferreira, V., Sotelo, G., Cabral, A., Correia, W., and Pacheco, O. (2014). Deployment of smart metering in the búzios city. In *Transmission & Distribution Conference and Exposition-Latin America (PES T&D-LA), 2014 IEEE PES*, pages 1–6. IEEE.

Fox, A., Patterson, D. A., and Joseph, S. (2013). *Engineering software as a service: An Agile approach using Cloud Computing*. Strawberry Canyon LLC.

Fox, A., Patterson, D. A., and Joseph, S. (2015). *Construindo Software como Serviço: Uma Abordagem Ágil Usando Computação em Nuvem*. Strawberry Canyon LLC.

Franke, T., Lukowicz, P., and Blanke, U. (2015). Smart crowds in smart cities: real life, city scale deployments of a smartphone based participatory crowd management platform. *Journal of Internet Services and Applications*, 6(1):1–19.

Galache, J. A., Yonezawa, T., Gurgen, L., Pavia, D., Grella, M., and Maeomichi, H. (2014). Clout: Leveraging cloud computing techniques for improving management of massive iot data. In *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*, pages 324–327.

Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanovic, N., and Meijers, E. (2007). Smart cities-ranking of european medium-sized cities. Technical report, Vienna University of Technology.

Girtelschmid, S., Steinbauer, M., Kumar, V., Fensel, A., and Kotsis, G. (2013). Big data in large scale intelligent smart city installations. In *Proceedings of International Conference on Information Integration and Web-based Applications and Services*, IIWAS '13, pages 428:428–428:432, New York, NY, USA. ACM.

Goldman, A., Kon, F., Junior, F. P., Polato, I., and de Fátima Pereira, R. (2012). Apache hadoop: conceitos teóricos e práticos, evolução e novas possibilidades. *XXXI Jornadas de atualizaçoes em informatica*.

Guan, L. (2012). Smart steps to a battery city. *Government News*, 32(2):24–27.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

Guo, B., Wang, Z., Yu, Z., Wang, Y., Yen, N. Y., Huang, R., and Zhou, X. (2015). Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Comput. Surv.*, 48(1):7:1–7:31.

Gurgen, L., Gunalp, O., Benazzouz, Y., and Gallissot, M. (2013). Self-aware cyber-physical systems and applications in smart buildings and cities. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1149–1154.

Hall, P. et al. (2000). Creative cities and economic development. *Urban studies*, 37(4):639–649.

Hancke, G. P., Hancke Jr., G. P., et al. (2012). The role of advanced sensing in smart cities. *Sensors*, 13(1):393–425.

Handte, M., Iqbal, M. U., Wagner, S., Apolinarski, W., Marrón, P. J., Navarro, E. M. M., Martinez, S., Barthelemy, S. I., and Fernández, M. G. (2014). Crowd density estimation for public transport vehicles. In *EDBT/ICDT Workshops*, pages 315–322.

Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnanam, J., Paraszczak, J., and Williams, P. (2010). Foundations for smarter cities. *IBM Journal of Research and Development*, 54(4):1–16.

Hernández-Muñoz, J. M., Vercher, J. B., Muñoz, L., Galache, J. A., Presser, M., Hernández Gómez, L. A., and Pettersson, J. (2011). Smart cities at the forefront of the future internet. volume 6656, pages 447–462.

Hussain, A., Wenbi, R., da Silva, A. L., Nadher, M., and Mudhish, M. (2015). Health and emergency-care platform for the elderly and disabled people in the smart city. *Journal of Systems and Software*, 110:253 – 263.

Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadist, P., Autili, M., Gerosa, M. A., and Hamida, A. (2011). Service-oriented middleware for the future internet: state of the art and research directions. *Journal of Internet Services and Applications*, 2(1):23–45.

Janajreh, I., Su, L., and Alan, F. (2013). Wind energy assessment: Masdar city case study. *Renewable energy*, 52:8–15.

Khan, Z., Anjum, A., and Kiani, S. L. (2013). Cloud based big data analytics for smart future cities. In *Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on*, pages 381–386.

Khan, Z., Anjum, A., Soomro, K., and Tahir, M. A. (2015). Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing*, 4(1):1–11.

Kshetri, N., Alcantara, L. L., and Park, Y. (2014). Development of a smart city and its adoption and acceptance: the case of new songdo. *Communications & Strategies*, (96):113.

Le-Phuoc, D., Nguyen-Mau, H. Q., Parreira, J. X., and Hauswirth, M. (2012). A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web*, 16:42–51.

Lee, Y. W. and Rho, S. (2010). U-city portal for smart ubiquitous middleware. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 1, pages 609–613.

Leite, L., Moreira, C. E., Cordeiro, D., Gerosa, M. A., and Kon, F. (2014). Deploying large-scale service compositions on the cloud with the choreos enactment engine. In *13th IEEE International Symposium on Network Computing and Applications (NCA)*, pages 121–128. IEEE.

Liu, P. and Peng, Z. (2013). Smart cities in china. *IEEE Computer Society*, 16.

Manville, C., Cochrane, G., Cave, J., Millard, J., Pederson, J. K., Thaarup, R. K., Liebe, A., Wissner, M., Massink, R., and Kotterink, B. (2014). Mapping smart cities in the eu.

March, H. and Ribera-Fumaz, R. (2016). Smart contradictions: The politics of making barcelona a self-sufficient city. *European Urban and Regional Studies*, 23(4):816–830.

Mitton, N., Papavassiliou, S., Puliafito, A., and Trivedi, K. (2012). Combining cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking*, 2012(1).

Mylonas, G., Theodoridis, E., and Munoz, L. (2015). Integrating smartphones into the smartsantander infrastructure. *Internet Computing, IEEE*, 19(2):48–56.

Naphade, M., Banavar, G., Harrison, C., Paraszczak, J., and Morris, R. (2011). Smarter cities and their innovation challenges. *Computer*, 44(6):32–39.

Nitti, M., Pilloni, V., Giusto, D., and Popescu, V. (2017). Iot architecture for a sustainable tourism application in a smart city environment. *Mobile Information Systems*, 2017.

Papa, R., Gargiulo, C., and Galderisi, A. (2013). Towards an urban planners' perspective on smart city. *TeMA Journal of Land Use, Mobility and Environment*, 6(01):5–17.

Parkavi, A. and Vetrivelan, N. (2013). A smart citizen information system using hadoop: A case study. In *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, pages 1–3.

Pereira, R. L., Sousa, P. C., Barata, R., Oliveira, A., and Monsieur, G. (2015). Citysdk tourism api-building value around open data. *Journal of Internet Services and Applications*, 6(1):1–13.

Perera, C., Zaslavsky, A. B., Christen, P., and Georgakopoulos, D. (2014). Sensing as a service model for smart cities supported by internet of things. *Trans. Emerging Telecommunications Technologies*, 25(1):81–93.

Pérez-González, D. and Díaz-Díaz, R. (2015). Public services provided with ict in the smart city environment: The case of spanish cities. *Journal of Universal Computer Science*, 21(2):248–267.

Petrolo, R., Loscri, V., and Mitton, N. (2014). Towards a Cloud of Things Smart City. *IEEE COMSOC MMTC E-Letter*, 9(5):44–48.

Piro, G., Cianci, I., Grieco, L. A., Boggia, G., and Camarda, P. (2014). Information centric services in smart cities. *Journal of Systems and Software*, 88(0):169 – 188.

Polato, I., Ré, R., Goldman, A., and Kon, F. (2014). A comprehensive view of hadoop research—a systematic literature review. *Journal of Network and Computer Applications*, 46:1–25.

Privat, G., Zhao, M., and Lemke, L. (2014). Towards a shared software infrastructure for smart homes, smart buildings and smart cities. In *International Workshop on Emerging Trends in the Engineering of Cyber-Physical Systems, Berlin*.

Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., et al. (2014). Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238.

Stephenson, M., Di Lorenzo, G., and Aonghusa, P. M. (2012). Open innovation portal: A collaborative platform for open city data sharing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 522–524. IEEE.

Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision and challenges for realising the internet of things.

Takahashi, K., Yamamoto, S., Okushi, A., Matsumoto, S., and Nakamura, M. (2012). Design and implementation of service api for large-scale house log in smart city cloud. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 815–820.

Tei, K. and Gurgen, L. (2014). Clout: Cloud of things for empowering the citizen clout in smart cities. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 369–370. IEEE.

Thornton, S. (2013). Chicago's windygrid: Taking situational awareness to a new level.

United Nations (2009). Urban and rural areas 2009.

Vakali, A., Anthopoulos, L., and Krco, S. (2014). Smart cities data streams integration: Experimenting with internet of things and social data flows. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, WIMS '14, pages 60:1–60:5, New York, NY, USA. ACM.

Van der Meer, T. G., Gelders, D., and Rotthier, S. (2014). E-democracy: exploring the current stage of e-government. *Journal of Information Policy*, 4:489–506.

Villanueva, F. J., Santofimia, M. J., Villa, D., Barba, J., and Lopez, J. C. (2013). Civitas: The smart city middleware, from sensors to big data. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pages 445–450.

Vlahogianni, E. I., Kepaptsoglou, K., Tsetsos, V., and Karlaftis, M. G. (2014). Exploiting new sensor technologies for real-time parking prediction in urban areas. In *Transportation Research Board 93rd Annual Meeting Compendium of Papers*, pages 14–1673.

Wan, J., Li, D., Zou, C., and Zhou, K. (2012). M2m communications for smart city: An event-based architecture. In *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on*, pages 895–900.

Washburn, D., Sindhu, U., Balaouras, S., Dines, R., Hayes, N., and Nelson, L. (2009). Helping cios understand "smart city" initiatives. *Growth*, 17:2.

Wenge, R., Zhang, X., Cooper, D., Chao, L., and Hao, S. (2014). Smart city architecture: A technology guide for implementation and design challenges. *Communications, China*, 11(3):56–69.

Wolff, A., Kortuem, G., and Cavero, J. (2015). Towards smart city education. In *2015 Sustainable Internet and ICT for Sustainability (SustainIT)*, pages 1–3.

Wu, C., Birch, D., Silva, D., Lee, C.-H., Tsinalis, O., and Guo, Y. (2014). Concinnity: A generic platform for big sensor data applications. *Cloud Computing, IEEE*, 1(2):42–50.

Yamamoto, S., Matsumoto, S., Saiki, S., and Nakamura, M. (2014). Using materialized view as a service of scallop4sc for smart city application services. In *Soft Computing in Big Data Processing*, pages 51–60. Springer.

Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., and David, B. (2015). A literature survey on smart cities. *Science China Information Sciences*, pages 1–18.

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, volume 10, page 10.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *Internet of Things Journal, IEEE*, 1(1):22–32.

Chapter 5

Algoritmos e Modelos de Programação em Big Data

Fabio Porto (LNCC)

Abstract

Big Data is a phenomenon that currently attracts a huge attention both in academia and industry. It is understood as an important asset as it provides data about all sorts of activities, in different granularity from individuals to huge groups and from all sorts of phenomena. The data deluge resulting from the Big Data phenomenon needs to be processed and interpreted. The later however is challenging as data do not fit in memory of a single computer and the type of analysis usually involves scanning almost all the available data. In order to deal with these challenges new parallel data processing models have emerged enabling the distribution of data and processing, potentially through thousands of machines. In this course, we wil present the MapReduce programming model and its implementation in the Spark Framework. We will also discuss algorithms that can help implementing applications on top of this model.

Resumo

Big Data é um fenômeno que atrai grande atenção no meio acadêmico e nas empresas. É considerado um importante ativo, presente em todas as atividades, e envolvendo dados sobre indivíduos, assim como sobre grupos, como aqueles em redes sociais. Para ser transformado em informação, o grande volume de dados proporcionado pelo fenômeno Big Data precisa ser processado e interpretado. O processamento, no entanto, impõe um grande desafio, uma vez que a quantidade de dados é tal que não cabe na memória principal de uma máquina, ainda que potente, e as análises via de regra varrem todo o conjunto disponível. De forma a lidar com esse desafio, foram propostos frameworks de processamento paralelo, como o MapReduce, que oferece uma interface simples de programação e um sistema de processamento que escala a até milhares de máquinas. O framework Apache Spark é um dos que se destaca neste panorama. Neste curso, apresentaremos o modelo de programação MapReduce e sua implementação Apache Spark. Discutiremos ainda algoritmos que auxiliam no desenvolvimento de aplicações neste paradigma.

5.1. Introdução

O fenômeno Big Data se apresenta como uma das novidades dessa década com maior impacto na sociedade. Realçado pela mídia [Economist, 2010, Nature, 2008], Big Data passou a ser um modelo de negócios e ciência em quase todos os domínios. Na ciências, podemos citar a astronomia como uma das primeiras a adotar a abordagem baseada em dados para avaliação de hipóteses científicas. Levantamentos digitais, tais como o Sloan Digital Sky Survey [SDSS,], se baseiam inteiramente em dados sobre corpos celestes extraídos de imagens capturadas pelo telescópio Sloan Foundation, localizado no Novo México. Igualmente, na bioinformática, bases de dados como as do GeneBank, no NCBI, crescem continuamente com a publicação de dados de sequenciamento genético de novos organismos, permitindo a investigação por cientistas e empresas. É claro que não se pode deixar de mencionar as grandes empresas ligadas a Web, como a Google, e as redes sociais, como o Facebook. Atualmente, essas empresas são os maiores investidores em processos automatizados puramente baseadas na análise de dados de registros de operações realizadas em seus sites. Processos como os de recomendação comercial, seguindo característica de seu perfil navegacional são usados na tomada de decisão sobre que anuncio apresentar para cada um dos usuários desses sistemas. Bom, é oportuno, no entanto, indagar o que é Big Data e, principalmente, como essa abordagem afeta o modelo de desenvolvimento de aplicações que pretende extrair conhecimento deste grande volume de dados? O grupo Gartner [1] caracterizou Big Data como um fenômeno ligado a produção de dados nas atividades em que aparecem 5 V's: Volume, Variedade, Velocidade, Valor, Veracidade. Ainda que todas essas características sejam de fato relevantes no contexto de Big Data, o volume de dados aparece como a que ganhou maior destaque inicialmente, principalmente pelo desafio que arquivos com volumes muito grandes impõem para a quantidade de memória principal disponível em um único computador. Além dos desafio impostos pelos 5V's, um ponto de suma importância a considerar no contexto de Big Data é o tipo de acesso aos dados. Em aplicações tradicionais, o modelo de processamento principal tem por alvo um indivíduo ou grupos de indivíduos. Em Big Data,informações sobre um indivíduo, em geral, tem pouca importância, que não seja para sua contribuição a um agregado de interesse. As aplicações que se debruçam sobre grandes volumes de dados estão interessadas em tendências, ou em padrões que emergem a partir do dado agregado. Neste contexto, os padrões de acesso envolvem a varredura de grandes partes dos dados, combinada com algum tipo de investigação específica complementar à informações disponível no arquivo de base da exploração. Notem o termo *exploração* que denota o tipo de experiência de acesso em dados neste contexto.

Dessa forma, neste curso, vamos considerar a seguinte definição para Big Data:

Definition 5.1.1. Big data é o desafio imposto para se processar grandes arquivos, cujo tamanho extrapola em muito a quantidade de memória disponível em uma única máquina. Além disso, consideramos que o processamento de dados em grandes volumes se interessa à características que emergem ao se varrer uma boa parte de seus dados.

Uma vez definida a classe de problemas que nos é de interesse, podemos enriquecer nosso entendimento do problema avaliando elementos relevantes para o desenvolvimento de aplicações que respondam ao desafio Big Data. Claramente, um primeiro ponto

[1] www.gartner.com

seria avaliar a arquitetura computacional a abrigar o ambiente de processamento. A premissa de que os dados a serem processados não cabem na memória principal de um único computador se baseia no fato de que o crescimento do volume de dados disponível para exploração não segue a lei de Moore [Moore, 2000] e, ainda que tenhamos computadores cada vez com maior volume de memória a ser alocado, o conjunto de dados disponível tende a superá-lo. Neste sentido, o grupo da Google que propôs o arcabouço de software baseado no modelo *MapReduce* (MR), considerou aplicações implementadas sobre um cluster com até milhares de máquinas simples [Dean and Ghemawat, 2008]. Apesar da maioria das aplicações não utilizar um número tão grande de máquinas, a possibilidade de escalar o processamento sobre a quantidade de nós disponíveis é bastante atraente, sem contar com a alternativa de uso da nuvem com alocação de recursos de grandes provedores. Desta forma, a partir do arcabouço Apache Hadoop, considerou-se a plataforma de cluster de servidores sem compartilhamento como o padrão de fato para processamento Big Data. Dois outros elementos principais, entre vários componentes do que se passou a chamar de Ecosistema Big Data, merecem realce neste momento. O primeiro diz respeito ao gerenciamento dos arquivos por entre as máquinas componentes do cluster. Os arcabouços MR [2] priorizam a execução dos procedimentos da aplicação de forma a minimizar a transferência de dados, optando se possível pela alocação do procedimento no mesmo nó em que os dados estejam distribuídos. Para esse fim, foi projetado o sistema de arquivos distribuídos *Google File System* (GFS) [Ghemawat et al., 2003], posteriormente implementado no arcabouço Hadoop como *Haddop File System (HDFS)*.Dessa forma, o modelo de programação considera arquivos distribuídos por nós de um cluster e processos alocados a estes nós. A escalabilidade da estratégia para tamanhos crescentes de arquivos é proporcionada pela divisão do arquivo pelos nós de processamento, considerando-se que os problemas sendo tratados possam ser resolvidos de forma paralela e distribuída. O segundo elemento dessa abordagem esta ligado a extensibilidade do arcabouço para diferentes problemas que compartilhem a característica de independência de processos paralelos. Este aspecto do problema é tratado pela adoção do modelo de programação funcional tendo por base as funções de *Map* e *Reduce*. Neste modelo, conforme veremos na seção 5.4, o usuário fornece procedimentos da aplicação modelados para processar cada item do conjunto de entrada (Map), e um conjunto de elementos da entrada (Reduce). A partir dessa simplificação, uma grande gama de problemas pode ser resolvida fornecendo-se o comportamento para as respectivas funções.

Neste contexto, definimos o tema geral deste curso. Dada um ambiente de processamento de grandes volumes de dados, principalmente norteado por soluções com arcabouços MR, especificam-se algoritmos que se adequam ao modelo de programação subjacente e forneçam soluções eficientes para a implementação de aplicações.

O restante deste curso encontra-se dividido da seguinte forma. Na seção 5.2 apresentamos um exemplo de aplicação no domínio da astronomia que servirá de base para nossas discussões. Em seguida, na seção 5.3 discutimos o ambiente para arcabouços MR, incluindo: a arquitetura , o ecosistema MR, e seus principais módulos tomando como base o arcabouço Apache Spark. Na seção 5.4, apresentamos a API de programação com suas funções principais. Uma vez apresentado o ambiente de execução e a API para especifi-

[2]Usaremos o termo *arcabouços MR* para nos referirmos aos arcabouços que implementam o modelo MapReduce

Figure 5.1. Dilúvio de Dados na Midia

cação de aplicações, passamos a discutir algoritmos de apoio e sua implicação no modelo MR. Neste sentido, a seção 5.5 discute estratégias para o particionamento dos dados pelos nós do cluster. Em seguida, na seção 5.6 discutimos as estruturas de indexação QuadTree e PH-Tree e KD-Tree, de apoio para consultas de intervalos e vizinhança. A seção 5.7 discute o tratamento de redução de dimensionalidade. Finalmente, a seção 5.8 trata de algoritmos que otimizam as funções de agregação. A seção 5.9 tesse alguns comentários finais.

5.2. Exemplo de Aplicação

Nesta seção, descrevemos uma aplicação da astronomia, *Constellation Queries* (CQ) [Porto et al., 2017], cujo objetivo é varrer grandes arquivos contendo objetos estelares identificados por telescópios em levantamentos astronômicos, tal como o SDSS [SDSS,], buscando por ocorrência do fenômeno conhecido como lentes gravitacionais [3]. A aplicação busca por padrões geométricos definidos a partir da composição de corpos estelares. A Einstein cross é uma dessas formas representantes do fenômeno de lentes gravitacionais. Seu padrão geométrico é composto por quatro corpos estelares, veja na Figura 5.3. Consideram-se soluções candidatas todas as composições cujas formas se assemelhem à forma geométrica fornecida e que guardem com esta uma relação de escala. Esta última

[3]https://www.cfa.harvard.edu/castles/

é função da razão entre as distâncias entre dois elementos correspondentes, entre a forma fornecida como padrão e a forma candidata. Neste sentido, o *dataflow* CQ implementa o processo de busca por lentes gravitacionais. Sua representação gráfica aparece na Figura 5.2

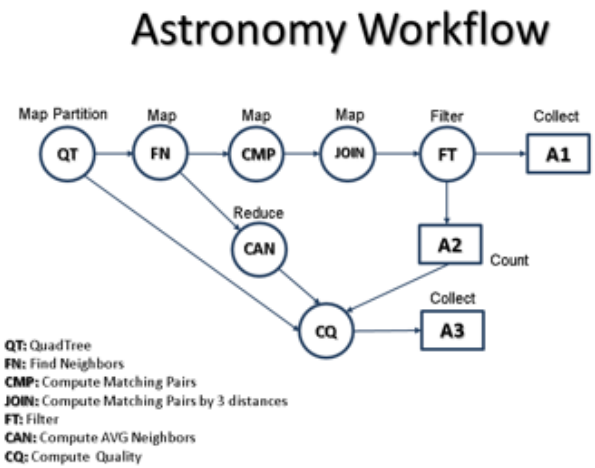

Figure 5.2. Dataflow - Constellation Queries

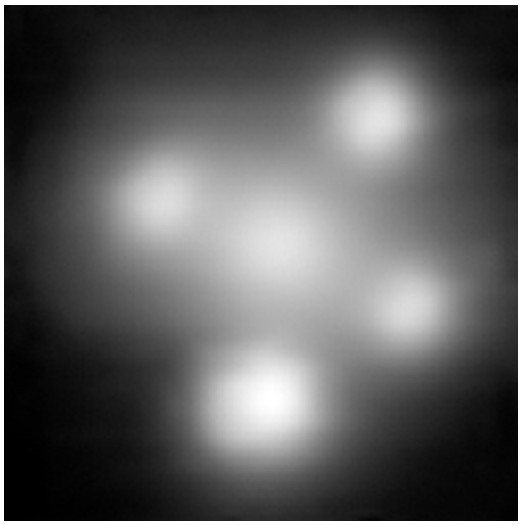

Figure 5.3. Einstein Cross

[Eigenbrod et al., 2008]

Ao longo do texto, detalheremos o dataflow *CQ*. Por hora, é suficiente perceber

que o dataflow define uma estrutura em grafo direcionado. Os nós do grafo correspondem a operações de transformação de dados e as arestas estabelecem uma relação produtor-consumidor. Os elementos quadrangulares são operações de produção de resultados finais.

5.3. Arquitetura

Arcabouços MR foram projetados em conformidade com arquitetura de paralelismo de dados. A Figura 5.4 [Ozsu and Valduriez, 1990] apresenta os principais modelos arquiteturais de clusters para processamento paralelo. Na Figura 5.4 (a), ilustra-se a arquitetura de memória compartilhada. Neste modelo de arquitetura, nós de computação possuem CPU e disco e compartilham uma área de memória obtida pela composição de áreas das máquinas envolvidas. Um exemplo importante deste tipo de arquitetura é o Módulo *fatnode* do sistema de super-computação Santos-Dumont. Esta arquitetura privilegia programas desenvolvidos para perceberem uma grande área de memória global com mecanismos de controle de acesso concorrente entre os diversos processos paralelos em execução. Em seguida, na ilustração (b) apresenta-se a arquitetura de disco compartilhado. Esta é a arquitetura paralela mais comum entre *clusters*. Basicamente, cada nó possui sua própria área de memória e CPU, porém compartilha o acesso aos dados com todos os outros nós através de um sistema de arquivo distribuído, como, por exemplo, o Lustre [4]. Todo acesso a dados requer a transferência pela rede entre o nó em que o arquivo se encontra e o solicitante. Usualmente, instala-se mecanismos de alta-velocidade entre os nós de processamento. Por último, na arquitetura ilustrada em (c), encontramos a arquitetura *sem compartilhamento*. Este é o modelo adotado por arcabouços MR. O sistema de arquivos distribuído oferece uma visão integrada de arquivos particionados e distribuídos pelos nós do *cluster*. Um módulo centralizado pode tomar conta dos fragmentos armazenados pelos vários nós de processamento. Adicionalmente, escalonadores de execução de jobs, como o *YARN*, procuram alocar os programas em nós onde haja partição do arquivo de entrada. Assim arquivos grandes possuem número de partições (veja seção 5.5) bem superior ao número de nós de processamento, fazendo com que os nós realizem trabalho durante todo o tempo de processamento do arquivo. Cada processo ativado, acessará o bloco local do arquivo de entrada e carregá-lo-a em sua área de memória local para processamento pelo diferentes núcleos da máquina.

5.3.1. O Sistema de Arquivos HDFS

O *Hadoop File System* (HDFS) é o sistema de arquivos distribuídos utilizado pela maioria dos arcabouços MR atuais, incluindo Apache Hadoop, Apache Spark e Apache Flink. O sistema foi projetado para aplicações de grandes volumes de dados priorizando: distribuição de dados e com isso, influenciando paralelismo de tarefas; tolerância à falhas através de réplicas de arquivos; e impedimento de atualização, eliminando controle de concorrência.

O sistema adota o modelo (grava-um-lê-muitos) de forma que um arquivo HDFS não sofre atualizações, uma vez que tenha sido criado, podendo ser lido múltiplas vezes. Veremos, quando discutirmos o sistema Spark, que arquivos do HDFS são lidos por trans-

[4] lustre.org

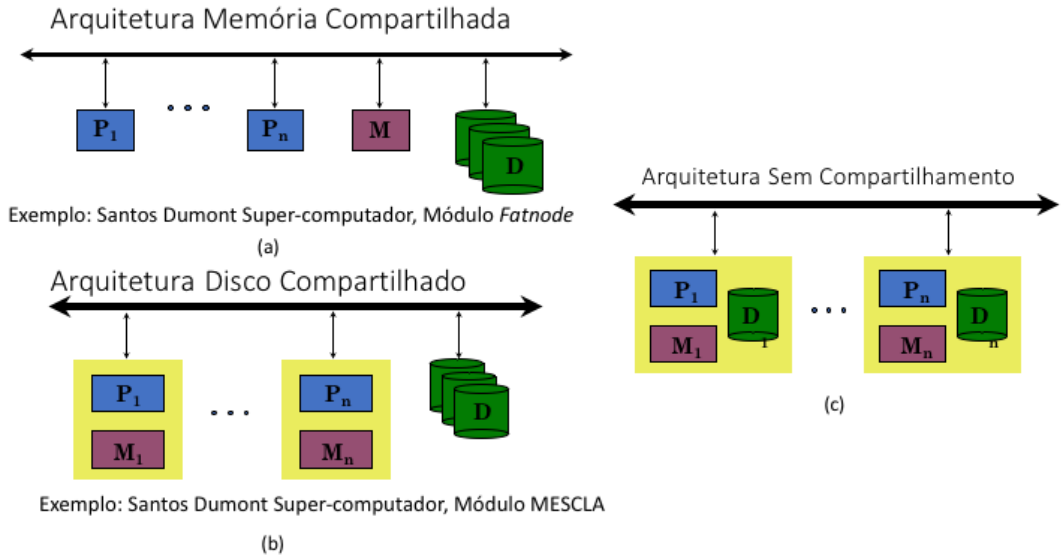

Figure 5.4. Arquitetura para Processamento Paralelo

formações que os processam e produzem novos arquivos, em memória ou em disco. Desta forma, o modelo não considera processos de atualização dos arquivos existentes.

O sistema está estruturado por um módulo servindo de catálogo geral *Nó de Nomes*, executando no nó computacional Mestre, e um módulo presente em cada um dos nós escravos, chamado *Nó de Dados*, ver Figura 5.5. Um arquivo no sistema HDFS é referenciado pelo seu caminho e nome individual. O sistema de caminhos obedece a uma estrutura hierárquica, comum a maioria dos sistemas de arquivos.

Um arquivo no HDFS é particionado em blocos de tamanho fixo típico de 64MB ou 128 MB. Blocos de arquivos são distribuídos pelos nós do cluster. Blocos em um nó são gerenciados pelo respectivo *Nó de nomes*. Novamente referindo-se ao processamento, blocos do HDFS são a unidade de processamento de tarefas no Spark. Sua distribuição pelos nós do cluster dirigem a execução paralela de processos que consomem blocos dos arquivos.

Adicionalmente, o sistema HDFS permite a réplica de blocos por nós do cluster. Inicialmente, o sistema é configurado para que cada bloco tenha duas réplicas, totalizando 3 réplicas por bloco alocadas em diferentes nós. Quando um nó falha durante a execução de um job, o *Nó de Nomes* informa a localização de réplicas dos blocos alocados ao nó apresentando problemas, permitindo que o escalonador re-inicie a tarefa em andamento em um dos nós contendo réplicas.

Finalmente, o HDFS é o componente fundamental no ecosistema de arcabouços MR. O sistema foi desenvolvido em Java, permitindo fácil integração com aplicações nas linguagens Scala, Java e com conectores para Python. As aplicações desenvolvidas utilizando-se de APIs nestas linguagens podem ser integradas aos arcabouços, permitindo a paralelização de suas tarefas, segundo o modelo MR.

[6]https://www.ibm.com/developerworks/br/library/waintrohdfs/

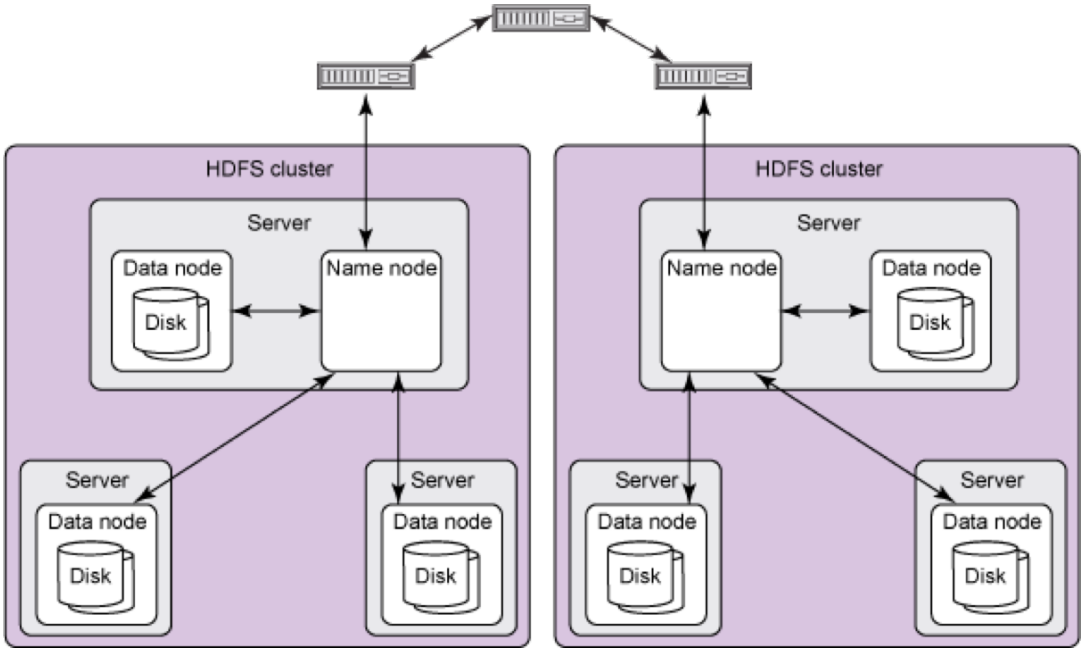

Figure 5.5. Arquitetura do Sistema HDFS - Fonte: IBM [6]

5.3.2. Utilizando o sistema HDFS

O *HDFS* pode ser utilizado como um sistema de arquivo adaptado a partir do protocolo *POSIX*. Todos os comandos são executados a partir do script *Hadoop*. Assim, pode-se executar:

```
hadoop [comando] [opçoes-genericas] [opçoes-comando]

hadoop fs -ls /user/hadoop/file1

hadoop fs -mkdir /user/hadoop/dir1

hadoop fs -cp /user/hadoop/file1 /user/hadoop/file2

hadoop fs -tail /user/hadoop/file1
```

5.4. O modelo de Programação MapReduce

Os arcabouços MR, como o Apache hadoop e o Apache Spark, foram desenvolvidos considerando a arquitetura paralela *sem-compartilhamento* conforme discutido em 5.3. Neste sentido, os sistemas utilizam a distribuição de dados provida por um sistema de arquivos distribuídos, como o HDFS. A distribuição de blocos de dados pelos nós do cluster direciona a alocação dos processos da aplicação para os mesmos nós. A este princípio chamamos de *Localidade de dados*.

Table 5.1. Comandos HDFS

Comando	Descrição	Exemplo
fs	realiza operações sobre os arquivos no HDFS	hadoop fs -cp /user/hadoop/file1 /user/hadoop/file2
put	copia arquivos para o HDFS do sistema de arquivos local	hadoop fs -put localfile1 localfile2 /user/hadoop/hadoopdir
copytolocal	copia arquivos do HDFS para o sistema de arquivos local	hadoop fs -copytolocal /user/hadoop/file localfile

Definition 5.4.1. *Localidade de Dados* é uma estratégia de processamento de dados distribuídos em que os processos são alocados, prioritariamente, nos nós em que os dados se encontram, minimizando a transferência de dados entre nós.

Precisamos, no entanto, entender que tipo de processos estão envolvidos em processamento de grande volumes de dados. Na seção 5.2, apresentamos um dataflow representando a aplicação de busca por lentes gravitacionais. Podemos iniciar a discussão sobre modelos de processamento de grandes volumes de dados comparando: *Workflow Científicos*, *Dataflows* e *Consultas em Banco de dados*. Enquanto todos esses modelos tratam de processamento de grandes volumes de dados, suas diferenças são marcantes.

- Workflows Científicos - são conjuntos de programas escritos em linguagens diversas e de forma autônoma, colocados em um mesmo processo de forma a definirem uma relação de produtor-consumidor de dados. A heterogeneidade de dados e o uso de programas intensivos de CPU, levam a implementações que não favorecem necessariamente a localidade de dados [Ogasawara et al., 2013, Ludäscher et al., 2006].
- Dataflows - são processos que implementam uma aplicação e são desenvolvidos em uma mesma linguagem de programação, de forma funcional, estabelecendo relação de produtor-consumidor entre as funções através de arquivos. O processamento é considerado intensivo de dados.
- Consulta de banco de dados - é a expressão de interesse em um subconjunto do banco de dados cujas operações são conhecidas e compõem a álgebra do modelo de dados. Uma consulta de banco de dados é tradicionalmente traduzida para um plano de consulta composto por operadores e uma relação de produtor-consumidor entre eles. No entanto, diferentemente das anteriores, as operações têm semântica conhecida.

Neste sentido, em arcabouços MR, uma aplicação é especificada como um dataflow, podendo ser representado por um grafo $D = (O, E)$, onde O é um conjunto de operações e E é um subconjunto de $O \times O$ definindo a relação de produtor-consumidor entre os elementos de O. Cada operação $o \in O$ está associada a um regra de transformação de dados, definida na linguagem de especificação do dataflow.

Definition 5.4.2. Transformações são funções $F = \{f_1, f_2, \ldots, f_n\}$ que recebem um ou mais arquivos de entrada e produzem um ou mais arquivos de saída. O comportamento de uma função $f_i \in F$ é conhecido e estabelece uma relação entre cada elemento do arquivo de entrada e elementos do arquivo de saída.

Tendo introduzido esses conceitos, podemos agora discutir sua implementação no arcabouço Apache Spark.

5.4.1. Apache Spark

O Apache Spark é um arcabouço MR escrito na linguagem Scala e descrito na tese de Doutorado de Matei Zaharia e apresentado em [Zaharia et al., 2010]. O arcabouço é integrado a várias fontes de dados, incluindo o *HDFS*. Seu modelo de programação baseado em *Localidade de Dados* é bastante semelhante ao do Hadoop porém otimiza o processamento de um *dataflow* mantendo os arquivos intermediários em memória. A motivação principal é que *dataflows* apresentam sequências de operações que podem ser executadas, uma após a outra, em um mesmo nó de processamento, evitando a transferência de arquivos entre nós. Neste cenário, a manutenção dos arquivos intermediários entre funções na memória principal pode levar a ordens de magnitude de ganho em tempo de processamento. Em *Spark* operações são classificadas em dois tipos: transformações e ações. As transformações recebem um ou mais *RDDs* de entrada e produzem um *RDD* de saída. Já uma *ação* recebe um *RDD* na entrada e produz um valor de tipo primitivo, por exemplo um inteiro para a ação de *count()*.

```
# Map transformando um RDD "data" com textos em linhas,
# separando por tabs e gerando um novo RDD "reviews"

reviews = data.map(lambda x: x.split("\tablename") )
```

Retomando o dataflow da figura 5.2, temos a seguinte divisão das operações do dataflow: Transformações = $< QT, FN, CMP, JOIN, FT >$, Ações: $< A_1, A_2, A_3, CAN, CQ >$

Spark processa *dataflows* através de uma estratégia conhecida como *avaliação tardia* (AT). Uma vez que precisa identificar os trechos do dataflow sujeitos a execução em uma mesmo fluxo, é necessário estabelecer as *fronteiras de localidade de dados*.

Definition 5.4.3. Uma fronteira de localidade de dados é um fragmento do dataflow que pode ser executado sem que haja necessidade de transferência de dados entre nós de um cluster. Em Spark, uma fronteira de localidade de dados define *estágios de processamento*.

Desta forma, o processo de *avaliação tardia* permite que otimizações sejam aplicadas sobre o fragmento do dataflow delimitado pela fronteira de localidade de dados. Um efeito secundário da AT é que transformações apenas são realizadas quando o sistema de avaliação encontra uma ação, fechando a fronteira relativa aquele fragmento.

5.4.2. Resilient Distributed Datasets (RDD)

O arcabouço *Spark* estendeu o conceito de *localidade de dados* proporcionada pelo *HDFS* de forma a otimizar a comunicação entre operações que estabeleçam um fluxo de processamento no modelo produtor-consumidor.

Um RDD é uma estrutura de dados distribuída. Recuperando o modelo de distribuição de dados do *HDFS*, o RDD segue uma estratégia semelhante com distribuição pela memória principal dos nós de um cluster. Assim, um RDD corresponde a um conjunto de objetos distribuídos e alocados em unidades de blocos pela área de memória disponível no cluster. Em Spark, blocos recebem o nome de *partições*, sendo essencialmente o mesmo conceito que blocos em HDFS, apesar de a princípio não serem persistentes. Adicionalmente, RDDs são não atualizáveis. A partir de um RDD podem-se aplicar transformações, gerando um novo RDD, ou ações, veja Figura 5.6. Estas últimas retornam o comando e os dados para o nós mestre.

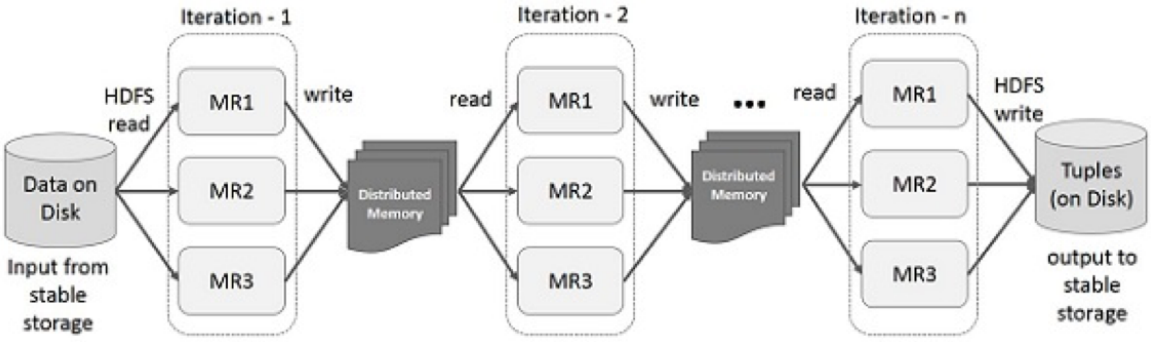

Figure 5.6. RDD em Memória

Transformações em Spark operam sobre RDDs, recebendo-os na entrada e produzindo-os na saída. RDD é um tipo de dado em Spark que além de sua estruturação em blocos contendo objetos, definidos em Java, Scala ou Python, itera sobre o conjunto de forma implícita. Assim uma transformação ou ação em *Spark* agindo sobre um RDD é invocada tantas vezes quantos forem os objetos alocados ao RDD. No código abaixo, um dataflow especifica um processo em que dois RDDs são juntados e, em seguida, o resultado é agrupado por uma chave. O RDD de agrupamento é filtrado por uma condição e o número de objetos no RDD filtrado é retornado. Notem a estrutura de fluxo de processamento explicitamente representados através do operador ".".

```
rdd1.join(rdd2)
     .groupby(..)
     .filter(...)
     .count()
```

Retomando a questão da avaliação tardia, dois efeitos imediatos deste modelo de execução sobre RDDs são (1) re-execução de um fragmento para reconstrução de uma partição de um RDD. Dado que RDDs são, a princípio, mantidos em memória, em um dataflow com mais de uma ação e que tenha uma dependência de dados entre RDDs usados no fragmento de uma ação e funções no fragmento de ações posteriores, Spark pode reconstruir o RDD inicial. Para isso, Spark mantém proveniência de construção associada a cada partição de um RDD (2).

No dataflow exemplo da figura 5.2, temos três ações: A_1, A_2, A_3. Considerando um escalonamento parcial: $< QT, FN, CMP, JOIN, FT, A1, CAN, CQ, A_2 >$, devido a AT,

o RDD produzido por FT será reconstruído para ser consumido por CQ quando da ação A_2, apesar de FT já ter sido construído quando da ação A_1.

5.4.3. Funções Spark

Como discutido, Spark oferece um conjunto de funções de segunda ordem como API para interface com as linguagens de programação Java, Python e Scala. As funções são ditas de segunda ordem pois invocam as implementações do usuário que fornecem o comportamento de primeira ordem a ser aplicado aos objetos dos RDDs. Assim, as funções de segunda ordem definem o modelo de transformação ou ação a ser aplicado enquanto a de primeira ordem fornece a implementação para tal operação.

Neste contexto a API Spark inclui as transformações conforme aparecem resumidamente na tabela 5.2:

Table 5.2. Transformações

Comando	Descrição
map(func)	transforma um RDD_i em um RDD_j
filter(func)	retorna um RDD em que os objetos atendem à condição imposta por *func*
flatmap(func)	produz um RDD em que o processamento de cada objeto do RDD de entrada pode gerar 0 ou mais objetos no RDD de saída
mappartition(func)	processa todos os objetos da partição do RDD na função *func*. Permite que se mantenha um estado de processamento entre o processamento de objetos do RDD de entrada.
union(RDD)	executa um operação binária em que o RDD de entrada é unido ao RDD passado como parâmetro.
groupbykey()	opera sobre um RDD do tipo (chave, valor) agregando pelos valores de uma mesma chave e produzindo um RDD <chave, <listavalores>

Igualmente, podemos listar as seguintes ações típicas, conforme a tabela 5.3.

Exemplo de código com a ação *Reduce* 5.7:

5.5. Algoritmos de Particionamento

Nesta seção discutiremos sobre algoritmos de particionamento de dados.

Definition 5.5.1. Dado um arquivo $D = \{d_1, d_2, ..., d_n\}$, onde d_i é um elemento do conjunto, e um cluster $N = \{n_1, n_2, ..., n_k\}$, onde n_j é um nó de um cluster, *Particionamento* é uma função $f : D \Rightarrow N$ que determina a alocação de um objeto em um nó de um cluster.

Como discutido em 5.4, os arcabouços MR se baseiam no critério de *localidade de dados* 5.4.1 como estratégia para processamento eficiente de grandes volumes de dados. O sistema HDFS 5.5, por exemplo, utiliza uma estratégia balanceada de distribuição de

Table 5.3. Ações em Spark - resumido

Comando	Descrição
reduce(func)	recebe uma função com dois parâmetros. O primeiro é um acumulador e o segundo é a variável cujo valor será acumulado durante a iteração pela partição
collect()	recupera todo o conteúdo do RDD nas diversas partições retornando ao mestre como uma lista
count()	retorna o número de entradas do RDD
take(n)	retorna uma lista com as *n* primeiras entradas do RDD
saveAsTextFile(path)	retorna um arquivo texto com o conteúdo do RDD no sistema de arquivo local ou HDFS

```
# reduce numbers 1 to 10 by adding them up
>>> x = sc.parallelize([1,2,3,4,5,6,7,8,9,10], 2)
>>> cSum = x.reduce(lambda accum, n: accum + n)
>>> print(cSum)
55

# reduce numbers 1 to 10 by multiplying them
>>> cMul = x.reduce(lambda accum, n: accum * n)
>>> print(cMul)
3628800

# by defining a lambda reduce function
>>> def cumulativeSum(accum, n):
...     return accum + n
...
>>> cSum = x.reduce(cumulativeSum)
>>> print(cSum)
55
```

Figure 5.7. Ação Reduce em Spark-Python

dados em que cada nó recebe blocos de mesmo tamanho. Da mesma forma, as partições

em memória utilizadas em *Spark* dividem os arquivos em unidades também de mesmo tamanho físico e as distribui, usando uma função de *hashing* como particionamento. Em [Oliveira et al., 2015], mostramos que quando o critério de particionamento atende à características de processamento dos dados, há um ganho significativo de processamento. Considere por exemplo, a aplicação exemplo 5.2. A formação de soluções se baseia em objetos localizados a uma certa distância uns dos outros, conforme estipulado pelo padrão geométrico fornecido. Neste contexto, não faz sentido buscar por vizinhos que estejam a uma distância além da maior distância no padrão geométrico sendo procurado. Essa restrição, que aparece em muitas aplicações com critérios de vizinhança, permite que os dados sejam particionados pelos nós de um cluster seguindo uma orientação de vizinhança, por exemplo, mantendo objetos com uma distância de até um ε em uma mesma partição, veja por exemplo a Figura 5.8.

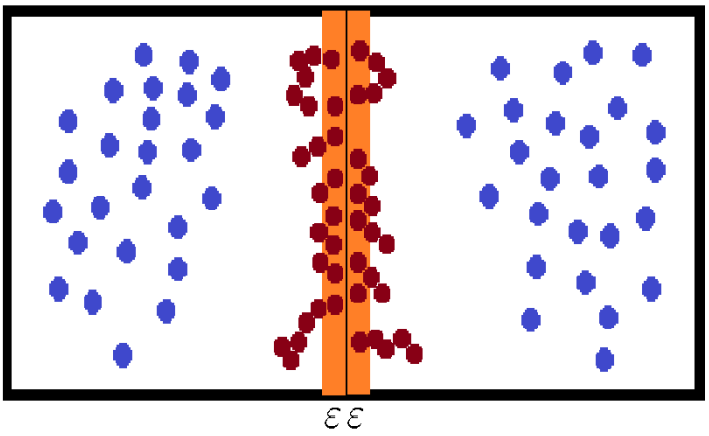

Figure 5.8. Fronteira em Partição

[7][Pires, 2016]

Na Figura 5.8, vê se uma região do espaço 2D, representada pela retângulo da figura com uma partição separando-a em duas regiões, não necessariamente de mesmo tamanho. Os elementos em azul são aqueles fora da zona fronteiriça. Já os elementos em roxo estão na zona de fronteira ou são efetados por ela. Neste caso, utilizarmos a fronteira delineada como critério espacial para o particionamento, precisaríamos incluir em cada região os objetos da região vizinha localizados na área de fronteira, (i.e. objetos roxos na figura 5.8).

5.5.1. O Algoritmo FRANCE

Em [Gaspar and Porto, 2014], propõe-se o *FRANCE*(FRAgmeNtador de Catálogos Espaciais, um algoritmo iterativo para particionar dados em histogramas *equi-depth*. Formalizamos o particionamento, segundo o FRANCE conforme a definição 1.

Definition 5.5.2. (Particionamento) Dado um objeto $o_{ji} < x, y, a_1, a_2, ..., a_t >$ tal que a_z, $1 \leq z \leq t$, é um atributo de o_{ji}, e $x \in X$ e $y \in Y$, tal que X e Y são as dimensões espaciais; um particionamento P é uma lista de valores $< v_1, v_2, ..., v_g >$ tal que $v_h \in x$ e $1 \leq h \leq g$. Desta forma, uma partição P_r, tal que $1 \leq r \leq g-2$ apresenta objetos vizinhos em uma região do espaço delimitada por P_r e P_{r+1}.

Considerando um contexto de dados espaciais 2D, o algoritmo calcula iterativamente os pontos de fragmentação segundo uma das dimensões, se preocupando em manter as partições com uma quantidade equivalente de objetos (com tolerância δ).

O *France* assume que se conhece o tamanho de arquivo e o número de partições desejado. Estas últimas são geradas iterativamente até que a quantidade de elementos seja próxima em δ do balanceamento perfeito. Consideramos perfeito, balanceamentos em que as partições apresentam n/g elementos, onde n é o total de elementos e g a quantidade de partições.

Primeiramente, divide-se, em uma das dimensões, a área espacial coberta pelo *dataset* em duas partições, cada uma cobrindo metade dos objetos. Uma variável *diff* é definida como metade do tamanho de cada partição. Para cada partição, enquanto não atingir um particionamento com n/g objetos (com uma tolerância de δ definida a priori), ela será reduzida por $diff$. Para cada passo, $diff = diff/2$. Quando a partição tiver a quantidade de objetos dentro do limite de δ, ela será fixada. O algoritmo executa recursivamente no espaço disponível entre as partições fixas.

O valor de δ é alterável no código e, tem como valor *default*, $0,5\%$ da quantidade de objetos desejada. Logo, todas as partições geradas terão $(n/g) \pm 0,005 \times (n/g)$ elementos. Com esse δ em 0,5% temos uma variação muito pequena no tamanho das partições.

O FRANCE gera como saída os valores da coordenada x onde ocorreram as divisões do espaço. A partir desses valores, podemos definir formalmente uma fronteira através da Definição 5.5.3.

Definition 5.5.3. Uma fronteira f_i definida segundo um valor de particionamento v_i, pertencente ao conjunto de partições P, na dimensão D, contem um conjunto de objetos O_{f_i} cujos valores da coordenada x estão entre $v_i - \varepsilon \leq x \leq v_i + \varepsilon$.

O pseudocódigo é apresentado no Algoritmo 1 e o seu passo a passo é ilustrado na Figura 5.9, onde os retângulos vermelhos correspondem às partições com uma quantidade de elementos maior que $n/x + \delta$ e que ainda precisam ser divididas para conter o número de objetos ideal. Os retângulos verdes representam as partições que contêm a quantidade de objetos dentro do limite de $n/x + \delta$ e não precisam mais ser subdivididas, ou seja, elas são partições fixadas. Para maiores detalhes, a implementação do FRANCE que fizemos em java está disponível à comunidade na internet[8].

5.5.2. Particionamento de grafos. *HDRF: Stream-Based Partitioning for Power-Law Graphs*.

Nos últimos anos, ao passo que ciência e tecnologia evoluem, nossa sociedade tem se deparado cada vez mais com a geração de grande quantidade de dados. Mais ainda, parte relevante desses dados é disposta estruturalmente como redes de larga escala. Além da grande dimensão que essas redes, representadas por grafos, possuem, sua presença se dá nos mais diversos domínios do conhecimento, tais como biologia, ciência da computação, química e sociologia [Lovász et al., 2009]. Nesse contexto, pode-se citar as redes sociais

[8] http://github.com/vinipires/FRANCE

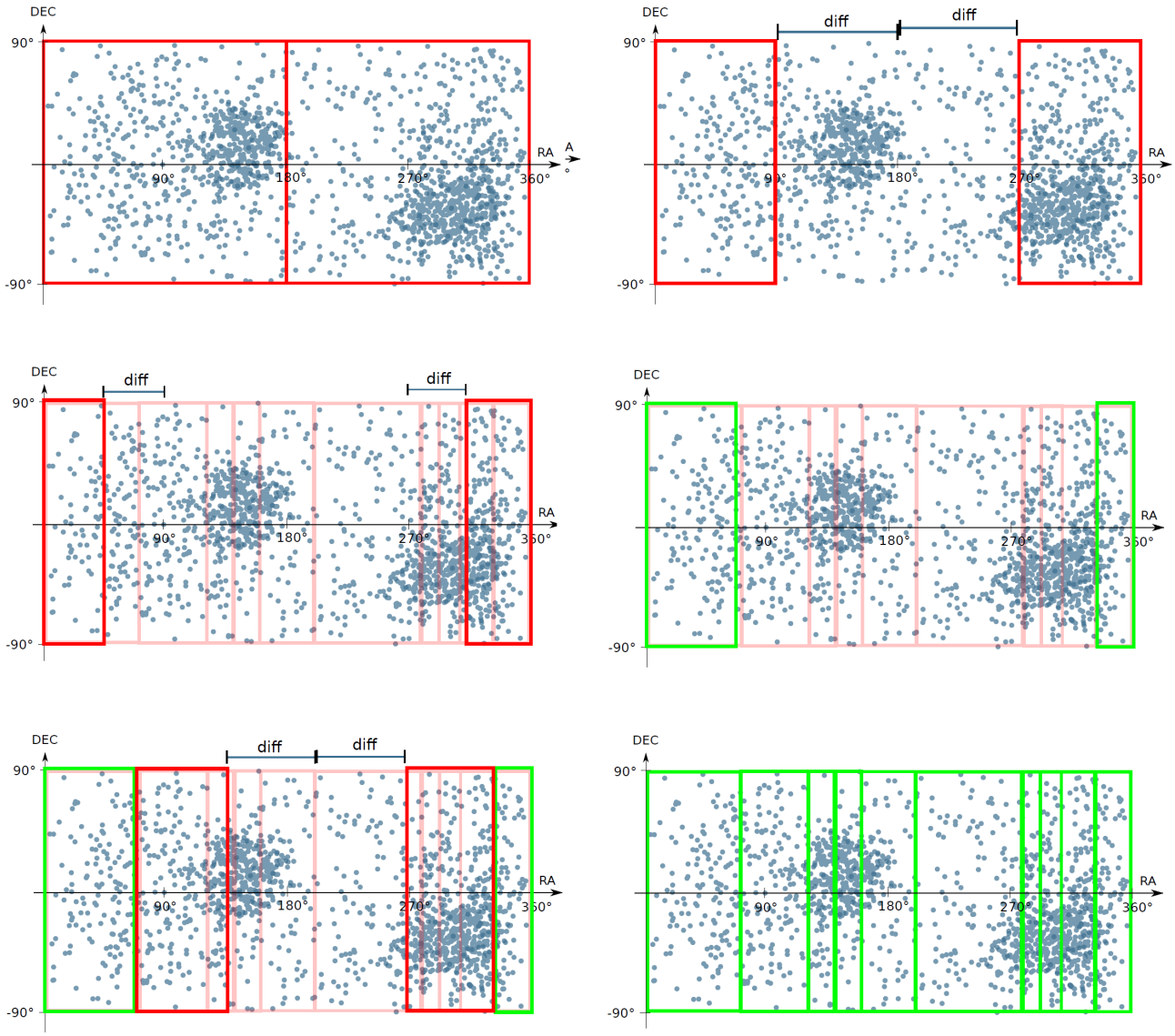

Figure 5.9. Passo a passo do FRANCE

online, como o *Facebook* que conta atualmente com cerca de 2 bilhões de usuários mensalmente ativos, cada qual com, em média, 155 ligações de amizade [Facebook, 2017]. Dada a relevância dessas redes, sua larga escala e usualmente padrões de conexão não triviais, suas análises demandam ferramental computacional compatível com o processamento em ambientes distribuídos e paralelos. Com isso, uma gama de plataformas de *software* projetadas para esse tipo de ambiente tem surgido recentemente [Guo et al., 2014].

Com a necessidade do uso de ambientes distribuídos e paralelos na análise de grafos de larga escala, um dos pré-requisitos para execução eficiente, dos algoritmos envolvidos é o particionamento dos grafos [Verma et al., 2017]. Intuitivamente, se quer particionar o grafo de modo que haja poucas arestas entre as partições, a fim de que o custo de comunicação seja reduzido. Aliás, ao processar um grafo em paralelo em k elementos

Algorithm 1 FRANCE: Algoritmo de Particionamento Espacial

Entrada 1 : *Arquivo de Entrada*
Entrada 2 : Quantidade de partições
function FRANCE(dataset,x) ▷ x é a quantidade partições desejada
    $objetopparticao \leftarrow \frac{|dataset|}{x}$
    startL=endL;startR=endR
    diffL=endL-startL
    diffR=endR=startR
    QTObjeL= computeQtdObj(StartL, endL)
    QTObjeR= computeQtdObj(endR, startR)
    for all $partition \in partitions$ **do**
        while $objetopparticao \notin QTObjeL \pm \delta \vee objetopparticao \notin QTObjeR \pm \delta$ objetos **do**
            Partição é reduzida/aumentada por $diff[L|R]$
            Update $diff[L|R]$
            Update endR, endL
        end while
        $Store\ part[partition] \leftarrow endL$
        $Store\ part[x - partition] \leftarrow endR$
    end for
    return part
end function
Saída: fronteiras;

de processamento ou em n nós computacionais, se deseja que o grafo seja particionado em k ou n partições de relativamente mesmo tamanho, de modo que seja minimizado o desbalanceamento de carga. Desse modo, haja vista a relevância das aplicações envolvendo redes de larga escala, assim como as dificuldades inerentes de seu particionamento pelo recurso computacional, técnicas destinadas à divisão de grafos em larga escala no âmbito de computação distribuída e paralela têm sido tópico de desenvolvimento e pesquisa. Nesse cenário, pode-se situar essas técnicas em duas classes [Gonzalez et al., 2012]: *corte em aresta* e *corte em vértice*, as quais são discutidas a seguir.

De modo geral, nas abordagens de corte em aresta, atribui-se os vértices às partições, enquanto as arestas são possivelmente estendidas pelas divisões. Enquanto isso, nas técnicas de corte em vértices, as arestas são atribuídas às partições, ao passo que os vértices são possivelmente estendidos entre as divisões. Vale atentar que em ambas estratégias, a extensão de arestas ou vértices pelas partições contribui no sobre-custo de armazenamento, comunicação e sincronização, posto que a informação da adjacência (corte em aresta) deve ser mantida em cada partição, e vértices devem ser replicados pelas divisões (corte em vértice). Como resultado de suas características, essas duas estratégias correlacionam-se com o grafo de modo distinto [Verma et al., 2017]. Técnicas de corte em aresta são preferíveis para grafos em que a maioria dos vértices seja de grau baixo, já que todas as arestas adjacentes a determinado vértice são alocadas na mesma partição. Entretanto, para grafos cuja distribuição de grau segue uma lei de potência [Barabási and Albert, 1999], nos quais estão presentes vértices com grau muito acima da média, estratégias baseadas em corte em vértice podem ser mais indicadas, uma vez que permitem melhor balanceamento de carga ao distribuir a carga desses vértices de grau elevado por várias partições.

Apesar da aplicabilidade das técnicas baseadas em corte em aresta, com a pro-

fusão de redes reais (grafos) cuja distribuição de grau segue uma lei de potência em diversas áreas do conhecimento [Barabási and Albert, 1999], abordagens de particionamento baseadas em corte em vértice vem ganhando notoriedade e sendo aplicadas em sistemas de processamento distribuído de grafos [Petroni et al., 2015]. Contudo, como esse tipo de estratégia lança mão à replicação de vértices, a sincronização entre as réplicas pode dificultar o desempenho computacional eficiente, haja vista a necessidade de coordenação e troca de dados entre os vértices replicados várias vezes no decorrer da execução dos algoritmos de análise. Portanto, em vista a essas dificuldades, uma estratégia eficiente no particionamento de grafos que se baseie em corte em vértice precisa resolver o problema de minimização *balanced k-way vertex cut*, isto é, além de manter a carga de trabalho por partição p o mais próximo possível do ótimo teórico $|E|/|P|$, deve-se minimizar o número de partições distintas $|A(v)|$ que cada vértice v foi atribuído:

$$\min \frac{1}{|V|} \sum_{v \in V} |A(v)| \quad \text{s.t} \quad \max_{p \in P} |p| < \sigma \frac{|E|}{|P|}$$

Diante do cenário exposto, [Petroni et al., 2015] propuseram um algoritmo guloso de particionamento de grafos com foco em *streaming - High degree (are) Replicated First* (HDRF) - para resolução do problema apresentado de *balanced k-way vertex cut*. Como o algoritmo HDRF é focado no processamento em *streaming*, sua entrada é um fluxo de arestas que forma o grafo a ser particionado, sendo realizada apenas uma passagem no grafo, isto é, dado que uma aresta já foi avaliada e atribuída a uma partição, o algoritmo posteriormente não altera de forma alguma essa atribuição. Além disso, HDRF, de forma gulosa, prioriza a replicação de vértices de grau relativamente mais alto. Como esses vértices de grau mais alto (*hubs*) servem muitas vezes como pontes entre subgrafos densos compostos por vértices de grau médio, ao priorizar a partição pelos *hubs* intrinsecamente almeja-se que as partições reflitam os agrupamentos observados no grafo sendo particionado. Desse modo, com base nas premissas apresentadas, o algoritmo funciona como a seguir.

Ao processar uma aresta e que conecta dois vértices v_i e v_j, inicialmente o algoritmo HDRF incrementa em um o grau parcial $\delta(v_i)$ e $\delta(v_j)$ desses vértices. Note que o grau parcial de um vértice é a quantidade de arestas a que ele pertence e que já foram processadas até o momento. Posteriormente, os graus de v_i e v_j são normalizados de modo que $\theta(v_i) = \delta(v_i)/[\delta(v_i) + \delta(v_j)] = 1 - \delta(v_j)$. Logo em seguida, o algoritmo HDRF calcula a pontuação C^{HDRF} para todas as partições $p \in P$, atribuindo portanto a aresta e a partição p^* que maximiza C^{HDRF}. Deve-se atentar que a pontuação é a combinação linear entre um fator de replicação, em que prioriza-se a cópia de *hubs* entre as partições, e um fator de balanceamento, no qual partições mais vazias são privilegiadas.

$$C^{HDRF}(v_i, v_j, p) = \overbrace{g(v_i, v_j, p)}^{\text{fator de replicação}} + \overbrace{\lambda \cdot \frac{\text{maxsize} - |p|}{\varepsilon + \text{maxsize} - \text{minsize}}}^{\text{fator de balanceamento}}$$
$$g(v_i, v_j, p) = p \in A(v_i) \cdot (2 - \theta(v_i)) + p \in A(v_j) \cdot (2 - \theta(v_j))$$

Vale notar ainda que alguns parâmetros devem ser informados ao algoritmo. O

tamanho máximo maxsize e mínimo minsize das partições. Um valor constante ε pequeno. O número de partições $|P|$. E por fim, o parâmetro λ, o qual permite o controle da influência do desbalanceamento de carga no tamanho da partição. Esse parâmetro é utilizado para lidar com problemas decorrentes de simetria e ordem na qual as arestas são processadas. O comportamento do algoritmo HDRF em relação a λ pode ser sintetizado nesses casos:

$$\begin{cases} \lambda = 0, & \text{agnóstico ao balanceamento de carga} \\ 0 < \lambda \leq 1, & \text{balanceamento utilizado para quebrar a simetria} \\ \lambda > 1, & \text{importância dada ao balanceamento proporcional a } \lambda \\ \lambda \to \infty & \text{atribuição aleatória das arestas} \end{cases}$$

Enfim, apesar da eficiência computacional demonstrada empiricamente em [Petroni et al., 2015], a necessidade de intervenção do usuário na parametrização de λ, a fim de lidar com particionamentos ineficientes decorrentes da ordem em que as arestas são processadas, pode ser considerada uma desvantagem e um tópico de investigação futura. Além disso, os autores deixam claro que implementações eficientes distribuídas e paralelas do algoritmo são trabalhos em aberto.

5.5.3. Uso do Algoritmo de Particionamento em Spark

Spark oferece dois métodos básicos de particionamento: *Hash* e *Range*.

A classe raiz de particionamento em Spark é a *Partitioner* que aparece especializada nas classes *HashPartitioner* e *RangePartitioner*. O *France* pode ser implementado sobre-escrevendo-se o método *getPartition(Object key)*.

5.6. Estruturas de Indexação

Nesta seção, trataremos do apoio ao processamento de grandes volumes através de estruturas de indexação. Em arcabouços MR, tal como o Spark, arquivos são distribuídos pelos nós de processamento, como vimos em 5.3.2. Apesar da aplicação típica envolver transformações aplicadas sobre RDDs, as funções de transformação podem se valer de uma estrutura de *lookup* para complementar a informação principal sendo extraída do RDD, ou ainda realizar a busca por vizinhos. Em nossa aplicação exemplo 5.2, a primeira tarefa constrói uma *Quadtree* [Samet, 2011] a ser usada para separação de estrelas em regiões e usar estas últimas como agregadores, veja seção 5.8. Cada região possui um centro geométrico se comportando como representante do conjunto de estrelas cobertos pelo seu quadrante espacial. Além disso, a estrutura pode responder a consultas buscando por vizinhos a uma distância indicada. De uma maneira geral, no processamento de grandes volumes de dados, estruturas de indexação podem ser utilizadas:

- como mecanismos de particionamento dos dados, permitindo paralelismo de processos. Este é o caso de índices baseados em *hashing* da chave de acesso.

5.6.1. Patricia-Hypercube Tree

Dados multidimensionais são comuns em várias aplicações, tais como simulações numéricas, sistemas de informação geográficas e na astronomia. Comumente, os dados multidimensionais se apresentam com um conjunto de dimensões espaço-temporais, embora possam existir outras dimensões dependendo do contexto da aplicação.

Existem diversas estruturas para indexação deste tipo de dados. Podemos citar como as mais predominantes, as R-Trees, KD-Trees e Quad-Trees. Porém, outra alternativa interessante, a PH-Tree (Patricia Hypercube Tree) foi apresentada mais recentemente [Zäschke et al., 2014]. A PH-Tree se baseia no conceito de árvores de prefixo binárias, que têm por objetivo reduzir o custo de armazenamento de dados. Neste tipo de árvore, fazemos uso do compartilhamento de prefixos comuns, diminuindo o tamanho total de armazenamento de dados multidimensionais. Além disso, a PH-Tree se baseia em cubos multidimensionais, permitindo assim uma navegação mais rápida pelos dados em comparação com outros tipos de árvores binárias.

A PH-Tree foi projetada com intuito inicial de servir como um método de indexação multidimensional para dados em memória. Embora, de acordo com os seus autores, no caso de conjuntos de dados com várias dimensões, a PH-Tree também seria adequada para indexação em disco, porque, neste caso os nós da árvore iriam conter uma quantidade grande o suficiente de registros para serem divididos de maneira eficiente em páginas do disco.

Uma PH-Tree é similar a uma Quadtree, embora utilize cubos multidimensionais e se baseie no conceito de compartilhamento de prefixos. A PH-Tree segue algumas filosofias caraterísticas. Ao contrário da abordagem da KD-Tree, em que apenas uma dimensão é particionada em cada nó, a PH-Tree segue o conceito implementado na Quatree, em que todas as dimensões são divididas em cada nó. Isto faz com que o acesso aos dados seja independente da ordem na qual as dimensões são armazenadas.

Outra característica da PH-Tree, é sua altura limitada e seu número de nós reduzido em comparação a outros tipos de árvores para dados multidimensionais. Isto ocorre porque cada nó da PH-Tree pode ter até 2^k filhos (sendo k o número de dimensões), e a altura máxima da árvore é igual ao número de bits do valor mais longo armazenado. Além disso, a PH-Tree é uma árvore inerentemente desbalanceada, e não requer algoritmos de rebalaceamento para operações de inserção e remoção de nós. Como a altura máxima da árvore é limitada, o problema da degeneração também fica limitado.

A estrutura interna da PH-Tree é determinada apenas pelo conjunto de dados presente nela. A ordem em que os dados são inseridos não é relevante na configuração final, isto é, se um mesmo conjunto de valores for inserido em uma ordem diferente, a estrutura final será a mesma. Operações de atualização (inserções e remoções) envolvem apenas a manipulação de no máximo dois nós, haja visto que a PH-Tree, pois nenhuma operação de balanceamento posterior é necessária.

A PH-Tree armazena entradas de dados, que são conjuntos de valores. Por exemplo, um ponto em 2D é armazenado como uma entrada com dois valores, cada um representando uma dimensão de um ponto. Os valores são armazenados em sua forma binária como uma cadeia de bits. Para árvores contendo dados com mais de uma dimensão, as

cadeias de bits são armazenadas em paralelo, como mostra a Fig. 5.10. No topo da árvore, temos um nó raiz com apenas uma única referência na segunda posição. A posição é calculada a partir do primeiro bit de cada um dos dois valores da entrada bi-dimensional. Usando esta abordagem, o vetor de posições com as referências se transforma em um cubo multidimensional. Abaixo do nó raiz, existe um prefixo para o sub-nó, consistindo de 0 para ambos os valores.

Cada nó da PH-Tree então contém um cubo multidimensional de referências. O tamanho desse vetor de referências é igual a 2^k. Cada posição desse vetor está relacionada ao endereço no cubo multidimensional, isto é, uma combinação de bits para cada dimensão. Associado a cada posição do vetor de referências, podemos ter um posfixo, indicando um nó folha. O posfixo é a parte final de um valor armazenado. Para nós intermediários, ao invés de um posfixo, temos um infixo e uma referência a outro cubo multidimensional/nó da PH-Tree.

Os bits entre um nó e seu nó pai são chamados de infixo. Chamamos de prefixo, todos os bits acima de um nó, formado pela concatenação dos infixos e endereços de cubos multidimensionais. Se concatenarmos o prefixo de um nó, com os bits do endereço do cubo multidimensional e o posfixo nós temos um conjunto de valores multidimensionais armazenados na arvóre.

Os cubos multidimensionais associados aos nós da PH-Tree podem ser esparsos, isso ocorre principalmente quando a árvore não tem entradas suficientes, com prefixos suficientemente variados. Neste caso podem-se adotar representações variadas para o vetor de referências baseado em um esquema de chave valor (chamado de linearized hypercube - LHC) ao invés do cubo multidimensional baseado em um vetor de referências (hypercube - HC). A Figura 5.10 mostra um exemplo de um PH-Tree de Exemplo contendo quatro entradas bi-dimensionais. São elas: (0001, 1000), (0011, 1000), (0011, 1010).

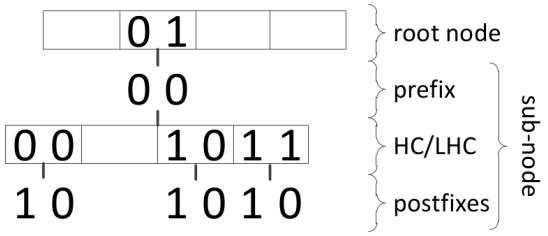

Figure 5.10. Exemplo de PH-Tree [1]

As PH-Trees suportam basicamente dois tipos de operações de consulta. As consultas pontuais em que buscamos pela existência de uma entrada específica. Esta busca é trivial, começando a partir do nó raiz, e através de comparações e buscas nos nós intermediários, é possível determinar a existência de uma entrada, avaliando apenas um subconjunto pequeno do número total de nós. De forma análoga é possível também executar as chamadas window queries, em que todos os valores dentro de uma janela multidimensional que determina um intervalo em cada dimensão devem ser retornados.

Testes comparativos mostraram que PH-Trees ocupam significativamente menos espaço que KD-Trees, e em alguns casos, até mesmo menos espaço que armazenamento

simples não indexado. Além disso, os autores demonstraram que a PH-Tree oferece um desempenho competitivo para atualizações, consultas pontuais e window queries. Sendo uma alternativa interessante para uma representação indexada dos dados com baixo custo de armazenamento.

5.6.2. Randomized KD-tree

O algoritmo KD-tree é uma popular estrutura de organização de dados multidimensionais, que tem por principal característica ser um algoritmo de simples implementação e com custo computacional pouco elevado, sendo utilizada em uma vasta gama de aplicações como processamento de imagens e localização espacial.

O procedimento de construção desta estrutura se inicia com a listagem das dimensões em uma dada sequência, em geral ordenada pelas dimensões de maior variância. Seguindo esta ordenação tem-se um processo de divisão dos dados, onde estes são fragmentados na mediana de uma dada dimensão dando origem a dois subconjuntos. A rotina então se repete de maneira recursiva para a próxima dimensão da lista ordenada, sendo particionado neste momento os subconjuntos gerados pelas divisões anteriores. Este processo se repete até que cada conjunto se remeta a um único elemento, tendo por fim uma árvore indicando as sucessivas divisões do espaço em dimensões alternadas. O Algoritmo 2 resume o processo.

Algorithm 2 Construção KD-tree(dataset $\in R^k$)

```
p=lista das dimensões ordenada pela variância
π(ROOT)=dataset
h ← 0
while árvore não construída do
    for i = 1...k do
        for cada nó x no nível h da árvore do
            medianNode=median(π(x),p(i))
            dividir π(x) em π(x_l) e π(x_r), com x_l(i) < medianNode e x_r ≥ medianNode, x_l ∈ π(x_l), x_r ∈ π(x_r)
        end for
        h++
    end for
end while
return KD-tree
```

Nesta estrutura a busca consiste em percorrer a árvore verificando as faixas de valores relativas a cada nó até se atingir um nó folha, conferindo se este registro atende a consulta especificada. Para caso de buscas por vizinhança é ainda necessário se verificar um subconjunto de nós adjacentes a folha encontrada, dado que este nó não é necessariamente o ponto mais próximo à coordenada da busca realizada.

Em se tratando destas buscas por vizinhança, o algoritmo pode apresentar algumas dificuldades em termos de desempenho, dado que a análise do subconjunto de pontos se trata de um procedimento recursivo e a quantidade de pontos verificada é proporcional ao número de dimensões do dado armazenado. Desta forma, a busca pode se tornar custosa quando se tratam de bases de elevada dimensão e um grande número de dados.

Para estas bases de dimensão elevada, uma possibilidade frequentemente utilizada para melhorar o desempenho das consultas é a utilização de buscas aproximadas. Em buscas aproximadas abre-se mão da garantia de fornecimento da resposta correta, buscando um ganho de desempenho ao se verificar um subconjunto dos pontos necessários que ainda permita certo grau de certeza da qualidade da resposta. Utilizando-se deste artifício há um *trade off* entre a qualidade da resposta fornecida e o custo computacional ao se realizar a busca, sendo uma decisão de projeto definir este melhor ajuste.

Neste contexto, a implementação de Randomized KD-trees [SilpaAnan and C., 2008] tem como objetivo aprimorar a precisão de buscas aproximadas para uma mesma quantidade de pontos verificados. Isto é feito com a construção de múltiplas KD-trees apresentando subdivisões distintas dos dados. Para criação destas árvores distintas são utilizadas diferentes ordens no processo de divisão dos dados pelas dimensões, sendo a ordem determinada por sucessivos sorteios em um subconjunto contendo as dimensões de maior variância ainda não selecionadas. O Algoritmo 3 destaca a mudança no processo de construção entre a KD-tree tradicional e uma RKD-tree.

Algorithm 3 Construção RKD-tree(dataset $\in R^k$)

```
1: vecVar(dataset)
2: for i = 1...k do
3:     p(i)=sorteio entre as D dimensões de maior variância não selecionadas
4: end for
5: RKD-tree ← Construir KD-tree com a sequência das divisões entre dimensões dada por p
6: return RKD-tree
```

A combinação de um conjunto de árvores RKD-tree permite a existência de múltiplas visões da distribuição dos dados, podendo a busca ser realizada em paralelo nestas árvores e de forma a ocorrer a verificação de subconjuntos distintos de pontos próximos à coordenada da consulta sem necessidade de um processo recursivo intensivo em cada uma destas individualmente.

Além de ser capaz de reduzir o número de nós analisados necessários para se obter determinada precisão da consulta, as RKD-trees ainda apresentam estrutura de processamento adequada a implementação de paralelismo e utilização em grandes bases de dados. Em termos de escalabilidade do processo, é possível distribuir as árvores em múltiplas unidades de processamento e realizar as buscas em paralelo, de forma a diminuir o tempo de processamento total de consulta as múltiplas árvores. Para as grandes bases de dados, outra possibilidade é a divisão dos dados entre múltiplas máquinas de forma que estes dados possam ser mantidos em memória nestas unidades de processamento. Desta forma, pode ser construída uma arquitetura com os dados particionados em múltiplos conjuntos de máquinas com cada máquina deste conjunto possuindo uma árvore de indexação para estes dados. Este processo permite que as consultas ocorram de maneira altamente paralelizável e que mesmo bases de dados com um número muito grande de elementos sejam mantidas constantemente em memória [Muja and Lowe, 2014].

Em termos de desempenho, esta abordagem se mostra competitiva tanto em tempo de processamento quanto em capacidade de paralelismo do processo de busca, apresentando desempenho superior ao da KD-tree tradicional e competindo com outras estruturas

utilizadas neste tipo de aplicação como o *Locality Sensitive Hashing* e a *Priority Search K-means Tree*.

5.7. Redução de Dimensionalidade

Desenvolvida em 1901, o PCA (Principal Component Analysis) é uma tradicional ferramenta de transformação para bases de dados, sendo utilizada em diversas áreas incluindo processamento de imagem, visualização de dados, recuperação de informação e redução de dimensionalidade.

O procedimento de execução busca uma transformação linear ortonormal dos dados que permita sua representação por meio de direções mais representativas deste conjunto. Em termos o PCA pode ser posto como a busca por uma transformação C, tendo:

$$X = YC$$

onde X é uma melhor representação da base de dados original Y. Neste sentido é pressuposto que as relações entre os atributos são lineares, guiando-se por aqueles de maior variância na análise. Outra característica desejada pela matriz X é que seus atributos possuam reduzida covariância, tendo seus valores o mais desacoplados possível [Shlens, 2003].

Dado o processo de construção é comum a utilização do PCA como técnica de redução da dimensionalidade dos dados, selecionando um subconjunto das componentes principais de maior variância e representando a base de dados apenas por este conjunto reduzido, com erro de representação relativo ao número de elementos selecionados.

Tendo em vista a definição generalista do procedimento PCA, foram desenvolvidas técnicas diversas para obtenção da matriz de transformação dos dados, podendo ser citadas dentre estas a decomposição da matriz de covariância e a decomposição por vetores singulares.

No contexto de *Big Data*, uma dificuldade que se impõe é a dimensão da base de dados, dado que os processos de construção da base modificada, envolvem, *a priori*, o processamento sobre todos os dados disponíveis, sendo relevante a escolha por uma técnica com maior capacidade de paralelização.

Neste sentido, a técnica sPCA [Elgamal et al., 2015] se apresenta como uma implementação do procedimento *Probabilistc PCA* voltado ao processamento em grandes bases de dados. O *Probabilistc PCA* é outro método de obtenção de aplicação do PCA que visualiza a obtenção da matriz de transformação como um processo inverso, onde dadas as múltiplas observações da base de dados, assume-se a existência de uma distribuição de probabilidade destes dados e busca-se, a partir da função de probabilidade advinda do processo, a estimativa de probabilidade máxima (MLE) da distribuição. A busca pela estimativa de probabilidade máxima é um procedimento recorrente em estatística, sendo possível se inspirar de algoritmos já existentes para resolução deste problema.

Em particular o sPCA, se utiliza de um procedimento iterativo que busca aproximações sucessivas para a matriz de transformação C, obtida pelo PCA originalmente. A técnica possui versões utilizando os principais frameworks para *Big Data*, *MapReduce*

e *Spark*, onde é buscada uma implementação com melhor desempenho tendo em vista a utilização em ambientes de arquitetura distribuída. O Algoritmo 4 mostra um esquema da implementação para o sPCA destacando em negrito as rotinas realizadas de maneira distribuída.

Algorithm 4 sPCA(dataset $\in R^k$)

1: $C = normrnd(D,d)$
2: $ss = normrnd(1,1)$
3: Y_m=**meanJob**(Y)
4: ss1=**FnormJob**(Y)
5: **while** condição de parada não satisfeita **do**
6: $M = C^T * C + ss * I$
7: $CM = C * M^{-1}$
8: $X_m = Y_m * CM$
9: $\{XtX, YtY\} =$ **YtXJob**(Y, Y_m, X_m, CM)
10: $XtX += ss * M^{-1}$
11: $C = YtY/XtX$
12: $ss2 =$ **trace**$(XtX * C^T * C)$
13: $ss3 =$ **ss3Job**(Y, Y_m, X_m, CM, C)
14: $ss = (ss1 + ss2 - 2 * ss3)/N/D$
15: **end while**
16: **return** KD-tree

Trata-se de um procedimento envolvendo diversas operações de produtos entre matrizes, mas havendo apenas alguns passos onde é necessário a paralelização das rotinas dada a dimensão das matrizes. Dentre as otimizações propostas no sPCA podem ser citados os seguintes passos: manutenção da esparsidade da base de dados através da propagação da media dos atributos da base de dados original, minimização da transferência de dados intermediários através do recômputo da matriz X, cálculo de operações de produto de matrizes de forma mais eficiente tomando proveito de uma possível esparsidade da base de dados original e das dimensões das matrizes envolvidas e o cômputo da norma de Froebenius de maneira paralelizável.

Verificando os testes de desempenho, as implementações em ambas as plataformas apresentaram desempenho competitivo, superando inclusive ao de ferramentas frequentemente utilizadas em projetos de *Big Data*, a implementação do método Stochastic SVD no Mahout-PCA (*MapReduce*) e a implementação da decomposição da matriz de covariância na biblioteca MLlib-PCA (*Spark*).

5.8. Clusterização

Uma das técnicas importantes na interpretação de grandes volumes de dados é o agrupamento de itens de dados semelhantes. Segundo [Berkhin, 2006], clusterização, ou agrupamento, é uma divisão de dados em grupos de objetos semelhantes. Cada grupo, chamado de cluster, é composto por objetos que são semelhantes entre si e muito diferente de objetos de outros grupos. A clusterização modela dados através de clusters. A modelagem de dados coloca a clusterização em uma perspectiva histórica enraizada na matemática, estatística e análise numérica. Do ponto de vista prático, o agrupamento desempenha um

papel de destaque em aplicações de mineração de dados, tais como a exploração científica de dados, recuperação de informação e de mineração de texto, aplicações de banco de dados espaciais, análise de Web, CRM, marketing, diagnósticos médicos, biologia computacional, e muitos outros.

Um exemplo simples de divisão de dados em grupos está disposto na Figura 5.11 e foi apresentado em [Matteucci, 2013]. Neste caso, identificam-se facilmente os 4 grupos em que os dados podem ser divididos; o critério de similaridade é a distância: dois ou mais objetos pertencem ao mesmo conjunto se eles são "próximos" de acordo com uma dada métrica de distância (neste caso, a distância geométrica). Esse tipo de agrupamento é chamado de clusterização baseada em distância.

Outro tipo de agrupamento é o agrupamento conceitual: dois ou mais objetos pertencem ao mesmo cluster se este define um conceito comum a todos os objetos. Em outras palavras, os objetos são agrupados de acordo com sua adequação aos conceitos descritivos, e não de acordo com as medidas de similaridade simples.

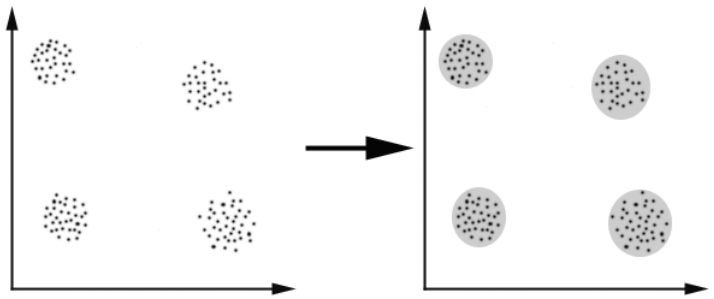

Figure 5.11. Exemplo simples de clusterização [Matteucci, 2013]

Segundo [Matteucci, 2013], não existe um "melhor" critério absoluto para decidir o que constitui um bom agrupamento que seja independente do objetivo final do agrupamento. Portanto, é o usuário que deve fornecer este critério, de tal maneira que o resultado do agrupamento atenderá às suas necessidades. Por exemplo, um usuário poderia estar interessado em encontrar representantes de grupos homogêneos (redução de dados), ou em buscar "agrupamentos naturais" e descrever suas propriedades desconhecidas (tipos de dados "naturais"), ou por buscar agrupamentos úteis e apropriados (classes de dados "úteis") ou ainda em buscar objetos de dados incomuns (detecção de outliers).

Quanto à classificação dos algoritmos de clusterização, [Berkhin, 2006] não a considera simples, pois as classificações se sobrepõem. São elas:

- métodos hieráquicos;
- métodos de particionamento;
- métodos baseados em grid;
- métodos baseados em co-ocorrência de dados categóricos;
- clusterização baseada em restrição;
- algoritmos de agrupamento usados em aprendizado de máquina;
- algoritmos de agrupamento escaláveis;

- algoritmos para dados de alta dimensionalidade.

No entanto, conforme [Berkhin, 2006], as técnicas de clusterização são tradicionalmente, amplamente divididas em hierárquicas e de particionamento. Cada uma deles se subdivide em várias diferentes técnicas. Enquanto algoritmos hierárquicos constroem clusters gradualmente, algoritmos de particionamento *aprendem* clusters diretamente. Ao fazer isso, eles tentam descobrir clusters iterativamente realocando pontos entre os subgrupos, ou tentam identificar os clusters como áreas altamente povoadas com dados. De acordo [Ochi et al., 2004], os algoritmos do primeiro tipo funcionam da seguinte forma: o conjunto de elementos é dividido em k subconjuntos, podendo k ser conhecido ou não, e cada configuração obtida é avaliada através de uma função-objetivo. Caso a avaliação da clusterização indique que a configuração não atende ao problema em questão, uma nova configuração é obtida realocando pontos entre os clusters, e o processo continua de forma iterativa até que algum critério de parada seja alcançado. Nesse esquema de realocação dos elementos entre os clusters, também conhecido como otimização iterativa [Ochi et al., 2004], os clusters podem ser melhorados gradativamente, o que não ocorre nos métodos hierárquicos. O k-médias, ou *k-means* [MacQueen, 1967] é um exemplo deste tipo de algoritmo.

[Berkhin, 2006] ainda comenta que algoritmos de particionamento do segundo tipo tentam descobrir componentes conexas densas de dados, estas são flexíveis quanto à sua forma. Um exemplo desse tipo de algoritmo bastante difundido é o DBSCAN [Ester et al., 1996a]. Esses algoritmos são menos sensíveis à *outliers* e podem descobrir conjuntos de formas irregulares. Eles geralmente trabalham com dados de baixa dimensionalidade de atributos numéricos, conhecidos como dados espaciais.

Já na clusterização hierárquica, conforme [Ochi et al., 2004], os clusters vão sendo formados gradativamente através de aglomerações ou divisões de elementos clusters, gerando uma hierarquia de clusters, normalmente representada através de uma estrutura em árvore, conforme exemplificado na Figura 5.12. Nessa classe de algoritmos, cada cluster com tamanho maior que 1 pode ser considerado como sendo composto por clusters menores.

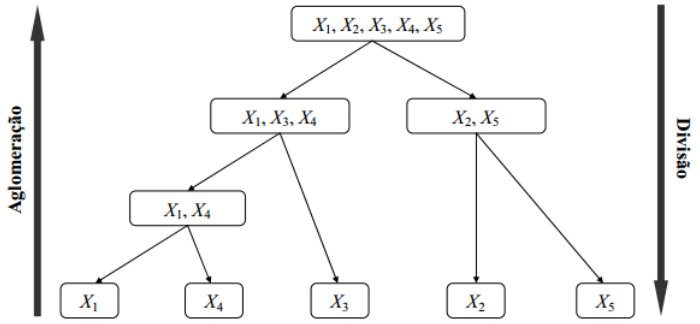

Figure 5.12. Exemplo de árvore de clusters na clusterização hierárquca [Ochi et al., 2004]

Nesses algoritmos existem dois tipos de abordagem: *bottom-up* e *top-down*. De acordo com [Ochi et al., 2004], nos algoritmos de aglomeração, que utilizam uma abor-

dagem *bottom-up*, cada elemento do conjunto é, inicialmente, associado a um cluster distinto, e novos clusters vão sendo formados pela união dos clusters existentes. Esta união ocorre de acordo com alguma medida que forneça a informação sobre quais deles estão mais próximos uns dos outros. Nos algoritmos de divisão, com uma abordagem top-down, inicialmente tem-se um único cluster contendo todos os elementos do conjunto e, a cada passo, são efetuadas divisões, formando novos clusters de tamanhos menores, conforme critérios pré-estabelecidos.

5.8.1. K-Means

K-means, ou k-médias, [MacQueen, 1967] é um algoritmo bastante difundido na literatura e um dos mais simples algoritmos de aprendizagem não supervisionada que resolvem o problema de clusterização. O objetivo é classificar um conjunto de elementos em um número k de clusters, dado como entrada. A idéia principal é definir *k centróides*, um para cada cluster, geralmente aleatórios, mas a melhor escolha é colocá-los longe um do outro. O próximo passo é levar cada ponto pertencente a um determinado conjunto de dados e associá-lo ao centróide mais próximo. Quando nenhum ponto está pendente, a primeira etapa é concluída e uma clusterização parcial é feita. Neste ponto é preciso voltar a calcular os k novos centróides dos clusters resultantes da etapa anterior. Logo após, uma re-alocação tem de ser feita entre os mesmos pontos e o novo centróide mais próximo. Esses passos são repetidos fazendo com que os k centróides mudem sua localização passo a passo até que não haja mais mudanças. Em outras palavras, os centróides não se movem mais [Matteucci, 2013].

O algoritmo visa minimizar uma função objetivo, neste caso uma função do erro quadrado. [Matteucci, 2013] descreve a função objetivo como sendo $J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^j - c_j||^2$, onde $||x_i^j - c_j||^2$ é a medida de distância escolhida entre um ponto x_i^j e o centróide do cluster c_j. Essa função é um indicador da distância dos n pontos e seus respectivos centróides.

Os passos do algoritmo são [Fontana and Naldi, 2009]:

1. Atribuem-se valores iniciais de centróides seguindo algum critério, por exemplo, sorteio aleatório desses valores dentro dos limites de domínio de cada atributo.
2. Atribui-se cada objeto ao cluster cujo centróide esteja mais próximo ao objeto.
3. Recalcula-se o valor do centróide de cada cluster, como sendo a média dos objetos atuais do cluster.
4. Repete-se os passos 2 e 3 até que os clusters se estabilizem.

Em [Fontana and Naldi, 2009], apresenta-se a Figura 5.13 ilustrando a execução do K-means. Nas Figuras 5.13 (a), 5.13 (b), 5.13 (c), mostra-se a execução dos passos 1, 2 e 3 respectivamente. Na Figura 5.13 (d) apresenta-se a repetição dos passos 2 e 3; e nas figuras 5.13 (e) e 5.13 (f) ilustra-se a repetição dos passos 2 e 3, respectivamente.

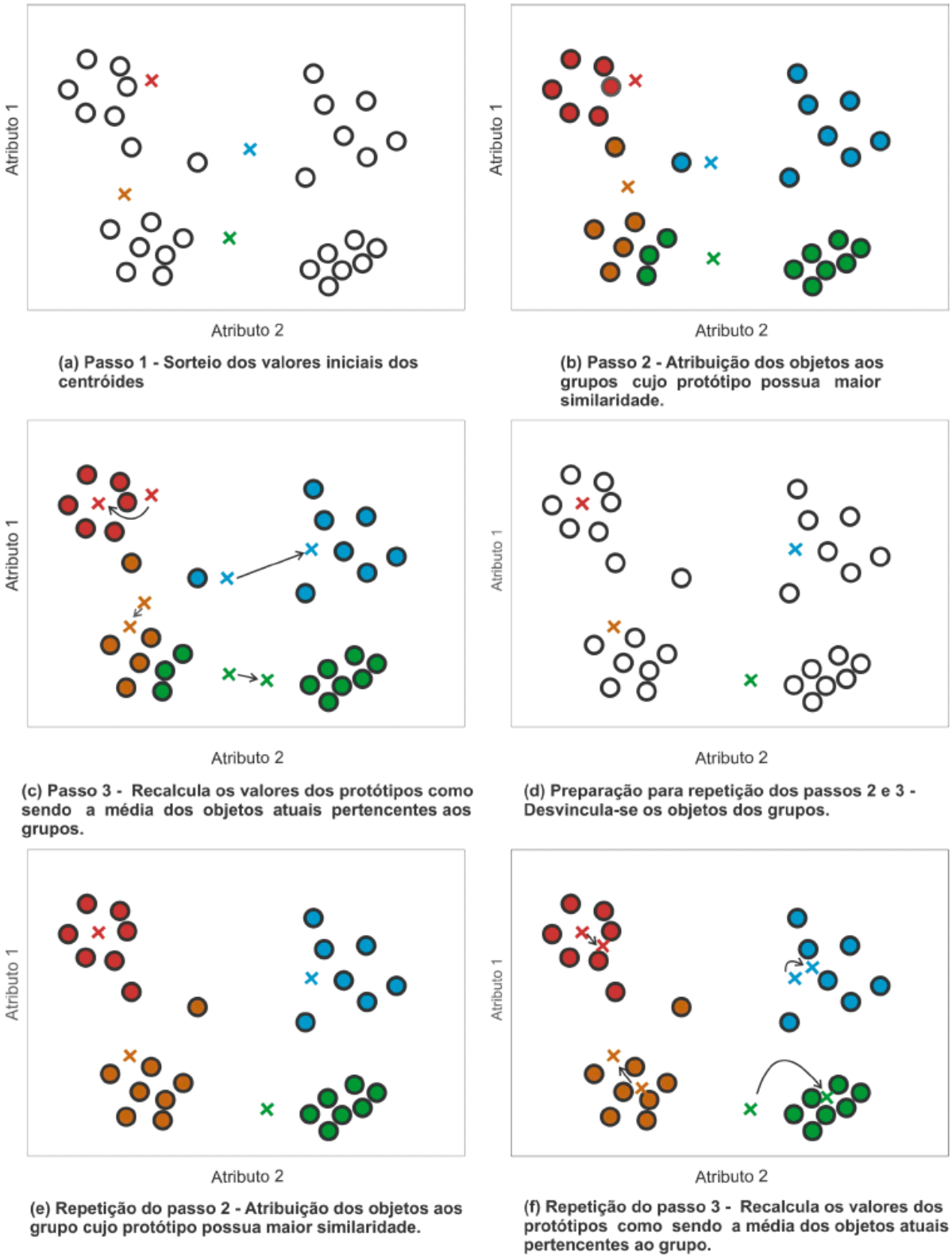

Figure 5.13. Exemplo de execução do k-means. Fonte: [Fontana and Naldi, 2009]

O K-means possui complexidade computacional equivalente a $O(n \times k \times i)$, onde

n é o número de objetos, k é o número de clusters e i o número de iterações. A distância entre os n objetos até cada um dos k centróides é calculada a cada iteração i. O número de dimensões também influência na complexidade do algoritmo, pois se o objeto tem d dimensões, a comparação entre dois desses objetos está em $O(d)$. Logo, para objetos que tenham 2 ou mais dimensões, a complexidade computacional do K-means passa a ser $O(n \times k \times i \times d)$.

5.8.2. Agrupamento de séries temporais. *YADING: Fast Clustering of Large-Scale Time Series Data*

O conceito de série temporal — uma sequência discreta de observações de algum fenômeno feita ao longo tempo — é utilizado em vários domínios do conhecimento, tais como, comunicação, finanças e economia [Shumway and Stoffer, 2010]. Ao passo que os intervalos de tempo entre as observações diminuem, assim como as características observadas aumentam, a escala e dimensionalidade das séries temporais compostas por essas observações demandam que novos algoritmos para tarefas de análise sejam desenvolvidos a fim de se obter eficiência e escalabilidade computacional. Aliás, em muitos cenários, o grau de eficiência computacional deve habilitar análises interativas em tempo real. Nesse contexto, o agrupamento de séries temporais é uma tarefa de análise relevante, o qual consiste em identificar, com base em algum critério de similaridade, grupos de instâncias (observações) homogêneas [Liao, 2005]. Por exemplo, em determinado *data center*, a cada intervalo de tempo predeterminado, podem ser medidas métricas de desempenho como percentual de uso de CPU e memória principal. Ao realizar o agrupamento das séries temporais formadas por essas medições ao longo do tempo, pode ser possível diagnosticar problemas de desempenho e compreender o status dos serviços disponíveis no *data center*.

Em face ao exposto, [Ding et al., 2015] propõem uma técnica de agrupamento que automaticamente congrega séries temporais de larga escala e dimensionalidade com altos resultados de desempenho computacional e qualidade: YADING. A técnica proposta consiste em três passos no agrupamento do conjunto de dados formado por séries temporais: (i) redução, (ii) agrupamento e (iii) atribuição do agrupamento. Em termos gerais, na etapa de redução, o conjunto de dados de entrada é amostrado e tem sua dimensionalidade reduzida. Já na etapa de agrupamento, o conjunto reduzido é agrupado. Na etapa final, o agrupamento previamente realizado é refletido no conjunto total dos dados. Os três passos da técnica YADING são discutidos a seguir, mas antes, para melhor compreensão, define-se formalmente o conceito de série temporal. Uma série temporal é um conjunto de dados $\mathbf{T}_{NxD} = \{T_1, T_2, ..., T_N\}$, onde N é o número de instâncias presentes no conjunto de dados, e D é a dimensão da série temporal, sendo cada instância $T_i = (t_{i1}, t_{i2}, ..., t_{iD})$.

Na primeira fase da técnica YADING, é realizada a redução do conjunto de entrada $\mathbf{T}_{NxD}$, isto é, identifica-se dois parâmetros $s, d \in \mathbf{N}$, onde $s \leq N$ é a quantidade de instâncias amostrada, enquanto $d \leq D$ é a quantidade reduzida de dimensões. Fundamentalmente a metodologia adotada na obtenção de s e d visa preservar a distribuição adjacente a $\mathbf{T}_{NxD}$ no conjunto de dados reduzido $\mathbf{T}_{sxd}$. Assim, para se obter s, assume-se que todo $T_i \in \mathbf{T}_{NxD}$ pertença a k grupos conhecidos. Partindo desse princípio, demonstra-se analiticamente que para $s = 2000$, isto é, ao serem escolhidos aleatoriamente duas mil instâncias de séries temporais, com 95% de confiança, consegue-se acurácia no agrupa-

mento próximo ao que seria realizado no conjunto total. É válido atentar para a relevância desse resultado: s independe de N. Além do processo de amostragem aleatória de T_i lançando mão a s, o método *Piecewise Aggregate Approximation of time series* (PAA) é utilizado na redução computacionalmente eficiente de dimensionalidade de $\mathbf{T}_{NxD}$. Apesar de computacionalmente eficiente, o método PAA demanda o parâmetro d, isto é, para quantas dimensões será reduzido $\mathbf{T}_{NxD}$. Nesse sentido, a técnica YADING utiliza uma abordagem em que, primeiramente são encontradas as frequências típicas de cada instância, sendo elas posteriormente ordenadas em forma de distribuição. A partir dessa distribuição seleciona-se um valor de percentil acima de 80%, o qual é empregado como d. Em conclusão, o tempo total para realizar o processo de redução de dados é de $\mathcal{O}(sD\log(D)+ND)$. Vale observar que o maior custo computacional é referente a aplicação do método PAA ($\mathcal{O}(ND)$) haja vista a necessidade de reduzir a dimensionalidade de cada instância do conjunto de dados.

Na segunda fase da técnica YADING, o conjunto amostrado de séries temporais é agrupado. Nesse cenário, deve-se atentar que os dados de séries temporais podem apresentar formas e densidades variadas, assim como perturbação de fase e ruído aleatório. Essas características possivelmente impactam na acurácia do agrupamento. Com isso, no intuito de garantir robustez no agrupamento dos dados assim como eficiência computacional, a técnica YADING emprega uma metodologia em que combina-se a métrica computacionalmente eficiente de distância L_1 e um método de agrupamento multi-densidade, sendo as estimativas de densidades ponto chave na técnica. Para realizar essas estimativas, primeiramente, para cada instância de série temporal em $T_i \in \mathbf{T}_{sxd}$ computa-se sua distância L_1 em relação a seus k vizinhos mais próximos, sendo esses valores listados em ordem decrescente em uma curva chamada de $kdis$. Os raios de densidade são definidos como os pontos de inflexão na curva $kdis$, que encontrados, são então empregados juntamente com o parâmetro $minPoints = 4$ pelo algoritmo DBSCAN [Ester et al., 1996b] no agrupamento. A propósito, é importante notar que se uma instância T_i foi agrupada com um raio de densidade, ela não é considerada para os demais raios. Enfim, o tempo total para realizar o processo de agrupamento é de $\mathcal{O}(s^2\log(s))$. Perceba que seria muito custoso realizar o agrupamento no conjunto de dados total.

Na última fase da técnica YADING o agrupamento realizado é refletido ao conjunto de dados total, isto é, os rótulos de grupo definidos para $\mathbf{T}_{sxd}$ são estendidos para $\mathbf{T}_{NxD}$. Com esse objetivo, a técnica YADING, para cada instância não rotulada T_i encontra a rotulada mais próxima T_j. Caso a distância entre T_i e T_j seja inferior ao raio de densidade utilizado no agrupamento da instância rotulada T_j, T_i recebe o rótulo de T_j. Caso contrário, T_j é considerada como ruído. A fim de evitar que seja computada a distância entre cada instância não rotulada e cada instância rotulada adota-se uma estratégia de poda que se baseia no fato de que se a distância entre uma instância não rotulada T_i e uma rotulada T_j é superior ao raio de densidade r utilizado no agrupamento de T_j, pela desigualdade triangular, tem-se que T_i está também distante mais do que r dos vizinhos rotulados de T_j. Assim, pode-se economizar a computação de distância entre T_i e os vizinhos de T_j. Nesse cenário, é construída uma estrutura chamada de *Sorted Neighbor Graph* (SNG), onde para cada ponto de núcleo definido na fase de agrupamento é armazenada suas distâncias em ordem ascendente para os outros elementos do conjunto de dados amostrado. Empiricamente, percebeu-se que o uso de SNG apresenta ganhos de

desempenho entre 2-4 vezes na prática. Por fim, tem-se que o tempo computacional da fase de atribuição é da ordem de $\mathcal{O}(Nsd)$.

Enfim, a metodologia de ponta a ponta no agrupamento de séries temporais empregada por YADING demonstra-se computacionalmente eficiente e provê resultados de qualidade. Mais especificamente, a independência do tamanho de amostragem, e a metodologia proposta de agrupamento multi-densidade são contribuições relevantes da técnica. Tem-se ainda como ponto positivo complexidade aproximadamente linear da técnica em relação à dimensão e ao tamanho de $\mathbf{T}_{NxD}$. Por fim, os trabalhos futuros vão da adaptação da técnica para conjuntos de dados compostos por séries temporais de escala e dimensão diversa, assim como, o uso da metodologia de YADING no congregação de séries temporais em *streaming*. Além desses pontos, [Mirzanurov et al., 2016] propõem melhorias concernentes à estrutura SNG e à acurácia da técnica.

5.9. Comentários Finais

A análise e interpretação de grandes volumes de dados é uma das atividades mais importantes na nova economia digital em que o bem de maior valor é a informação. Neste sentido, academia e as empresas redirecionam suas atividades de forma a prover técnicas, ferramentas e ambientes computacionais para fazer face aos desafios dessa nova ciência. Em particular, rotulou-se as atividade nessa área numa nova área denominada Ciência de Dados que integra: ciência da computação, estatística e matemática. Neste contexto, os arcabouços MR tomam importância significativa por aliarem propriedades importantes, tais como: escalabilidade, tolerância à falhas, extensibilidade para diferentes aplicações e flexibilidade com relação ao ambiente computacional. Apesar dessas qualidade, o desenvolvimento de aplicações neste novo contexto requer a reavaliação de algoritmos e sua contextualização sob os desafios de grandes volumes de dados. Neste curso, apresentamos algumas classes de algoritmos importantes para o tratamento de aplicações em grandes volumes. Algoritmos de particionamento de dados adicionam critérios semânticos para a distribuição dos dados pelas nós de processamento. Aspectos, por exemplo como o de vizinhança são beneficiados por particionamentos que os levem em consideração. Adicionalmente, técnicas de indexação eficiente permitem ações de busca por chave ou mesmo de forma espacial adaptadas ao contexto dos grandes volumes. Finalmente, agregar dados segundo critérios de semelhança permite reduzir a dimensionalidade e usar representantes em análise sobre a massa de dados. Obviamente, a lista de algoritmos não para por ai. Podemos investigar técnicas de dimensão de dimensionalidade, amostragem de dados, além de predição. Esse grande conjunto de técnicas forma essa nova disciplina de processamento de grandes volumes de dados.

References

[Barabási and Albert, 1999] Barabási, A. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

[Berkhin, 2006] Berkhin, P. (2006). A Survey of Clustering Data Mining Techniques. *Grouping Multidimensional Data*, pages 25–71.

[Dean and Ghemawat, 2008] Dean, J. and Ghemawat, S. (2008). Mapreduce: Simplified

data processing on large clusters. *Commun. ACM*, 51(1):107–113.

[Ding et al., 2015] Ding, R., Wang, Q., Dang, Y., Fu, Q., Zhang, H., and Zhang, D. (2015). Yading: fast clustering of large-scale time series data. *Proceedings of the VLDB Endowment*, 8(5):473–484.

[Economist, 2010] Economist, T. (2010). The data deluge. *The Economist*.

[Eigenbrod et al., 2008] Eigenbrod, A., Courbin, F., Sluse, D., Meylan, G., and Agol, E. (2008). Microlensing variability in the gravitationally lensed quasar qso 2237+0305 ≡ the einstein cross. i. spectrophotometric monitoring with the vlt. *Astronomy & Astrophysics*, 480(3):647–661.

[Elgamal et al., 2015] Elgamal, T., Yabandeh, M., Aboulnaga, A., Mustafa, W., and Hefeeda, M. (2015). spca: Scalable principal component analysis for big data on distributed platforms. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 79–91, New York, NY, USA. ACM.

[Ester et al., 1996a] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996a). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

[Ester et al., 1996b] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996b). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

[Facebook, 2017] Facebook (2017). Company info.

[Fontana and Naldi, 2009] Fontana, A. and Naldi, M. C. (2009). Estudo e comparação de métodos para estimação de números de grupos em problemas de agrupamento de dados. ICMC.

[Gaspar and Porto, 2014] Gaspar, D. and Porto, F. (2014). A multi-dimensional equi-depth partitioning strategy for astronomy catalog data.

[Ghemawat et al., 2003] Ghemawat, S., Gobioff, H., and Leung, S.-T. (2003). The google file system. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, pages 29–43, New York, NY, USA. ACM.

[Gonzalez et al., 2012] Gonzalez, J., Low, Y., Gu, H., Bickson, D., and Guestrin, C. (2012). Powergraph: Distributed graph-parallel computation on natural graphs. In *Proc. USENIX Symp. on Operating Systems Design and Implementation (OSDI)*, pages 17–30.

[Guo et al., 2014] Guo, Y., Biczak, M., Varbanescu, A., Iosup, A., Martella, C., and Willke, T. (2014). How well do graph-processing platforms perform? an empirical performance evaluation and analysis. In *Proc. IEEE Int. Parallel and Distributed Processing Symp. (IPDS)*, pages 395–404.

[Liao, 2005] Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874.

[Lovász et al., 2009] Lovász, L. et al. (2009). Very large graphs. *Current Developments in Math.*, 2008:67–128.

[Ludäscher et al., 2006] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., and Zhao, Y. (2006). Scientific workflow management and the kepler system: Research articles. *Concurr. Comput. : Pract. Exper.*, 18(10):1039–1065.

[MacQueen, 1967] MacQueen, J. B. (1967). Some Methods for Classification and Analysis of MultiVariate Observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

[Matteucci, 2013] Matteucci, M. (2013). A Tutorial on Clustering Algorithms.

[Mirzanurov et al., 2016] Mirzanurov, D., Nawaz, W., Lee, J., and Qu, Q. (2016). An effective cluster assignment strategy for large time series data. In *International Conference on Web-Age Information Management*, pages 330–341. Springer.

[Moore, 2000] Moore, G. E. (2000). Readings in computer architecture. chapter Cramming More Components Onto Integrated Circuits, pages 56–59. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Muja and Lowe, 2014] Muja, M. and Lowe, D. (2014). Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):107–113.

[Nature, 2008] Nature (2008). Big data welcome to the petacentre. *Nature*, 455.

[Ochi et al., 2004] Ochi, L. S., Dias, C. R., and Soares, S. S. F. (2004). *Clusterização em Mineração de Dados*.

[Ogasawara et al., 2013] Ogasawara, E., Dias, J., Silva, V., Chirigati, F., de Oliveira, D., Porto, F., Valduriez, P., and Mattoso, M. (2013). Chiron: a parallel engine for algebraic scientific workflows. *Concurrency and Computation: Practice and Experience*, 25(16):2327–2341.

[Oliveira et al., 2015] Oliveira, D., Boeres, C., Fausti, A., and Porto, F. (2015). Avaliação da localidade de dados intermediários na execução paralela de workflows bigdata. In *XXX Simpósio Brasileiro de Banco de Dados, SBBD 2015, Petrópolis, Rio de Janeiro, Brasil, October 13-16, 2015.*, pages 29–40.

[Ozsu and Valduriez, 1990] Ozsu, T. and Valduriez, P. (1990). *Principles of Distributed Databases*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Petroni et al., 2015] Petroni, F., Querzoni, L., Daudjee, K., Kamali, S., and Iacoboni, G. (2015). Hdrf: Stream-based partitioning for power-law graphs. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 243–252, New York, NY, USA. ACM.

[Pires, 2016] Pires, V. (2016). *NACLUSTER: Resolvendo Entidades em Larga Escala a partir de Multiplos Catalogos da Astronomia*. PhD thesis, Universidade Federal do Ceará.

[Porto et al., 2017] Porto, F., Khatibi, A., Nobre, J. R., Ogasawara, E., Valduriez, P., and Shasha, D. (2017). Constellation Queries over Big Data. *ArXiv e-prints*.

[Samet, 2011] Samet, H. (2011). *The Design and Analysis of Spatial Data Structures*. Springer New York.

[SDSS,] SDSS. Sloan digital sky survey.

[Shlens, 2003] Shlens, J. (2003). A tutorial on principal component analysis: derivation, discussion and singular value decomposition.

[Shumway and Stoffer, 2010] Shumway, R. H. and Stoffer, D. S. (2010). *Time series analysis and its applications: with R examples*. Springer Science & Business Media.

[SilpaAnan and C., 2008] SilpaAnan and C., Hartley, R. (2008). Optimised kd-trees for fast image descriptor matching. In *26th IEEE Conference on Computer Vision and Pattern Recognition*.

[Verma et al., 2017] Verma, S., Leslie, L. M., Shin, Y., and Gupta, I. (2017). An experimental comparison of partitioning strategies in distributed graph processing. *Proc. VLDB Endow.*, 10(5):493–504.

[Zaharia et al., 2010] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA. USENIX Association.

[Zäschke et al., 2014] Zäschke, T., Zimmerli, C., and Norrie, M. C. (2014). The ph-tree: A space-efficient storage structure and multi-dimensional index. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 397–408, New York, NY, USA. ACM.

Chapter
6

Deep Learning - Teoria e Prática

Cristina Nader Vasconcelos, Esteban Walter Gonzalez Clua

Abstract

Deep Learning is bringing a major IT revolution in recent years, opening up new horizons and possibilities for many different areas and applications. Thanks to the investment of large companies such as NVIDIA, Google, Microsoft, IBM, among others, many tools and frameworks are becoming accessible for the friendly application of Deep Learning in solving many different problems. This short course will present basic concepts of neural networks and deep neural networks, an overview of some libraries and tools and a brief introduction to GPU architectures and how they support the area.

Resumo

Redes Neurais Profundas vêm provocando uma grande revolução na indústria de TI nos últimos anos, abrindo diversos horizontes e possibilidades para as mais variadas áreas e aplicações. Graças ao investimento de grandes empresas, como NVIDIA, Google, Microsoft, IBM, dentre outras, inúmeras ferramentas e plataformas vêm se tornando acessíveis para aplicar Deep Learning em soluções de diversos problemas. Neste mini-curso serão apresentados conceitos básicos de redes neurais profundas, algumas bibliotecas e ferramentas, arquiteturas de GPUs e como as mesmas são capazes de viabilizar a área.

6.1. Introdução

A Inteligência Artificial (IA) tem sido há longo tempo almejada e buscada pelos humanos. O HAL (o computador de bordo da *Discovery One*, do filme "2001 - Uma Odisséia no Espaço"), a Skynet (sistema de IA do filme Exterminador do Futuro) ou a Matrix (do filme homônimo) nos fascinam e ao mesmo tempo nos assustam com uma visão do que pode ser um futuro controlado pelas máquinas. Embora pesquisas na área de IA venham sendo desenvolvidas há décadas, nos últimos anos experimentamos uma verdadeira revolução, levada adiante por grandes empresas e com

investimentos bilionários: NVIDIA, Google, IBM, Microsoft, Amazon, Facebook, Baidu, entre diversas outras. Em alguns casos, o tema de IA não se trata apenas de uma das muitas áreas de atuação destas empresas, mas vem se tornando a mais importante em seu direcionamento.

Apesar de "tarefas inteligentes" exigirem diversos graus de atuação e de não nos aprofundarmos na comparação com a inteligência humana neste texto, estamos aqui interessados em mecanismos que possibilitem a resolução de tarefas pelo aprendizado dos padrões que compõem o problema a ser resolvido: se queremos preparar uma receita, precisamos reconhecer os ingredientes, as suas quantidades e inclusive onde se encontram na dispensa; se queremos atravessar uma rua, temos que reconhecer a sinalização do local, os carros que estão nas vias e o local correto para a travessia. Podemos observar que em diferentes tarefas faz-se necessário como estágio fundamental o reconhecimento de padrões que regem a tomada de decisão entre a observação do contexto em que queremos resolvê-la (fornecido por um sinal de entrada) até a produção da saída desejada.

Em alto nível, podemos dizer que a revolução da IA que estamos vivendo se dá pelo desenvolvimento de sistemas com capacidade de aprender padrões intrínsecos que regem uma determinada tarefa pela observação de grandes volumes de dados e da sua capacidade de generalização dos modelos aprendidos nesses sistemas para inferir a resposta esperada em novos casos.

Através da análise de padrões, os sistemas baseados em técnicas de *Deep Learning* são capazes de reconhecer, traduzir, sintetizar e até prever sinais das mais diferentes naturezas. Têm sido usadas na analise de sinais visuais como imagens e vídeos, sinais de áudio, de linguagem natural, entre outros tipos de sinais e além da combinação de sinais de natureza distintas.

Inúmeras são as aplicações nas quais as técnicas de *Deep Learning* já são utilizadas com sucesso, obtendo resultados impensáveis de serem obtidos em curto prazo até recentemente. A listagem a seguir inclui apenas algumas delas, como forma de inspiração ao leitor. Técnicas de *Deep Learning* têm sido usadas para:

- interpretação de caracteres em diferentes alfabetos e tipografias a partir de imagens [LeCun et al. 1989, Ciresan et al. 2011a, Ciresan et al. 2012];
- reconhecimento de fala [Chan et al. 2016], verificação e identificação do orador [Heigold et al. 2016];
- tradução entre diferentes línguas [Britz et al. 2017] e geração de linguagem natural modelando explicitamente propriedades como estilo e assunto [Bowman et al. 2016];
- produzir respostas a e-mails [Kannan et al. 2016] e ainda construir agentes capazes de interagir em linguagem natural [Shao et al. 2017];
- detecção de pedestres em imagens [Sermanet et al. 2013], classificação de poses das mãos [Tompson et al. 2014], e ainda estimar pose do corpo em imagens com múltiplas pessoas [Papandreou et al. 2017];

- segmentação e rotulação em tempo real de cenas pela análise de imagens de profundidade [Couprie et al. 2013] e também análises especializadas em diversos outros tipos de objetos como reconhecimento de sinais de trânsito [Ciresan et al. 2011b] e de números de imóveis [Sermanet et al. 2012] além de poderem identificar múltiplos objetos, contá-los e localizá-los [Eslami et al. 2016];

- controlar veículos autônomos e robôs seja sua navegação [Gupta et al. 2017, Giusti et al. 2016] ou seus mecanismos de interação com o ambiente [Levine et al. 2016];

- detecção de mitose em imagens histológicas de câncer de mama [Ciresan et al. 2013], detecção de retinopatia diabética em fotografias de fundo de retina [Gulshan et al. 2016], classificação de lesões de pele [Esteva et al. 2017], entre diversas outras aplicações biomédicas;

- produzir imagens com super resolução [Dahl et al. 2017] ou ainda sintetizar imagens foto-realistas [Odena et al. 2016];

- simular em tempo real fluidos e fumaça [Tompson et al. 2016] bem como criar uma reconstrução em 3D a partir de uma visualização 2D de objetos [Yan et al. 2016];

- prever a continuidade em sequências de vídeo [Villegas et al. 2017];

- aprender e aplicar estilos artísticos seja na criação de imagens [Dumoulin et al. 2017] ou na geração de música [Jaques et al. 2016];

- jogar videogames [Mnih et al. 2013] e jogos de tabuleiro [Silver et al. 2016];

- encontrar provas de teoremas de primeira ordem [Loos et al. 2017];

- produzir computadores capazes de aprender sua própria programação, como uma Máquina de Turing Neural [Kaiser and Sutskever 2016].

As redes neurais são as protagonistas nesta revolução produzida pela aprendizagem de máquina. A modelagem tradicional de sistemas com aprendizagem de máquina consiste em duas fases. Primeiramente são extraídos a partir dos dados brutos (uma imagem, por exemplo) que compõe o sinal de entrada um conjunto de medições que caracterizem as propriedades que se deseja avaliar do problema (chamada etapa de extração de características). A partir das medições uma amostra que se deseja avaliar passa a ser representada tipicamente por um vetor descritor que representa a assinatura daquela amostra segundo as medições estabelecidas. Na sequência é aplicado um algoritmo de aprendizagem de máquina para a tomada de decisão sobre o conjunto de medições levantado.

O grande desafio nesta abordagem clássica é que a etapa de extração de características não é trivial, já que nela estão subentendidos conhecimentos especialistas. O que devemos observar na análise de uma imagem médica para fazer um diagnóstico? O que devemos observar de um áudio para identificar o orador? O que devemos observar de um texto para verificar a sua autoria?

Além disso, mesmo em casos em que são conhecidos os critérios especialistas adotados em uma determinada análise, encontrar uma formalização matemática que suporte as medições desejadas sobre o elemento em questão com todas as suas possíveis variações válidas também não é trivial. Em grande parte dos problemas de análise de sinais biológicos é comum que existam uma série de variações válidas e frequentes que o tornam ainda mais difíceis de serem modelados. Supondo que conseguimos modelar a forma 3D de um determinado cachorro para responder a pergunta: essa foto contém um cachorro? E se alterarmos a iluminação, continua sendo um cachorro? E o cenário, pose, escala com que o cachorro aparece na imagem, influenciam nessa decisão? Como deve ainda ser considerado um cachorro, tais invariâncias devem ser incorporadas ao modelo matemático.

Esse problema se estende pra outros tipos de sinais e tarefas. Ao analisar áudio poderíamos indagar sobre sotaque, intonação, velocidade da fala, ruído ambiente, taxa de captura, entre outras variações. De acordo com o sinal observado e o problema que se deseja resolver existem outras tantas variações a serem incorporadas ao modelo matemático de suporte a sistemas para tratamento de sinais biológicos, podendo até por vezes não serem explicitamente conhecidas dada a complexidade da tarefa ou do sinal.

Nas abordagens conhecidas como *Deep Learning* para solução dos mais diferenciados problemas, a etapa de descrição formal do modelo pela formulação matemática e implementação do conhecimento especialista para extração de características é substituída por um processo de treinamento de redes neurais de maneira integrada a etapa de tomada de decisão. O treinamento permite observar as correlações existentes em um grande volume de dados para aprender diretamente do dado bruto quais padrões são relevantes a tarefa e assim ajustar os parâmetros da rede para que uma vez treinada passe a fornecer um mapeamento entre um novo dado de entrada e sua saída no problema modelado.

Esta possibilidade de solução integrada vem destacando um conjunto de técnicas que atendem ao nome *Deep Learning* (DL) como uma das abordagens mais populares e bem sucedidas para tarefas envolvendo aprendizado de padrões. Existem 3 fatores que se alinharam para que o DL se tornasse viável atualmente: (1) uma grande quantidade de dados disponíveis (Big Data); (2) Desenvolvimento de novas técnicas de DL; (3) Supercomputação de forma acessível. Neste último quesito é onde as GPUS se tornam importantes protagonistas nesta revolução: além de serem processadores altamente paralelos (arquiteturas como a NVIDIA Pascal podem chegar a 11 TFlops de processamento numa única placa, mais do que o dobro do supercomputador Laurence, maior computador do mundo no ano 2000), o "DNA" de suas arquiteturas permitem que diversas tarefas típicas de DL sejam beneficiadas de forma direta pelo alto paralelismo disponível. Para se ter uma ideia, em 2012 a Google apresentou um sistema de DL para análise de imagens no qual usavam 1000 servidores de CPUs, com um custo de 5 milhões de dólares e consumo de energia de 600KW. No ano seguinte, resolveram o mesmo problema usando 3 servidores, com 12 GPUs, a um custo de 33 mil dólares e com uma energia de 4KW.

Sobre o primeiro dos 3 fatores listados, também é notável que as comunidades

de pesquisa em reconhecimento de padrões nos diferentes tipos de sinais (áudio, texto e imagem) tenham se organizado para promover desafios com grandes bancos de dados anotados. A anotação desses bancos consiste em criar pares em que cada exemplo de entrada é associado a resposta esperada, muitas vezes fornecida por um especialista. A partir desses bancos é possível padronizar a avaliação de diferentes algoritmos de maneira a convergir esforços de pesquisa em uma plataforma de comparação em comum. É o caso da Imagenet Large Scale Visual Recognition Challenge [ima], que em 2012 destacou a capacidade das Redes Neurais de Convolução frente as abordagens de modelagem de conhecimento especialista na tarefa de classificação de um conjunto de 1000 categorias de objetos fornecendo para treinamento uma base de imagens com 1.2 milhão de imagens anotadas.

Outra frente importante tem sido as bibliotecas disponibilizadas para a comunidade. Sejam as desenvolvidas por Universidades ou por empresas, viabilizaram acelerar a pesquisa na área e impulsionaram o crescente número de aplicações desenvolvidas.

Este capítulo de livro, embora de maneira resumida, busca cobrir os temas mais abordados nas soluções vigentes com o uso de *Deep Learning*. Não tem a pretensão de cobrir toda a área, mas de servir como convite para um primeiro contato do leitor e de instigar sua busca pelas fronteiras do DL. São apresentados os fundamentos necessários para compreensão do que constitui as chamadas Redes Neurais na Seção 6.2, aprofundando duas de suas possíveis modelagens ditas Redes Neurais Alimentadas para Frente (na Seção 6.2.2.1) e Redes Neurais Recorrentes (na Seção 6.2.2.3). Em seguida, a Seção 6.3 apresenta as chamadas Redes Neurais de Convolução como ilustração do primeiro modelo, enquanto que a Seção 6.4 ilustra o segundo modelo apresentando as chamadas redes *Long-Short Term Mermory*. Buscando cobrir os bastidores de sua implementação eficiente, a Seção 6.5 apresenta a arquitetura das GPUs. Por fim, uma breve apresentação sobre as ferramentas de desenvolvimento é apresentada na Seção 6.6.

6.2. Fundamentos

As diferentes soluções baseadas em aprendizado profundo são construídas em cima de variações das chamadas redes neurais artificiais. Seus fundamentos são apresentados nesta seção sem a pretensão de cobrir os múltiplos aspectos teóricos da área.

Diferentemente de algoritmos programados para traduzir um modelo de co/-nhe/-ci/-men/-to especialista na solução de um problema, redes neurais simulam o cérebro humano no sentido de que aprendem a realizar uma tarefa.

Em um contexto mais amplo do que as redes neurais artificiais, a área de aprendizado de máquina classifica os algoritmos de aprendizado em três categorias:

- **Aprendizado supervisionado**: aplicável quando são conhecidos pares associando entradas às respectivas respostas desejadas. Algoritmos nesta categoria buscam aprender uma função que mapeia as entradas nas saídas desejadas. Uma vez aprendida, tal função de mapeamento pode ser aplicada a novas entradas. O processo de treinamento continua até que o modelo atinja um nível

de precisão desejado nos dados de teste ou que uma quantidade de iterações seja atingida.

- **Aprendizado não-supervisionado:** aplicável quando não são fornecidas saídas desejadas, mas se deseja encontrar ou fazer inferências sobre padrões inerentes ao comportamento das amostras de entrada. Os algoritmos mais conhecidos nesta categoria são os de clusterização, responsáveis pela organização das amostras de entrada em um conjunto de grupos, ditos *clusters*.

- **Aprendizado por reforço:** aplicável quando se deseja um algoritmo que interage com um ambiente para tomar decisões. Nesta categoria o aprendizado se dá expondo o algoritmo a um ambiente no qual ele pode praticar por tentativa e erro, recebendo recompensa ou punições como *feedback* pelas decisões tomadas. Com o passar do tempo, passa a acumular conhecimento com a experiência passada, ou seja, aprende a preferir o tipo certo de ação e evitar as que induzem ao erro, tentando capturar o melhor conhecimento/estratégia para tomar decisões futuras.

Existem abordagens de redes neurais profundas para tratamento de problemas nas três categorias citadas. Sua construção parte de elementos simples, mas que em grande quantidade e propriamente dispostos e adaptados interagem entre si modelando diferentes comportamentos.

6.2.1. O Perceptron

Diferentes topologias de Redes Neurais Artificiais (RNAs) podem ser construídas utilizando-se variações e combinações da mesma peça básica: um neurônio artificial.

Em alto nível, podemos pensar que um neurônio artificial é um mecanismo capaz de processar um conjunto de sinais que recebe como entrada na forma do vetor $X = \{x_1, x_2, ..., x_N\}$ para produzir um sinal de saída (y).

Assim como neurônios biológicos possuem dendritos por onde recebem estímulos, um corpo no qual tais sinais são processados, e um axônio responsável por emitir para fora do neurônio um sinal de saída, também os neurônios artificiais possuem um conjunto de canais de entrada, etapas de processamento e uma saída que pode ser ligada a outros neurônios (Figura 6.1).

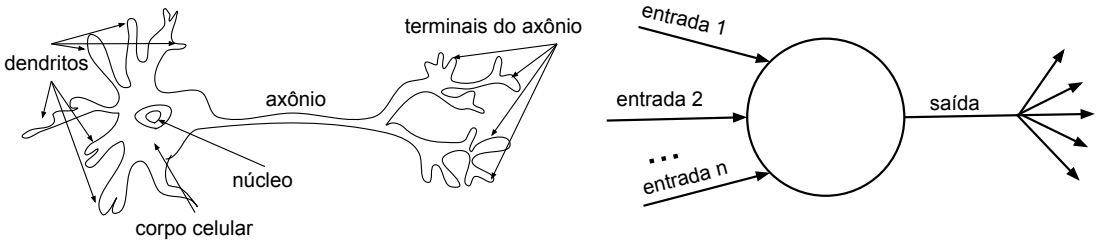

Figura 6.1: Neurônio Biológico e Neurônio Artificial

Ainda hoje utilizamos os fundamentos do modelo básico de neurônio artificial proposto em 1957 por Frank Rosenblatt, denominado Perceptron [Rosenblatt 1961]. Partindo de conceitos propostos pela neurociência, Rosenblatt introduziu a ideia de associar um peso a cada conexão de entrada do neurônio artificial de maneira a ponderar sua influência para uma determinada tarefa (Figura 6.2).

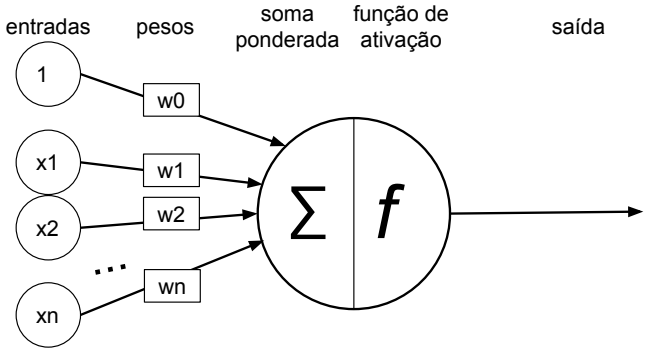

Figura 6.2: Perceptron: a combinação linear das entradas x_i ponderadas pelos pesos w_i é transformada pela função de ativação f na saída y emitida pelo neurônio

Assim, um neurônio que possui N conexões de entrada, associadas respectivamente aos pesos $W = \{w_i\}, 1 < i <= N$ produz uma transformação linear do sinal de entrada $X = \{x_i\}, 1 < i <= N$ descrita pela equação:

$$z = W \cdot X + b = \sum_{i=1}^{N} w_i x_i + b = w_1 x_1 + w_2 x_2 + ... + w_N x_N + b \quad (1)$$

Observe que além do vetor X de sinais de entrada e do vetor W de seus respectivos pesos é comum acrescentar ao neurônio um termo b chamado de bias. O bias não depende do sinal de entrada, mas aumenta os graus de liberdade da fronteira de decisão representada pelo neurônio, uma vez que permite que ela não necessariamente passe pela origem do sistema de coordenadas do sinal de entrada.

Para uniformizar a nomenclatura, é comum encontrarmos na literatura a inclusão do valor constante 1 como primeiro elemento do conjunto de entrada de maneira que b passa a ser representado pelo peso associado a tal entrada constante e passa a ser representado como o peso w_0. Com isso, a Equação 1 passa a ser reescrita como:

$$z = W \cdot X = \sum_{i=0}^{N} w_i x_i = w_0 1 + w_1 x_1 + w_2 x_2 + ... + w_N x_N \quad (2)$$

O neurônio artificial aplica uma nova transformação ao resultado da combinação linear dos sinais de entrada. Tal transformação é tipicamente não linear de maneira a aumentar o poder de expressão do neurônio e realizada pela função de ativação f.

Diferentes funções de ativação são utilizadas na formulação dos neurônios artificiais tais como a identidade, a tangente hiperbólica, a sigmoide hiperbólica, e a retificada linear (ReLU) (Figura 6.3).

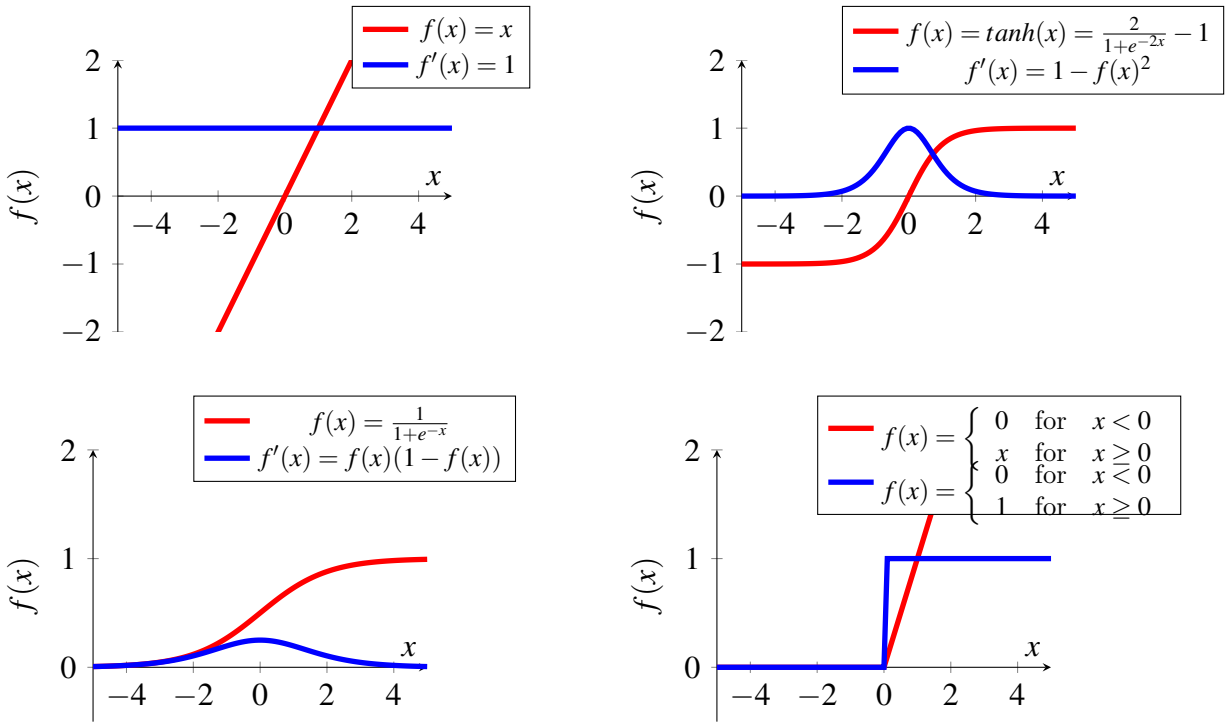

Figura 6.3: Exemplos de diferentes funções de ativação: função identidade, tangente hiperbólica, sigmoide hiperbólica (função logística), e a retificação linear (ReLU)

O perceptron de Rosenblatt foi formulado visando a construção de um classificador binário, ou seja, capaz de produzir como saída os valores 0 e 1. Nele a função de ativação desempenha o papel de aplicação de um limiar. A função de ativação adotada no perceptron é chamada de Função Degrau de Heaviside e definida como (Figura 6.4):

$$f(z) = \begin{cases} 1, & se\ W \cdot X + b > 0 \\ 0, & cc. \end{cases} \tag{3}$$

O perceptron desta forma definido consegue resolver problemas binários, no qual as possíveis saídas são linearmente separáveis. Portanto, é capaz de aprender as funções booleanas de negação (NOT), "e"(AND) e "ou"(OR). Entretanto, utilizando-se apenas um perceptron não é possível modelar a função booleana "ou exclusivo" (XOR), nem sua negação (NXOR). A Figura 6.5 ilustra fronteiras de decisão no espaço de entrada 2D estabelecidas por perceptrons recebendo duas entradas booleanas x_1 e x_2 e considerando $x_0 = 1$ como entrada constante de ativação do bias. Observe que são necessárias duas retas para modelar a operação XOR (portanto não sendo possível de representar com um único perceptron), enquanto que basta uma reta para definir as operações AND, OR e NOT. Tal limitação é

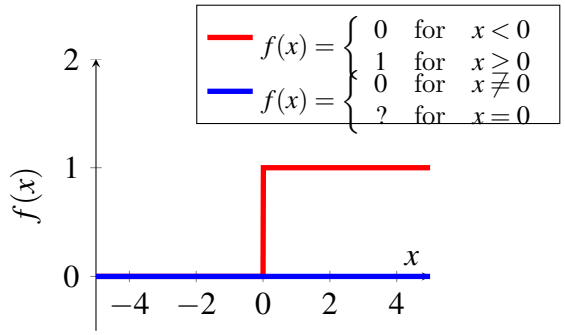

Figura 6.4: Função de ativação dos perceptrons: Função Degrau de Heaviside

desfeita pela combinação de mais perceptrons em camadas, conforme apresentado nas próximas seções.

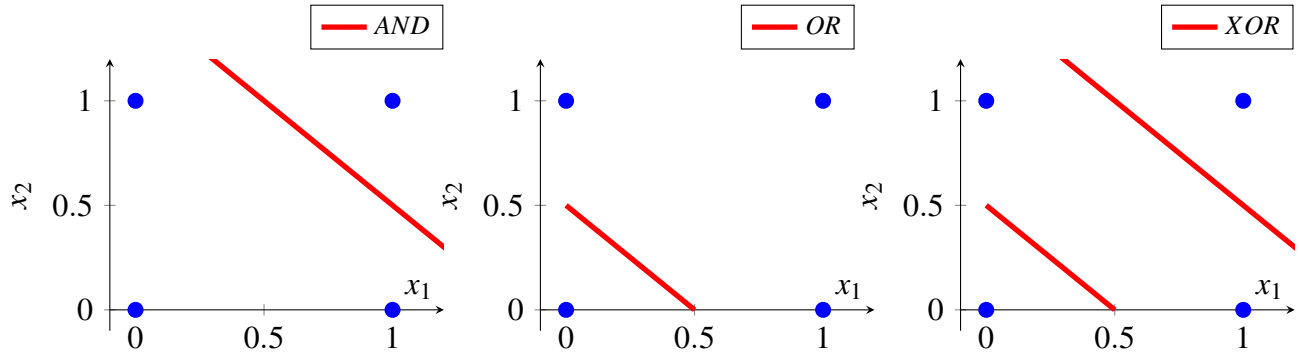

Figura 6.5: AND e OR: Operações booleanas modeladas por um único perceptron; XOR: duas retas são necessárias

6.2.1.1. Treinando o Perceptron

Em uma abordagem diferente da modelagem e programação de algoritmos específicos para resolver um problema, a mágica dos neurônios artificiais acontece quando os neurônios aprendem a resolvê-los.

No caso dos perceptrons, o processo de aprendizado é capaz de encontrar retas que separam entradas 2D, planos no caso de entradas 3D, ou ainda hiperplanos em dimensões maiores. Ao fim do aprendizado, podem ser utilizados para, dada um nova amostra, determinarem se esta se encontra de um lado ou de outro da fronteira de decisão aprendida.

O aprendizado parte de um conjunto de amostras de entrada sobre as quais são conhecidas as saídas desejadas. Cada amostra de treinamento é composta por um par (X,Y), no qual X é um vetor n-dimensional descrevendo um sinal de entrada, e Y é a resposta correspondente desejada.

Os parâmetros W e b configuram a fronteira de decisão modelada pelo neurônio, portanto, são os parâmetros a serem aprendidos durante o treinamento. Para treinar

um perceptron é comum inicializar seus n pesos do conjunto W com valores aleatórios (tipicamente entre -1 e 1) e deve-se associar também um valor inicial para seu bias b (tipicamente zero).

Uma vez inicializados, o treinamento em si consiste de duas etapas a serem alternadas repetidas vezes. A cada iteração t: (i) a primeira etapa processa um sinal de entrada X, produzindo um valor de saída $o(t)$; (ii) a segunda etapa ajusta os parâmetros do neurônio de acordo com a comparação entre a saída obtida $o(t)$ e o resultado desejado Y;

Ao executar a primeira etapa os valores dos parâmetros W e b são mantidos fixos e usados para ponderar os elementos de uma amostra de treinamento, que são então somados ao bias e aplicados à função degrau, obtendo-se uma saída $o(t) = f(z(X(t)))$ (vide equações 2 e 3).

Na segunda etapa acontece o aprendizado propriamente dito. Nela, o perceptron ajusta os pesos e bias observando quão diferente é o resultado $o(t)$ obtido com seus parâmetros atuais da resposta desejada Y fornecida no par de treinamento. O erro obtido na iteração t é calculado como $e(t) = Y - o(t)$ e é utilizado para atualizar os parâmetros por:

$$W(t+1) = W(t) + \alpha(Y - o(t))X \tag{4}$$

onde α representa a taxa de aprendizado (*learning rate*), t representa o passo da iteração, X representa o vetor com os canais da amostra de entrada acrescido de $x_0 = 1$ e W representa o vetor de pesos acrescido de $w_0 = b$.

O valor de α é um parâmetro escolhido previamente ao treinamento utilizando diferentes políticas e de alta influência ao processo de aprendizado. Ele estabelece o fator com que uma amostra contribui em uma iteração para a atualização dos parâmetros do perceptron. Logo, a Equação 4 pode ser interpretada como uma atualização proporcional ao erro da amostra avaliada (termo $y - o(t)$) e a quanto cada conexão de entrada x_i contribuiu para o erro uma vez que está relacionada a um w_i correspondente.

O processo de treinamento do perceptron é repetido até que por uma época o erro de treinamento seja inferior do que um limiar pré-estabelecido, ou ainda, pode ser repetido por uma quantidade pré-determinada de épocas. Denomina-se época, uma passagem do treinamento sob o conjunto de amostras completo.

6.2.2. Redes Neurais Artificiais

Existem diferentes maneiras de se conectar um conjunto de neurônios para formação de uma Rede Neural Artificial - RNA. A arquitetura de uma RNA define o padrão de disposição e conexão de seus neurônios. Focaremos nossa abordagem em dois tipos de arquiteturas diferenciadas pelo direcionamento de suas conexões.

A Subseção 6.2.2.1 introduz as arquiteturas nas quais o sinal é transmitido em uma única direção e por este motivo são chamadas de Redes Neurais Alimentadas para Frente (do inglês *Feedforward Neural Networks*).

Em seguida, a Subseção 6.2.2.3 introduz as arquiteturas nas quais há neurônios cujas saídas realimentam a si próprios e a outros neurônios de maneira a criar ciclos. Por esse motivo são chamadas Redes Neurais de Retroalimentação ou ainda Redes Neurais Recorrentes (do inglês *Recurrent Neural Networks*).

6.2.2.1. Redes Neurais Alimentadas para Frente

As Redes Neurais Alimentadas para Frente são também chamadas de Redes Neurais de Múltiplas Camadas ou ainda de Perceptron Múltiplas Camadas (MLP do inglês *multi-layer perceptron*). Adotaremos a sigla MLP embora não necessariamente utilizam neurônios idênticos aos perceptrons originais por substituir a função de ativação original.

Nas MLPs os neurônios são organizados em camadas sequenciais de maneira que o sinal fornecido como entrada é transmitido em uma direção apenas (Figura 6.6). Nessas arquiteturas os neurônios de uma mesma camada não são conectados entre si. Cada neurônio recebe sinais de entrada vindos de neurônios de camadas anteriores e por sua vez transmite o sinal por ele produzido para neurônios de camadas seguintes.

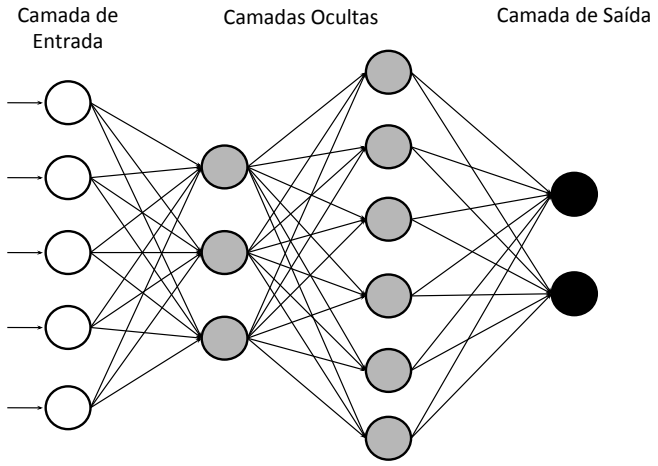

Figura 6.6: Estrutura de uma rede neural de múltiplas camadas. Os neurônios estão representados por círculos e as conexões por setas, todas em uma mesma direção, sem a formação de ciclos.

A estrutura em camadas é composta por: (i) uma camada inicial dita camada de entrada, responsável pela leitura dos dados a serem processados; (ii) uma ou mais camadas intermediárias, chamadas camadas escondidas, as quais não são nem de entrada nem de saída e são responsáveis pelo processamento propriamente dito; (iii) uma última camada responsável por emitir o resultado do processamento, ou seja, a saída da rede.

Deste ponto em diante do texto vamos incluir um novo índice j para di/-fe/-ren/-ciar de qual neurônio nos referimos. O conjunto formado pelos pesos associados às conexões (indexadas por i) de entrada do neurônio j passa a ser definido como

$W_j = \{w_{ji}\}$ (Figura 6.7) e o conjunto das ativações que alimentam tais conexões como $X_j = \{x_{ji}\}$.

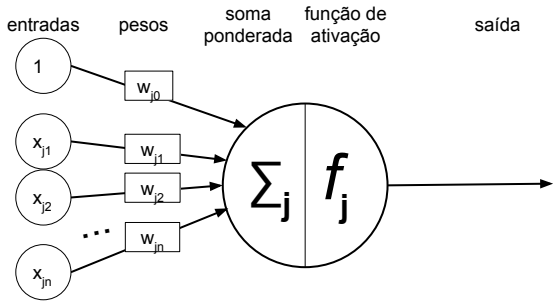

Figura 6.7: Neurônio j e suas conexões de entrada de dados, cada uma associada a um índice i

Nesta notação, a combinação linear das N entradas de um neurônio j passa a ser escrita como:

$$z_j = W_j \cdot X_j = \sum_{i=1}^{N} w_{ji} x_{ji} \tag{5}$$

Enquanto que a função de ativação f, a qual produz a saída o_j emitida pelo neurônio, passa a ser escrita como:

$$f(z_j) = f(W_j \cdot X_j) = f(\sum_{i=1}^{N} w_{ji} x_{ji}) = o_j \tag{6}$$

O resultado obtido pelo neurônio j passa a alimentar outros neurônios de camadas seguintes. Supondo existir uma conexão de índice i a partir do neurônio j para o neurônio k, tal conexão é ativada com valor o_j. Logo, temos $o_j = x_{ki}$ e portanto o_j passa a fazer parte do conjunto X_k. Uma vez que a saída de um neurônio j pode estar conectada a diferentes neurônios em camadas seguintes, os elementos dos conjuntos de ativações de diferentes neurônios não são necessariamente únicos.

Podemos descrever o algoritmo de propagação de um sinal X por uma rede neural alimentada para frente de L camadas ocultas, parametrizada por $L+1$ matrizes de pesos W como:

Algorithm 1 Propagação MLP

procedure MLPFORWARD(X, W)
  $x(0) \leftarrow X$.
  $c \leftarrow 1$.
  for $c <= L+1$ **do**
    $z(c) \leftarrow W(c-1)x(c-1)$ ▷ combinação linear
    $x(c) \leftarrow f(z(c))$ ▷ transformação não-linear
  return $o \leftarrow x(L+1)$

Partindo de um modelo de arquitetura, a topologia de uma RNA descreve sua composição estrutural específica, tal como o número de neurônios utilizados, a(s) função(ões) de ativação escolhida(s).

Sobre a formulação de uma MLP para um determinado problema, a definição da topologia das suas camadas de entrada e saída é normalmente simples e intuitiva de ser definida. Isso porque essas camadas estão diretamente relacionadas respectivamente a dimensionalidade dos dados que queremos fornecer para a rede e a dimensionalidade da solução desejada.

As mais diferentes arquiteturas com alimentação para frente exploram variações nas camadas ocultas, uma vez que ao modificá-la altera-se de fato o mapeamento modelado pela rede. A Seção 6.3 apresenta um tipo de arquitetura de alimentação para frente inspirada no processamento biológico de dados visuais.

Uma vez definida uma arquitetura, a escolha do número de camadas ocultas e do número e disposição de seus neurônios e conexões é muitas vezes realizada por experimentação.

6.2.2.2. Treinando Redes Alimentadas para Frente: Algoritmo de Retropropagação

Vimos na Seção 6.2.1.1 o algoritmo para treinamento de perceptrons, os quais também podem ser pensados como redes neurais simples de uma única camada oculta. Nesta seção apresentamos o algoritmo de Retropropagação (do inglês *Backpropagation*) para treinamento de redes neurais de múltiplas camadas.

Para realizar o treinamento de tais redes acrescenta-se uma extensão após a sua camada de saída (ilustrada na Figura 6.8), responsável por calcular uma função E de erro ou perda (do inglês *loss function*). É essa função de erro que se deseja minimizar ao longo do processo de treinamento.

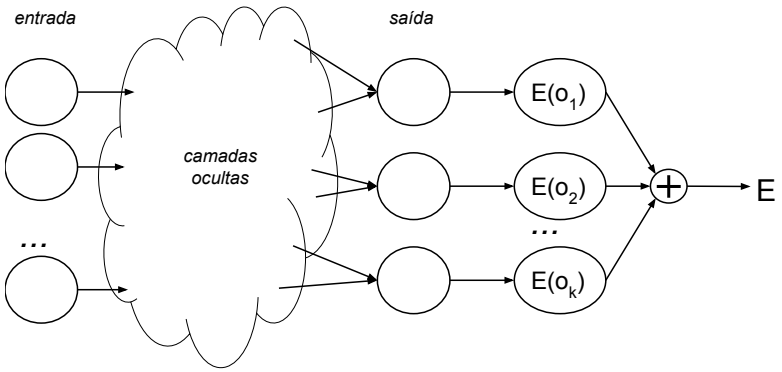

Figura 6.8: Extensão da rede criada para o cálculo do erro.

Uma vez definida uma rede MLP, os parâmetros a serem ajustados para diminuir o erro segundo a função E adotada são os pesos associados as suas conexões. Portanto, são eles os parâmetros que se deseja aprender com o processo de treinamento.

O primeiro passo do algoritmo de retropropagação é a inicialização dos valores dos pesos da rede. É comum o uso de pesos iniciais aleatórios em torno de zero (como

por exemplo amostrados de uma distribuição normal uniforme de média zero e desvio padrão um).

Uma vez concluída a inicialização, o algoritmo passa a repetir as seguintes duas etapas até que o critério de parada escolhido seja atingido (Figura 6.9):

- **Etapa de propagação do sinal**: fixando-se os pesos, um sinal de entrada $X(t)$ é processado na direção estabelecida pelas conexões, até produzir a saída $o(t)$ correspondente a tal amostra de treinamento. O erro de treinamento da iteração t é então calculado comparando a resposta da rede $o(t)$ com a saída desejada $y(t)$ fornecida no par de treinamento utilizando-se a função de erro $E(t)$ previamente escolhida.

- **Etapa de retropropagação do erro**: partindo-se da camada de saída da rede é feita uma retropropagação do erro na direção oposta à estabelecida pelas conexões até atingir os neurônios da camada de entrada. Uma vez estimado quanto cada conexão influenciou o erro obtido, os pesos são atualizados.

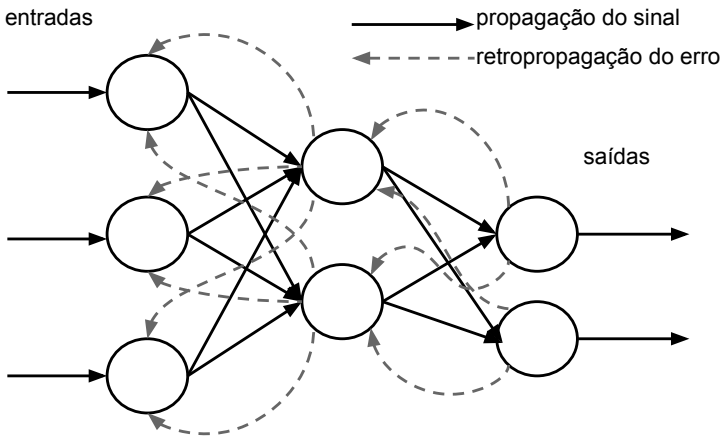

Figura 6.9: Fluxograma do algoritmo de retro-propagação: propagação do sinal na direção das conexões e retro-propagação do erro na direção oposta

Na descrição em alto nível apresentada, falta detalhar como são atualizados os pesos. O algoritmo de retropropagação realiza um processo de otimização da função E, ou seja, busca minimizá-la, sendo os pesos as variáveis a serem manipuladas com esse objetivo. Assumiremos uma otimização de gradiente descendente estocástico online no qual os pesos são atualizados baseados na observação de uma amostra de treinamento por vez, apresentadas em ordem aleatória.

Para ajustar um determinado w_{ji} segundo a otimização de gradiente descendente, o algoritmo estima como este peso influenciou o erro obtido na iteração atual t. Para isso calcula localmente a inclinação de $E(t)$ em relação a $w_{ji}(t)$ enquanto assume fixos os demais pesos. Matematicamente, a tangente da função de erro avaliada em $E(t)$ em relação a um determinado peso $w_{ji}(t)$ é obtida pela derivada parcial de $E(t)$ em relação a tal peso: $\partial E(t)/\partial w_{ji}(t)$. Por sua vez, o vetor gradiente da

função de erro em relação aos pesos da rede é formado pelas derivadas parciais de $E(t)$ por cada um dos elementos de $W(t)$:

$$\nabla E(t) = \left\langle \frac{\partial E(t)}{\partial w_{1,1}(t)}, \frac{\partial E(t)}{\partial w_{1,2}(t)}, \frac{\partial E(t)}{\partial w_{1,\dots}}, \frac{\partial E(t)}{\partial w_{2,1}(t)}, \frac{\partial E(t)}{\partial w_{2,2}(t)}, \frac{\partial E(t)}{\partial w_{2,\dots}}, \cdots, \frac{\partial E(t)}{\partial w_{n,1}(t)}, \frac{\partial E(t)}{\partial w_{n,2}(t)}, \frac{\partial E(t)}{\partial w_{n,\dots}}, \right\rangle \quad (7)$$

O vetor gradiente indica a direção e o sentido de maior inclinação de $E(t)$ em relação a um conjunto de parâmetros. Por conta dessa propriedade, os métodos conhecidos como gradiente descendente utilizam $-\nabla E(t)$ para caminhar em E de maneira a minimizá-la. Operam alternando entre o cálculo do erro, o respectivo vetor gradiente em relação aos parâmetros observados, e a atualização de tais parâmetros por um pequeno passo na direção negativa a encontrada.

Com isso, no treinamento de redes MLP seus pesos são atualizados somando-se a eles o negativo do vetor $\nabla E(t)$ ponderado pela taxa de aprendizado α a qual regula o tamanho do passo a ser dado na direção do gradiente:

$$W(t+1) = W(t) - \alpha \nabla E(t) \quad (8)$$

Em relação a um determinado peso $w_{ij}(t)$ podemos escrever:

$$w_{ji}(t+1) = w_{ji}(t) - \alpha \frac{\partial E(t)}{\partial w_{ji}(t)} \quad (9)$$

Falta ainda detalhar como o algoritmo de retropropagação encontra as derivadas parciais $\partial E(t)/\partial w_{ji}(t)$ em uma determinada iteração. O algoritmo de retropropagação propõe o uso da regra da cadeia para abrir $\partial E(t)/\partial w_{ji}(t)$ nos passos de processamento realizados pela rede entre a ativação da conexão i de alimentação do neurônio j até a saída e cálculo da função de erro.

Supondo j um neurônio da camada de saída, z_j a combinação linear de suas entradas (Equação 5), f_j a função de ativação aplicada ao resultado de z_t (Equação 6) e supondo uma função de erro atuando sobre as saídas dos neurônios da última camada, podemos reescrever $\partial E(t)/\partial w_{ji}(t)$ como (Figura 6.10):

$$\frac{\partial E(t)}{\partial w_{ji}(t)} = \frac{\partial E(t)}{\partial f_j(t)} \frac{\partial f_j(t)}{\partial z_j(t)} \frac{\partial z_j(t)}{\partial w_{ji}(t)} \quad (10)$$

onde:

- o termo $\partial E(t)/\partial f_j(t)$ pode ser obtido diretamente da derivação da função de erro em relação a saída produzida pelo neurônio j, avaliada em $f_z(t)$;
- o termo $\partial f_j(t)/\partial z_j(t)$ pode ser obtido pela derivação da função de ativação adotada, avaliada em $z_j(t)$;
- e por sua vez o termo $\partial z_j(t)/\partial w_{ji}(t)$, por representar a derivação da combinação linear das entradas de j em relação ao peso $w_{ji}(t)$, assume o valor fornecido como entrada para tal conexão, portanto $x_{ji}(t)$.

Com esses valores, pode-se aplicar a regra definida pela Equação 9 para atualizar os pesos da última camada. É comum reescrevê-la de maneira a isolar os termos que não dependem de $w_{ji}(t)$ usando:

$$\delta_j = \frac{\partial E(t)}{\partial z_j(t)} \tag{11}$$

e reescrevemos as Equações 10 e 8 respectivamente como:

$$\frac{\partial E(t)}{\partial w_{ji}(t)} = \delta_j(t) x_{ji}(t) \tag{12}$$

$$W(t+1) = W(t) - \alpha \delta_j(t) * X_j \tag{13}$$

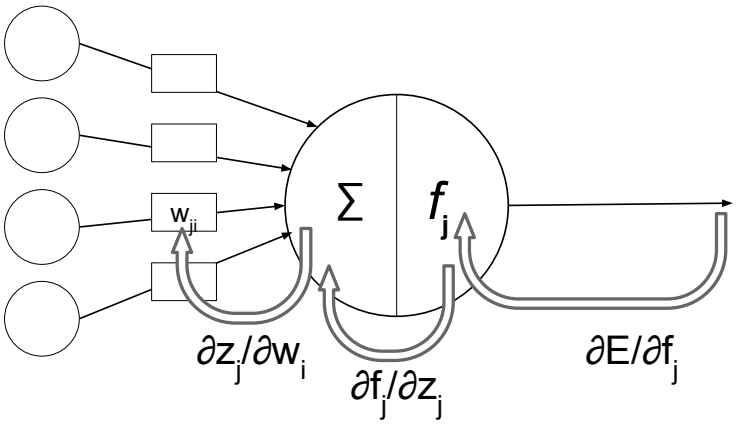

Figura 6.10: Reescrita da derivada parcial do erro em relação aos neurônios da última camada

Pesos associados a conexões que alimentam neurônios de outras camadas continuam esse processo de retropropagação. Tomando como ponto de partida a Equação 10 para avaliar o que ocorre em neurônios das demais camadas, observamos que o termo $\frac{\partial f_j(t)}{\partial z_j(t)}$ pode ser calculado uma vez conhecida a função de ativação do neurônio, e $\frac{\partial z_j(t)}{\partial w_{ji}(t)}$ é obtido como o valor do sinal que ativou a conexão, portanto $x_{ji}(t)$.

Entretanto, o termo $\frac{\partial E(t)}{\partial f_j(t)}$ não é mais obtido diretamente da derivação de E, uma vez que neurônios das camadas ocultas não produzem diretamente a saída da rede. O sinal produzido como sua saída em f_j se propaga por n conexões para neurônios da(s) camada(s) seguinte(s), na forma de ativações $x_k i$. Portanto $\frac{\partial E(t)}{\partial f_j(t)}$ precisa ser reescrito como a soma das derivadas parciais de E em relação a essas n conexões por onde $f_j(t)$ é propagado.

$$\frac{\partial E(t)}{\partial f_j(t)} = \sum_{k=1}^{n} \frac{\partial E(t)}{\partial x_{ki}(t)} = \sum_{k=1}^{n} \frac{\partial E(t)}{\partial z_k(t)} \frac{\partial z_k(t)}{\partial x_{ki}(t)} = \sum_{k=1}^{n} \frac{\partial E(t)}{\partial z_k(t)} w_{ki}(t) = \sum_{k=1}^{n} \delta_k(t) w_{ki}(t) \tag{14}$$

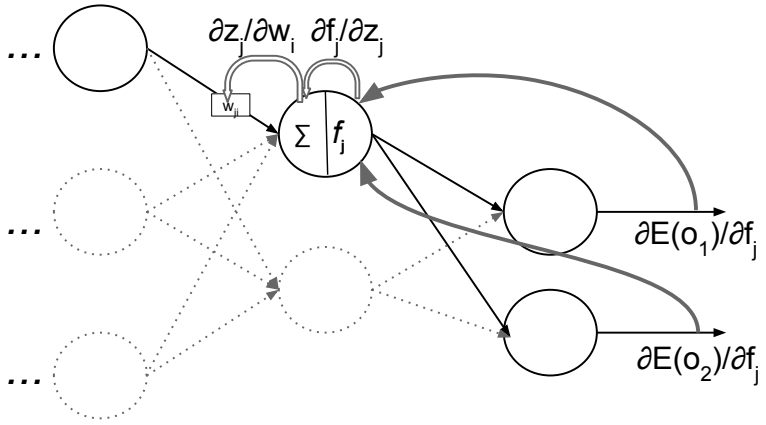

Figura 6.11: Reescrita da derivada parcial do erro em relação aos neurônios de camadas escondidas

Usando a definição de $\delta_j(t)$ (Equação 11) podemos ainda escrever:

$$\delta_j(t) = \frac{\partial f_j(t)}{\partial z_j(t)} \sum_{k=1}^{n} \delta_k(t) w_{ki}(t) \tag{15}$$

De onde observamos que o algoritmo de retropropagação do erro requer que as funções de ativação adotadas para os neurônios sejam diferenciáveis. Com a Equação 15 é possível retropropagar os δ de camadas mais profundas para anteriores, e atualizar os pesos aplicando-as na Equação 13.

Logo, o algoritmo de retropropagação pode ser descrito como:

- Aplica-se uma amostra de treinamento pela rede propagando-a usando as equações 5 e 6 para encontrar as ativações de todos os neurônios das camadas escondidas e de saída.
- Os valores de δ_k para todos os neurônios de saída são calculados usando as equações 11 e 10.
- Os δs encontrados são retropropagados usando a Equação 15 para obter δ_j de cada neurônio de camadas escondidas.
- Para aplicar a atualização dos pesos, usa-se a Equação 14 e os valores precomputados dos δs na Equação 9.

6.2.2.3. Redes Neurais Recorrentes

Nesta seção apresentamos uma introdução ao modelo de redes com retroalimentação chamadas Redes Neurais Recorrentes (do inglês *Recurrent Neural Networks*-RNN). Diferentemente das redes alimentadas para frente, em RNNs a saída de alguns neurônios retro-alimentam a si próprios e a outros neurônios formando ciclos

Algorithm 2 Retro-propagação MLP

```
1: procedure MLPBACK(X, W0)
2:     dx(L+1) ← dE(x(L+1), y)/dx(L+1)
3:     for c from L+1 downto 1 do
4:         dz(c) ← df/dz(z(c)) · dx(c)                    ▷ δ(c)
5:         dx(c−1) ← W^T(c−1) dz(c)
6:         dW(c−1) ← dz(c) x^T(c−1)
       return dW[0 ··· L]
```

no percurso de processamento. Tais arranjos formadores de ciclos permitem que haja informação persistente entre computações, ou seja, valores que se propagam ao longo de uma sequência de análises na forma de ativações das conexões de retro-alimentação. Esses valores são considerados estados escondidos ou ocultos por não terem sidos fornecidos como sinais de entrada, nem serem emitidos como de saída da rede.

Por permitirem que informações sejam passadas de uma etapa de processamento para próxima, RNNs são capazes de lidar com sinais na forma de sequências. Seja X o vetor com o sinal de entrada da rede composto por uma sequência de N_s elementos, onde cada elemento é indexado como $X(t)$ e é descrito por N_c valores ou canais. Logo, a sequência completa é composta por um vetor de tamanho $N_s * N_c$.

Uma mesma RNN pode lidar com sequências de tamanhos variados, não exigindo que N_s seja fixo. Isso porque a RNN é modelada para receber um dos elementos $X(t)$ da sequência por vez (de onde sua camada de entrada tem N_c neurônios), e para ser replicada ao longo da sequência.

Como ilustração de uma topologia de RNN, a Figura 6.12 apresenta uma rede capaz de processar sequências de tamanho qualquer contendo elementos descritos cada um por 2 valores. A RNN ilustrada possui uma camada de entrada de dois neurônios (em correspondência ao número de canais do sinal de entrada), ligados a uma camada escondida de 3 neurônios interconectados entre si e a uma camada de saída formada por 2 neurônios. Observe na Figura 6.13 que a rede é replicada para processar os elementos da sequência de entrada, indexados na ilustração como $t-1$, t, $t+1$.

Embora outras construções possam ser elaboradas, por exemplo aumentando o número de camadas escondidas, assumiremos por simplicidade um modelo geral de uma RNN com uma única camada escondida, com uma quantidade de neurônios qualquer, na qual seus neurônios possuem uma conexão para si próprios, para os outros neurônios da mesma camada e para os neurônios de saída da rede.

Os neurônios recursivos de uma RNN seguem em alto nível o modelo geral de neurônios artificiais no sentido de aplicarem uma combinação linear de suas entradas, seguida de uma transformação não-linear. Entretanto, não somente utilizam de tal combinação de transformações para produzir seu valor de saída, mas também para atualizar os estados escondidos, que irão realimentar a si próprios e a outros neurônios da rede no tratamento do próximo elemento da sequência (Figura 6.14).

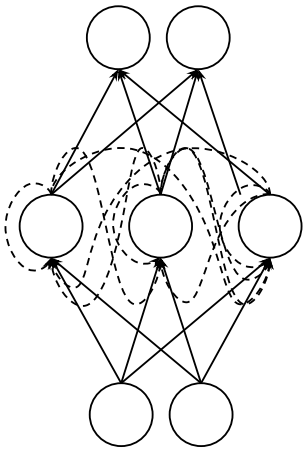

Figura 6.12: Ilustração de topologia RNN com 2 neurônios na camada de entrada, 3 na camada escondida, responsáveis pela recorrência, e 2 na camada de saída

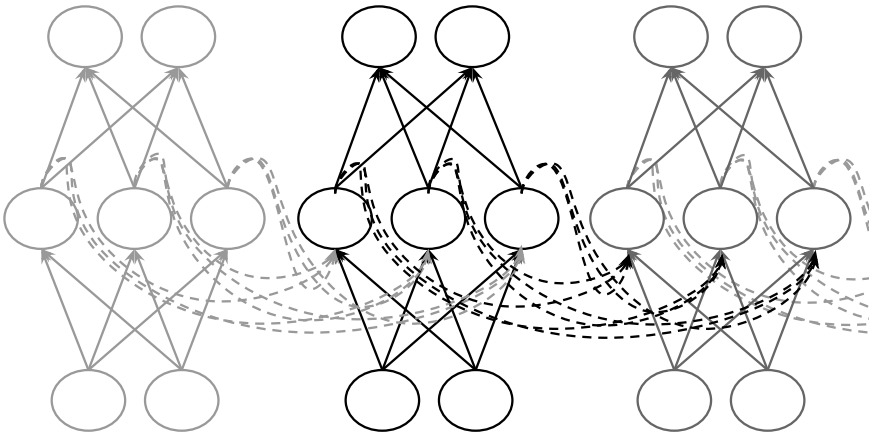

Figura 6.13: RNN da Figura 6.12, replicada em $t-1$, t e $t+1$.

O estado $h(t)$ produzido por um neurônio recursivo de função de ativação f_h, na iteração t pode ser encontrado como:

$$
\begin{aligned}
h(1) &= f_h(W_h h(0) + W_e X(1) + b_e) \\
h(2) &= f_h(W_h h(1) + W_e X(2) + b_e) \\
&= f_h(W_h(f_h(W_h h(0) + W_e X(1) + b_e)) + W_e X(2) + b_e) \\
h(3) &= f_h(W_h h(2) + W_e X(3) + b_e) \\
&= f_h(W_h(f_h(W_h(f_h(W_h h(0) + W_e X(1) + b_e)) + W_e X(2) + b_e)) + W_e X(3) + b_e) \\
h(t) &= f_h(W_h h(t-1) + W_e X(t) + b_e) \qquad (16) \\
&= f_h(W_h(f_h(W_h \cdots (f_h(W_h h(0) + W_e X(1) + b_e)) \cdots + W_e X(t-1) + b_e)) + W_e X(t) + b_e) \qquad (17)
\end{aligned}
$$

Onde,

- $h0$ é o estado inicial dos estados escondidos, o qual pode ser inicializado com zeros, um valor fornecido ou aprendido ;

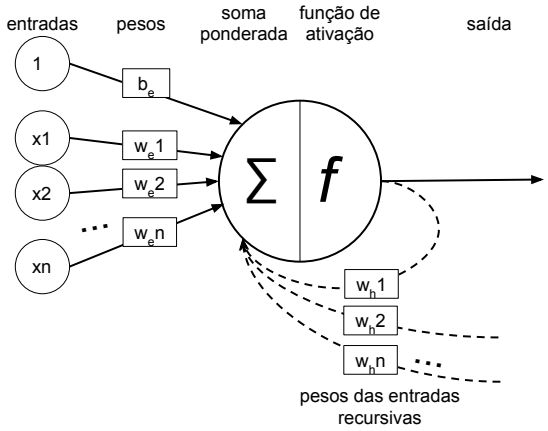

Figura 6.14: Neurônio recursivo

- W_e, b_e: representam respectivamente a matriz de pesos e o vetor de bias das conexões alimentadas com o sinal de entrada;
- W_h: representa a matriz de pesos das conexões recursivas;
- f_h: representa a função de transformação não-linear dos neurônios recursivos

Já a saída produzida pela rede na iteração t, a partir do valor produzido pelo estado escondido é encontrada por:

$$o(t) = f_s(W_s h(t) + b_s) \tag{18}$$

onde,

- W_s, b_s: representam respectivamente a matriz de pesos e o vetor de bias das conexões partindo dos neurônios recursivos para os neurônios da camada de saída;
- f_s: representa a função de transformação não-linear dos neurônios de saída.

Segundo este modelo, dada uma sequência X, o processamento realizado por uma RNN de uma única camada escondida pode ser descrito como:

Portanto, RNNs são parametrizadas por 3 matrizes de pesos (W_e,W_h e W_s), 2 vetores bias (b_e e b_s) e, opcionalmente, um vetor de valores para definir seu estado escondido inicial ($h0$). São esses os parâmetros a serem aprendidos por um processo de treinamento.

6.2.2.4. Treinando Redes Neurais Recorrentes

Assim como no treinamento de redes alimentadas para frente, também em RNNs o treinamento parte de uma função de perda ou erro E. Como a saída esperada neste caso é uma sequência, a função de perda tipicamente é formulada como a soma

Algorithm 3 Processamento realizado por uma RNN

1: **procedure** RNNRUN(X, N_s, $h0$)
2: $h(0) \leftarrow h0$. ▷ inicializa estado oculto
3: $t \leftarrow 1$.
4: **for** $t <= N_s$ **do**
5: $z_h(t) \leftarrow W_e X(t) + W_h h(t-1) + b_e$ ▷ combinação linear do sinal de entrada e estado oculto
6: $h(t) = f_h(z_h(t))$ ▷ atualiza estado oculto
7: $z_s(t) \leftarrow W_s h(t) + b_s$ ▷ combinação linear dos resultados da camada escondida
8: $o(t) = f_s(z_s(t))$ ▷ calcula saída
9: $t \leftarrow t+1$

dos erros de cada instante t comparando elemento a elemento a sequência desejada $y = \{y(1), y(2), ..., y(N_s)\}$ com a produzida pela rede $o = \{o(1), o(2), ..., o(N_s)\}$. Mais formalmente:

$$E(o,y) = \sum_{t=1}^{N_s} E(o(t), y(t)) \tag{19}$$

O treinamento da RNN pode ser realizado como uma otimização por gradiente descendente, de maneira semelhante ao que foi apresentado na Seção 6.2.2.2. Neste caso o vetor gradiente da função de erro é calculado em relação aos 6 parâmetros de configuração da RNN que se deseja aprender.

De maneira semelhante ao treinamento apresentado para MLP, o algoritmo chamado Retropropagação Através do Tempo (do inglês *backpropagation through time*) aplica a regra da cadeia para decompor as derivadas parciais da função de erro da Equação 19 nas etapas de processamento da RNN da saída da rede até os respectivos parâmetros. Para isso, desenrola a RNN em uma rede sem laços. Isso é possível porque a RNN aplicada a uma sequência pode ser pensada como cópias de uma mesma rede, cada uma passando uma mensagem a cópia sucessora. O desenrolar é feito replicando a rede RNN e seus parâmetros em tantas cópias quantos forem os elementos da sequência que se deseja processar, e transformando as conexões de retroalimentação em conexões de alimentação para frente entre o neurônio de uma determinada cópias e os neurônios da cópia seguinte ao longo da sequência de redes criada. Assim, 'desenrola-se' a RNN ao longo do tempo em uma rede de alimentação para frente, mas com parâmetros comuns compartilhados entre as cópias (Figura 6.16).

Enquanto as ativações que trafegam na rede são dinâmicas e estão relacionadas a um determinado instante de processamento, por outro lado os parâmetros de configuração da rede são fixos durante todo o processamento de uma sequência e portanto são compartilhados nas N_s cópias da rede. Por esse motivo, no cálculo das derivadas do erro parcial em cada um deles, tais erros são acumulados ao longo da

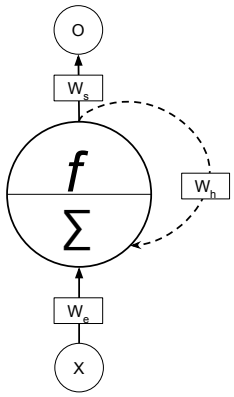

Figura 6.15: Ilustração simplificada da RNN agrupando conexões da mesma natureza

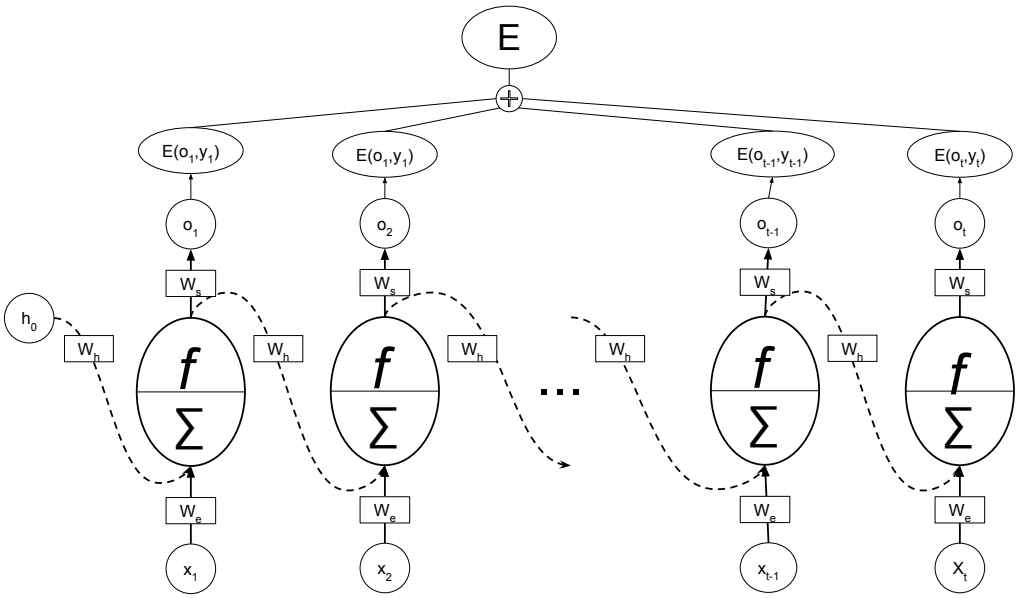

Figura 6.16: Erro aplicado ao modelo de RNN simplificado 'desenrolado'

sequência:

$$\frac{\partial E(o,y)}{\partial W_s} = \sum_{i=1}^{T} \frac{\partial E(o(t),y(t))}{\partial z_s(t)} h(t)^\intercal = \sum_{i=1}^{T} \delta z_s(t) h(t)^\intercal \tag{20}$$

$$\frac{\partial E(o,y)}{\partial b_s} = \sum_{i=1}^{T} \frac{\partial E(o(t),y(t))}{\partial z_s(t)} = \sum_{i=1}^{T} \delta z_s(t) \tag{21}$$

$$\frac{\partial E(o,y)}{\partial W_h} = \sum_{i=1}^{T} \frac{\partial E(o(t),y(t))}{\partial z_h(t)} h(t-1)^\intercal = \sum_{i=1}^{T} \delta z_h(t) h(t-1)^\intercal \tag{22}$$

$$\frac{\partial E(o,y)}{\partial W_e} = \sum_{i=1}^{T} \frac{\partial E(o(t),y(t))}{\partial z_h(t)} X(t)^\intercal = \sum_{i=1}^{T} \delta z_h(t) X(t)^\intercal \tag{23}$$

$$\frac{\partial E(o,y)}{\partial b_e} = \sum_{i=1}^{T} \frac{\partial E(o(t),y(t))}{\partial z_s(t)} = \sum_{i=1}^{T} \delta z_h(t) \tag{24}$$

Assim, utilizando a regra da cadeia, o algoritmo de retropropagação do erro através do tempo é descrito em pseudo-código como:

Algorithm 4 Retropropagação através do tempo

1: **procedure** RNNBACK(X, N_s, $h0$)
2: o = RNNrun(X, N_s, $h0$)
3: $t \leftarrow N_s$.
4: **for** $t > 0$ **do**
5: $dz_s(t) \leftarrow df_s(z_s(t)) \cdot dL(o(t), y(t))/do(t)$ ▷ δz_s
6: $db_s \leftarrow db_s + dz_s(t)$
7: $dW_s \leftarrow dW_s + dz_s(t)h(t)^\intercal$
8: $dh(t) \leftarrow dh(t) + W_s^\intercal dz_s(t)$ ▷ parte 1
9: $dz_h(t) \leftarrow df_h(z_h(t))dh(t)$ ▷ δz_h
10: $dW_e \leftarrow dW_e + dz_h(t)X(t)^\intercal$
11: $db_e \leftarrow db_e + dz_h(t)$
12: $dW_h \leftarrow dW_h + dz_h(t)h(t-1)^\intercal$
13: $dh(t-1) \leftarrow W_h^\intercal dz_h(t)$ ▷ parte 2
14: $t \leftarrow t-1$
return $[dW_e, dW_h, dW_s, db_e, db_s, dh(0)]$

Uma observação é que, assim como o sinal percorre a rede em duas direções (direção de emissão da saída $o(t)$ e direção de atualização do estado oculto $h(t)$), também o erro em relação a h deve ser retropropagado por esses dois caminhos. Eles podem ser identificados respectivamente no pseudo código como a parte 1 e parte 2 de contribuição para a dh.

Na prática, o treinamento de RNNs apresenta dois problemas, inicialmente apontados por Bengio et al. [Bengio et al. 1994], decorrentes da propagação do erro através de múltiplas iterações não-lineares: explosão do gradiente e fuga/diluição do gradiente. Seu detalhamento formal pode ser consultado em [Pascanu et al. 2013].

O problema dito explosão do gradiente refere-se a observação de um grande aumento da norma do vetor gradiente durante o treinamento de relações de longo prazo. Neste efeito, as derivadas da função de perda de um determinado instante em relação às ativações de muitos passos atrás podem ser exponencialmente grandes. Tal aumento é causado por um efeito borboleta na retropropagação do erro por longas cadeias, onde uma pequena alteração resulta em um grande efeito muitas iterações depois.

O problema da diluição/desaparecimento do gradiente refere-se ao comportamento oposto e mais frequentemente observado, quando componentes da retropropagação do gradiente de longo prazo vão exponencialmente rápido para a norma 0, impossibilitando o modelo de aprender correlação entre eventos temporalmente distantes.

Mesmo quando os parâmetros assumem valores que estabilizam o gradiente (sem sumir ou explodir), a dificuldade do aprendizado de dependências de longo alcance permanece. Isso pelo fato de que a multiplicação repetida das Matrizes

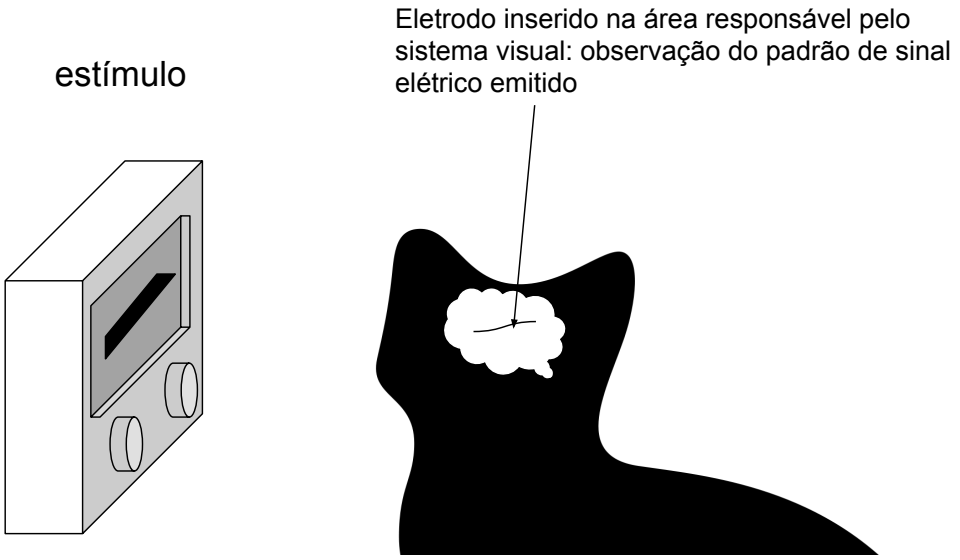

Figura 6.17: Observação dos campos receptivos de mamíferos com estímulo de diferentes padrões visuais [Hubel and Wiesel 1959, Hubel and Wiesel 1968, Hubel and Wiesel 2005]

Jacobianas [1], por conta da retro-propagação do gradiente por diversas iterações, produz pesos exponencialmente menores para aprendizado de dependências a longo prazo quando comparados a dependências de curto prazo.

Tais motivos fazem com que seja difícil treinar RNNs em sua formulação original em sequências com dependências temporais de longo alcance. Felizmente, há arquiteturas que buscam reduzir tais problemas, conforme apresentado na Seção 6.4.

6.3. Redes Neurais de Convolução

Esta seção apresenta as chamadas Redes Neurais de Convolução (do inglês *Convolutional Neural Networks – CNN ou ConvNets*), que são redes com arquiteturas de alimentação para frente especialmente projetadas para processamento de sinais visuais.

CNNs foram inspiradas pelas descobertas dos neurocientistas Hubel e Wiesel nas décadas de 1950 e 1960 sobre a organização do córtex visual de animais (Figura 6.17) [Hubel and Wiesel 1959, Hubel and Wiesel 1968, Hubel and Wiesel 2005]. Nele há neurônios que individualmente respondem a pequenas regiões do campo visual. A subregião do campo visual observado que dispara um determinado neurônio é chamada seu campo receptivo.

Observaram também um padrão relacionando a distribuição espacial dos

[1] A Matriz Jacobiana é composta pelas derivadas parciais de um ponto específico x de uma função de múltiplas variáveis. No algoritmo de retropropagação ela indica como uma pequena mudança no estado h se propaga (e se retro-propaga) e é multiplicada por si mesma por conta da propagação do erro pelas conexões recorrentes.

neurônios com a de seus respectivos campos receptivos segundo o qual neurônios vizinhos apresentam campos receptivos semelhantes e com sobreposição entre si, de maneira a formar em conjunto um mapa completo do campo visual observado. Diferentes mapas são formados com variação do padrão de tamanho e localização dos campos receptivos e em níveis crescentes de abstração semântica. Essas observações guiaram a base da modelagem das arquiteturas de CNNs, conforme descrito a seguir (Figura 6.18).

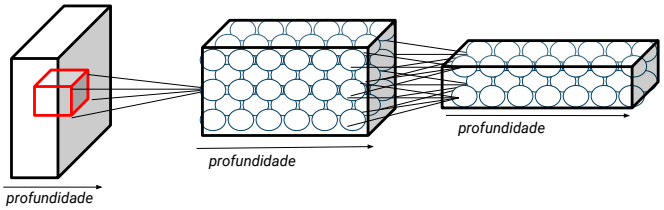

Figura 6.18: Organização espacial dos neurônios em CNNs em grades de altura e largura coerentes com o sinal de recebido como entrada por cada camada e profundidade correspondente ao número de mapas produzidos pela camada

Com exceção da primeira e das últimas camadas de uma CNN, os neurônios de cada camada são organizados em uma grade com altura e largura suficientes para cobertura do sinal de alimentação correspondente e profundidade representando diferentes processamentos sobre tal sinal. Nesta organização, uma fatia da grade é obtida fixando-se uma profundidade. Neurônios vizinhos no interior de uma fatia possuem sobreposição entre suas conexões de entrada, simulando a sobreposição de campos receptivos do córtex. Já os neurônios em uma determinada altura e largura fixas, possuem o mesmo campo receptivo, mas realizam diferentes operações sobre o campo observado (Figura 6.19).

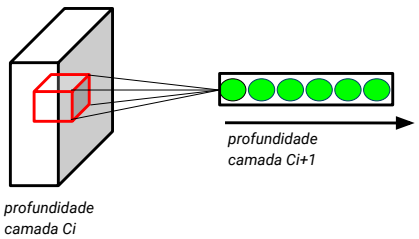

Figura 6.19: Neurônios de uma mesma altura e largura em uma determinada camada da CNN compartilham campos receptivos, mas realizam diferentes operações sobre o sinal de entrada

Os campos receptivos, por sua vez, são modelados limitando as conexões de entrada em cada neurônio apenas a elementos no interior de uma determinada vizinhança estabelecida sobre os sinais de camadas anteriores. Observe que essa modelagem das conexões apenas sobre uma vizinhança difere do modelo mais geral de redes alimentadas para frente no qual um neurônios de uma determinada camada é conectado a todos os neurônios da camada anterior e por esse motivo são ditas camadas completamente conectadas.

Sobre as atividades desempenhadas pelas células do córtex visual, Hubel e Wiesel observaram dois tipos de comportamento: o das células simples, que têm seu disparo maximizado quando seu campo receptivo apresenta arestas em orientações particulares, e o das células complexas, cujo disparo é insensível a posição exata das arestas no campo observado, e que cobrem campos receptivos maiores do que os das células simples. Esses dois tipos de células inspiram respectivamente as chamadas camadas de convolução e camadas de agrupamento (do inglês *pooling*) presentes de maneira intercalada em redes CNN.

As camadas de convolução de uma CNN realizam a extração de padrões visuais tais como arestas, texturas, motivos e ainda outros padrões visuais de maior significado semântico. Quando a aplicação da CNN não se tratar de imagens, as camadas de convolução são responsáveis pela extração de características relevantes ao tipo de sinal sendo processado. Para esse fim, desempenham operações semelhantes a aplicação de filtros pelo operador de convolução adotado na área de Processamento de Imagens e de Sinais em geral (Figura 6.23). A convolução de um sinal 1D de domínio discreto F por um filtro W de tamanho $s = (2*k+1)$ é o operador linear definido como:

$$(F*W)[x] = \sum_{i=-k}^{k} F[x-i]W[i] \tag{25}$$

A convolução é replicada por toda a extensão de f variando-se x para obter o sinal de saída 1D. A Figura 6.20 ilustra a aplicação de um filtro onde $s = 3$.

Figura 6.20: Convolução de um sinal 1D por um filtro de tamanho 3

De maneira semelhante, a convolução pode ser definida para operar em sinais com mais dimensões. Seja F um sinal 2D e W um filtro de tamanho $(2*k+1)\times(2*k+1)$, a convolução $(F*W)$ é o operador linear definido como:

$$(F*W)[x,y] = \sum_{i=-k}^{k}\sum_{j=-k}^{k} F[x-i][y-j]W[i][j] \tag{26}$$

A convolução é replicada por toda a extensão de F variando-se x,y para obter o sinal de saída 2D (Figura 6.21).

Enquanto na área de Processamento de Sinais os valores dos filtros são cuidadosamente definidos por especialistas de maneira a representar uma transformação desejada, as CNNs realizam convoluções por filtros cujos pesos (W) são aprendidos pelo processo de treinamento.

O conjunto de neurônios de uma fatia replica o mesmo operador de convolução a diferentes regiões do sinal de entrada de maneira a representar a aplicação de um determinado filtro sobre todo o sinal (Figura 6.22). Para isso, os neurônios de uma

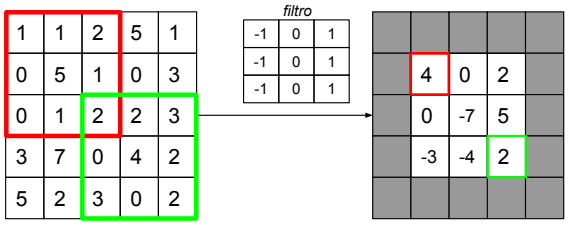

Figura 6.21: Convolução de um sinal 2D por um filtro de tamanho 3x3. Elementos em cinza descartados pelo filtro ultrapassar a borda. De maneira alternativa, podem ser calculados supondo zero elementos fora do mapa de características (*zero padding*)

fatia compartilham entre si seus pesos, fazendo com que a união de suas saídas formem um mapa de uma determinada característica extraída do sinal observado.

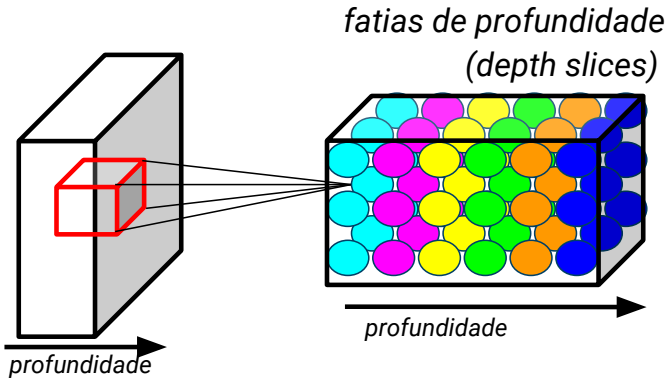

Figura 6.22: Fatiamento dos neurônios de uma CNN: neurônios de uma mesma fatia compartilham pesos, mas aplicam a convolução a diferentes campos receptivos de maneira a cobrir toda o sinal de entrada

Como consequência, juntas as respostas às fatias de uma grade produzem um conjunto de diferentes mapas de características sobre o sinal observado. A Figura 6.23 ilustra um exemplo de conjunto de filtros aprendidos em uma primeira camada de convolução de filtros de $7x7$ elementos.

Sobre o compartilhamento de pesos, cabe ressaltar que além de simular convoluções, é a propriedade responsável por viabilizar a aplicação de redes neurais em dados espacialmente densos, tal como imagens, pela significativa redução de parâmetros na rede. Essa redução acontece em comparação ao modelo padrão de redes alimentadas para frente, já que ao invés de cada neurônio receber como entrada todos os elementos produzidos na camada anterior, em CNNs para a produção de cada mapa de características o número de pesos é limitado ao tamanho do filtro aplicado (altura do filtro$\times$ largura do filtro $\times$ profundidade da camada anterior).

Sobre a profundidade dos filtros, tipicamente a primeira camada de convolução processa uma quantidade pequena de canais de entrada (um para imagens em tons de cinza, três para canais em cores, entre outros), enquanto que as demais camadas passam a processar a quantidade de fatias geradas como mapas de carac-

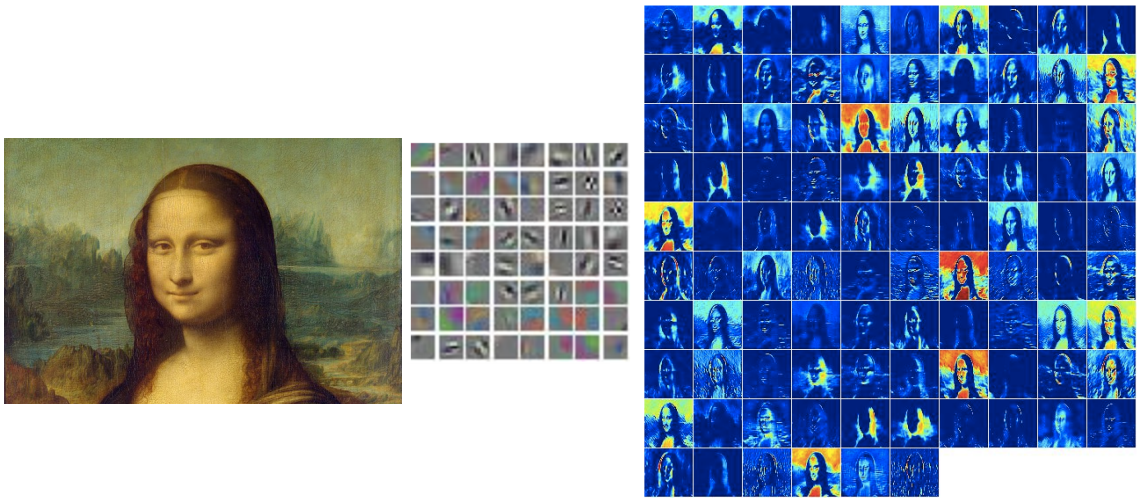

Figura 6.23: Exemplo de filtros aprendidos pela primeira camada de convolução e saída da ativação desses filtros ao apresentar uma face para a rede

terísticas distintos na camada que as alimenta. Ressaltando que a convolução opera ponderando elementos no interior mas também entre fatias.

Ao longo da CNN, é construída uma hierarquia de características, na qual as camadas iniciais extraem características elementares, como bordas e cantos, que são combinadas sucessivamente para a extração de características de mais alto nível semântico nas camadas seguintes.

Os neurônios da camada de convolução se completam com a escolha de uma função de ativação, sendo a rectificação linear (do inglês *Rectified Linear Unit* – ReLU) a mais popular em CNNs recentes. Pautada em observações biológicas, foi proposta por Hahnloser et al. [Hahnloser et al. 2000] e é definida como:

$$f(x) = max(0,x) \tag{27}$$

A ReLU é diferenciável para todo x diferente de zero, sendo sua derivada constante em 1 para $x > 0$, e zero nos demais casos. Essa função tem importantes propriedades tais como não saturar, não apresentar problemas de explosão nem desaparecimento do gradiente, produzir ativações esparsas, além de simplificar os passos de propagação e retro-propagação do gradiente ao diminuir a complexidade dos cálculos envolvidos. Por esses motivos é observado que o uso da ReLU acelera o aprendizado em CNNs profundas, e tem permitido o treinamento de topologias cada vez mais profundas (ou seja, com crescente número de camadas).

Neurônios com ativação ReLU podem vir a 'morrer' e passar a não aprender durante os passos seguintes do treinamento. Esse fenômeno ocorre quando todas as amostras apresentadas a rede são classificadas por tal neurônio com zero. Esse caso induz gradientes nulos e portanto, a parada do aprendizado. Para que um neurônio seja considerado vivo durante o treinamento, basta que ao longo de uma época pelo menos algumas das amostras sejam por ele classificadas com saídas não-zero, pois em consequência induzem sua atualização.

Nas CNNs, seguindo a observação do que acontece em neurônios biológicos, as camadas de convolução são intercaladas com camadas com maior campo receptivo. Para isso, as camadas de agrupamento (do inglês *pooling*) de CNNs realizam uma operação de re-escalonamento do sinal (do inglês *downsampling*) ao longo das dimensões de altura e largura da grade, mantendo o número de fatias inalterado.

O agrupamento de ativações vizinhas nos mapas de características é feito em sub-regiões de tamanho pré-determinado na topologia da rede (como um hiperparâmetro), com ou sem sobreposição entre tais sub-regiões de acordo com o passo entre regiões. Dado um sinal de entrada de dimensões $w_i \times h_i \times d_i$, um passo s e um campo receptivo de tamanho f, após o agrupamento obtêm-se uma saída de dimensão $w_o \times h_o \times d_o$ por:

$$w_o = (w_i - f) \div s + 1 \tag{28}$$

$$h_o = (h_i - f) \div s + 1 \tag{29}$$

$$d_o = d_i \tag{30}$$

Ao mesmo tempo que o agrupamento produz neurônios com maior campo receptivo, impõe uma invariância na localização das características no interior das subregiões agrupadas.

Sobre a operação realizada pelos neurônios das camadas de agrupamento, em CNNs recentes a operação de máximo local tem sido a mais popular (Figura 6.24) embora historicamente outras operações tenham sido usadas, como por exemplo agrupamento *L2-norm* ou média local [LeCun et al. 1998]. Neste caso são chamadas camadas de *max-pooling*.

Ainda que tenham sido inspiradas pela organização do córtex visual de organismos vivos, as CNNs têm se mostrado robustas no processamento de uma variedade de outros sinais descritos na forma matricial, não necessariamente imagens, abrindo seu leque de aplicações. Como exemplos de sinais analisados com sucesso por CNNs podemos citar entre outros: sequências de linguagem natural na forma de vetores 1D; imagens em tons de cinza e espectogramas de áudio na forma de vetores 2D; imagens com profundidade, vídeos e dados volumétricos na forma de vetores 3D, entre outros.

Esse sucesso se dá por conta de quatro ideias chaves da arquitetura CNN descritas nesta seção que tomam proveito das características de sinais naturais: adoção de conexões locais, compartilhamento de pesos, agrupamento (*pooling*) e a construção em múltiplas camadas provendo uma hierarquia de características.

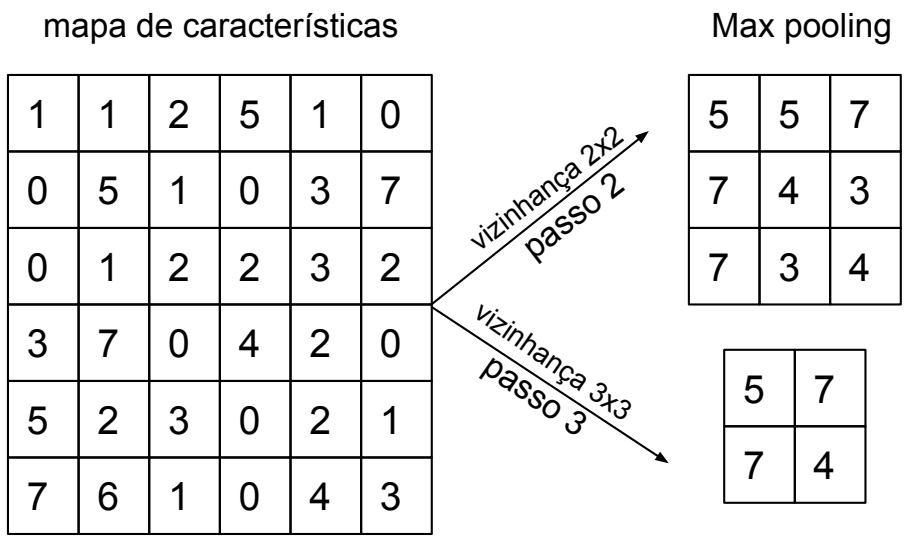

Figura 6.24: Ilustração aplicação de agrupamentos sobre mapa de características de $6x6$, com agrupamento por filtro de $2x2$ e passo de agrupamento 2, e agrupamento por filtro de $3x3$ e passo de agrupamento 3

A última camada de CNNs está associada ao problema que se deseja resolver. Em problemas de regressão, em que as saídas são valores reais, tipicamente é adotada função de perda Euclidiana nos valores emitidos pelos neurônios desta camada.

Em problemas de classificação, a última camada é normalmente definida como uma camada com k neurônios, onde cada um representa uma determinada classe, completamente conectada à anterior. Para predição de K classes mutualmente exclusivas, adota-se tipicamente nessa camada a função de ativação *Softmax*, também chamada de função exponencial normalizada que é definida como:

$$\sigma(Z_c) = \frac{e^{Z_c}}{\sum_{j=1}^{K} e^{Z_j}} \ \texttt{para c} = 1, \cdots, k. \tag{31}$$

onde Z_c representa a transformação linear do neurônio c, ou seja, a combinação linear do vetor de ativações X produzidas pela camada anterior pelo pesos W_c de suas conexões.

A função *Softmax* escala o vetor de saída da última camada da CNN de maneira que seus elementos pertencem ao intervalo $[0,1]$ e que somados sejam 1. Com isso, modela uma distribuição de probabilidade discreta da saída da rede Y pertencer a uma classe c entre as k possíveis saídas da rede, observadas as ativações X recebidas pela última camada:

$$P(X = c|\mathbf{X}) = \frac{e^{\mathbf{X}^\intercal \mathbf{W}_c}}{\sum_{j=1}^{K} e^{\mathbf{X}^\intercal \mathbf{W}_j}} = \tag{32}$$

É importante ainda mencionar que, com a intenção de melhorar a performance das CNNs, durante seu treinamento é comum a adoção do método de *dropout* [Srivastava et al. 2014]. O método de *dropout* busca impedir que um neurônio dependa fortemente de algum outro para fazer suas previsões em um fenômeno de coadaptação.

O *dropout* consiste do desligamento aleatório de neurônios antes de cada passo de propagação de amostras durante o treinamento. Os neurônios desligados não contribuem para o passo de propagação do sinal e desta forma não contribuem durante a retropropagação do erro. Com isso, a cada iteração do treinamento, o mecanismo de sorteio simula uma rede com topologia diferente, e força a quebra de dependências. Assim, faz com que os neurônios aprendam características mais robustas e relevantes a partir das diferentes combinações geradas.

6.3.1. Estudo de caso

Existem diversas arquiteturas de CNNs em um enorme leque de aplicações. Estão a seguir descritas em alto nível algumas delas, selecionadas por questões históricas do desenvolvimento de CNNs ou ainda por sua relevância frente a desafios de classificação de imagens em grandes bases. Mais especificamente, estão descritas as arquiteturas vencedoras das edições 2012 a 2015 do desafio de classificação do *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* [Russakovsky et al. 2015]). São elas:

- LeNet [LeCun et al. 2001]: arquitetura responsável por realizar as primeiras aplicações de sucesso de CNNs como o reconhecimento de dígitos em processamento de cheques. O uso de camadas alternadas simulando células simples e complexas já tinha sido usado anteriormente no modelo denominado neocognitron [Fukushima 1980], que é treinado por uma heurística. A LeNet introduz o treinamento de CNNs pelo algoritmo de retropropagação, e o compartilhamento dos pesos conforme adotado até hoje [LeCun et al. 1989].

  A arquitetura LeNet-5 consiste de uma rede de 7-camadas mais a entrada (duas de convolução alternadas com duas de agrupamento, três completamente conectadas sendo a última a da camada de saída) sendo usada para classificar dígitos em imagens monocromáticas de 32×32 pixels (Figura 6.25).

- AlexNet [Krizhevsky et al. 2012]: arquitetura vencedora do desafio ILSVRC em 2012, no qual obteve 16% de erro em comparação ao segundo lugar com 26% error na classificação *top-5*, chamando a atenção para o potencial das CNNs profundas. Em comparação com a LeNet, a Alexnet é maior, mais profunda, trabalhando com imagens coloridas e em maior resolução, possui maior número de mapas de características empilhados por camada e se diferencia também por adotar a função de ativação ReLU, agrupamento por *max-pooling* além de ser treinada em GPU. A arquitetura AlexNet consiste de cinco camadas de convolução alternadas com agrupamento, seguidas de três camadas completamente conectadas seguidas da camada de saída com função de ativação *Softmax*(Figura 6.26).

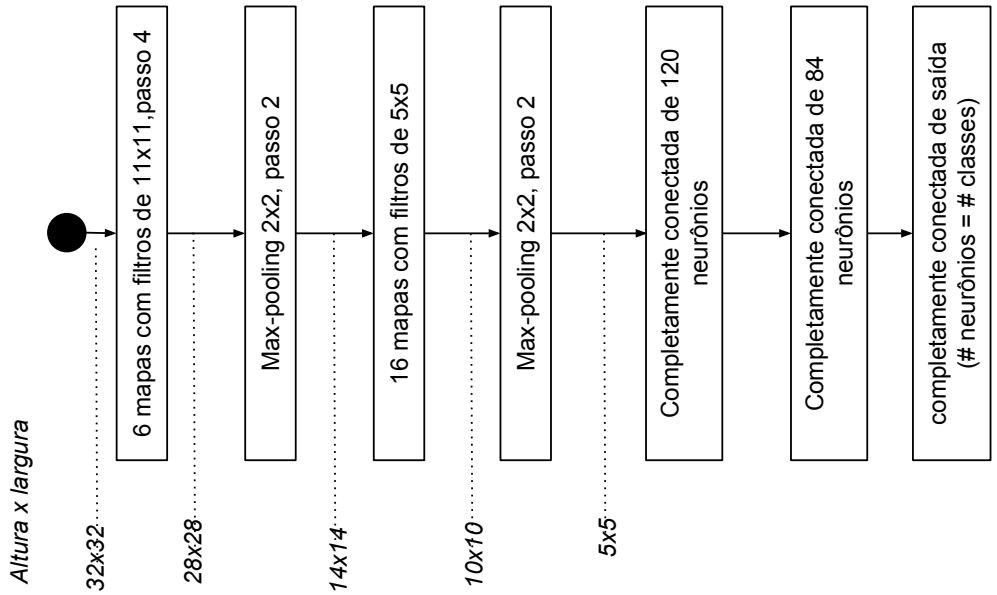

Figura 6.25: Ilustração da arquitetura LeNet-5 [LeCun et al. 2001]

- ZF Net[Zeiler and Fergus 2013]: arquitetura vencedora do ILSVRC 2013 fez importantes melhorias sobre os hiperparâmetros que definem a AlexNet expandindo o tamanho das camadas de convolução intermediárias e reduzindo a passada e o tamanho dos filtros da primeira camada de convolução. Este trabalho também propõe camadas de reversão da rede, na estrutura chamada deconvnet, permitindo a visualização das ativações produzidas pelos mapas de características ao longo da hierarquia de camadas da CNN.

- GoogLeNet [Szegedy et al. 2015a]: arquitetura de 22 camadas vencedora da edição de 2014 do ILSVRC, desenvolvida por pesquisadores do Google (Figura 6.27). Propõe um módulo denominado *Inception* o qual reduziu consideravelmente o número de parâmetros da rede em comparação com a arquitetura AlexNet (de $60M$ para $4M$). O módulo inception combina filtros de diferentes tamanhos criando camadas mistas e mais largas inspirados na proposta de rede dentro da rede [Lin et al. 2013]. Além disso, elimina as camadas completamente conectadas tipicamente usadas no topo de CNNs, e coloca em seu lugar uma camada de agrupamento por média (*average pooling*), com isso eliminando mais parâmetros de treinamento. Melhorias do modelo original da GoogLeNet foram propostas por [Szegedy et al. 2015b, Szegedy et al. 2016]

- VGGNet [Simonyan and Zisserman 2014]: arquitetura vencedora do desafio de localização e segundo lugar no desafio de classificação da edição do ILSVRC 2014. Apontou a profundidade da rede como componente crucial para melhoria de performance. A VGGNet é uma arquitetura extremamente homogênea em dois modelos, com 16 ou 19 camadas de convolução alternadas com agrupamento, seguidas por 3 camadas completamente conectadas no final da rede. Utiliza filtros de 3×3 por toda a rede com agrupamentos em janelas 2×2. Em comparação a GoogLeNet, possui maior demanda por memória e parâmetros

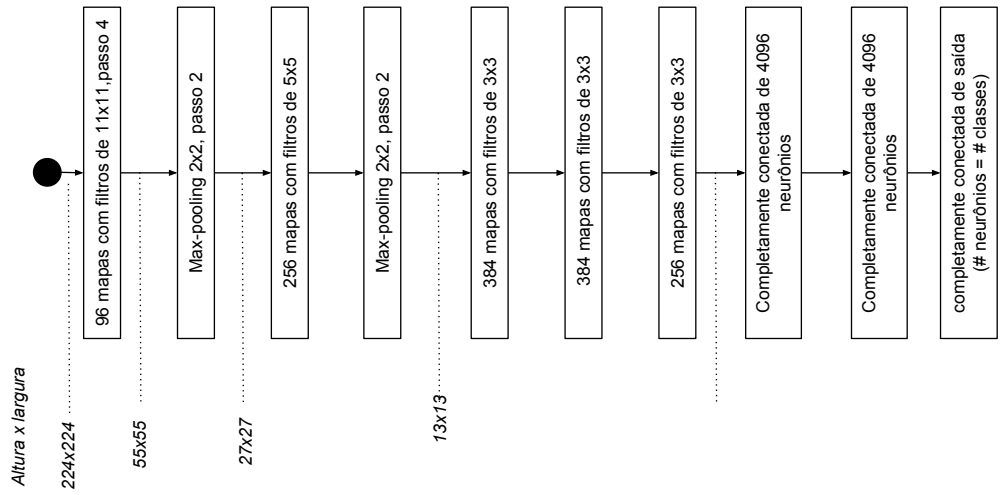

Figura 6.26: Ilustração da arquitetura AlexNet proposta em [Krizhevsky et al. 2012]

(140M), mas muitos provindos das camadas completamente conectadas, o que pode ser substituido sem perda de performance.

- ResNet [He et al. 2015]: arquitetura vencedora da edição ILSVRC 2015. Suas principais contribuições são a inclusão de conexões especiais chamadas de *skip connections* além do uso de normalização de lotes. Juntas permitiram o treinamento de redes consideravelmente mais profundas chegando a 1K camadas. A ResNet é composta por múltiplas de camadas de convolução com filtros 3×3 (com exceção da primeira) e agrupamento 2×2, combinadas com *skip connections*. A Figura 6.28 ilustra a arquitetura ResNet com 34 camadas, sendo a vencedora do ILSVRC composta por 152 camadas. Seguindo a linha do estado da arte até então, as redes ResNet não possuem camadas completamente conectadas no final da rede. Sendo sua versão melhorada descrita no artigo [He et al. 2016].

6.4. Redes Neurais Recorrentes: LSTMs

Conforme apresentado na Seção 6.2.2.3, o aprendizado de dependências de longo prazo em redes neurais recorrentes é afetado pelo fato de que a repetida multiplicação das matrizes Jacobianas para a retro-propagação dos gradientes ao longo das conexões recursivas tendem numericamente a explodir ou a desaparecer.

Diferentes abordagens foram desenvolvidas buscando contornar os problemas no aprendizado de dependências de longo prazo, embora ainda seja considerado um dos principais desafios de pesquisa na área. Entre as abordagens desenvolvidas podemos citar: as *Echo State Networks – ESN* [Jaeger and Haas 2004]; as *Liquid State Machines – LSM* [Maass et al. 2002]; redes com conexões entre instantes de tempo não consecutivos (maiores do que entre t e $t+1$) como o uso de *skip connections* [Lin et al. 1995]; as redes com barreira (*gates*) como nos modelos *long-shot term memory – LSTM* [Hochreiter and Schmidhuber 1997] e *Gated Recurrent Units* (GRU) [Cho et al. 2014]; entre outros. O modelo proposto pelas redes LSTM tem

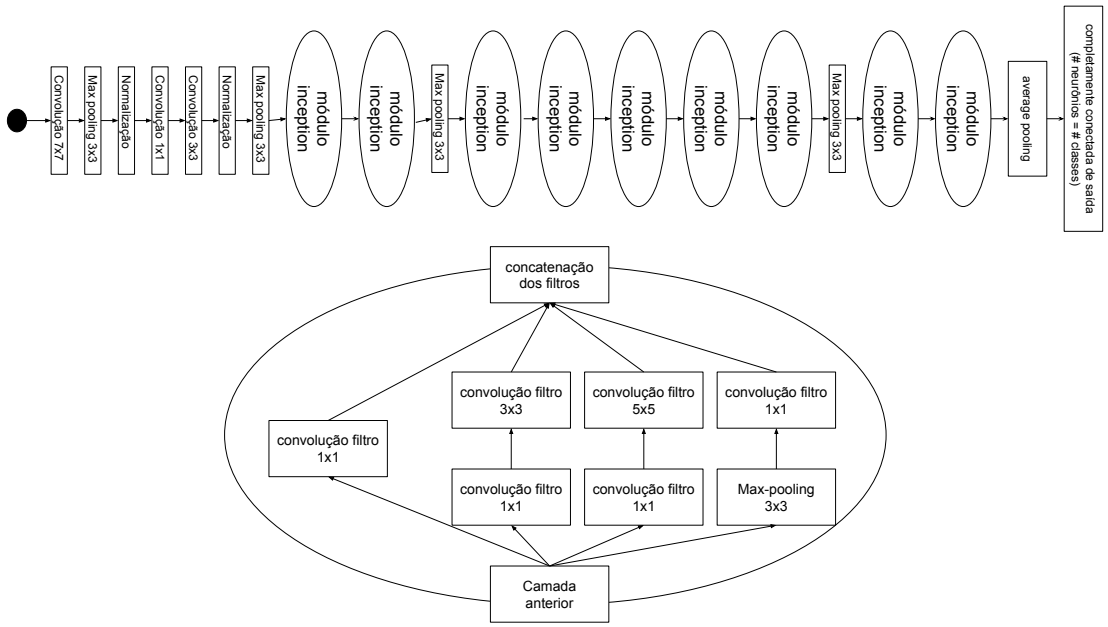

Figura 6.27: Ilustração da Arquitetura GoogLeNet e seu módulo Inception propostos em [Szegedy et al. 2015a]

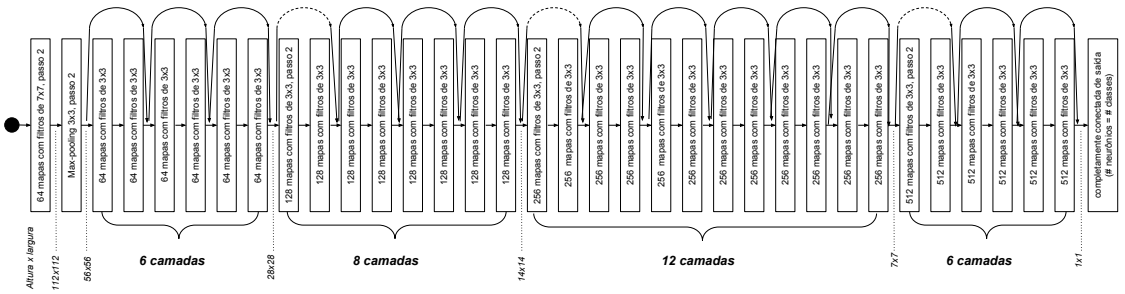

Figura 6.28: Ilustração da arquitetura ResNet de 34 camadas proposta em [He et al. 2015]. Camadas de convolução com filtros em sua maioria de tamanho 3x3, e inclusão das *skip connections* a cada par de camadas de convolução

sido a proposta com maior adoção e em diferentes aplicações e por esse motivo é apresentado nesta seção.

A ideia básica das LSTMs para tratar os problemas de treinamento de dependências de longo prazo de RNNs é baseada na criação de caminhos pelos quais a retropropagação do gradiente possa fluir.

Sua arquitetura é baseada no conceito de células que externamente funcionam como neurônios de uma RNN no sentido que recebem as mesmas entradas (diretas ou recorrentes) e emitem as mesmas saídas de um neurônio da RNN.

A diferença está no funcionamento interno de tais células. Os neurônios de uma RNN tradicional aplicam uma função de ativação não linear sobre a combinação linear de suas entradas (diretas e recorrentes), conforme apresentado na Seção 6.2.2.3. Já nas células da rede LSTM, além da recorrência original de RNN (que

passa a ser considerada uma recorrência externa a célula) possuem também conexões de recorrência internas a própria célula. Essas novas conexões possuem o papel de controlar a leitura e escrita do estado atual, criando um mecanismo dinâmico de controle do fluxo da informação, simulando barreiras (do inglês *gates*) no caminho da retro-alimentação dos estados.

As barreiras em uma LSTM criam conexões de maneira que a rede permite guardar informações durante um longo período de tempo. Além de poder guardar, uma vez que a informação for utilizada, também criam um mecanismo para que a rede possa esquecer uma determinada informação (associando zero ao estado correspondente). A LSTM é capaz de aprender quando o estado pode ser esquecido, portanto reagindo ao contexto da sequência sendo analisada. Assim, a LSTM consegue controlar dinamicamente de acordo com o contexto quando guardar ou esquecer informações mantidas em seus estados.

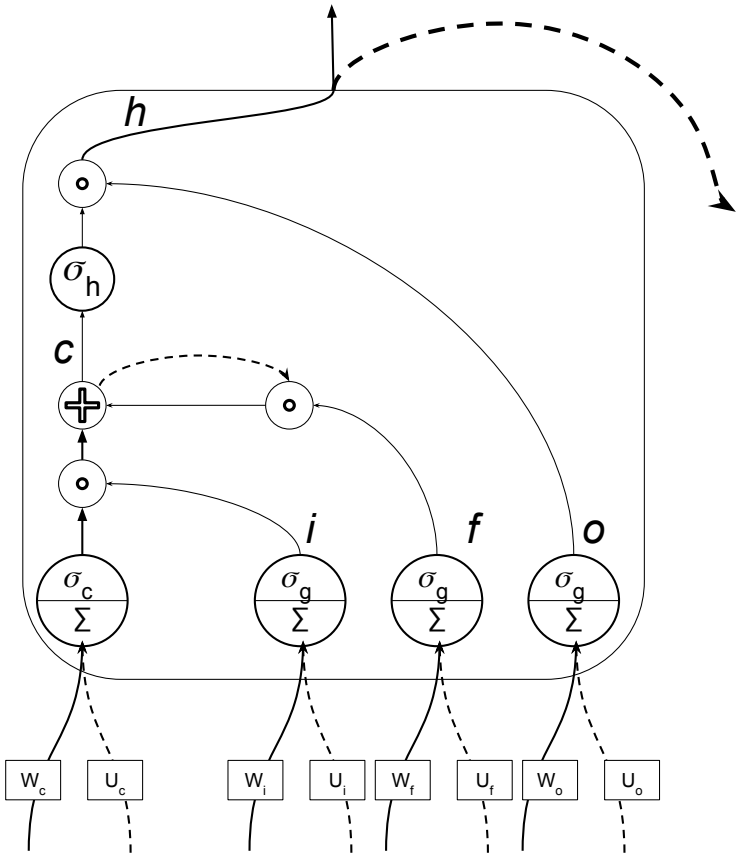

Figura 6.29: Ilustração de uma célula LSTM que mantém as entradas e saídas originais de uma RNN tradicional, mas cria internamente mecanismos de barreira (*gates*) que controlam o fluxo da informação i, f e o, para controle de escrita, limpeza e leitura da célula c

O ponto central da célula LSTM é seu estado $S(t)$, o qual pode fluir através da célula para permitir que a informação trafegue tanto diretamente sem ser alterada, ou ainda permitindo ser alterada de acordo com as conexões de barreira.

Mais especificamente, as conexões de barreira de uma célula LSTM são de três tipos definidas a seguir. Uma barreira i que limita a entrada de informação para a célula, controlando portanto a operação de escrita na célula (do inglês *input gate*). Uma barreira o que limita a saída da célula, controlando a leitura/consulta da informação contida na célula (do inglês *output gate*). Uma barreira f que gerencia a manutenção ou esquecimento de valores na célula, controlando a limpeza da informação contida na célula (do inglês *forget gate*). Sendo cada uma das três controladas por seus próprios pesos aprendidos durante o treinamento, de maneira a reagirem com o contexto (informação de entrada) para controle de suas funções. São definidas por:

$$f(t) = \sigma_g(W_f x(t) + U_f h(t-1) + b_f) \tag{33}$$
$$i(t) = \sigma_g(W_i x(t) + U_i h(t-1) + b_i) \tag{34}$$
$$o(t) = \sigma_g(W_o x(t) + U_o h(t-1) + b_o) \tag{35}$$

onde W_f,W_i,W_o são os pesos e b_f,b_i,b_o seus bias associados as conexões de entrada para as respectivas barreiras f, i e o; enquanto que U_f,U_i,U_o representam os pesos associados a retroalimentação da saída da célula para suas barreiras. As funções de ativação σ_g tipicamente usadas nas barreiras são do tipo função sigmóide (portanto, emitem saída no intervalo $[0,1]$). Com essa formulação, as três barreiras reagem dinamicamente a cada instante t em reação ao sinal recebido como entrada $x(t)$ e também a saída produzida pela célula no instante anterior $h(t-1)$.

O funcionamento da célula c por sua vez é definido por:

$$c(t) = f(t) \circ c(t-1) + i(t) \circ \sigma_c(W_c x(t) + U_c h(t-1) + b_c) \tag{36}$$

onde σ_c é tipicamente uma tangente hiperbólica e isoladamente realiza um processamento equivalente ao de neurônios de camadas escondidas em RNNs tradicionais uma vez que combina as ativações de entradas diretas $x(t)$ ponderadas por pesos W_c e bias b_c com as recorrentes ponderadas pelos pesos U_c. A diferença da LSTM está no fato que o valor produzido como saída de $\sigma_c(t)$ pode ou não ser acumulado ao estado corrente, de acordo com a barreira de entrada $i(t)$ e a manutenção os não do estado da célula ao longo de repetidas iterações é controlada pela barreira f.

Por fim, a saída da rede é controlada pela barreira o usando:

$$h(t) = o(t) \circ \sigma_h(c(t)) \tag{37}$$

onde $h(t)$ representa o vetor de saída e a função de ativação σ_h tanto é definida como uma tangente hiperbólica, como também por $\sigma_h(x) = x$ na variação das LSTM denominadas *peephole LSTM*.

O acréscimo das conexões de retroalimentação que exercem o papel de barreiras faz com que a célula LSTM possua mais parâmetros do que o modelo original RNN (apresentado na Seção 6.2.2.3). Além disso, enquanto no modelo original os pesos de atualização dos estados ocultos são fixos ao longo das iterações, ao aplicar-se a LSTM em uma sequência, a atualização dos estados ocultos ocorre dinamicamente. Isso porque ao longo do tempo tal atualização é condicionada ao contexto por seus

pesos estarem controlados por outras unidades escondidas que são as próprias barreiras.

Com essa modelagem, a LSTM consegue aprender dependências de longo prazo mais facilmente ao produzir o chamado Carrossel de Erro Constante (CEC) pois cria um mecanismo capaz de induzir derivadas de valor constante próximo de 1 ao longo de diversas iterações quando as próprias barreiras são ativadas com valores próximos de 1 fazendo com que a informação seja propagada por mais tempo que em RNNs.

6.5. GPUs

As GPUs foram parcialmente responsáveis pela revolução em Deep Learning. Como apresentado anteriormente, grande parte da matemática envolvida em redes neurais profundas corresponde a cálculos de matrizes. As arquiteturas das GPUs permitem em alguns casos acelerar algumas centenas de vezes a velocidade destes cálculos, tornando viável a resolução de problemas no que se refere ao tempo.

Neste capítulo iremos apresentar resumidamente a arquitetura das GPUs, bem como uma breve introdução em como programa-las no nível mais baixo de sua arquitetura. Embora na maioria das vezes os usuários de deep learning não venham a precisar entrar neste nível, uma compreensão básica da arquitetura pode ajudar nas buscas de otimizações ou na manipulação das bibliotecas, bem como entender de onde vem a aceleração das GPUs.

6.5.1. Breve História

O termo GPU surgiu em 1999, quando a NVIDIA lançou a GeForce 256 e a ATI a Radeon 7500. Estas duas placas gráficas passaram a se denominar processadores ao invés de simples aceleradores, pois implementavam tarefas complexas do pipeline gráfico, incluindo estágios de iluminação, processamento de vértices e pixel. O pipeline gráfico introduzido pelas GPUs recebeu o nome de pipeline de função fixa, pois todos os algoritmos de transformação geométrica, iluminação e rasterização eram implementados diretamente no hardware e manipulados via APIs gráficas. A natureza dos métodos numéricos do pipeline gráfico implicam uma resolução de cálculos de vetores e matrizes.

Em 2001 surge o pipeline gráfico programável, possibilitando que o desenvolvedor possa programar os estágios de vértice e de pixel do pipeline gráfico. Esta programação passou a ser feita por programas denominados de shaders, que eram funções invocadas pelas APIs gráficas em algum dos estágios do pipeline, possibilitando que os mesmos pudessem ser customizados e alterados. O hardware das placas desta época eram compostos por dois conjuntos de processadores, um conjunto de processadores dedicados ao processamento dos vértices e o outro conjunto de processadores de pixel. Neste momento percebeu-se que as GPUs eram poderosos processadores para resolver métodos numéricos não atrelados ao pipeline gráfico, especialmente em casos onde operações matriciais eram intensas e grandes. Em 2006 finalmente foram lançadas as primeiras GPUs com arquiteturas unificadas: os processadores de vértice e pixel passaram a ser um único processador mais genérico e

com capacidade computacional mais abrangente. Desde então, a arquitetura CUDA vem possibilitando que as GPUs possam encapsular milhares de processadores num único dispositivo.

6.5.2. Arquitetura da GPU

A arquitetura da GPU é classificada como um modelo do tipo Single Instruction, Multiple Threads (SIMT). Isto quer dizer que diversas threads executam a mesma instrução a cada ciclo de clock. Além disso, no modelo de programação do CUDA, as thread são organizadas em blocos, formando uma organização lógica das instâncias do kernel. Na parte de hardware, as GPUs são organizadas em multiprocessadores (SM - symmetric multiprocessors) idênticos [4], onde cada um dos SMs é composto por centenas de cores, memórias e registradores. Estes valores dependem da arquitetura e geração da GPU.

A execução das threads envolve o escalonamento das mesmas em dois níveis: no primeiro nível, os blocos de threads são escalonados nos SMs; e, no segundo nível, as threads de cada bloco são escalonadas nos núcleos (cores). O número de SMs pode variar de acordo com o modelo e geração da GPU. Muitas mudanças no projeto de um SM foram feitas desde a primeira versão da GPU a fim de melhorar a performance e o consumo de energia destes processadores.

Há um limite de threads que podem ser alocadas em uma GPU. Atualmente o limite de threads por blocos é de 2048. Embora haja um limite de quantidade de blocos, por ser um número muito grande (2.147.483.647), considera-se este como infinito. Por fim, conceitualmente os blocos são agrupados formando uma grid.

A GPU tem diferentes níveis hierárquicos de memórias, cada um com capacidade e velocidade de acesso diferente e são organizadas da seguinte forma:

- A memória global, que é a memória principal e pode chegar a uma capacidade de 12GB na arquitetura Maxwell. Os dados contidos nesta memória podem ser acessados por qualquer thread de qualquer bloco. Tem uma latência maior que as demais memórias e os dados armazenados podem ser vistos por diferentes kernels . A gerência desta memória, como alocação e desalocação, são feitas pela CPU.

- A memória compartilhada é uma memória de baixa latência e uma velocidade de acesso muito rápida. Cada SM tem seu próprio espaço de memória compartilhada e seu escopo é o do bloco, isto é, quando um bloco é desalocado de um SM, seus dados são apagados.

- A memória local é chamada assim porque o seu âmbito é local para a thread e não por causa de sua localização física. A memória local é off-chip, tornando o acesso a ela tão caro como o acesso à memória global. Esta memória é acessível somente para uma thread específica e os dados persistem apenas durante a execução desta thread.

- A Memória de textura corresponde a uma memória só de leitura e em cache. Ela é otimizada para localidade espacial 2D, permitindo que threads que estão próximas possam usar a mesma operação de leitura para os dados correspondentes, no caso de haver coalescência. Essa memória também é capaz de realizar interpolação

de dados, operação típica na resolução de textura anti- aliasing.

- A memória constante também é armazenada temporariamente em cache e seu acesso custa uma operação de leitura a partir do cache, caso seja evitado um cache miss. É uma pequena memória, com capacidade de 64 KB na Kepler, por exemplo.

A CPU é capaz de ler e escrever na memória global da GPU, sendo que este processo ocorre através do barramento de dados PCI-EXPRESS. Toda a comunicação entre o host (CPU) e o device (GPU) é feita através deste canal.

O escalonamento dentro de cada SM é feito em grupos de 32 threads. Assim, 32 threads consecutivas executam a mesma instrução. Ao fim da execução da última instrução, outras 32 threads consecutivas são escalonadas. Este procedimento de escalonamento é conhecido como warp. Neste sentido recomenda-se instanciar a quantidade de threads em um bloco como sendo múltiplo de 32. O melhor padrão de acesso à memória ocorre quando os dados da memória estão alinhados com as threads de um warp, ou seja, o acesso é coalescente. No procedimento de escalonamento são criados dois índices: um índice é referente ao bloco ao qual uma determinada thread pertence e o outro é o índice da thread dentro do bloco que ela pertence. Como 32 threads de um mesmo warp executam as mesma instruções, uma importante otimização é garantir que estas threads tenham o mesmo caminho de dados. Neste sentido, se há divergência no fluxo de execução do código em uma ou mais thread de um mesmo warp , algumas threads podem ficar ociosas, serializando a execução. Chama-se a este fenômeno de divergência.

Em versões mais antigas do CUDA, só era permitido criar grids com o mesmo kernel , ou seja, não existia a possibilidade de executar kernels concorrentes. Com o lançamento da arquitetura FERMI, passou a ser possível executar mais de um kernel ao mesmo tempo. Contudo, apenas a arquitetura Kepler realmente implementa a execução de kernels concorrentes a nível de hardware. Kernels concorrentes são escalonados internamente pela GPU e um SM pode executar apenas um mesmo kernel. Na prática, isto significa que cada bloco pode ter apenas threads instanciadas de um mesmo kernel e o kernel concorrente é escalonado em outro SM.

No lançamento da arquitetura Kepler foi introduzida a tecnologia Hyper-Q, que permite diferentes programas ou threads de CPU dispararem diferentes kernels em uma mesma GPU. Até então, quando um kernel era disparado, a GPU ficava ocupada executando este kernel e não recebia outros kernels de outros programas ou threads de CPU, sendo necessário um sincronismo por barreira para que uma grid fosse finalizada antes de inicializar outra.

6.5.3. CUDA

Pode-se utilizar a GPU de diversas maneiras diferentes. Em alguns casos serão usadas bibliotecas que se utilizam intensivamente da GPU para executar as suas funções, como é o caso do cuDNN. Em outros casos, o nível de abstração pode ser maior ainda, onde serão usados frontends que por sua vez usam bibliotecas e que por sua vez usam as GPUs, como é o caso do DIGITS. Entretanto, sempre que

for necessário programar diretamente na GPU, é necessário utilizar sua linguagem nativa, portanto CUDA ou OpenCL. Nesta seção estaremos focando em arquiteturas NVIDIA, portanto arquiteturas baseadas em CUDA.

Aplicações desenvolvidas para GPUs envolvem partes de código em CPU e partes de código para GPU, uma vez que as CPUs são responsáveis por gerenciamento das GPUs. Este gerenciamento consiste em alocação de memória, cópia de dados e chamada do kernel. Após a computação da GPU, a CPU copia de volta os dados da memória da GPU para a CPU. A comunicação (processo de cópia) ocorre entre a memória principal da CPU e a memória global da GPU.

A arquitetura atual da GPU PASCAL é capaz de alcançar 11 TFlops em processamento e possui largura de banda capaz de copiar até 720 GB por segundo de dados da CPU para a GPU e vice-versa. Em outras palavras, é possível copiar até 56 bilhões de números ponto flutuante por segundo, o que é uma quantidade de 65 vezes menor que a capacidade de processamento de pontos flutuantes por segundo. Por esta razão, a otimização em minimizar o tráfego de dados de/para a GPU é uma obrigação a fim de atingir o máximo de desempenho. O código de GPU é basicamente uma função, que é chamada pela CPU. Esta função é chamada de kernel e o código da figura 6.30 é um exemplo de “Hello world” em CUDA.

```
2 // GPU code
3 __global__ void the_kernel(void){
4    }
5
6 // CPU Code
7 int main(void) {
8       the_kernel<<<1,1>>>();
9       printf("Hello World!");
10 }
```

Figura 6.30: Código de um programa Hello World em CUDA.

A declaração que indica a função kernel é a declaração ”global”. Esta declaração indica que a função é executada na GPU, sendo chamada pela CPU. Neste exemplo, nenhuma computação está sendo realizada, pois não há sentido em imprimir de forma paralela várias mensagens de Hello world, o exemplo é para efeitos didáticos.

Ao chamar um kernel é necessário indicar quantas threads serão instanciadas. Como já apresentado, há uma organização das threads em blocos e o total de threads instanciadas é dado por: total de threads = quantidade de blocos x threads por blocos. No exemplo do Código da figura , as threads são instanciadas na linha 7 através da sintaxe: KERNEL <1,1>, onde o primeiro parâmetro é o número de blocos e o segundo indica o número de threads por bloco. Estes mesmos parâmetros podem também serem interpretados como: o primeiro parâmetro indica a dimensão do bloco, enquanto o segundo parâmetro descreve a dimensão do bloco.

O Código da figura 6.31 é um exemplo de aplicação que tem o objetivo de somar dois valores, onde o kernel executa uma simples operação de soma na GPU. Embora este exemplo ainda não aborde as questões inerentes ao paralelismo da GPU, pois apenas instancia uma única thread, é um exemplo que apresenta um importante estágio das aplicações de GPU que é a comunicação entre GPU e CPU.

O Código da figura 6.31, em linhas gerais, ilustra todos os estágios que

```
1 // GPU code
2 __global__ void sumKernel (int *a, int *b, int *c) {
3    *c = *a + *b;
4 }
5
6 // CPU Code
7 int main(void) {
8    int a, b, c;
9    int *d_a, *d_b, *d_c;
10   int size = sizeof(int);
11
12   // Allocate space for device
13   cudaMalloc((void **)&d_a, size);
14   cudaMalloc((void **)&d_b, size);
15   cudaMalloc((void **)&d_c, size);
16
17   // Give some initial values
18   a = 10; b = 20;
19
20   // CPU -> GPU
21   cudaMemcpy(d_a, &a, size, cudaMemcpyHostToDevice);
22   cudaMemcpy(d_b, &b, size, cudaMemcpyHostToDevice);
23
24   // kernel execution: 1 thread
25   sumKernel<<<1,1>>>(d_a, d_b, d_c);
26   // GPU -> CPU
27   cudaMemcpy(&c, d_c, size, cudaMemcpyDeviceToHost);
28   // Clean memory
29   cudaFree(d_a);cudaFree(d_b); cudaFree(d_c);
30 }
```

Figura 6.31: código completo de um programa simples para somar dois valores na GPU

envolvem uma aplicação:

a. Definição do kernel (linhas 2 a 4);

b. Declaração das variáveis de GPU (linha 9). Como a GPU tem espaços de memória diferente é necessário criar e alocar variáveis em espaços diferentes da CPU;

c. Alocação da memória da GPU utilizada no kernel. A gerência da memória da GPU é de responsabilidade da CPU. Portanto, tanto neste passo (alocação) quanto no passo de desalocação, os comandos são executados pela CPU, conforme linhas 13 à 15.O cudaMalloc tomará conta da referência para as variáveis declaradas em (b);

d. Enviar dados da CPU para a GPU (linhas 21 e 22): função cudaMemcpy irá transferir os dados da memória de CPU pra a GPU. Note que o último argumento é usado para dizer em que direção ocorre a cópia, neste caso DeviceToHost;

e. Execução do Kernel (linha 25): o kernel será executado criando um número total de threads equivalente a blocos x números de threads por bloco;

f. Enviar os dados de volta da GPU para a CPU (linha 29), usando novamente a função cudaMemcpy;

g. Liberar a memória da GPU (linha 29);

Como mencionado anteriormente, GPU computing é uma computação híbrida e deve tratar com pelo menos duas arquiteturas de hardware. Neste sentido, cada hardware tem sua própria memória e sempre que se requer enviar uma tarefa de um dispositivo para outro é necessário fazer a transferência de dados pelas memórias. Transferir dados da CPU para a GPU é normalmente um dos gargalos dos programas e deve ser minimizado. Num futuro próximo pretende-se que os sistemas tenham apenas uma única memória, evitando a tarefa de transferência.

Os kernels de CUDA são lançados de forma assíncrona em relação ao host. Isto significa que uma vez chamado o kernel, a CPU está livre para continuar processando o que vier na sequência. Entretanto, se logo após chamar o kernel a CPU requerer resultados desta chamada, deve-se inserir um ponto de sincronismo, de forma a esperar que a GPU entregue os resultados antes de usá-los. Esta tarefa é feita usando o comando cudaDeviceSynchronize. Note que no código da figura 6.31 não há este ponto de sincronismo. Isto ocorre porque a função cudaMemcpy já inclui uma sincronização dentro de sua implementação.

Finalmente, vamos incluir paralelismo no exemplo. Obviamente somar dois números não é uma tarefa paralelizável, portanto vamos agora somar 2 vetores de tamanho N. O código da Figura 6.32 mostra um kernel que faz esta tarefa.

```
1 __global__ void Vecadd(int *d_a, int *d_b, int *d_c) {
2    int i = threadIdx.x;
3    d_c[i] = d_a[i] + d_b[i]
```

Figura 6.32: Kernel para somar dois vetores de forma paralela.

Este kernel mostra uma palavra reservada importante em CUDA, chamada threadIdx. Cada thread de um bloco irá receber seu índice individual. A atribuição deste índice é feita pelo escalonador de warps e independentemente do número de threads criados, tem tempo constante. Neste sentido, suponhamos que se desejem criar N threads. De forma paralela, teremos a seguinte execução:

Thread 0: dc[0] = da[0] + db[0]

Thread 1: dc[1] = da[1] + db[1]

...

Thread N: dc[N] = da[N] + db[N]

Se o número de threads é menor que o número de núcleos disponíveis em cada bloco em execução, podemos dizer que a soma será feita numa única passada na GPU. Entretanto, se o N for maior, uma fila de threads será criada e haverá listas de threads para serem chamadas em warps distintos. Para somar o vetor de N, uma maneira possível de chamar o kernel seria:

VecAdd < < <1, > > >(da, db, dc);

Esta chamada cria um bloco, sendo que este possui N threads dentro. Enquanto aqui já há um paralelismo ocorrendo, ainda estamos usando um número

limitado de recursos, já que há apenas um bloco sendo usado, o que significa que tudo está recaindo num mesmo SM. Há também outra limitação, que consiste no número máximo de threads que se pode criar por bloco, sendo em algumas arquiteturas 1024 e em outras (mais modernas) 2048. Isto significa que se o vetor tiver mais do que esta quantidade de elementos, este kernel não poderá tratar o vetor. Uma possibilidade seria em cada thread somar mais do que um elemento do vetor. Mesmo assim, ainda estaríamos usando um só SM, portanto não usaríamos todos os recursos da GPU.

Para um uso completo da GPU, é necessário criar também diversos blocos, conforme a chamada de kernel abaixo:

VecAdd < < <K,L> > >(da, db, dc);

Neste caso estamos criando K blocos, cada um composto de L threads. Assim sendo, o tamanho do Grid corresponde a N=L.L, que é o número de elementos do vetor. Para fazer isto, é necessário realizar um pequeno ajuste no kernel, de forma que os blocos sejam usados adequadamente para mapear partes distintas do vetor. O código da Figura 6.32 mostra este kernel.

```
__global__ void Vecadd(int *d_a, int *d_b, int *d_c) {
     int i= threadIdx.x + blockIdx.x * blockDim.x;
     d_c[i] = d_a[i] + d_b[i];
}
```

Figura 6.33: Kernel para somar dois vetores usando mais de um bloco.

O escalonador de warps cria L threads para cada bloco, de forma que cada bloco tem o mesmo conjunto de números threadIdx para suas threads. Entretanto, o escalonador atribui diferentes índices de blocos para cada bloco. Estes índices podem ser compostos com os índices das threads para criar índices únicos para cada elemento do vetor. Enquanto este kernel é mais eficiente que o apresentado anteriormente, ainda há um problema a ser tratado, que é o caso de N não ser múltiplo da tupla K,L. Suponha, por exemplo, que N=101. Neste caso, poderíamos atribuir K=5 e L=21. Porém seriam criados 105 threads e para o caso do blocoIdx.x=4 e threadIdx.x=20 teriamos um índice i=104, que levaria a um elemento do vetor que não existe. Para evitar este problema, é comum colocar um condicional referente ao índice calculado, conforme pode ser visto na figura 6.34

```
1 __global__ void Vecadd(int *d_a, int *d_b, int *d_c) {
2 int i= threadIdx.x + blockIdx.x * blockDim.x;
3    if (i < N)
4       d_c[i] = d_a[i] + d_b[i];
5 }
```

Figura 6.34: Kernel para somar dois vetores usando mais de um bloco.

É importante mencionar que nestes exemplos estamos usando a memória global para armazenar os vetores. Enquanto esta memória é grande e capaz de

ser acessada por qualquer thread de qualquer bloco, a mesma tem uma latência grande e pode demandar bastante tempo para o código. A memória compartilhada é uma alternativa eficiente para acesso a memória. Enquanto uma leitura de um valor na memória global pode levar até 400 ciclos de máquina, um acesso memória compartilhada pode requerer apenas 4 ciclos. A maior restrição presente na memória compartilhada são o seu tamanho pequeno (96Kb na arquitetura Pascal) e pouca persistência (quando um bloco termina de ser executado, os dados da memória compartilhada são apagados, pois um nobo bloco irá entrar no SM). Entender como funciona e como desenvolver programas com esta memória está além do propósito deste texto, mas o leitor interessado pode ler mais em [Clua and Zamith 2015]

6.6. Ferramentas

Avanços em ferramentas e bibliotecas para o desenvolvimento de redes neurais profundas permitem a engenheiros, cientistas e entusiastas explorar soluções para diferentes aplicações na área de Aprendizado de Máquina, tais como classificação de imagens e vídeo, processamento natural de linguagem e reconhecimento de áudio. Essas ferramentas permitem que usuários treinem, desenvolvam e testem redes neurais profundas utilizando todo poder computacional proporcionado pelas GPUs. As bibliotecas de redes neurais profundas mais recentes apresentam uma forma de interação em alto nível onde o usuário deve se preocupar apenas com a modelagem da rede, sendo transparentes ao usuário as etapas de mais baixo nível de programação e otimização computacional. Entre as bibliotecas mais populares encontram-se: Caffe, CNTK, Tensor Flow, Torch e Theano.

Durante o minicurso será apresentada uma introdução à biblioteca Caffe [caf] e um estudo de caso será desenvolvido usando a ferramenta Digits NVidia DIGITS [dig]. O material da parte experimental está disponível em:

`http://www2.ic.uff.br/~gpu/learn-gpu-computing/deep-learning/`

6.7. Outras considerações

Este capítulo procurou apresentar conceitos fundamentais ao aprendizado profundo, sem a pretensão de cobrir toda a área. Alguns temas importantes não foram apresentados ou detalhados, tais como técnicas de aprendizado não-supervisionado, aprendizado por reforço, transferência de conhecimento, mecanismos de atenção, redes generativas adversárias, dentre outros. Tal diversidade ilustra a abrangência de inúmeras pesquisas na área e a importância que a mesma possui na indústria da tecnologia. Embora haja uma grande proliferação de aplicações de redes profundas nos últimos anos, boa parte da fundamentação vem sendo desenvolvida já há alguns anos e deve ser consultada como embasamento e fonte de inspiração para novas soluções. Dentro de sua limitação, este texto buscou ser um convite inicial ao leitor que deseja se iniciar por tais investigações.

References

[caf] Caffenet. `http://caffe.berkeleyvision.org/`. Último acesso em 10/4/2017.

[ima] Imagenet. `http://www.image-net.org/`. Último acesso em 10/4/2017.

[dig] Página oficial do projeto digits da nvidia. `https://developer.nvidia.com`. Digits, 2016. Último acesso em 10/7/2016.

[Bengio et al. 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw.*, 5(2):157–166.

[Bowman et al. 2016] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space.

[Britz et al. 2017] Britz, D., Goldie, A., Luong, T., and Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv*.

[Chan et al. 2016] Chan, W., Jaitly, N., Le, Q. V., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*.

[Cho et al. 2014] Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

[Ciresan et al. 2013] Ciresan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013 - 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part II*, pages 411–418.

[Ciresan et al. 2011a] Ciresan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2011a). Convolutional neural network committees for handwritten character classification. In *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*, pages 1135–1139.

[Ciresan et al. 2011b] Ciresan, D. C., Meier, U., Masci, J., and Schmidhuber, J. (2011b). A committee of neural networks for traffic sign classification. In *The 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, California, USA, July 31 - August 5, 2011*, pages 1918–1921.

[Ciresan et al. 2012] Ciresan, D. C., Meier, U., and Schmidhuber, J. (2012). Transfer learning for latin and chinese characters with deep neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, June 10-15, 2012*, pages 1–6.

[Clua and Zamith 2015] Clua, E. and Zamith, M. (2015). Programming in cuda for kepler and maxwell architecture. In *Revista de Informática Teórica e Aplicada*, volume 22, pages 34–42.

[Couprie et al. 2013] Couprie, C., Farabet, C., Najman, L., and LeCun, Y. (2013). Indoor semantic segmentation using depth information. In *International Conference on Learning Representations (ICLR2013)*.

[Dahl et al. 2017] Dahl, R., Norouzi, M., and Shlens, J. (2017). Pixel recursive super resolution. *arXiv*.

[Dumoulin et al. 2017] Dumoulin, V., Shlens, J., and Kudlur, M. (2017). A learned representation for artistic style. *ICLR*.

[Eslami et al. 2016] Eslami, S. M. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. (2016). Attend, infer, repeat: Fast scene understanding with generative models.

[Esteva et al. 2017] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118.

[Fukushima 1980] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

[Giusti et al. 2016] Giusti, A., Guzzi, J., Ciresan, D. C., He, F., Rodriguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Caro, G. D., Scaramuzza, D., and Gambardella, L. M. (2016). A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667.

[Gulshan et al. 2016] Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. Q., Mega, J., and Webster, D. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*.

[Gupta et al. 2017] Gupta, S., Davidson, J., Levine, S., Sukthankar, R., and Malik, J. (2017). Cognitive mapping and planning for visual navigation.

[Hahnloser et al. 2000] Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951.

[He et al. 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

[He et al. 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. *CoRR*, abs/1603.05027.

[Heigold et al. 2016] Heigold, G., Moreno, I., Bengio, S., and Shazeer, N. M. (2016). End-to-end text-dependent speaker verification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[Hochreiter and Schmidhuber 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

[Hubel and Wiesel 1959] Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591.

[Hubel and Wiesel 1968] Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195:215–243.

[Hubel and Wiesel 2005] Hubel, M. and Wiesel, T. N. (2005). *Brain and Visual Perception*. Oxford Univeristy Press.

[Jaeger and Haas 2004] Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, pages 78–80.

[Jaques et al. 2016] Jaques, N., Gu, S., Turner, R. E., and Eck, D. (2016). Generating music by fine-tuning recurrent neural networks with reinforcement learning. In *Deep Reinforcement Learning Workshop, NIPS*.

[Kaiser and Sutskever 2016] Kaiser, L. and Sutskever, I. (2016). Neural gpus learn algorithms. In *International Conference on Learning Representations*.

[Kannan et al. 2016] Kannan, A., Kurach, K., Ravi, S., Kaufman, T., Miklos, B., Corrado, G., Tomkins, A., Lukacs, L., Ganea, M., Young, P., and Ramavajjala, V. (2016). Smart reply: Automated response suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2016)*.

[Krizhevsky et al. 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[LeCun et al. 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

[LeCun et al. 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[LeCun et al. 2001] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (2001). Gradient-based learning applied to document recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press.

[Levine et al. 2016] Levine, S., Sampedro, P. P., Krizhevsky, A., and Quillen, D. (2016). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection.

[Lin et al. 2013] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *CoRR*, abs/1312.4400.

[Lin et al. 1995] Lin, T., Horne, B. G., Tiňo, P., and Giles, C. L. (1995). Learning long-term dependencies is not as difficult with narx recurrent neural networks. Technical report, College Park, MD, USA.

[Loos et al. 2017] Loos, S., Irving, G., Szegedy, C., and Kaliszyk, C. (2017). Deep network guided proof search.

[Maass et al. 2002] Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.*, 14(11):2531–2560.

[Mnih et al. 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*.

[Odena et al. 2016] Odena, A., Olah, C., and Shlens, J. (2016). Conditional image synthesis with auxiliary classifier gans. *arXiv*.

[Papandreou et al. 2017] Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., and Murphy, K. (2017). Towards accurate multi-person pose estimation in the wild.

[Pascanu et al. 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1310–1318.

[Rosenblatt 1961] Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document.

[Russakovsky et al. 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Sermanet et al. 2012] Sermanet, P., Chintala, S., and LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification. In *International Conference on Pattern Recognition (ICPR 2012)*.

[Sermanet et al. 2013] Sermanet, P., Kavukcuoglu, K., Chintala, S., and LeCun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR'13)*. IEEE. Video part 1; Video part 2.

[Shao et al. 2017] Shao, L., Gouws, S., Britz, D., Goldie, A., Strope, B., and Kurzweil, R. (2017). Generating long and diverse responses with neural conversation models. *arXiv*.

[Silver et al. 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503.

[Simonyan and Zisserman 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

[Srivastava et al. 2014] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

[Szegedy et al. 2016] Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.

[Szegedy et al. 2015a] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015a). Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.

[Szegedy et al. 2015b] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015b). Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567.

[Tompson et al. 2016] Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. (2016). Accelerating eulerian fluid simulation with convolutional networks. *arXiv*.

[Tompson et al. 2014] Tompson, J., Stein, M., Lecun, Y., and Perlin, K. (2014). Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.*, 33(5):169:1–169:10.

[Villegas et al. 2017] Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. (2017). Decomposing motion and content for natural video sequence prediction.

[Yan et al. 2016] Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. (2016). Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision.

[Zeiler and Fergus 2013] Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901.