

UM FRAMEWORK MULTITEMÁTICO PARA
COLETA DE INFORMAÇÃO GEOGRÁFICA
PRODUZIDA POR VOLUNTÁRIOS

MICHELE BRITO PINHEIRO

**UM FRAMEWORK MULTITEMÁTICO PARA
COLETA DE INFORMAÇÃO GEOGRÁFICA
PRODUZIDA POR VOLUNTÁRIOS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: CLODOVEU AUGUSTO DAVIS JÚNIOR

Belo Horizonte

Agosto de 2016

© 2016, Michele Brito Pinheiro.
Todos os direitos reservados.

Pinheiro, Michele Brito

P654f Um framework multitemático para coleta de
informação geográfica produzida por voluntários /
Michele Brito Pinheiro. — Belo Horizonte, 2016
xx, 68 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais

Orientador: Clodoveu Augusto Davis Júnior

1. Computação — Teses. 2. Sistemas de Informação
Geográfica 3. Crowdsourcing 4. Sistemas de recuperação
da informação 5. Redes Sociais On-line I. Orientador.
II. Título.

CDU 519.6*74 (043)



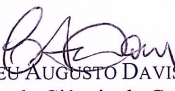
UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO


FOLHA DE APROVAÇÃO

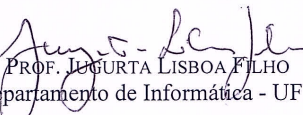
Um framework multitemático para coleta de informação geográfica produzida
por voluntários

MICHELE BRITO PINHEIRO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. CLODOVEU AUGUSTO DAVIS JÚNIOR - Orientador
Departamento de Ciência da Computação - UFMG


PROFA. GISELE LOBO PAPPA
Departamento de Ciência da Computação - UFMG


PROF. JURGATA LISBOA FILHO
Departamento de Informática - UFV

Belo Horizonte, 26 de agosto de 2016.

Agradecimentos

Primeiramente, devo agradecer ao meu orientador Clodoveu A. Davis Jr., pelo suporte, paciência, compreensão, e por acreditar no meu trabalho em um momento importante da minha vida.

À professora Gisele L. Pappa e ao professor Jugurta Lisboa Filho por terem aceitado o convite para participar dessa banca de avaliação.

Aos meus pais, Kátia e Eurípedes, para os quais todos os agradecimentos que eu possa fazer nunca serão suficientes. O imenso amor e dedicação, a paciência e o apoio deles a mim, me nutrem cotidianamente e me garantem a firmeza para enfrentar todos os desafios que me propus ao longo da vida, mesmo vivendo a tanto tempo distante. Às minhas irmãs amadas, Marina e Vanessa, cujo companheirismo e lealdade nunca fazem me sentir sozinha. O carinho e o apoio delas também foram ingredientes essenciais nesse processo.

Aos meus amigos pelo carinho, suporte, amizade, companheirismo. Em especial agradeço à Alexandre e Rodrigo, por me ajudarem em diversos momentos da minha vida - e nesse trabalho, assim como seus pais Ilka e Lauro que sempre me receberam com muito carinho em sua casa.

Aos colegas de laboratório, com a convivência dos quais tive diversos aprendizados e bons momentos, em especial: Michele Brandão, Vitor Mangaravite, Daniel Xavier e Wellington Dores.

A Deus, por me guiar, por sempre está presente em minha vida, apresentando sua luz e esperança tanto nos momentos quanto nos de dificuldade. Sem Ele esse trabalho não poderia ser realizado.

À CAPES por permitir que esse trabalho pudesse ser desenvolvido pelo seu financiamento.

*“Somewhere over the rainbow, way up high
There’s a land that I’ve heard of
Once in a lullaby.”*

(Harold Arlen, E.Y. Harburg)

Resumo

Desde que os recursos associados à *Web 2.0* começaram a permitir que cidadãos comuns publicassem conteúdo produzido por eles mesmos, vêm se intensificando as iniciativas de coleta de informação geográfica produzida por voluntários. Nessa vertente do *crowd-sourcing*, cidadãos são vistos como “sensores” espacialmente distribuídos, e habilitados por aplicações voltadas à coleta de dados de variados tipos. No entanto, desafios referentes à cobertura espacial, à qualidade e à motivação dos voluntários demandam soluções criativas e variadas para o desenvolvimento dos instrumentos de coleta. A presente dissertação apresenta um *framework* para geração de tais instrumentos, projetado para permitir a escolha entre múltiplas funções de coleta e alternativas de representação espacial de eventos, fenômenos e objetos geográficos. O *framework* traz também recursos para reconhecer e valorizar a participação dos voluntários. A criação de recursos para coleta de dados sobre múltiplos temas é feita interativamente na *Web*, sem a necessidade de programação. Um estudo de caso voltado à coleta de dados sobre mobilidade urbana é apresentado como exemplo de aplicação do *framework*.

Abstract

Since the resources associated with the Web 2.0 had allowed ordinary people to publish content produced by themselves, there have been growing number of initiatives to gather geographic information produced by volunteers. In this model of crowdsourcing, citizens are seen as spatially distributed “sensors”, enabled to collect different kinds of data through these applications. However, challenges related to spatial coverage, quality and motivation of volunteers require creative and varied solutions for the development of data collection instruments. This work presents a framework for generating such instruments, designed to allow the user to choose between multiple functions to gather information and alternatives to spatially represented events, phenomena and geographic objects. The framework also provides resources to recognize and value the participation of volunteers. Interactively through the web, it is possible to create resources for collecting information on multiple themes without the need of programming. We present a case study aimed at collecting data on urban mobility as the framework application example.

Lista de Figuras

2.1	<i>Crowdsourcing</i> e <i>Crowdsensing</i> segundo a motivação dos usuários	8
3.1	Organização em categorias vs. temas	25
3.2	Visão geral da arquitetura	32
3.3	Diagrama de classes simplificado	33
3.4	Diagrama de casos de uso de papéis dos usuários	35
5.1	Tela inicial sobre a visão do administrador	54
5.2	Informações básicas do tema	54
5.3	Atributos descritivos do tema	55
5.4	Configuração da pontuação por ações.	56
5.5	Configuração das categorias de <i>expertise</i>	57
5.6	Listagem dos temas	57
5.7	Detalhes do tema	58
5.8	Controles de criação de uma nova contribuição	58
5.9	Formulário de nova contribuição	59
5.10	Informações sobre a nova contribuição	59
5.11	<i>Feedback</i>	60

Lista de Tabelas

2.1	Motivação dos voluntários	13
2.2	Estratégias de alocação para jogos geográficos [Antoniou & Schlieder, 2014]	15
4.1	Alternativas analisadas <i>versus</i> utilizadas	41
4.2	Valores de Referência para Grupo de Especialização	43
4.3	Valores de Referência para Recompensas	43
4.4	Rotas de registro e autorização de acesso dos voluntários	45
4.5	Rotas de manipulação dos temas	46
4.6	Rotas para listagem de ações e grupos de <i>expertise</i>	46
4.7	Rotas para manipular as contribuições	47
4.8	Rotas para manipular comentários, <i>wikis</i> , anexos e avaliação das contribuições	48
4.9	Rotas de administração de usuários	49

Sumário

Agradecimentos	vii
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	3
1.1 Objetivos	5
1.2 Estrutura do trabalho	6
2 Trabalhos Relacionados	7
2.1 Crowdsourcing e Crowdsensing	7
2.1.1 Neogeografia	8
2.1.2 PPGIS	9
2.1.3 Citizen Science	10
2.2 Problemas e limitações das contribuições voluntárias	11
2.3 Soluções para aplicações de VGI	13
2.3.1 VGI sob demanda	14
2.3.2 <i>Gamificação</i>	14
2.3.3 Verificação por pares	15
2.3.4 Verificação automática	16
2.4 A relação entre VGI e IDE	17
2.5 <i>Frameworks</i>	18
3 Abordagem	23
3.1 Temas	23

3.2	Requisitos	27
3.2.1	Requisitos Funcionais	27
3.2.2	Requisitos Não-Funcionais	30
3.3	Arquitetura	31
3.4	Modelagem dos Dados	32
3.5	Papeis dos Usuários	34
3.6	Comentários finais	36
4	Implementação	37
4.1	Avaliação do Ambiente de Desenvolvimento	37
4.1.1	Serviços	37
4.1.2	Aplicação	40
4.2	Implementação das camadas	42
4.2.1	Camada de Persistência	42
4.2.2	Camada de Serviços	43
4.2.3	Camada de Aplicação	49
5	Estudo de caso	53
6	Conclusão e Trabalhos Futuros	61
6.1	Trabalhos futuros	62
	Referências Bibliográficas	65

Lista de Siglas

ACL *Access Control List*

API *Application Programming Interface*

DER Departamento de Estradas e Rodagem

DOM *Document Object Model*

FGDC *Federal Geographic Data Committee*

HTML *HyperText Markup Language*

HTTP *Hypertext Transfer Protocol*

IDE Infraestrutura de Dados Espaciais

IG Informação Geográfica

INSPIRE *Infrastructure for Spatial Information in Europe*

JWT *JSON Web Tokens*

MVC *Model-View-Controller*

NSDI *National Spatial Data Infrastructure*

ONGs Organizações Não Governamentais

ORM *Object-Relational Mapping*

PPGIS *Public Participation Geographic Information Systems*

SGBD Sistema de Gerenciamento de Banco de Dados

SIG Sistema de Informação Geográfica

SMS Short Message Service

SPA *Single Page Application*

SQL *Structured Query Language*

VGI *Volunteered Geographic Information*

WWW *World Wide Web*

Capítulo 1

Introdução

O termo *web 2.0* foi pela primeira vez utilizado por volta de 2004 [O’Reilly, 2005]. Desde então, tornou-se sinônimo de uma revolução no uso da Internet. A partir do advento da *web 2.0*, passa a ser viável que usuários publiquem conteúdo produzido por eles mesmos diretamente e sem nenhum controle centralizado. Dessa forma, a *web 2.0* veio reforçar a perspectiva inicial de Berners-Lee, considerado o criador da *World Wide Web* (WWW), de que a Web se tornaria um meio colaborativo onde todos podem se “encontrar, ler e escrever” [Richardson, 2009]. Ela trouxe impactos também sobre o fluxo, volume e estruturação dos dados que são publicados online. A variedade de formas de publicação de dados produzidos por usuários comuns vem aumentando desde então, variando desde simples postagens em *blogs* a *check-ins* em mídias sociais como o Facebook¹ e o Foursquare². Diante do potencial de engajamento de usuários conectados, surgiram projetos de coleta e organização de dados baseados na contribuição direta de pessoas comuns. Esse fenômeno recebeu vários nomes, sendo *crowdsourcing* o mais conhecido deles [Mateveli et al., 2015]. Atualmente, muitas fontes de informação disponíveis online são majoritariamente compostas por contribuições gratuitas e anônimas de grandes quantidades de pessoas conectadas. Exemplos incluem bases de conhecimento, como a Wikipédia³, e sistemas de mapeamento colaborativo, como o OpenStreetMap⁴.

Para grande parte das atividades humanas, as dimensões tempo e espaço são bastante relevantes. O surgimento de recursos tais como os sistemas de mapas digitais online (Google Maps, Bing Maps, entre outros) assim como a ampla disponibilidade de dispositivos de posicionamento global (como o GPS) embarcados em computado-

¹Facebook - <http://www.facebook.com/>

²Foursquare - <http://www.foursquare.com>

³Wikipédia - <https://www.wikipedia.org/>

⁴Open Street Map - <http://www.openstreetmap.org/>

res móveis e smartphones, deram origem a uma grande variedade de novas aplicações de uso cotidiano e popular. A manutenção e atualização dos dados usados por essas aplicações, por outro lado, requer um grande esforço e muitos recursos, já que se exige grande detalhamento (escalas grandes, com muitos detalhes e grande precisão de posicionamento) ao mesmo tempo que se espera ampla cobertura (essencialmente todas as regiões em que a atividade humana é intensa, como as áreas urbanas). Agências nacionais de mapeamento não atingem, via de regra, tamanho detalhamento nos dados que produzem. Órgãos públicos municipais, que precisam desse tipo de detalhamento, têm grandes dificuldades operacionais e financeiras para manter suas informações geográficas atualizadas. Assim, percebeu-se que a participação de cidadãos comuns, vistos como “sensores” espacialmente distribuídos, poderia ser decisiva para esforços de coleta e manutenção de informação geográfica. Como no caso de *crowdsourcing*, esse movimento recebeu vários nomes, tendo se notabilizado a expressão *Volunteered Geographic Information (VGI)* [Goodchild, 2007].

Assim, do ponto de vista de Informação Geográfica, a onda de aplicações de *crowdsourcing* representou uma grande mudança em potencial, uma vez que os métodos tradicionais de produção de dados por levantamento em campo ou por meio de projetos de sensoriamento remoto eram demorados e representavam grandes custos. A nova alternativa permitiria que qualquer indivíduo minimamente equipado, mas não necessariamente capacitado tecnicamente, se tornasse uma potencial fonte de dados [Goodchild, 2007]. Fonte essa com maior alcance, pela força dos números, e custo extremamente baixo. Todavia, a falta de conhecimento técnico dos voluntários levantou outras questões, ligadas à qualidade dos dados coletados [Flanagin & Metzger, 2008]. Percebeu-se também uma potencial dificuldade na cobertura de áreas menos populosas, ou dotadas de um volume menor de voluntários [Elwood et al., 2012], e uma queda brusca de interesse em projetos de contribuição voluntária [Antoniou & Schlieder, 2014], causada pela saturação das contribuições em pequenas áreas de seu maior interesse [Haklay, 2010; Antoniou & Schlieder, 2014].

Os problemas típicos de VGI, como os indicados no parágrafo anterior, têm sido enfrentados criativamente nos projetos por meio da adoção de métodos e técnicas variados, incorporados às soluções tecnológicas. Por exemplo, existem variados métodos de identificação (ou não) de voluntários, validação de contribuições por especialistas ou por outros usuários [Goodchild & Li, 2012; Flanagin & Metzger, 2008; Maué, 2007], incentivo e premiação à participação qualificada [Chen et al., 2014], *gameficação* [Martella et al., 2015; Antoniou & Schlieder, 2014], realimentação da informação consolidada para o contribuinte individual, e muitas outras.

1.1 Objetivos

A hipótese investigada nessa dissertação é a de que é viável encapsular os variados mecanismos de contribuição voluntária em um arcabouço abrangente, a partir do qual a criação de sistemas de coleta de informação geográfica fornecida por voluntários pode ser feita escolhendo entre alternativas metodológicas adequadas aos temas de coleta e as características esperadas da comunidade de voluntários. Esses mecanismos incluem, entre outros, fatores tais como a escolha de temas e sub-temas, a escolha do tipo mais adequado de representação espacial (e.g., ponto, linha, polígono), a determinação de atributos descritivos relacionados ao fenômeno ou evento coletado e o estabelecimento de mecanismos de acompanhamento da reputação de usuários.

Dessa maneira, o objetivo deste trabalho é estudar as possíveis variações estruturais, metodológicas e procedimentais para a criação de recursos de contribuição voluntária de dados geográficos. Essas variações são organizadas em um conjunto de funções, componentes de um *framework* de métodos, a partir do qual se pode conceber e criar rapidamente novos instrumentos de coleta de informação geográfica por voluntários. Um aspecto essencial desse *framework* é a possibilidade de seu uso em projetos multitemáticos, sem perder a especificidade de cada um de seus temas componentes, além de oferecer recursos flexíveis de construção dos instrumentos de coleta.

Nesse sentido, são definidos os seguintes objetivos específicos:

- Estudo sobre os diferentes tipos de cenários de utilização das aplicações de VGI descritos na literatura, assim como as principais questões associadas à coleta e métodos aplicados a esses contextos individuais;
- Definição das principais funções que compõem o *framework*, estabelecendo assim um escopo de requisitos, a partir da análise da literatura;
- Modelagem da arquitetura e dos componentes;
- Desenvolvimento, precedido por uma análise das tecnologias que mais se adequam ao cenário delimitado e às características pretendidas para o *framework*;
- Proposição de estudo de caso e exemplos para validação do *framework* na criação de aplicações de VGI.

1.2 Estrutura do trabalho

O restante desta dissertação está organizado como se segue. O Capítulo 2 apresenta uma contextualização acerca das aplicações de VGI assim como os principais problemas e *frameworks* descritos na literatura, delineando assim um conjunto de funções. O Capítulo 3 apresenta a abordagem proposta para o desenvolvimento do *framework*. Posteriormente, no Capítulo 4, é apresentada uma análise para escolha das tecnologias usadas e como elas foram combinadas para desenvolver a aplicação final. No Capítulo 5 são apresentados estudos de caso que exemplificam e validam o *framework*, assim como discutem os resultados obtidos. Por fim, no Capítulo 6 são apresentadas conclusões e descritos trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

O presente capítulo apresenta uma contextualização do *crowdsourcing* aplicado a coleta de informação geográfica, assim como as principais discussões envolvidas nesse processo e a viabilidade de utilização desse tipo de dado. Por fim, é apresentada uma visão geral dos *frameworks* descritos em trabalhos científicos.

2.1 Crowdsourcing e Crowdsensing

Após o aparecimento das aplicações de *crowdsourcing* de informação geográfica, surgiram inúmeras variações e especializações nesse contexto. Mateveli et al. [2015] ponderam sobre a forma da participação voluntária do usuário nesse tipo de iniciativa, propondo assim uma taxonomia para essas aplicações.

Do ponto de vista dos métodos de coleta, os autores dividem o universo de aplicações entre *crowdsensing* e *crowdsourcing*. No primeiro caso, as informações são coletadas através de sensores, podendo ou não apresentar informações secundárias sobre a forma textual, visual ou auditiva. Já o segundo apresenta apenas informações geradas pelos usuários através de *blog*, *microblogs*, *wikis*, etc.

O segundo corte no universo de aplicações dessa natureza se dá pelo prisma do papel do usuário, que pode assumir uma posição passiva ou ativa no momento da coleta. No primeiro caso, o usuário fornece informações sem que tenha consciência do ato, como dados de navegação na *Web* e informações obtidas por aplicativos *mobile* em *background*, com ou sem o consentimento explícito do usuário. No segundo caso, o usuário tem plena consciência de estar realizando uma contribuição e age nessa direção, como por exemplo ao criar um novo local na Wikimapia, realizar *check-in* no Foursquare¹, ou emitir um comentário em uma página da *Web*.

¹Foursquare - <https://pt.foursquare.com/>

Os autores ainda consideram iniciativas orientadas por tarefa, em que os voluntários são convocados a cobrir demandas por informação previamente organizadas em uma lista. Pela taxonomia proposta, aplicações de VGI são classificadas como *crowdsourcing* ativo, ou seja, a fonte primária de informação são voluntários humanos, e sua contribuição é dada conscientemente. A Figura 2.1 ilustra a taxonomia proposta por Mateveli et al. [2015].

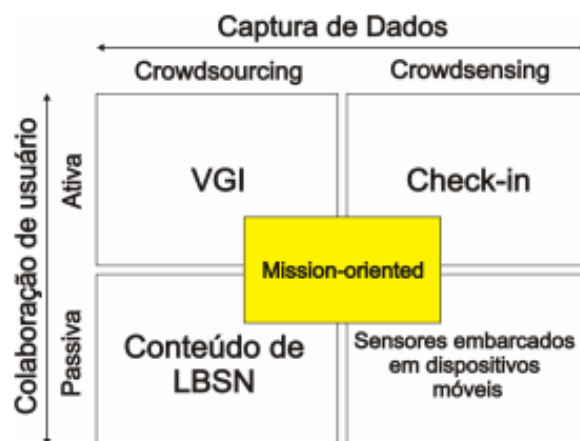


Figura 2.1: *Crowdsourcing* e *Crowdsensing* segundo a motivação dos usuários

Sob essa ótica, o presente trabalho se dedica apenas a tratar de aplicações onde o usuário se voluntaria à fornecer ativamente informações sobre um determinado local. Dentro desse sub-contexto de *crowdsourcing*, a literatura traz diferentes tipos de denominações e classificações do *crowdsourcing* ativo, como *Neogeografia*, *Public Participation Geographic Information Systems (PPGIS)* e *Citizen Science*, além do já citado VGI. Essas propostas são abordadas nas próximas seções.

2.1.1 Neogeografia

Após o surgimento de ferramentas como *Google Earth* e *Google Maps*, e da popularização da utilização de mapas em sites pessoais através da API do *Google Maps*, tornou-se evidente a existência de um novo contexto, onde a informação geográfica estava sendo produzida. Via de regra, essas iniciativas eram criadas por usuários que possuíam pouca ou nenhuma familiaridade com os métodos tradicionais de coleta de dados utilizados por profissionais com os SIGs. Por outro lado, devido à sua grande popularização, surgiram os “geógrafos amadores” que, devido à sua grande quantidade e capacidade de geração de informação georreferenciada, deram origem ao termo *Neogeografia*. Mais especificamente, o termo reflete o conjunto de técnicas e ferramentas que não se encaixam na geografia tradicional [Turner, 2006]. Apesar dos indivíduos envolvidos com a

Neogeografia não serem necessariamente geógrafos, Goodchild [2009] destaca que essas pessoas podem dispor de uma “*expertise*” local, ou familiaridade sobre o espaço que frequenta (espaço ativo), tendo então seu devido valor.

2.1.2 PPGIS

Algumas iniciativas de colaboração voluntária perceberam a possibilidade de incentivar a participação da comunidade para intervir politicamente em ações governamentais de seu potencial interesse. Dessa forma, técnicas de coleta de contribuições e opiniões surgem como meio de expressão da opinião pública, configurando as iniciativas denominadas *Public Participation Geographic Information Systems* (PPGIS). O termo PPGIS surgiu antes mesmo da designação VGI ser concebida, quando diversos projetos emergiram, em geral organizados por ONGs, com a intenção de alertar as autoridades sobre a necessidade de melhorias em determinadas áreas [Drew, 2003; Elwood, 2006]. Nesse âmbito se inserem iniciativas como Wikicrimes, Strepitus², Mapa Opinião³, Common Census⁴ e Ushahidi⁵. Além dessas, existem aplicações voltadas para gestão urbana e tomada de decisões governamentais, como descrito por Lin [2013] e Johnson & Sieber [2013].

Okolloh [2009] apresenta um histórico de criação e implementação do Ushahidi, uma aplicação voltada para a participação pública em situações de crise. A aplicação surgiu em meio aos conflitos do Quênia, com o objetivo de relatar e mapear “testemunhos” da população sobre atos de violência cometidos pelo governo local, em um cenário onde informações estão desestruturadas pois os meios de comunicação foram censurados. Seu desenvolvimento foi integralmente realizado por voluntários, do registro do domínio e sua hospedagem à implementação do código de contribuição por SMS (serviço de mensagens curtas). As informações são fornecidas pelos usuários de duas formas, pela interface *Web* e por envio de SMS, sendo esta última verificada com maior cuidado pela rede de colaboradores, que comparam as mensagens com outras fontes de dados, antes de publicá-las na plataforma. Quando o relato aparenta possuir credibilidade, mas não pode ser verificado, este é publicado com o indicativo de informação não confirmada.

A exemplo desse tipo aplicação, Vellozo et al. [2013] desenvolveram para ambientes *Web* e móvel (Android e iOS), uma aplicação que permite reportar denúncias e reclamações sobre o ruído urbano, denominada Strepitus. O mesmo grupo de pesquisa

²Strepitus - <http://geo.lbd.dcc.ufmg.br/strepitus/>

³Mapa Opinião - <http://geo.lbd.dcc.ufmg.br/mapaopinio>

⁴Common Census - <http://commoncensus.org/>

⁵Ushahidi - <http://www.ushahidi.com>

desenvolveu posteriormente o Mapa Opinião [Soares et al., 2014], uma ferramenta que tem por objetivo a criação de um canal de comunicação e denúncia do mau uso do dinheiro público. Nesse aplicativo é possível marcar locais no mapa ou que necessitam de maior investimento por parte do governo, seja onde a intervenção não condiz com o volume de dinheiro gasto, ou onde as intervenções foram positivas e trouxeram benefício à comunidade.

O Wikicrimes [Furtado et al., 2010] é outra aplicação que ganhou destaque por permitir o registro da ocorrência de crimes, de forma independente dos registros oficiais de segurança pública. A informação assim coletada, portanto, assume um caráter complementar em relação às ocorrências policiais, por refletir em parte a subnotificação de crimes de baixo impacto, e de reforço em relação aos dados oficiais, por indicar concentrações de crimes no espaço urbano. Além disso, cumpre um papel educativo para a população, uma vez que informações detalhadas e especializadas sobre ocorrências policiais não são divulgadas amplamente pelas instituições de segurança pública brasileiras. A iniciativa constantemente realiza parcerias com diversos grupos da sociedade de forma a promover a divulgação da aplicação e sua utilização. Além disso foram criados serviços de alerta aos usuários da ocorrência de crimes em determinadas regiões, incentivando desta forma a ajuda mútua.

2.1.3 Citizen Science

As tecnologias que trouxeram a Neogeografia também incentivaram o desenvolvimento da *citizen science*, ou ciência produzida por cidadãos comuns. Ao contrario da Neogeografia, os usuários se submetem a um rigor técnico maior, apesar de se considerarem cientistas “não profissionais”. A princípio o termo *citizen science* representa projetos em diversas áreas que tem um foco científico de coleta análise e disseminação de dados, que são conduzidos por voluntários que não se denominam como cientistas. Haklay [2013] destaca que a diferenciação entre um projeto colaborativo e um projeto de *citizen science* algumas vezes pode ser bem clara, como na catalogação de espécies de plantas, mas por vezes não é tão fácil de caracterizar, como no caso da coleta de dados para o OpenStreetMap. O autor também descreve a vertente do *citizen science* geográfico, em que a dimensão geográfica faz parte dos dados coletados para análise de forma indireta (como fotos de plantas que eventualmente terão geotags, por exemplo) ou de forma direta, onde a leitura da posição geográfica é um pré-requisito na obtenção do dado (local onde uma espécie de planta específica foi encontrada, por exemplo). Nesse aspecto, as aplicações de VGI podem se tornar uma ferramenta de coleta de dados científicos, desde que seja bem definido, e adotado, um protocolo de obtenção

das informações.

2.2 Problemas e limitações das contribuições voluntárias

Devido à natureza voluntária das aplicações de VGI, não é possível garantir que os dados resultantes tenham qualidades como abrangência de cobertura e acurácia posicional. Pela natureza dos participantes e dos instrumentos de coleta, os dados nos quais tais restrições são feitas não serão coletados com a ajuda de voluntários; é necessário adotar os procedimentos convencionais e profissionais capacitados nesses casos. As contribuições voluntárias são tipicamente melhor vocacionadas para processos em que os dados estão amplamente distribuídos, estão ao alcance de cidadãos comuns em suas atividades diárias, e não exigem instrumentos sofisticados de coleta. Dentro da lógica de *crowdsourcing*, algumas deficiências podem ser superadas pela validação ou confirmação de dados por outros voluntários. Assim, são listadas a seguir um conjunto de questões e limitações relativas ao processo de obtenção e, principalmente, à qualidade das informações coletadas por voluntários.

- **Cobertura dos dados.** Por construção, as aplicações de *crowdsourcing* dependem da atividade do grupo de usuários que se voluntariam na tarefa de criação e disseminação de informação. Do ponto de vista geográfico, essa atividade se torna perceptível através da distribuição geográfica dos dados, sendo evidenciada pela alta concentração e detalhamento de algumas regiões e a presença de “buracos” devido a pouca informação em outras. Sui et al. [2013] atribui as grandes discrepâncias entre as áreas de grande atividade e baixa atividade à facilidade de acesso aos meios de criação de informação, utilizando como exemplo a diferença percentual de acesso à *Internet* da população entre países do “Norte Global” (América do Norte e Europa) e “Sul Global” (América Latina, África e Ásia). Além disso a baixa quantidade de informações e sua atualização pode estar associada à condição socioeconômica da região, como verificado por Haklay [2010]. Segundo Steve Coast, fundador do OpenStreetMap, há regiões que ninguém quer mapear, mas são principalmente regiões que as pessoas não estão interessadas em visitar [GISPro, 2007].

Por outro lado, tomando apenas o cenário de fácil acesso aos meios de contribuição, existem ainda regiões de grande atividade, que Haklay [2013] atribui à

grande concentração populacional, e regiões de intensa atividade ao ar livre, como parques.

- **Credibilidade.** Flanagin & Metzger [2008] analisam a credibilidade dos dados obtidos pelas aplicações de VGI. Primeiramente, os autores citam alguns pontos que tornam a credibilidade das informações coletadas questionáveis, como a falta de padrões mais rígidos de qualidade (como ocorrem em aplicações da Neogeografia), a dificuldade de se documentar as origens das fontes de informação, que podem estar indisponíveis, mascaradas ou completamente ausentes, e a inexistência de um indivíduo com conhecimento técnico específico no papel de responsável pela curadoria das informações. Outra análise realizada pelos autores identifica duas vertentes de credibilidade: a de credibilidade como uma percepção e a credibilidade como acurácia. No primeiro caso a credibilidade está associada com a ideia de “confiabilidade” e os autores abordam que esse conceito pode ser mais útil em aplicações com propósitos políticos e sociais. Já o segundo, como o nome indica, está ligado à ideia de precisão de posicionamento ou exatidão temática, sendo associado pelos autores à avaliação da produção de dados de natureza científica.
- **Precisão/Acurácia.** Como as informações obtidas pelas iniciativas de VGI podem ser provenientes de diversas fontes e instrumentos de coleta [Goodchild & Li, 2012], sua precisão pode ser indefinida ou ausente sob a forma de metadados. Além disso, algumas vertentes de *crowdsourcing* ativo tendem a apresentar dados com acurácia heterogênea, em parte devido à falta de processos científicos na coleta e medição dos dados [Haklay, 2010, 2013], o que também é causado pelo conhecimento científico limitado dos desenvolvedores dessas aplicações [Haklay, 2010; Tulloch, 2007], assim como o pouco conhecimento técnico dos envolvidos na coleta dos dados [Haklay, 2010].
- **Motivação dos voluntários.** Uma das questões que permeiam a contribuição voluntária geográfica consiste na motivação dos usuários em disponibilizarem informações. Com o intento de investigar esses motivos, Coleman et al. [2009] fazem uma análise de estudos empíricos realizados com outras formas de *crowdsourcing*, como as *wikis*. Primeiramente, os autores fazem um levantamento sobre como os usuários são geralmente classificados nesses estudos quanto ao seu engajamento e produtividade. Alguns estudos chegam a cinco categorias, que vão desde usuários que possuem pouco conhecimento, mas possuem algum interesse sobre o assunto e podem vir a contribuir, a usuários com grande experiência

profissional com o assunto. Além disso, os autores fazem uma distinção entre as aplicações de VGI, separando-as em três grupos (orientado ao mercado, redes sociais e civil/governamental, como PPGIS), destacando que, dependendo do grupo em que a aplicação se enquadra, as motivações dos usuários podem ser diferentes, e conseqüentemente, as variáveis que são avaliadas para classificá-los quanto à produção de informação podem ser diferentes. Partindo dos conjuntos de interação dos usuários, os autores traçam um conjunto de motivações (Tabela 2.1), tanto positivas quanto negativas, em que as positivas são aquelas em que os usuários visam a produção de informação útil à comunidade, e negativas aquelas em que o conteúdo gerado é não confiável.

Tabela 2.1: Motivação dos voluntários

Positivas
Altruísmo
Estímulo intelectual
Proteção ou garantia de investimento pessoal
Recompensa social
Aumento da reputação pessoal
Provem um espaço para auto-expressão criativa e independente
Orgulho de um espaço, e sua valorização
Negativas
Vandalismo
Agenda (movido por uma empresa interessada na desinformação)
Maliciosa / Criminal

Alguns autores ressaltam a importância da identificação da motivação dos usuários para o desenvolvimento de mecanismos de incentivo à utilização das ferramentas e engajamento [Maué, 2007], e mais ainda à formulação de técnicas de avaliação da credibilidade das contribuições, uma vez que o padrão de contribuição de informações corretas ou maliciosas pode ser identificado através da motivação para a qual essa informação foi passada. Adicionalmente, o grau de motivação permite inferir *scores* de confiabilidade aos dados dos usuários partindo de seu conjunto de motivações [Goodchild & Li, 2012; de Souza et al., 2014, 2013].

2.3 Soluções para aplicações de VGI

Dados os principais problemas inerentes das aplicações de VGI citados anteriormente, surgiram diversos mecanismos e técnicas que podem ser utilizados de forma comple-

mentar pelas ferramentas, a fim de buscar soluções para as questões indicadas na seção anterior. As seções a seguir descrevem algumas dessas técnicas.

2.3.1 VGI sob demanda

Como uma alternativa aos problemas de cobertura dos dados e motivação dos usuários, Chen et al. [2014] propõem uma abordagem para aplicações de VGI sob demanda orientado a tarefas. Nessa abordagem os usuários são capazes de criar missões sob a forma de perguntas sobre fenômenos ou eventos que podem estar ocorrendo em um determinado local. Na criação das tarefas, os autores têm a liberdade de delimitar o local sobre o qual desejam obter informações, bem como o formato multimídia das respostas (textual, som, imagem, vídeo, etc.). Os usuários voluntários que aceitam essas missões recebem em troca recompensas dentro da plataforma. As tarefas são então distribuídas aos voluntários, utilizando-se um sistema de recomendação, que leva em consideração a localização atual do usuário e os pontos ofertados pelo autor. O sistema funciona, assim, de forma similar a ferramentas como o *Amazon Turk*, onde as tarefas são recomendadas tendo em vista o quanto os “voluntários” gostariam de receber pela missão e o quanto o autor está disposto a recompensá-los. Ao final, a resposta entregue pelo usuário é obtida através da verificação se o usuário realmente esteve no local da missão, e através de um modelo probabilístico.

Apesar de se basear em um sistema para *crowdsensing* o mecanismo pode ser adaptado ao *crowdsourcing* ativo utilizando as informações de contribuição dos usuários como *features* para o sistema de recomendação de tarefas. A abordagem tem potencial para a solução dos problemas de cobertura dos dados, pois direciona os “sensores” (voluntários) a pontos onde existem pouca informação, e também motiva a utilização da ferramenta, pois funciona como um lembrete ou uma gameificação da ferramenta.

2.3.2 Gamificação

A *gamificação* de aplicações tem sido utilizada em diversos contextos com o objetivo de incentivar a participação de usuários através de desafios e recompensas. Do ponto de vista de *crowdsourcing*, a técnica é voltada para a resolução de problemas como a retenção de usuários menos engajados na utilização da ferramenta, a redução do contraste entre as áreas de menor e maior atividade de coleta [Antoniou & Schlieder, 2014], validação, integração e limpeza dos dados [Antoniou & Schlieder, 2014]. O primeiro problema é claramente solucionado através do fator motivacional, em que a gamificação se baseia. Os demais podem ser solucionados através do direcionamento dos usuários

(que nesse contexto são chamados de jogadores) através de desafios que podem ser vistos como tarefas. Martella et al. [2015] apresentam um conjunto de tarefas presentes em aplicações de VGI que, ao contrário do conceito apresentado na Seção 2.3.1, não se limitam apenas à coleta de informações em locais predeterminados, mas também realizam a validação de informações previamente coletadas e fazem de-duplicação e limpeza desses dados. Dessa forma, os usuários podem voluntariar suas habilidades cognitivas de forma similar ao retratado por Haklay [2013] e Grey [2009], porém na construção de metadados ligados à qualidade, que necessitam de uma avaliação manual. Do ponto de vista de cobertura dos dados, um dos principais mecanismos utilizados consiste na conquista de territórios [Martella et al., 2015], e também chamada de alocação de lugares [Antoniou & Schlieder, 2014]. Esse sistema consiste na atribuição de uma região a um usuário caso esse tenha atingido os requisitos necessários para tomar posse do terreno. No caso de Antoniou & Schlieder [2014], esses requisitos estão relacionados à presença do usuário nesses locais (Tabela 2.2), tendo portanto um viés mais próximo do *crowdsensing* que *crowdsourcing* ativo introduzido na Seção 2.1, no início do capítulo. Já Martella et al. [2015] baseiam a conquista de terrenos na quantidade de bens virtuais que o usuário possui, que no caso podem ser representados como a recompensa por execução de tarefas dentro da aplicação.

Alocação	Objetivos
Primeiro a visitar	Cobertura espacial
Enésimo a visitar	Balanceamento do jogo
Visitante mais frequente	Revisita frequente
Desalocação	Objetivos
Nunca	Simplicidade
Quando reivindicado	Dados mais atuais e balanceamento de jogo
Por decaimento de tempo	Balanceamento do Jogo

Tabela 2.2: Estratégias de alocação para jogos geográficos [Antoniou & Schlieder, 2014]

2.3.3 Verificação por pares

Uma das formas de avaliação das informações coletadas se baseia na utilização do próprio ambiente de *crowdsourcing* para sua validação [Goodchild & Li, 2012], onde os voluntários assumem tanto a tarefa de criação de novas contribuições quanto a de manutenção e curadoria dos dados. Assim sendo, Goodchild & Li [2012] apresentam duas abordagens: (i) realização da constante correção e complementação das informações fornecidas por outros usuários, e (ii) utilização de um meio social/meritocrático para identificação, filtragem, correção e remoção de problemas com as informações coletadas.

Do primeiro ponto de vista, Goodchild & Li [2012] analisam a Lei de Linus para o desenvolvimento de software livre, (“se existirem olhos o suficiente, todos os *bugs* serão óbvios” [Raymond, 1999]), amplamente utilizada para justificar a acurácia das informações de outros tipos de *crowdsourcing* como a Wikipédia, e sua aplicabilidade no contexto de VGI. Nesse sentido, mesmo que a maioria dos usuários apresentem pouco conhecimento sobre a ferramenta ou o tema da contribuição, a união de indivíduos trará a correção dos erros. Os autores mostram ainda que, apesar dessa estratégia parecer promissora num primeiro momento, existem casos em que os fatos mapeados são considerados “obscuros”, ou seja, são fenômenos que ocorrem em regiões pouco exploradas ou despovoadas, ou que se sucedem em um breve período de tempo, ou que apresentam propriedades de pouco interesse para a maior parte das pessoas.

Já a ótica social analisada por Goodchild & Li [2012] se apoia na noção de hierarquia dentro da rede de colaboração. Nesse sentido são definidos curadores ou pessoas consideradas confiáveis dentro da comunidade. A essas pessoas é conferido o poder de vetar e filtrar contribuições inválidas ou maliciosas. Dessa forma os autores destacam quatro modos através dos quais esses cargos podem ser atribuídos: (i) sistema meritocrático, ligado a mecanismos de reputação; (ii) eleição realizada por um grupo preexistente de curadores; (iii) híbrido de pessoas fixas e todos os demais usuários, e (iv) fixado pelo grupo desenvolvedor. A primeira estratégia traz à tona conceitos como reputação e confiabilidade, discutidos mais detalhadamente por Maué [2007]. Maué verifica que o sistema de reputação confere propriedades que vão além da qualidade, podendo ser utilizado para: (a) avaliar a confiabilidade dos usuários, (b) aferir os conhecimentos do usuário sobre os recursos da plataforma e técnicas de coleta de dados espaciais, (c) inferir o conhecimento local que o indivíduo possui, e (d) motivar os usuários. Como exemplos dessa abordagem destacam-se as aplicações OpenStreetMap, Wikimapia e ClickOnMap.

2.3.4 Verificação automática

A verificação automática se baseia na criação de sistemas de triagem automática [Goodchild & Li, 2012]. Para isso são utilizadas regras derivadas do contexto espacial, surgidas tanto de abstrações realizadas através de pesquisas científicas quanto por práticas amplamente adotadas por agências reguladoras. Essas regras variam de conceitos simples, como a restrição da geometria dos objetos espaciais ou ângulo máximo de curvatura em determinados tipos de vias, a mais abstratos, a exemplo do conceito inserido por Tobler [1970], na chamada Primeira Lei da Geografia:

“Todas as coisas estão relacionadas, mas coisas próximas estão mais relacionadas que coisas distantes”.

Dado esse conjunto de restrições, podem ser criados métodos de triagem automática, que rejeitariam contribuições em locais errados ou improváveis.

2.4 A relação entre VGI e IDE

Diversas aplicações de VGI emergiram após o surgimento de mecanismos que as tornaram possíveis como citado anteriormente, porém as mesmas se diferenciam quanto ao ambiente de desenvolvimento, a forma de coleta de dados, sua validação, etc. Nesse sentido [Castelein et al., 2010] realizam uma caracterização de cinco aplicações de contribuição voluntária geográfica, partindo dos conceitos empregados no desenvolvimento de Infraestrutura de Dados Espaciais (IDE)s.

Partindo dos métodos clássicos na formação de acervos de dados espaciais, Craglia [2007], analisa as diferenças entre VGI e IDE e questiona “VGI é uma gazela onde a IDE é o elefante?”, ilustrando sua percepção sobre a comparação dos dois contextos. Nesse panorama, a VGI apresenta um modelo de obtenção de dados rápido e flexível, com baixo custo, onde a informação se torna quase automaticamente disponível aos usuários, dependendo do tipo de validação realizada [Goodchild & Li, 2012]. Porém o controle de qualidade muitas vezes está submetido à avaliação de outros usuários, podendo a informação já ter sido analisada, ou estar incorreta esperando que algum avaliador a verifique. Outro problema é a falta de padronização entre os dados gerados pelas diversas aplicações, uma vez que cada uma delas foi criada e é mantida por usuários, grupos ou organizações distintas, prejudicando assim sua interoperabilidade [de Souza et al., 2013, 2014]. Já a formação de IDEs é caracterizada por uma coleta lenta e rígida, sobre um alto custo, em que a compilação dos resultados é um processo moroso, mas apresenta padrões de qualidade bem estabelecidos, típicos das instituições voltadas para cartografia oficial.

Tradicionalmente, os dados coletados por agências e empresas particulares, seguem um viés introduzido pelo *Federal Geographic Data Committee* (FGDC), que define um conjunto de temas primários necessários para suprir as demandas dos usuários [Committee et al., 1997], chamado de *framework* de dados. Esse conceito é adotado pela *National Spatial Data Infrastructure* (NSDI) norte-americana, que implementa sete camadas espaciais, em oposição à *Infrastructure for Spatial Information in Europe* (INSPIRE) que apresenta 34. Dentro desse escopo podemos destacar as primeiras aplicações de VGI, como OpenStreetMap e Wikimapia. Além delas existem também

aplicações voltadas à correção de dados espaciais, como o Google Map Maker ⁶, e o *The National Map Corps*, iniciativa da agência de mapeamento americana *United States Geological Survey*.

Existem ainda esforços sobre as aplicações de VGI visando a documentação das informações coletadas através de metadados, aproximando mais ainda esse tipo de fonte ao modelo de organização das IDEs. Nesse sentido de Souza et al. [2013] propõem um *template* para catalogação que leva em consideração cinco dimensões: Identificação, Horário da Contribuição, Geoposicionamento, Qualidade e Auditoria e Distribuição. A primeira delas, como o seu próprio nome revela, é utilizada para identificar e classificar a contribuição, sendo composta por variáveis como título, categoria, tipo, *software* e *website* de origem. A segunda representa a dimensão temporal, indicando quando a contribuição foi obtida, assim como suas atualizações. O geoposicionamento compõe, além da geometria do objeto, o seu retângulo envolvente. A qualidade é uma das dimensões complexas desse *template*, pois apresenta informações como o *score* da contribuição assim como o método de avaliação, e ainda o número de visualizações e edições de suas informações, além de informações adicionais que podem ser utilizadas para avaliar sua qualidade, como o nível de zoom do mapa em que o usuário realizou a contribuição. Por fim a Auditoria e Distribuição apresenta informações sobre o usuário responsável pela contribuição, como idade, *ranking* do usuário, assim como o tamanho da escala do *ranking*. Do ponto de vista da distribuição, o *template* agrega informações sobre o *website* que veicula essa informação.

2.5 Frameworks

Com o objetivo de facilitar e reduzir os custos na implementação de novas aplicações, cinco abordagens de *frameworks* foram propostas na literatura [Silva & Davis Jr, 2008; Okolloh, 2009; Sheppard, 2012; Davis Jr et al., 2013; de Souza et al., 2014]. Dentre estas, o projeto desenvolvido por Silva & Davis Jr [2008] oferece uma solução para ambiente *Web*. A aplicação permite criação de novos temas de coletas pelos usuários, assim como a customização dos elementos de interface. O foco deste trabalho está em viabilizar que pessoas com pouca experiência em desenvolvimento de software possam criar suas próprias iniciativas. Apesar de atender as necessidades do público comum, do ponto de vista dos desenvolvedores, ele possui pouca flexibilidade, uma vez que está preso a softwares proprietários, e suas versões de código aberto ainda são rudimentares, sendo a replicação em outros sistemas ainda muito trabalhosa.

⁶Google Map Maker - <http://www.google.com.br/mapmaker>

Como descrito anteriormente, o Ushahidi surgiu como uma iniciativa voltada para PPGIS, mas devido ao alcance dos serviços prestados, acabou tomando a direção de transformação em *framework* para o desenvolvimento de aplicações voltadas para o cenário de crise. Do ponto de vista operacional, o *framework* permite que a customização seja realizada por pessoas com alguma experiência de desenvolvimento. [Okolloh, 2009] apresenta diversas funcionalidades implementadas, dentre elas a utilização de SMS, devido a possíveis dificuldades de acesso à Internet nesses casos, e a forma com que novas contribuições são avaliadas.

Sheppard [2012] também apresenta um *framework* modularizado para ambiente móvel e *Web*. O foco dessa abordagem está em usuários mais experientes, com domínio sobre esses cenários de desenvolvimento, e em reduzir as dificuldades na confecção de novas aplicações, sem perder a flexibilidade de seus componentes. O autor utiliza HTML5.0, que confere um conjunto de funcionalidades *cross-plataform*, podendo ser interpretado e executado por navegadores em *desktops* e celulares. Além disso, parte dos componentes situados do lado do servidor foram desenvolvidos em Python utilizando o *framework* Django, o que garante a inclusão de novos módulos e a utilização de uma diversidade de bibliotecas para esse ambiente. O autor apresenta a organização de seu projeto em três principais subestruturas: “*wq.app*”, “*wq.db*”, “*wq.io*”. O componente *wq.app* refere-se à aplicação cliente, onde os usuários realizam suas contribuições. Como já citado anteriormente, a especificação do HTML5.0 possibilita a chamada de funções dos navegadores, que incluem a identificação do posicionamento geográfico, o armazenamento em cache ou em arquivos de persistência, além do envio de arquivos como imagens salvas no respectivo ambiente. Segundo o autor, essas quatro funções garantem o desenvolvimento de quaisquer aplicações simples de VGI, com capacidade de funcionar *off-line* e realizar sincronização, ficando a cargo do desenvolvedor acrescentar funções adicionais. O segundo componente recebe o nome de *wq.db* e realiza o papel de interlocutor entre o sistema de gerenciamento de banco de dados e o desenvolvedor. A união do *framework* Django com a ferramenta South permite a abstração na definição do modelo conceitual sem a necessidade de conhecer as nuances do gerenciador de bancos de dados, como Oracle, PostgreSQL/PostGIS ou MySQL por exemplo. Adicionalmente, o Django permite a utilização de uma biblioteca que realiza abstração em alto nível de protocolos OAuth para interação com redes sociais. Por fim, os autores definiram um componente de interoperabilidade *wq.io*, desenvolvido em Python, que permite que os usuários definam um processo de importação e exportação de arquivos do banco de dados da aplicação.

Seguindo a mesma linha, Davis Jr et al. [2013] apresentam uma abordagem flexível que prioriza o desenvolvedor e não o usuário final. Nesse trabalho são traçadas

funcionalidades básicas e é desenvolvida uma arquitetura em três camadas: *Dados*, *Serviços* e *Aplicações*. A camada de *Dados*, define uma modelagem conceitual que reflete as funcionalidades da aplicação, sendo incorporadas no modelo. As classes apontadas compreendem entidades voltadas ao armazenamento de contribuições e suas versões, controle de usuários e sua reputação, e por fim de avaliação de contribuições.

A camada de *Serviços* é dedicada a mediar a troca de informações entre as aplicações e os bancos de dados, podendo-se implementar qualquer tipo de protocolo de troca de mensagens sem alterar o esquema dos dados ou as aplicações já existentes. Como exemplo de componentes desta camada os autores citam os serviços padronizados pelo *Open Geospatial Consortium*, implementados pelo catálogo de serviço Geoserver.

A camada de *Aplicação* foi organizada utilizando o padrão de projeto *Model-View-Controller* (MVC), expressando uma modularização interna. Desta maneira sua estrutura interna pode ser reutilizada em outros projetos. As *Views* correspondem aos elementos responsáveis pela interface da aplicação. Estas são divididas por funcionalidades levantadas através dos requisitos do sistema. Os *Controllers*, são responsáveis por processar informações obtidas pelos *Models* e *Views*, além dos componentes de hardware e software do ambiente. Os resultados são devidamente repassados, concluindo assim a requisição realizada pelos demais elementos. Por fim, o *model* é responsável por se comunicar com a camada de serviços. Este componente, portanto, implementa um protocolo específico de troca de mensagens que podem ser interpretadas por um *web-service*.

Diferente das abordagens anteriores, o ClickOnMap [de Souza et al., 2014] tem o foco na criação de metadados e na avaliação das contribuições. Como foi citado anteriormente, um dos problemas da utilização de VGI em projetos científicos é a sua parca garantia de qualidade, credibilidade, autoridade e padronização. Baseando-se no DM4VGI [de Souza et al., 2013], do mesmo grupo de pesquisa, os autores desenvolveram uma ferramenta para criação de aplicações de VGI capaz de gerar automaticamente metadados sobre as contribuições. Devido ao fato da especificação do DM4VGI impor a documentação da qualidade da contribuição e nível de experiência do voluntário no qual a realizou, o ClickOnMap se utiliza desse modelo implementando métricas de qualidade para as contribuições baseadas no nível do voluntário, assim como na avaliação dos demais usuários. O *framework* implementa ainda meios de descrição da contribuição através de *wikis*, que podem ser revisadas pelos demais usuários, sendo também uma forma de avaliação da veracidade da contribuição.

O sistema também oferece um conjunto de ferramentas de análise como visualizações geográficas e gráficos, ambos com suporte à filtragem por tipo de contribuição. Além disso é possível realizar a busca sobre os metadados das contribuições.

Tendo analisado as funcionalidades descritas, a presente dissertação objetiva a criação de um *framework* que reduza o trabalho de desenvolvimento de novas aplicações, e o nível de conhecimento necessário sobre as tecnologias envolvidas nessa tarefa. Com esse intento, o trabalho propõe uma arquitetura baseada no trabalho de Vellozo et al. [2013], com os meios de validação e hierarquização dos usuários proposta por de Souza et al. [2013], que permita uma incorporação posterior de sistemas automatizados para a criação de metadados de Souza et al. [2014, 2013], e que atenda às diferentes modalidades de aplicação de VGI propostas por Sheppard [2012], encapsuladas sob a visão de múltiplos temas.

Capítulo 3

Abordagem

O capítulo anterior apresentou um contexto geral sobre as aplicações de *crowdsourcing* geográfico ativo e suas diversas facetas. Foram também apresentados seus principais problemas, em sua grande maioria decorrentes de seu processo peculiar de coleta de informação, assim como algumas abordagens para a solução desses problemas. Por fim, o capítulo apresentou uma análise de alguns *frameworks* existentes na literatura. Partindo desse contexto, o presente capítulo apresenta uma abordagem que combina parte das soluções para problemas típicos de aplicações de VGI com as propostas de *framework* de coleta de dados fornecidos por voluntários, em uma visão de múltiplos temas.

3.1 Temas

As iniciativas VGI em geral referem-se a um fenômeno, evento ou estrutura física sobre os quais se deseja obter dados, tendo em vista a distribuição geográfica em que ocorrem ou se situam. A esse tipo de entidade de interesse, o presente trabalho atribui o nome de *Tema*.

Os *frameworks* descritos anteriormente são em geral capazes de produzir aplicações e mecanismos de coleta de dados voltados a conjuntos de temas únicos, em que cada tema tem a mesma estrutura. Os temas são diferenciados nas contribuições genéricas através de atributos específicos como categorias, subcategorias e/ou tipos. Denominamos essa abordagem de *conjunto de temas*. Observamos que um dos problemas dessa abordagem é a ausência de especialização dos mecanismos de coleta para o cenário específico de cada categoria.

Considere-se, por exemplo, uma aplicação de VGI que tem por objetivo analisar a ocorrência de acidentes em vias urbanas e estradas. Existem dois tipos de consumidores

para as informações geradas por essa aplicação. O primeiro inclui os responsáveis pela iniciativa, que podem ser: (i) uma pesquisa científica sobre mobilidade urbana (*Citizen Science*), (ii) um grupo da sociedade interessado na tomada de medidas públicas para melhorias urbanas (PPGIS) ou (iii) um órgão do governo responsável pela engenharia de trânsito (Departamento de Estradas e Rodagem (DER), por exemplo). O segundo tipo de consumidor são os próprios voluntários, que podem ser notificados sobre novos acidentes, e sobre regiões onde devem possuir atenção redobrada devido ao índice de acidentes ou condições da pista.

Nesse exemplo podemos definir quatro temas distintos para serem mapeados, sendo o primeiro a ocorrência de *Acidentes* em si, acompanhado de estado de conservação da *Sinalização*, que pode ser um dos fatores causadores de acidentes, a presença de *Radares*, que pode ser um fator para redução do número de acidentes, e a *Taxa de fluxo de tráfego* (número de veículos / intervalo de tempo), que pode apresentar uma correlação com o índice de acidentes. Nesse caso os instrumentos necessários para a coleta informações sobre *Acidentes* são diferentes dos necessários para a catalogação do estado de conservação de placas de *Sinalização*. Usando os *frameworks* citados anteriormente, as contribuições sobre *Acidentes* e *Sinalização* apresentariam a mesma estrutura, sendo diferenciadas apenas por uma categoria ou tipo informado pelo usuário.

Esta dissertação propõe outra abordagem, que denominamos *multitemática*, pela qual é possível definir diferentes temas no mesmo instrumento de coleta, sendo cada qual projetado e detalhado individualmente. Consideramos importante que seja possível especializar os tipos de informações que deverão ser coletadas respeitando as necessidades individuais de cada tema, como foi exemplificado anteriormente. Adicionalmente, essa especialização permite a introdução de restrições aos tipos de dados a serem coletados bem como a utilização de artefatos complementares como *wiki*, *arquivos*, *comentários*. Além disso, o prisma de temas pode ser aplicado às demais funções do *framework*, definindo elementos como *representação geográfica*, *forma de visualização* e *sistema de reputação* aplicável à comunidade de voluntários envolvida.

Do ponto de vista de representação geográfica, é importante poder especificar o tipo geométrico desejado na contribuição para um tema, o que torna mais fácil a construção de visualizações. Por sua vez, as visualizações podem ser desenvolvidas levando em consideração a forma de representação escolhida, sendo também definidos parâmetros de apresentação como cor e simbologia. Além disso, pode-se desenvolver filtros de exibição, tendo em consideração o período de tempo em que a contribuição é válida ou relevante, como no caso de eventos como *Acidentes*, que ocorrem com duração mais curta se comparados com a vida útil da *Sinalização* de trânsito, mesmo sendo ambos representados usando pontos no espaço.

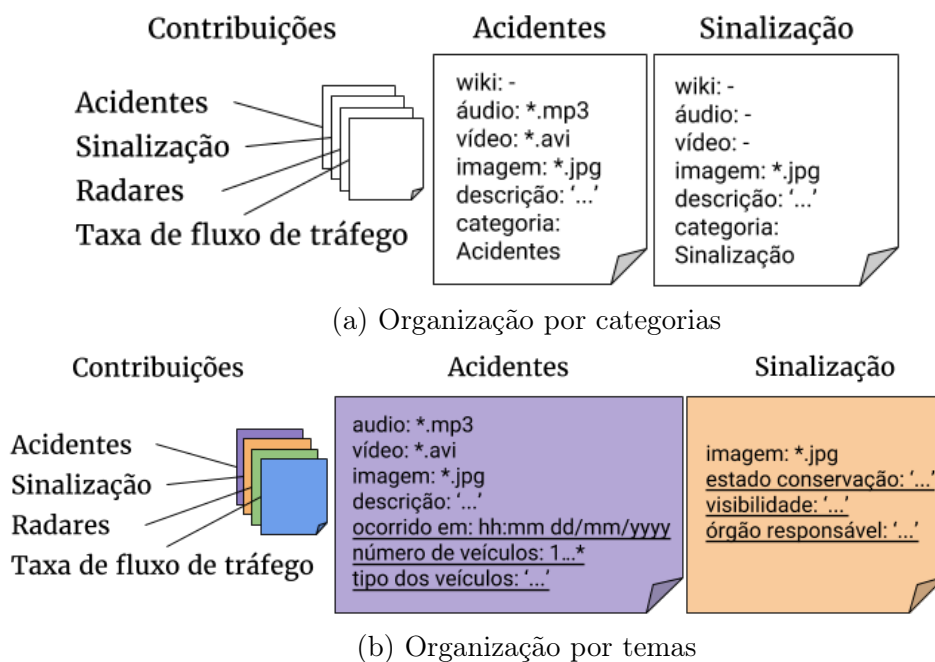


Figura 3.1: Organização em categorias vs. temas

No caso dos sistemas de reputação, o objetivo é definir uma forma de distribuir pontuação por ações levando em consideração os temas em que o voluntário colaborou. A partir da pontuação acumulada, é possível traçar o nível de envolvimento, conhecimento específico ou de experiência pessoal do voluntário, introduzindo a ideia de “expertise” vinculada a cada tema. Nas abordagens de conjunto de temas, como não é realizada essa distinção, as contribuições de um voluntário em temas diferentes são acumuladas e valorizadas da mesma maneira. Por exemplo, se um voluntário que contribui quase exclusivamente sobre *Acidentes* realiza uma contribuição eventual sobre a *Taxa de fluxo de tráfego*, essa contribuição não deveria ser valorizada da mesma forma que uma contribuição de um especialista nesse tema. Além da separação de contextos na valorização das ações dos voluntários, em nossa proposta torna-se possível também diferenciar o sistema de recompensas que atribui valor às contribuições, em um mecanismo de incentivo à participação. Nesse caso, a abordagem ameniza os efeitos de temas cujos fenômenos ocorrem com mais frequência, e por sua vez apresentam proporcionalmente maior volume de contribuições, se comparados com os demais. Dessa forma é possível calibrar os pesos dos temas a fim de se comparar o esforço de coleta entre voluntários exclusivos de temas distintos.

Em resumo, a separação da reputação por temas valoriza mais o domínio de conhecimento do usuário. Na abordagem multitemática podemos ter três tipos de cenários quanto ao conjunto de temas e com relação ao nível de conhecimento requerido

do voluntário para a realização de uma contribuição:

- Apenas temas que não necessitam capacitação técnica para contribuição;
- Apenas temas que exigem capacitação técnica;
- Misto, parte dos temas exigem alguma capacitação e parte não exige;

No primeiro caso podemos ver que aplicação multitemática está centrada nos princípios da Neogeografia, onde não existe nenhum tipo de restrição quanto ao conhecimento técnico do voluntário no momento da contribuição, e o foco está na produção de informação de forma irrestrita. Nesse caso, o nível de *expertise* entre temas pode perder seu significado, uma vez que todos os usuários têm igual capacidade de produzir informação em todos os temas.

No segundo caso podemos classificar a aplicação multitemática com características do *Citizen Science*, onde muitas vezes é exigido, como pré-requisito, que os voluntários possuam algum tipo de conhecimento técnico, fornecido ou não pelos administradores da aplicação. Nesse caso também o nível de *expertise* por tema pode não ser discriminativo, tendo em vista a qualidade da contribuição e o nível de conhecimento do voluntário, uma vez que todos podem estar sujeitos a um pré-treinamento em todos os temas, porém pode ser discriminativo quanto ao nível de envolvimento, dado o volume de contribuições. Além disso, as aplicações multitemáticas com características de *Citizen Science* podem ainda ser compostas por temas interdisciplinares. Nesse caso específico, a diferença entre níveis de especialidade de cada tema pode ser significativa, levando em consideração o conhecimento específico prévio sobre o tema dos voluntários.

No terceiro caso, a aplicação multitemática seria composta por uma mistura de temas com dinâmicas de Neogeografia e *Citizen Science*, podendo até mesmo apresentar um viés público-participativo (PPGIS). Nesse caso a diferença entre os níveis de especialidade em cada tema pode ser também ser significativo, uma vez que existem em um mesmo ambiente temas que exigem níveis distintos de capacitação, especialização de seus voluntários e volume de contribuições.

Através da separação de contextos, é possível ainda atribuir pesos diferentes às ações de cada tema, o que permite que o administrador da aplicação direcione a produção de um tipo de conteúdo (comentário, *wiki*, envio de arquivos, etc.) para um tema de forma independente dos demais. Além disso, é possível ajustar os pesos das ações de cada tema a fim de relativizar a pontuação final dos usuários, utilizando uma taxa de crescimento menor para temas sobre fenômenos que ocorrem com mais frequência, e de modo contrário, utilizando uma taxa de crescimento maior para os temas sobre fenômenos que ocorrem com maior frequência. Assim sendo, caso seja interessante a

criação de um *ranking* geral com todos os temas, é possível juntar a pontuação de cada tema para compor o *score* geral do voluntário.

O conceito de múltiplos temas pode se assemelhar com o apresentado pelo projeto *Crowdmap*, porém as diferenças se tornam visíveis com o nível de customização dos temas. Como o *Crowdmap* é fruto da ferramenta *Ushahidi*, esta ainda se baseia no conceito de testemunhos, o que limita o formato das informações associadas ao dado geográfico a um campo de texto que tem a mesma semântica de contribuição testemunhal para todos os temas criados. Além disso, não existe um sistema de pontuação e reputação que os usuários possam ter acesso para acompanhar sua progressão, o que poderia ser utilizado para analisar a qualidade das contribuições.

Além da liberdade de customização dos temas, do ponto de vista técnico nosso projeto permite que os administradores tenham total controle sobre a extensão de suas funcionalidades, assim como do armazenamento e análise dos dados.

Partindo da definição da abordagem multitemática, e considerando a argumentação exposta acima, as próximas seções apresentam elementos do projeto do *framework* proposto.

3.2 Requisitos

No processo de planejamento do *framework* foram levantados um conjunto de requisitos funcionais e não-funcionais. O primeiro grupo, como o próprio nome define, contém todas as funcionalidades que o sistema deve possuir e define o escopo dos módulos a serem implementados, visando oferecer alternativas para o tratamento dos problemas típicos de aplicações VGI. Já o segundo grupo inclui requisitos técnicos, que determinaram a escolha do ambiente de desenvolvimento, justificando assim as tomadas de decisões quanto ao uso das tecnologias disponíveis.

3.2.1 Requisitos Funcionais

1. **Identificação dos voluntários.** O voluntário deve ser capaz de se identificar no sistema [Davis Jr et al., 2013; Sheppard, 2012; de Souza et al., 2014], caso seja interessante para a dinâmica do tema em questão. Caso o voluntário se identifique, é possível atribuir autoria ao dados coletados, permitindo a criação de instrumentos para avaliação da credibilidade dos mesmos, como mencionado por Flanagan & Metzger [2008]. Caso seja interessante para a aplicação permitir a anonimização dos voluntários, as contribuições assim realizadas deverão ser identificadas como anônimas [Okolloh, 2009; de Souza et al., 2014].

2. **Criação de novos temas.** O administrador poderá optar entre as seguintes alternativas de representação geográfica para os temas: ponto, linha ou polígono. Poderá incluir uma série de atributos descritivos que serão vinculados a cada tema. Os tipos desses atributos deverão incluir strings, datas e valores numéricos. O administrador deverá também ser capaz de escolher se um tema permitirá o acréscimo de informações adicionais, como arquivos anexos e *Wikis*, como proposto por de Souza et al. [2014]. Adicionalmente, o administrador deve ser capaz de optar por permitir ou não que as contribuições sejam comentadas pelos voluntários. Dessa forma, temas que descrevem eventos podem omitir essa função [Sheppard, 2012], pois nesse caso predominam contribuições efêmeras.
3. **Definição de características da Visualização.** Em cada tema, será permitido escolher uma forma de visualização adequada à geometria, selecionando parâmetros visuais como cor e simbologia [Davis Jr et al., 2013; de Souza et al., 2014].
4. **Tempo de exibição.** O *framework* deve possibilitar os administradores da aplicação colaborativa delimitar um tempo em que as contribuições devem deixar de serem exibidas. Isso permite que temas para o mapeamento de eventos tenham suas contribuições omitidas quando o tempo de expiração for atingido desde sua contribuição, como foi utilizado por Vellozo et al. [2013].
5. **Definição dos valores das recompensas.** O autor da contribuição deve ser capaz de definir quantos pontos cada ação do voluntário sobre um tema será recompensada. Assim sendo, dado um conjunto de ações permitidas no sistema, cada tema deve possuir suas regras de pontuação próprias, que mais se adeque à natureza do mesmo.
6. **Pontuação dos voluntários.** Caso o voluntário se identifique na aplicação colaborativa, suas colaborações deverão ser pontuadas tanto em ações positivas quanto negativas [Coleman et al., 2009]. Assim sendo, conforme Coleman et al. [2009] constatou, as ações negativas devem acarretar em redução dos pontos e consequente redução do desempenho desse voluntário no tema em que a ação foi realizada. Dentre as colaborações do voluntário estão: criação de nova contribuição, comentário, avaliação de contribuições, envio de arquivos, edição de *wiki*, por exemplo. Como foi citado anteriormente as recompensas recebidas pelo voluntário devem variar conforme as regras de pontuação atribuídas ao tema em questão. No caso de colaborações de voluntários anônimos, a pontuação deve ser omitida. Esse processo se baseia no trabalho de de Souza et al. [2014], e tem relação direta com a credibilidade das contribuições do voluntário [Flanagin &

Metzger, 2008], assim como permite a motivação dos voluntários na obtenção de recompensas [Maué, 2007; Goodchild & Li, 2012]. Através dessa sistema de pontuação, o *framework* tornaria-se suscetível à futuras expansões como: gamificação [Antonioni & Schlieder, 2014; Martella et al., 2015], mecanismos de coleta orientados a missões [Chen et al., 2014] e introdução de um sistema de metadados [de Souza et al., 2013].

7. **Categorias de *expertise*.** Cada tema deve possuir um conjunto de categorias de *expertise* predeterminados, sendo possível modificar a pontuação mínima e máxima para que o voluntário pertença a uma categoria em cada tema. Esse requisito parte das categorias de voluntário descritas no trabalho de de Souza et al. [2014]. No caso de contribuições em temas anônimos, as categorias deverão ser atribuídas apenas à voluntários identificados.
8. **Sistema de reputação** O *framework* deve incluir um sistema de reputação. Esse sistema utilizará a pontuação obtida pelo voluntário através de suas ações para inferir sua reputação, como sugerido por de Souza et al. [2014, 2013]. Dessa forma, é possível introduzir uma estrutura hierárquica entre os voluntários [Maué, 2007; Goodchild & Li, 2012] utilizando categorias de *expertise* como variável de hierarquização.
9. **Feedback.** A aplicação colaborativa deve prover *feedbacks* aos voluntários sobre seu volume de contribuições e pontos adquiridos através de suas colaborações.
10. **Contribuições.** Uma vez que o tema foi definido, a aplicação colaborativa deve ser capaz de receber contribuições geograficamente localizadas, seguindo a geometria especificada na criação do tema. As contribuições devem ter sua autoria identificada no caso de voluntários registrados, e serem identificadas como anônimas caso contrário. No caso dos voluntários identificados, a aplicação colaborativa ainda deve recompensá-los com os pontos relativos à esse tema, pelo ato da contribuição.
11. **Avaliação das contribuições.** A aplicação colaborativa deve permitir que os voluntários sejam capazes de avaliar as contribuições de outros voluntários, atribuindo-lhes uma pontuação que reflita sua percepção de qualidade sobre a contribuição [de Souza et al., 2014], aproveitando assim o esforço cognitivo dos mesmos [Haklay, 2013; Goodchild & Li, 2012].

12. **Wikis.** O voluntário deve ser capaz de criar e editar as *wikis* dos temas que permitem tal mecanismo. Esse ato deve ser recompensado conforme as regras de pontuação do tema.
13. **Comentários.** Caso o tema possa receber comentários, a aplicação colaborativa deve ser capaz de incorporar esses comentários e relacionar os mesmos à contribuição em questão. Além disso, os voluntários que realizarem as contribuições devem obter uma recompensa referente à sua ação no tema específico.
14. **Anexos.** Se o tema for configurado para admitir a utilização de anexos para descrição de suas contribuições, a ferramenta deve ser capaz de receber e armazenar esses arquivos, assim como ser capaz de listar e apresentar esses anexos na exibição da contribuição. A cada anexo enviado o voluntário deverá receber a devida pontuação por sua colaboração [Coleman et al., 2009], que pode variar com as regras de pontuação atribuídas ao tema.
15. **Registro do histórico de contribuições.** Cada contribuição realizada pelo voluntário deve ser registrada pela aplicação colaborativa [de Souza et al., 2014], permitindo tanto a recuperação das contribuições para fins de análise temática, temporal ou espacial, bem como para fins de auditoria, como em um *log*.

3.2.2 Requisitos Não-Funcionais

1. **Extensibilidade.** O *framework* precisa ser extensível, para viabilizar o acréscimo de novas funções, refletindo novas técnicas para melhorar seu desempenho, bem como incentivar o engajamento dos voluntários e o aumento da qualidade das contribuições. Essas incluem, conforme discutido no Capítulo 2, funções voltadas para gamificação, contribuições orientadas a missões, etc.
2. **Flexibilidade.** O *framework* deve poder ser usado para gerar aplicações adaptáveis a diferentes configurações de ambientes, envolvendo diferentes sistemas operacionais, SGBDs e navegadores *Web*.
3. **Documentação.** Os componentes e ambiente de desenvolvimento devem ser criados ou escolhidos tendo em vista a documentação disponível para sua utilização, reduzindo o tempo de desenvolvimento em eventuais manutenções futuras do *framework*.
4. **Utilização de *Object-Relational Mapping* (ORM).** Em geral, as bibliotecas ORM dão suporte a diversos SGBDs e tornam as peculiaridades das linguagens

de consulta e descrição dos dados transparentes ao desenvolvedor, e em alguns casos dão ainda suporte a tipos espaciais. Assim sendo, esse requisito permite que *framework* seja mais flexível em relação ao tipo de ambiente à disposição do administrador.

5. **Funcionamento em navegadores Web.** Ambos o *framework* quanto aplicação colaborativa deve, no mínimo, funcionar em navegadores *Web*, sendo os instrumentos de colaboração relativos à esse cenário.
6. **Funcionamento em plataformas móveis.** A princípio o *framework* e a aplicação colaborativa deve poder ser utilizada em plataformas móveis, usando o navegador específico da plataforma. Em futuras expansões, pode-se desenvolver um *framework* para geração de aplicações colaborativas híbridas ou nativas para essa plataforma.
7. **Single Page Application (SPA).** O modelo de aplicação SPA foi desenvolvido para ambiente *Web* a fim de substituir o modelo antigo, onde o conteúdo estático (arquivos HTML e JavaScript) e dinâmico eram requisitados a cada renderização. As SPAs, como o próprio nome diz, têm seu conteúdo estático requisitado apenas uma vez. Quando o usuário navega em outras seções do *framework*, é requisitado apenas o conteúdo dinâmico. Dessa forma, as transições entre seções dos sistemas são mais rápidas, promovendo uma melhor navegabilidade.
8. **RESTful services.** O *framework* utiliza o modelo de arquitetura de serviço *RESTful* proposto por Fielding [2000]. O padrão *RESTful* é caracterizado pela divisão da arquitetura em camadas, sendo que em cada camada são definidos componentes que possuem um protocolo bem definido de comunicação. Outra característica desse padrão é que os componentes das camadas não armazenam o estado das demais, sendo portanto totalmente independentes. Assim, a utilização do padrão *RESTful* reduz a complexidade no desenvolvimento do *framework*, e promove a extensibilidade através do encapsulamento de funções em novos componentes.

3.3 Arquitetura

Seguindo a arquitetura proposta por Davis Jr et al. [2013], a abordagem pode ser dividida em três camadas: *aplicação*, *serviços* e *persistência*, como pode ser visto na Figura 3.2. Uma das vantagens dessa abordagem é a separação de responsabilidades

entre componentes, o que torna o desenvolvimento do *framework* mais conciso, reduz sua complexidade e torna mais simples a implementação de futuras extensões ao *framework*. Além disso, essa divisão permite que componentes sejam substituídos ou adicionados, como uma nova versão para outro ambiente (camada de aplicação), ou uma API desenvolvida em outra linguagem, por exemplo, contanto que o contrato da interface de comunicação entre as camadas se mantenha, não exigindo que os demais módulos sejam modificados.

A camada de persistência é composta por um SGBD, que necessariamente deve possuir suporte a dados espaciais. A modelagem do esquema relacional para esse componente será apresentada mais adiante ainda nesse capítulo (Seção 3.4).

Camada de Aplicação	Aplicação Web
Camada de Serviços	API
Camada de Persistência	SGBD

Figura 3.2: Visão geral da arquitetura

Acima da camada de persistência está a camada de serviços. Essa camada possui duas funções principais: limitar o acesso aos dados da camada de persistência e processar os dados antes que estes sejam enviados à camada de persistência. Nesse componente se encontra a maior parte da lógica necessária pela aplicação, como o controle do processo de criação de novos temas, controle da reputação dos voluntários, e armazenamento de anexos, por exemplo.

A última camada contém as aplicações. O presente trabalho considerou o desenvolvimento de uma aplicação *web*, que posteriormente pode ser migrada para ambiente *mobile*, de modo à atender aos requisitos apresentados. Nesse caso, como foi citado anteriormente, os demais módulos não precisarão ser modificados.

3.4 Modelagem dos Dados

Levando em consideração o conceito de tema apresentado anteriormente, assim como os requisitos funcionais, podemos expandir a modelagem dos dados a partir dos três módulos principais: *Tema*, *Reputação* e *Usuários* (Figura 3.3).

O módulo *Tema* engloba todas as classes básicas para se compor um tema, sendo essas: *Tema*, *Contribuição*, *Wiki*, *Arquivos*, *Comentários* e *Avaliação*. Dentre as classes

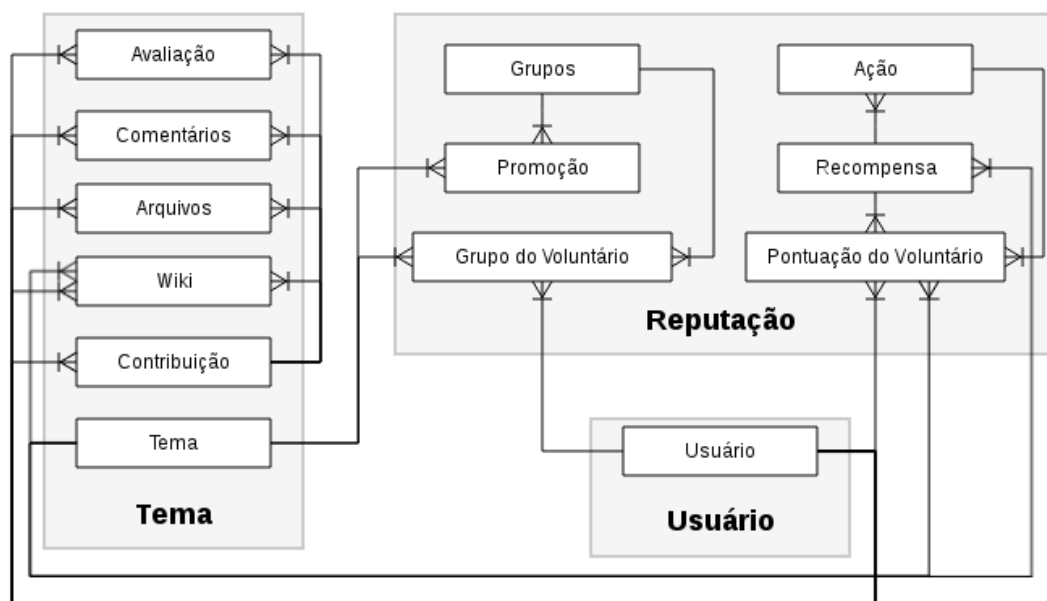


Figura 3.3: Diagrama de classes simplificado

listadas, a classe *Tema* talvez seja uma das mais importantes implementadas pelo *framework*. Essa classe descreve todas as configurações necessárias para compor um tema, como o nome, a geometria, os atributos da contribuição e se o tema permite a utilização de artefatos complementares (wiki, arquivos e comentários). A definição dos atributos em especial, é utilizada para a criação das tabelas de contribuição do respectivo tema e na definição dos elementos HTML utilizados na renderização do formulário da contribuição.

A classe *Contribuição* é portanto uma espécie de classe “genérica”, que assim como na definição de linguagens de programação, essa classe pode apresentar atributos diferentes dependendo do tema a que se refere. Apesar dessa flexibilidade, a classe mantém um conjunto de atributos fixos como identificação do voluntário, versão, data de criação, modificação e exclusão. Os demais atributos são definidos pela classe *Tema* como foi descrito anteriormente.

As classes *Wiki*, *Arquivos*, *Comentários*, *Avaliação* utilizam um artifício que permite a rastreabilidade das contribuições a que se referem, para isso as mesmas apresentam um atributo que representa a referência ao tema a qual a contribuição faz parte a o identificador da contribuição em si. Além disso cada uma dessas classes apresenta atributos específicos necessários para armazenar as informações de sua finalidade, como “corpo” para *Wiki*, endereço, tipo e formato de arquivo para *Arquivos*, comentário para *Comentários*, *score* mínimo, máximo e médio para *Avaliação*. As classes *Wiki* e *Avaliação* apresentam sempre o estado atual da desses artefatos com relação à contribuição,

assim sendo, ambas as possuem classes auxiliares que representam o histórico desses artefatos, como a avaliação de cada individuo sobre uma contribuição para se computar a *Avaliação* da contribuição, e a versão da *Wiki* após um usuário edita-la.

O módulo de *Usuário* inclui apenas a classe que recebe o mesmo nome do conceito (*Usuário*). A mesma é composta por atributos para descrição do usuário, como nome e sobrenome, e-mail e senha para acesso à aplicação, a posição inicial de visualização para o mapa, e atributos booleanos que indicam se ele é um administrador ou voluntário, se confirmou seu e-mail ou se foi banido da aplicação.

O último módulo, o de *Reputação*, engloba todas as classes necessárias para a criação de um sistema de reputação por temas. Assim sendo, o mesmo é composto pela classe *Ação*, que descreve as ações que um voluntário pode realizar ao contribuir em um tema, e pela classe *Grupo* que descreve os grupos de *expertise* que um voluntário pode fazer parte. Adicionalmente o módulo contém a classe *Recompensa*, que como o nome sugere, descreve as recompensas que o voluntário recebe ao realizar ações em um tema, onde a pontuação total é descrita pela classe *Pontuação do Voluntário*. Por fim, a progressão do voluntário nos grupos de *expertise* de um tema é delimitado através da classe *Promoção*, e a cada novo grupo atingido em um tema por um voluntário é descrito pela classe *Grupo do Voluntário*.

3.5 Papeis dos Usuários

Os papeis de usuários delimitam as funcionalidades às quais um grupo de usuários pode ter acesso. No caso da presente abordagem foram definidos três papeis: de voluntário anônimo e identificado, e administrador, como pode ser visto na Figura 3.4.

O voluntário anônimo é capaz de visualizar as contribuições existentes no sistema, mas com a restrição de identificação dos autores desses dados, comentários realizados e informações complementares como *wiki* e *anexos*. Esse tipo de voluntário pode realizar apenas contribuições geográficas que são identificadas como realizadas por voluntários anônimos.

O voluntário identificado pode realizar as funções mais simples do sistema, como realizar contribuições, visualizar contribuições, criar e editar *wikis*, comentar contribuições, avaliar contribuições, editar o próprio perfil, assim como visualiza-lo.

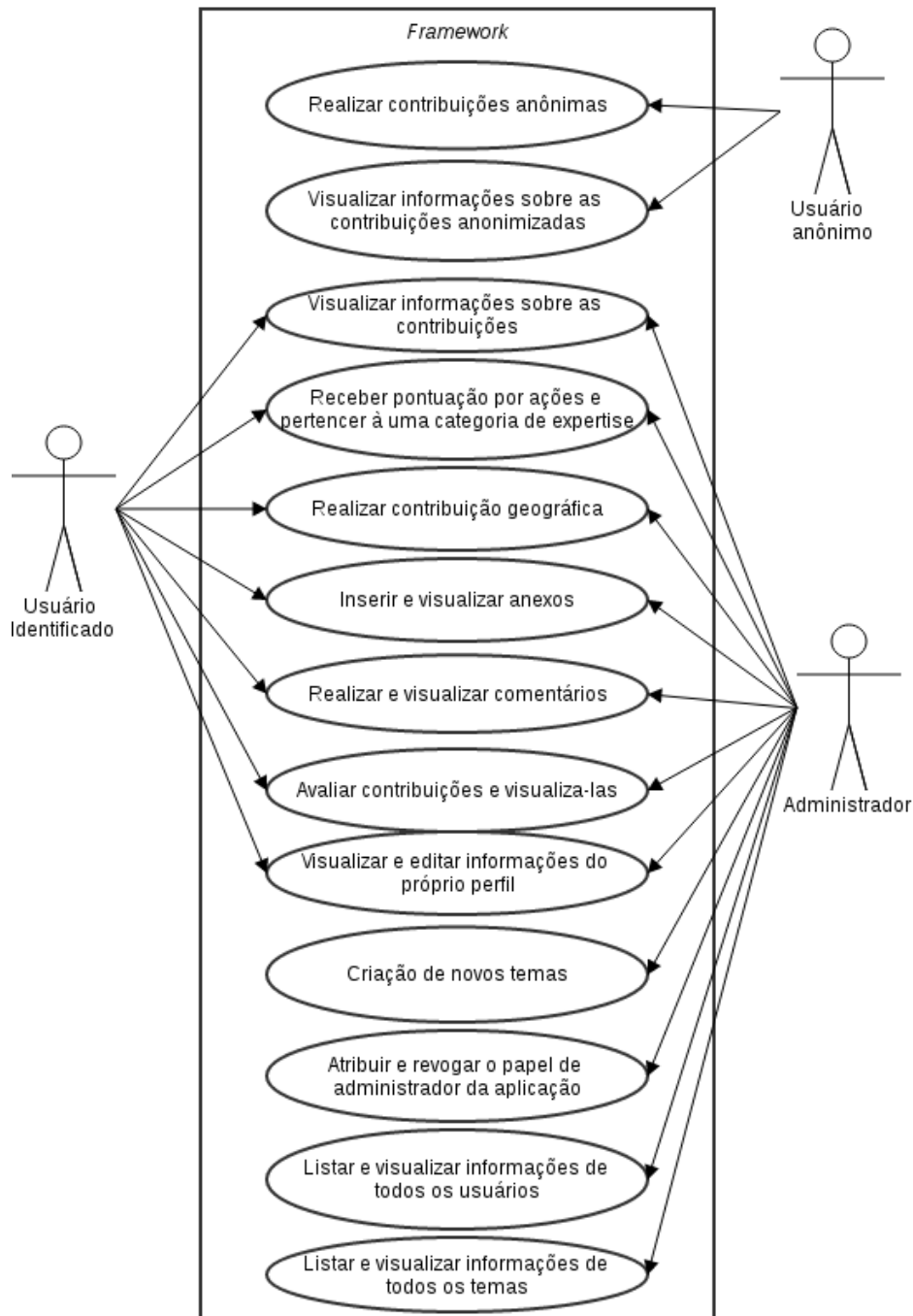


Figura 3.4: Diagrama de casos de uso de papéis dos usuários

Os administradores têm acesso a seções especiais do sistema. Nessa seções é possível criar e editar temas, visualizar informações dos voluntários, mudar a categoria de um usuário de voluntário para administrador, e também banir manualmente voluntários (em caso de mau comportamento).

3.6 Comentários finais

Seguindo a ordem das camadas propostas para a arquitetura do projeto, detalhadas neste capítulo, o próximo capítulo apresenta duas etapas do processo de implementação do *framework*: a avaliação do ambiente de desenvolvimento e implementação das funcionalidades em si.

A avaliação do ambiente será realizada utilizando como guia os requisitos não funcionais listados no presente capítulo. Dessa forma, as escolhas realizadas para os componentes de cada camada serão justificadas por esses requisitos.

Na segunda etapa, os requisitos funcionais, juntamente com a modelagem dos dados apresentada no capítulo atual, darão origem às funcionalidades dos componentes das camadas da arquitetura. Assim sendo, as funcionalidades serão detalhadas ao nível de implementação.

Capítulo 4

Implementação

O presente capítulo aborda os detalhes da implementação do projeto como um todo. Primeiramente, é apresentada uma avaliação dos ambientes de desenvolvimento guiada pelos requisitos não-funcionais abordados no capítulo anterior. Em seguida é apresentada a implementação dos módulos do *framework*, baseando-se nos requisitos funcionais.

4.1 Avaliação do Ambiente de Desenvolvimento

Para o desenvolvimento da aplicação, inicialmente foi realizada a escolha do ambiente de desenvolvimento, para isso realizou-se um processo de avaliação das tecnologias disponíveis para a execução do projeto. Como método de avaliação, foram desenvolvidos módulos básicos com funcionalidades que faziam parte do escopo definido pelos requisitos funcionais e, utilizando-se dos requisitos não-funcionais, as tecnologias foram então escolhidas.

4.1.1 Serviços

Como definido pelos requisitos não funcionais do capítulo anterior, a avaliação do ambiente para desenvolvimento do componente da camada de serviço foi limitada às alternativas que possuem bibliotecas de ORM e que comportem o modelo *RESTful*, e ainda que apresentem aspectos como flexibilidade e documentação completa. Assim sendo, a avaliação pode ser organizada em ambientes baseados nas linguagens de programação Python e JavaScript.

Do ponto de vista dos ambientes baseados em Python, foram testados os *frameworks* Django e Pyramid. Ambos os *frameworks* apresentam grande utilização por

desenvolvedores, e portanto oferecem considerável documentação. Outro ponto importante é o suporte a *plugins* de terceiros, que viabilizam a inclusão de novas funcionalidades de maneira simples e rápida. A principal diferença encontrada entre os dois *frameworks* foi a filosofia em que ambos são construídos. Enquanto o Django oferece aos desenvolvedores um menor esforço em termos de customização, fornecendo kits completos (a depender do tipo de aplicação que se deseja desenvolver), o Pyramid apresenta a política contrária, oferecendo o mínimo possível e pacotes individuais que abarcam apenas a funcionalidade desejada. Ambos os ambientes oferecem suporte a duas grandes bibliotecas ORM em Python: *Alchemy* e *Storm*.

Entre os dois *frameworks*, o Pyramid se mostrou uma melhor opção, devido seu perfil mais simples e enxuto, sem a perda de especialização.

Posteriormente, foi estudado o ambiente baseado em JavaScript. Esse ambiente é mais novo que o anterior, e é construído sobre o *NodeJS*, uma plataforma criada sobre o motor de *JavaScript* Chrome V8 desenvolvido pela *Google*. Assim como em Python, em Node os *scripts* escritos em JavaScript são interpretados e executados, porém são marcados pelo comportamento assíncrono inerente ao ambiente em JavaScript. Nos últimos anos esse ambiente tem sido muito utilizado tanto no desenvolvimento de APIs quanto de aplicações *Web*.

Assim como nas alternativas em Python, o ambiente Node permite que os desenvolvedores acrescentem bibliotecas com funcionalidades que podem ser agregadas aos projetos. Uma das vantagens do ambiente Node é o seu repositório oficial, chamado de *NPM*. Os projetos em Node são capazes de armazenar as versões das bibliotecas das quais a aplicação necessita para funcionar. Como o NPM se baseia em repositórios *git*, é possível recuperar as versões exatas dessas dependências. Dessa forma o ambiente Node permite que a aplicação seja facilmente distribuída aos usuários finais sem que esses tenham que lidar com quebras de dependência, reduzindo os problemas de descontinuidade das bibliotecas requeridas. Outra vantagem desse ambiente é a constante atualização: uma vez que o NPM se baseia em *git*, sua comunidade é muito ativa, além de apresentar um mecanismo muito simples para criação de novas bibliotecas. Nesse contexto, foram analisados os *frameworks* Express e Hapi para Node.

Express é um *framework* utilizado em Node para o desenvolvimento tanto de aplicações *Web* quanto de APIs. Assim como Django, ele utiliza o sistema de *middleware*, que consiste em sub-rotinas processadas antes das rotinas dos *endpoints* propriamente ditos. É possível, por exemplo, utilizar *middlewares* para a verificação de *tokens* das permissões de acesso a rotas reservadas do serviço. Outro ponto importante é que, devido à sua ampla utilização, existem inúmeras fontes de documentação sobre o funcionamento da biblioteca.

Devido ao seu grande poder de customização, o Express apresenta o contraponto de agregar grande complexidade ao desenvolvimento de serviços em grande escala, o que dificulta a manutenção e evolução dos mesmos. Esse é um dos pontos em que o Hapi, um outro *framework* para NodeJS, se sobressai. O Hapi apresenta outra filosofia, em que o desenvolvedor deve se focar apenas nas rotinas das rotas e não na configuração de desenvolvimento de *middlewares*. Este *framework* apresenta também uma ampla comunidade desenvolvedora que provê suporte a ele, além de ser utilizado por grandes empresas como Walmart e Netflix, que desenvolvem módulos e contribuem para a evolução do Hapi. Enquanto para Express é frequentemente necessário o desenvolvimento de um *middleware* ou a utilização de bibliotecas de pequenos desenvolvedores, o Hapi apresenta em geral um *plugin* desenvolvido por uma comunidade inteira, com amplo suporte e bem documentado. Mais ainda, existem grandes *plugins* para as mais diversas *features* que normalmente são utilizadas em serviços. Apesar de se tratar de um *framework* de mais alto nível, o mesmo não tem sua performance comprometida.

Dentre os *frameworks* para NodeJS, tendo em vista a complexidade do desenvolvimento da aplicação, o *Hapi* foi o que se apresentou mais adequado ao desenvolvimento da aplicação, dadas sua flexibilidade, documentação e simplicidade.

Dentre as bibliotecas ORM disponíveis para Node, as duas analisadas para utilização foram a *ORM2* e *Sequelize*. A primeira apresentava suporte aos SGBDs MariaDB, MySQL, PostgreSQL, SQLite e MongoDB, já a segunda a apenas a SGBDs relacionais, não fornecendo, portanto, uma interface programável para MongoDB. Após alguns testes com ambas as bibliotecas, ficou claro que elas não encapsulavam suporte a consultas espaciais e, além disso, a *ORM2* não garantia que as chaves estrangeiras fossem criadas em alguns casos. A solução para ambos os problemas foi a utilização da biblioteca *Sequelize* e a customização da mesma para o suporte à recuperação de dados espaciais. Essa contribuição inclusive foi oferecida aos desenvolvedores e utilizadores da biblioteca, que iniciaram um processo de inclusão dessa nova *feature* que, por sua vez, já está presente nas versões atuais da biblioteca ¹.

Devido ao sistema de controle de versão de bibliotecas, e a fim de manter a consistência entre as tecnologias usadas tanto no *frontend* quanto no *backend*, o ambiente baseado em NodeJS foi escolhido para o desenvolvimento da API, e o *framework* utilizado foi o *Hapi*.

¹Contribuição da autora ao Sequelize - <https://github.com/sequelize/sequelize/pull/3641>

4.1.2 Aplicação

Partindo dos requisitos não funcionais, o ambiente da camada de aplicação se restringe apenas à plataforma *Web*, mais especificamente em recursos que permitam o desenvolvimento de aplicações SPA. Sendo assim, foram analisados dois arcabouços em JavaScript: AngularJS e ReactJS. Ambos são largamente documentados e desenvolvidos em conjunto pela comunidade *open source* e, respectivamente, pelas empresas Google e Facebook.

O AngularJS se baseia no modelo MVC, onde o *model* pode ser representado pelos *services* e *factories* que encapsulam o processo de requisição dos dados utilizados pela aplicação. Como *view* ele utiliza um sistema de *templates*, que podem ser *strings* embutidos nos módulos desenvolvidos ou HTML. Além disso, a cada módulo pode ser atribuído um *controller* responsável pela estruturação dos dados requisitados pelos *services* e *factories* para posterior renderização na *view* correspondente.

A biblioteca ReactJS vem sendo desenvolvida pelo Facebook nos últimos dois anos e vem ganhando um grande volume de adeptos. Ao contrário do AngularJS, o ReactJS não apresenta uma estrutura de *framework*, pois esta biblioteca foi criada sobre o ponto de vista de componentes de interface. Sendo assim, um SPA pode ser engendrado através da junção de componentes. Um dos pontos-chaves da biblioteca é a utilização de DOMs virtuais, ou seja, que são criados artificialmente através de JavaScript. A vantagem do DOM virtual é o total controle sobre os componentes da página, o que resulta em uma renderização mais eficiente. Em geral, nas aplicações de SPA, quando um componente é atualizado devido à mudança de seu conteúdo (como a notificação de novo *e-mail* ou uma nova mensagem de chat) é necessária a re-renderização de toda a página. Como os componentes de React são todos mapeados, nessa mesma situação apenas os componentes afetados pela mudança de conteúdo serão renderizados novamente, otimizando assim a resposta da aplicação ao usuário. Um dos problemas dessa biblioteca é que ela é focada apenas em componentes, sendo necessária a utilização de outras bibliotecas para realização de tarefas simples, como o roteamento das *views* dentro da aplicação. Outro fato peculiar da biblioteca é o de que os componentes em React são mapeados utilizando a árvore de aninhamento deles, e a propagação das alterações dos componentes se dá a partir dos nós pais até os filhos, o que torna impossível que nós irmãos gerem ações que alterem uns aos outros, como no caso de botões em um formulário. Para isso, foi lançada também pelo Facebook uma arquitetura chamada Flux, que apresenta um sistema de mensagens que podem ser disparadas por quaisquer componentes e são capazes de afetar todos os componentes que escutam eventos específicos e que são entregues pelos mesmos. Por fim, apesar de sua

larga utilização, a biblioteca, até esse momento, está em versão de desenvolvimento. Ou seja, ainda apresenta instabilidade em sua API.

Em uma primeira abordagem, a biblioteca React teve seu uso considerado devido ao seu grau de otimização. No entanto, devido à sua complexidade e à constante mudança, decidiu-se utilizar a biblioteca Angular, que garantiria uma maior estabilidade e redução do tempo de desenvolvimento.

Para a renderização dos mapas, foi inicialmente utilizada a biblioteca Openlayers, que por muitos anos foi uma alternativa para a criação de visualizações de mapas sem estar atrelada a um serviço de *web mapping* como o Google Maps. Recentemente surgiu uma alternativa ao OpenLayers chamada Leaflet, que possui uma documentação também completa e funções de mais alto nível que o Openlayers. Uma das vantagens do Leaflet está em sua abertura ao acoplamento de *plugins*. Essa abertura permite o desenvolvimento independente de novas funcionalidade e, como resultado, existem inúmeros módulos que aumentam as possibilidades na criação de visualizações. O Openlayers, por outro lado, apresenta uma API que permite a manipulação dos componentes do mapa em nível mais baixo, o que aumenta seu poder de customização. Devido à simplicidade no desenvolvimento, foi adotada posteriormente a biblioteca Leaflet para a renderização dos mapas.

Tabela 4.1: Alternativas analisadas *versus* utilizadas

Camada	Tipo de biblioteca	Nome da biblioteca	Utilizada
Serviços	Framework para Python	Pyramid	
		Django	
	Framework para NodeJS	ExpressJS	
		HapiJS	x
	ORM para NodeJS	ORM2	
		Sequelize	x
Aplicação	Framework para SPA	AngularJS	x
		ReactJS	
	Renderização de mapas	Leaflet	x
		OpenLayers	
	Visualização de dados	d3	
		c3	x

Por fim, em resposta ao requisito funcional que inclui a exibição de estatísticas dos voluntários, foram avaliadas duas bibliotecas para a renderização de gráficos: *d3* e *c3*. O *d3* é uma biblioteca para renderização de visualizações que ganhou destaque por seu alto nível de customização. Devido à sua popularidade, ela apresenta uma extensa documentação para referência. Já o *c3* é na verdade um encapsulamento do *d3*, que

simplifica a interface de programação para a geração de gráficos, porém apresenta uma limitação de tipos de gráficos que podem ser desenvolvidos. Nesse ponto, foi adotado o *c3*, que para os propósitos da aplicação atende bem à função de renderização de gráficos, uma vez que inicialmente não estão planejadas visualizações mais complexas. Ademais, essa escolha não impede uma futura transição para visualizações mais complexas com o *d3*.

A Tabela 4.1 apresenta um resumo das alternativas adotadas para implementação do *framework*.

4.2 Implementação das camadas

Como abordado anteriormente, a aplicação pode ser dividida em três camadas: persistência, serviços e aplicação. Seguindo essa divisão, a presente seção tem como objetivo apresentar a implementação dos componentes de cada camada, utilizando o ambiente escolhido na seção anterior, bem como os requisitos funcionais abordados no Capítulo 3.

4.2.1 Camada de Persistência

Como apresentado na seção anterior, a comunicação entre a camada de persistência e de serviços é realizada através a ORM *Sequelize*. Essa biblioteca utiliza três tipos diferentes de arquivos: *migrations*, *seeds* e *models*, além das informações para conexão com o SGBD. Os dois primeiros arquivos são utilizados por uma ferramenta do *Sequelize* para criar tabelas e popular o banco, respectivamente. Dessa forma, o primeiro arquivo é preenchido com descrição das classes, e o segundo com os dados a inserir. Esses dois arquivos, juntamente com a ferramenta de inicialização, são utilizados para o mapeamento do esquema conceitual para o físico. O terceiro tipo de arquivo é utilizado pela ORM para identificar os atributos presentes em cada classe, permitindo assim que a ORM gere a devida consulta SQL. Esse arquivo será posteriormente utilizado pela camada de serviços que será abordada mais adiante (Seção 4.2.2).

O processo de implementação da camada de persistência consiste, portanto, na criação dos arquivos de *migration*, seguindo o esquema de dados abordado no capítulo anterior, assim como os *seeds* contendo valores de referência para o funcionamento correto da camada de serviços.

Uma das classes que contém valores de referência é a *Groups*, onde são detalhados os limites de pontuação pré-definidos mínimos e máximos para que o voluntário pertença a um determinado grupo, descritos na Tabela 4.2. A partir dessas informa-

ções, torna-se possível para o administrador identificar esses grupos no momento da customização dos valores.

Tabela 4.2: Valores de Referência para Grupo de Especialização

Nome	Min	Max
Malicious Collaborator	-	-1
Basic Collaborator	0	99
Legal Collaborator	100	249
Master Collaborator	250	499
Experienced Collaborator	500	799
Special Collaborator	800	2147483647

Seguindo a mesma lógica dos grupos de *expertise*, podemos citar também as ações dos voluntários, que da mesma forma possuem valores de referência que guiam as configurações específicas para o sistema de pontuação dos temas, como podem ser vistos na Tabela 4.3.

Tabela 4.3: Valores de Referência para Recompensas

Nome	Pontos
Collaboration performed	10
Comment performed	5
Evaluated collaboration	1
Extra files sent	2
Performed a Wiki review in a VGI	1
Had a collaboration evaluated	User own a weighted arithmetic mean of ratings
Made a false collaboration	-15
Made a malicious collaboration	-50
Made a criminal collaboration	User baned
Promoted by administrators	Add score of position

4.2.2 Camada de Serviços

Como detalhado anteriormente, a camada de serviços é composta por uma API que foi desenvolvida utilizando o *framework Hapi* e a ORM *Sequelize*. Esse serviço segue o padrão *RESTful* de Fielding [2000], que é *stateless*, ou seja, não mantém o estado dos componentes das camadas superior e inferior, que se comunicam através de mensagens completas (que contêm informação suficiente para que as requisições sejam processadas). Nesta seção é detalhada a implementação desse serviço, descrevendo a segurança do serviço, os *endpoints* implementados e os mecanismos para gratificação dos voluntários.

4.2.2.1 Segurança

Partindo do primeiro requisito funcional, que determina se voluntários precisam ou não ser identificados pelo sistema, foi desenvolvido um arcabouço de segurança que tem tanto a função de proteger os dados da aplicação de acessos indevidos, quanto identificar o voluntário. Dessa forma, na aplicação existem dois tipos de identificação, aquela em que verifica se o usuário está autorizado a acessar a aplicação, e aquela que verifica o papel do usuário e determina se ele tem permissão para acessar algum recurso.

Para a primeira identificação, o serviço conta com um sistema de autenticação de requisições baseado em *tokens*. Assim, as requisições aos recursos exclusivos a voluntários autenticados devem possuir no cabeçalho de suas mensagens do protocolo HTTP o atributo *Authentication* contendo o devido *token*. Como escolha de implementação, foi utilizado um *token* do tipo *JSON Web Tokens (JWT)*, que, como o próprio nome sugere, cria um *hash* a partir de um objeto JavaScript, que por sua vez contém informações como a identificação do usuário que requisitou o serviço, bem como seu tempo de expiração, que pode ser configurado. Assim sendo, para obter um *token*, o componente da camada de aplicação deve requisitá-lo ao serviço, através de um *endpoint* público. Para isso, o serviço demanda que a aplicação envie o *e-mail* do usuário e sua senha, que são verificados, e caso os mesmos coincidam com os presentes no banco de dados, o serviço retornará o respectivo *token*. A identificação do usuário se dá quando o mesmo realiza uma requisição a um *endpoint* protegido, onde o serviço realiza a decodificação do *hash*, obtendo assim a identificação do usuário que requisitou o recurso.

Para fazer a diferenciação dos papéis dos voluntários identificados, ou seja, que são capazes de acessar informações protegidas pela autenticação, o serviço utiliza uma lista de controle de acesso (*Access Control List (ACL)*). Essa lista basicamente representa uma *checklist* para cada papel, que identifica se os *endpoints* identificados com uma *tag* específica são acessíveis ou não pelo usuário. Dessa forma, uma vez que o usuário identificado tenha seu *token* decodificado, é possível identificar e verificar se seu papel permite que tenha acesso a um recurso específico.

Durante a verificação da validade do *token*, as informações do usuário que o expediu são sempre verificadas, assim como o seu nível de acesso garantido pelo papel. Como a constante verificação dessa informação no banco de dados pode levar a uma diminuição no tempo de resposta às requisições, utilizou-se nesse ponto um gerenciador *NoSql* em memória, baseado em chave-valor, chamado *Redis*. Esse banco de dados foi escolhido por sua interface de programação para Node ser bem documentada. Dessa forma, o tempo de resposta das requisições pode ser otimizado, uma vez que o tempo

de acesso direto (por chave) aos dados em memória é consideravelmente menor que o tempo de resposta às consultas ao banco de dados, o que também contribui para redução da carga de acesso a ele.

Tendo em vista esses dois métodos de segurança e identificação, o HapiJS conta com dois *plugins* para realizar essas validações. Em ambos os casos é necessária a criação de uma função de validação do usuário que requisitou a rota. Na autenticação dos *tokens*, basta definir uma função que realiza a verificação das informações presentes nele, como a preexistência do usuário na aplicação ou se o *token* já expirou. Já no caso do ACL é necessário definir um objeto contendo as permissões de acesso às *tags* citadas. Além das funções de validação, os *plugins* permitem que cada rota seja identificada com o tipo de validação necessária para seu acesso, como a autenticação do *token* e a *tag* do ACL.

Para o cadastro do usuário na aplicação e expedição dos *tokens* mediante a identificação do mesmo, foram criados dois *endpoints* (Tabela 4.4).

Tabela 4.4: Rotas de registro e autorização de acesso dos voluntários

<i>Rota</i>	Método	Ação
/users/register	POST	registra um novo usuário
/login	POST	realiza o processo de autorização (expedição de <i>token</i>)

4.2.2.2 Temas e configurações do sistema de reputação

Do ponto de vista dos temas, os requisitos funcionais definem que o administrador deve ser capaz de criar novos temas, especificando as informações que serão coletadas bem como a representação espacial e simbologia da visualização. Para isso foi desenvolvido um conjunto de rotas (Tabela 4.5) sobre o tema que podem ser utilizadas para a criação, listagem e visualização dos temas. Aqui foram excluídas as ações de atualização e exclusão, uma vez que os dois métodos podem gerar inconsistências nos dados coletados, devido à modificação nas estruturas de armazenamento.

Adicionalmente, foi desenvolvida uma rota de verificação dos temas em uso, que pode ser utilizada para verificar na aplicação se o nome de referência da tabela que será criada para as contribuições do tema escolhido pelo administrador já foi utilizado, mantendo a consistência de unicidade de nomes das tabelas no banco. Essa rota pode ser utilizada apenas por administradores.

Dentre as rotas implementadas, a de listagem e visualização não apresentam muita complexidade, sendo representado apenas o encapsulamento das consultas na tabela de temas, na qual a estrutura dessa entidade foi descrita no capítulo anterior

(Figura 3.3). O acesso a essas funções é livre para voluntários identificados, independentemente de seus papéis. No caso específico da listagem, o acesso é livre a voluntários anônimos, porém apenas temas abertos a esse público são retornados.

Tabela 4.5: Rotas de manipulação dos temas

<i>Endpoint</i>	Método	Ação
/themes/id	GET	Mostra as informações de um tema
/themes	GET	lista todos os temas
/themes	POST	Cria um novo tema
/themes/namesinuse	POST	Verifica se já existe um tema com essa referência para tabela

A rota de criação é mais complexa, uma vez que necessita gerar uma nova tabela utilizando os mecanismos fornecidos pela ORM, seguindo o esquema definido pelo administrador. Além disso, é necessário criar entradas nas tabelas de pontuação por ações nos temas (*theme_rewards*) e na pontuação para evolução da *expertise* nos temas (*theme_promotions*). Dessa forma, quando o voluntário identificado realizar uma ação referente ao tema, ele receberá a respectiva recompensa predefinida para o tema. A criação dessas entradas segue dois requisitos funcionais definidos no capítulo anterior, que estabelecem que o administrador deve ser capaz de configurar essas duas variáveis. Além disso, o voluntário poderá progredir nas categorias de *expertise*, seguindo as diretrizes de pontuação fixadas. Esse *endpoint* é de acesso exclusivo dos administradores.

Como na criação de um novo tema, o administrador deverá informar a configuração das pontuações recebidas pelos voluntários, e definir as regras de progressão nos níveis de *expertise*. Foram definidas duas rotas (Tabela 4.6) para referência de quais ações devem ser recompensadas e quais níveis de *expertise* estão disponíveis aos voluntários, assim como a pontuação para seus respectivos parâmetros. Assim sendo, os *endpoints* gerados são listados na Tabela 4.6, e ambos são de acesso restrito aos voluntários identificados.

Tabela 4.6: Rotas para listagem de ações e grupos de *expertise*

<i>Endpoint</i>	Método	Ação
/actions	GET	lista todos as ações padrão registradas
/groups	GET	lista todos os grupos padrão registrados

4.2.2.3 Colaborações dos voluntários e recompensas

Para atender ao requisito de criação de novas contribuições, foi desenvolvido um conjunto de rotas para manipulação das contribuições (Tabela 4.7), permitindo assim não

só sua criação quanto edição, listagem e exclusão. Para simplificar a estrutura das rotas, os temas nesse conjunto de *endpoints* são referenciados através de um identificador único na tabela de temas. Dessa forma, para cada operação, seja ela de listagem ou até mesmo de criação, é realizada uma busca pelo tema em questão e, assim, esse identificador pode ser traduzido no sufixo da tabela de contribuição do tema. Esse método se assemelha ao utilizado pelas tabelas de catálogo de vários SGBDs. Para tornar esse processo atômico foi utilizado o suporte do *Sequelize a transactions*. Apenas o método de exclusão de uso restrito aos administradores, portanto, os demais métodos são de livre acesso a todos os voluntários. Logo, estes podem criar, editar e visualizar as contribuições.

Tabela 4.7: Rotas para manipular as contribuições

<i>Endpoint</i>	Método	Ação
/contributions/th_id/con_id	GET	Mostra as informações de uma contribuição
/contributions/th_id/con_id	PUT	Atualiza as informações de uma contribuição
/contributions	POST	Cria uma nova contribuição
/contributions/th_id/con_id	DELETE	Remove uma contribuição da aplicação

Para atender aos requisitos de criação e listagem de comentários, arquivos e *wikis* e avaliação de contribuições, foi desenvolvido um conjunto de rotas destinadas a essas informações complementares (Tabela 4.8). Esses quatro itens apresentam um funcionamento similar e para tanto possuem escopo de *endpoints* bem parecidos. Assim como o conjunto de rotas das contribuições, essas rotas apresentam apenas o método reservado aos administradores, sendo os demais de livre acesso aos voluntários. Para esses métodos também foi utilizado o mecanismo de tradução do *id* do tema em sua referência, conseqüentemente cada instância dessas informações complementares pode referenciar diretamente os objetos da classe de contribuição do tema.

A fim de atender aos requisitos que estipulam a recompensa dos voluntários ao criarem esses tipos de informação, o serviço utiliza um suporte oferecido pela biblioteca *Sequelize*, onde é possível atribuir *hooks* (que apresentam funcionalidade similar aos *triggers* dos SGBDs), que disparam funções após eventos de inserção, atualização e exclusão desses dados. Assim sendo, em cada uma dessas ações, o serviço verifica respectiva pontuação para o tema e a adiciona aos pontos totais do voluntário identificado nesse tema. Os *hooks* verificam ainda se o voluntário identificado conseguiu pontos suficientes para evoluir de nível de *expertise*, realizando portanto o registro na tabela *user_groups*, caso essa condição seja atingida.

Tabela 4.8: Rotas para manipular comentários, *wikis*, anexos e avaliação das contribuições

<i>Endpoint</i>	Método	Ação
/comments/th_id/con_id	GET	Lista todos os comentários de uma contribuição
/comments/comm_id	PUT	Atualiza um comentário de uma contribuição
/comments	POST	Cria um novo comentário sobre uma contribuição
/comments/comm_id	DELETE	Remove um comentário de uma contribuição
/wiki/th_id/con_id	GET	Mostra as wikis de uma contribuição
/wiki/wiki_id	PUT	Atualiza a wiki de uma contribuição
/wiki	POST	Cria uma nova wiki sobre uma contribuição
/wiki/wiki_id	DELETE	Remove a wiki de uma contribuição
/files/th_id/con_id	GET	Lista todos os arquivos de uma contribuição
/files/file_id	PUT	Atualiza um arquivo de uma contribuição
/files	POST	Cria um novo arquivo sobre uma contribuição
/files/file_id	DELETE	Remove um arquivo de uma contribuição
/rating/th_id/con_id	GET	Retorna a avaliação de uma contribuição
/rating/mine/th_id/con_id	GET	Mostra a avaliação do voluntário solicitante sobre a contribuição
/rating/th_id	PUT	Atualiza a avaliação do voluntário de uma contribuição
/rating	POST	Cria a avaliação do voluntário sobre uma contribuição
/rating/th_id	DELETE	Remove a avaliação do voluntário de uma contribuição

4.2.2.4 Administração de Usuários

Outro requisito funcional listado define que os voluntários devem ter acesso às informações sobre suas contribuições. Para isso foi desenvolvido um conjunto de rotas de usuário, além de rotas adicionais que permitem que os voluntários identificados editem suas informações do perfil como nome, email e senha. Adicionalmente foram introduzidas outras rotas às quais apenas o administrador tem acesso, e permitem que esse liste todos os usuários da aplicação e edite-os, podendo modificar o *status* de adminis-

trador para voluntário, assim como banir voluntários manualmente se necessário. Os *endpoints* e suas funções específicas estão descritos na Tabela (4.9). Por medida de segurança, voluntários anônimos não têm acesso a nenhuma dessas rotas.

Tabela 4.9: Rotas de administração de usuários

<i>Endpoint</i>	Método	Ação
/users	GET	lista todos os usuários
/users/id	GET	Mostra as informações de um usuário
/users/me	GET	Mostra as informações do usuário da requisição
/users/id	PUT	Edita as informações de um usuário
/users/me	PUT	Edita as informações do usuário da requisição
/users	POST	Cria um novo usuário
/users/id	DELETE	Remove usuário da aplicação

4.2.2.5 *Web socket*

Para que a aplicação fosse capaz de atualizar dinamicamente no momento em que os voluntários distintos inserissem suas contribuições, foi criado um outro serviço junto à API, utilizando os recursos do *framework* Hapi em conjunto com a biblioteca de *web socket*. Basicamente, essa biblioteca é capaz de escutar e enviar eventos à aplicação. Dessa forma, quando o serviço identifica uma nova contribuição, uma mensagem é enviada a todos os clientes com as informações da nova contribuição, que posteriormente pode ser incluída na visualização do mapa das aplicações que estão escutando esse tipo de evento.

4.2.3 Camada de Aplicação

Como foi definido anteriormente pelos requisitos não funcionais, foi desenvolvida uma aplicação do tipo SPA, utilizando o *framework* AngularJS. Assim como explanado no capítulo anterior, os componentes dessa camada utilizam a camada de serviço para persistir os dados criados pelos usuários na camada de persistência. A modularização da aplicação foi dividida conforme as seções da aplicação, compondo o conjunto de módulos: *Contribution, Home, Login, Register, Themes, Users*. Além disso, seguindo o primeiro requisito listado no capítulo anterior, e os serviços desenvolvidos no capítulo atual, podemos agrupar essas seções com relação ao nível de acesso requerido para utilizá-las.

4.2.3.1 Seções Públicas

Dentre as *views* públicas estão *Login*, *Register*, *Home*. Como citado anteriormente, essas seções são derivadas do primeiro requisito funcional, que determina que tanto voluntários identificados quanto anônimos podem contribuir nos temas que assim o permitam.

A seções de *Login* e *Register*, como seus nomes já sugerem, foram criadas com a finalidade de permitir que voluntários novos se cadastrassem e pudessem utilizar esse registro tanto para identificar suas ações quanto para receber suas recompensas e progredir seu nível de *expertise*. Ambas são compostas por formulários de envio de informação e disparam requisições ao serviço. No caso da *view* de *Login*, os dados do usuário são enviados ao serviço. O serviço responde à requisição com uma mensagem contendo o *token* para autenticação das requisições posteriores em nome desse usuário pela aplicação. Sendo assim, a aplicação armazena esse *token* e concede ao usuário acesso às seções privadas da aplicação.

A seção *Home* é a única que apresenta um comportamento híbrido, se comportando como pública quando a aplicação apresenta temas que permitem contribuições anônimas. Além disso, quando a seção é visualizada por um voluntário anônimo, o menu superior apresenta apenas a opção de *login* e registro. Do ponto de vista do voluntário autenticado, a *view* apresenta as opções para navegar ao perfil do mesmo, retornar a seção *Home* e realizar *logout*. No caso dos administradores são incluídas ainda opções de acesso à seção de administração de usuários, administração de temas e criar novo tema.

Seguindo os requisitos funcionais para visualização das contribuições, esta seção apresenta a distribuição geográfica das contribuições utilizando a representação icônica/geográfica e cores escolhidas pelo administrador na criação do tema. Para isso o mapa de contribuições, que obtém suas informações através de *web socket*, como foi abordado anteriormente nesse capítulo, dessa forma as contribuições podem ser exibidas no momento em que são adicionadas para os demais voluntários.

Outro requisito atendido através desta seção é a visualização das informações da contribuição, que é realizada através de uma barra lateral que se expande ao clique do usuário sobre a classe de contribuição. Através dessa barra lateral também é possível realizar comentários e avaliar a contribuição, como previsto no conjunto de requisitos funcionais.

4.2.3.2 Seções Privadas

Dentre as seções privadas, a seção *Profile* é de livre acesso para tanto o voluntário identificado, quanto para o administrador. Essa seção permite que o voluntário visualize suas informações, assim como as estatísticas sobre sua contribuição, atendendo ao requisito que estabelece o *feedback* das ações do voluntário. Adicionalmente, a seção permite que o voluntário altere seus dados pessoais como nome, *e-mail* e senha.

As demais seções são de uso exclusivo do administrador da aplicação. A seção de voluntários permite que o administrador gerencie os usuários da aplicação, podendo lista-los ou visualizar suas informações. Além disso é possível modificar os voluntários identificados para administradores e vice-versa, ou, no outro extremo, banir os voluntários maliciosos manualmente.

A seção de *Themes* permite o gerenciamento dos temas por parte do administrador. Dentre as funcionalidades presentes nessa seção estão a listagem dos temas, assim como a visualização de suas estatísticas. Nessa seção existe também a *view* de criação de novos temas, que atende ao requisito de criação de temas. Nessa *view* é possível definir os atributos de visualização, as informações complementares da contribuição, a presença ou não de colaborações adicionais como comentários, anexos e wikis, assim como as configurações de pontuação e evolução nos níveis de *expertise*.

Dado o conjunto de funções que foram implementadas, o *framework* está completo, e o próximo capítulo apresenta seu funcionamento através de um estudo de caso sobre mobilidade urbana.

Capítulo 5

Estudo de caso

O presente capítulo aborda um estudo de caso sobre mobilidade urbana para avaliar e exemplificar a utilização da ferramenta.

Um dos problemas encontrados na mobilidade urbana é a superlotação dos ônibus. A fim de avaliar as funcionalidades implementadas pelo *framework*¹, o presente capítulo apresenta como tema um de contribuição voluntária pode ser desenvolvido, utilizando o exemplo da lotação dos ônibus. Nesse tema o usuário pode indicar a lotação aproximada de um veículo em determinado ponto de seu itinerário.

Como alvo de ação VGI, este tema apresenta diversas características típicas. Em primeiro lugar, propõe a coleta de informação distribuída espacial e temporalmente, por cidadãos que estejam vivenciando o problema da superlotação e queiram contribuir conscientemente para seu estudo. Além disso, trata-se de dado que seria de difícil obtenção junto às concessionárias de transportes urbanos, por expor deficiências do sistema. A limitação de cobertura, típica de projetos de VGI, não atrapalha o esforço de coleta desses dados, uma vez que as informações coletadas para determinadas linhas pode ser usado independentemente da existência de dados ou contribuintes em outras. Finalmente, o contraste entre dados fornecidos por múltiplos usuários pode ajudar a filtrar contribuições errôneas ou maliciosas. Dados fornecidos diretamente pelos usuários do sistema de transporte público podem ser um elemento fundamental na análise de múltiplas fontes heterogêneas de dados espaciais sobre o tema de mobilidade urbana, uma das vertentes mais interessantes da pesquisa em Computação Urbana [Foth et al., 2011; Zheng et al., 2014]. Um desafio de pesquisa em Computação Urbana nesse caso seria a integração de dados quantitativos e qualitativos fornecidos pelos cidadãos a dados oficiais (fornecidos pela concessionária de transporte público local), dados de sensores ambientais urbanos (laços no asfalto para contagem de veículos, dados me-

¹ThemeRise - <http://aqui.io/themerise>

teorológicos, câmeras de controle de trânsito), dados de posicionamento de veículos (trajetórias GPS em ônibus), e dados de *crowdsourcing* e *crowdsensing* obtidos em redes sociais como Waze e Twitter.

Como foi citado anteriormente, apenas os administradores são capazes de criar novos temas dentro da aplicação. Para isso, o mesmo deve utilizar a opção presente no menu superior, intitulado *Create Theme*, como mostra a Figura 5.1.

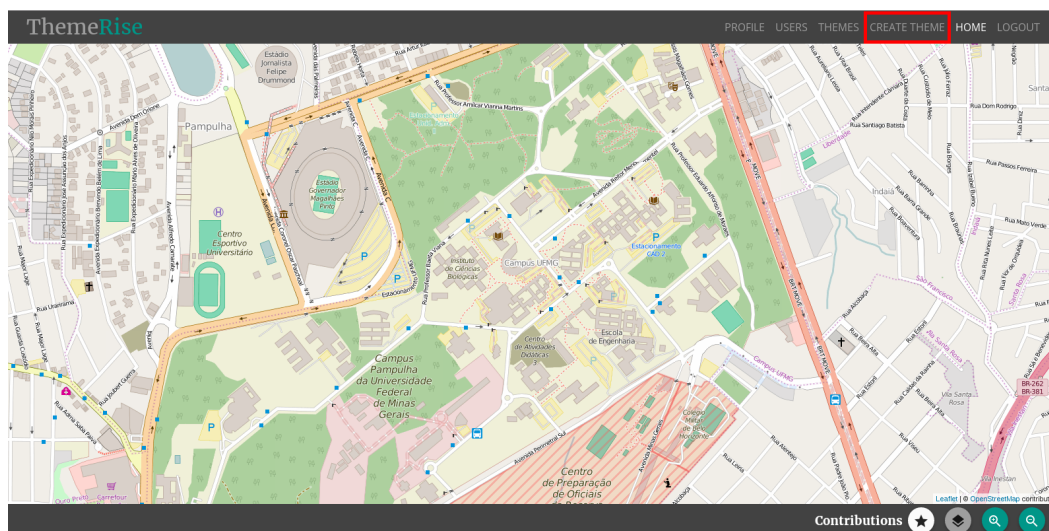


Figura 5.1: Tela inicial sobre a visão do administrador

Após selecionar essa opção, é apresentada ao usuário a tela de criação de temas. Nessa tela o administrador fornece informações básicas como nome do tema, nome de referência para a tabela, geometria, símbolo, cor e presença de comentários e *wikis* (Figura 5.2).

Theme Information

Name	Table reference
Lotação de ônibus	lotacao_onibus

.. point EPSG:4326 directions_bus Choose a color #141cd1

Description

Mapeamento da lotação dos ônibus municipais. Allow anonymous contribution

Stop showing the contribution Expires with Time unit

Theme Extras

Comments Wikis Files

Figura 5.2: Informações básicas do tema

Para o exemplo do estudo de caso, o tema foi intitulado “Lotação de ônibus”. O nome definido para a criação da tabela foi “lotacao_onibus”, com geometria sendo “ponto”. Foram também escolhidos um ícone com as feições de um ônibus, e a cor azul para representar as contribuições na visualização do mapa. Além do preenchimento da descrição, foi definido que esse tema permite que os usuários comentem e enviem anexos para a contribuição. Adicionalmente o tema deve permitir que usuários anônimos realizem contribuições, permitindo assim que o tema atinja um maior alcance.

Em seguida, o administrador deve informar quais são os atributos descritivos do tema. Como pode ser visto na Figura 5.3, são preenchidas as informações sobre o esquema físico da entidade que será criada no SGBD, como nome, tipo e tamanho, no caso de tipos textuais. O administrador deve também informar as opções de renderização do campo, como a legenda no formulário, a *tag* HTML que será utilizada pela contribuição, bem como o tamanho.

Theme Attributes					
Label	Column name	Html Type	Html Width	Schema Type	Schema Length
Linha de ônibus					
Attribute reference: linha,					
Schema type: string,					
Schema length: 255,					
Html type: Text,					
Html width: 40					
Número aproximado de pessoas					
Attribute reference: numero_pessoas,					
Schema type: integer,					
Schema length: -,					
Html type: Number,					
Html width: 40					
Observado em					
Attribute reference: data_observacao,					
Schema type: date,					
Schema length: -,					
Html type: DatePicker,					
Html width: 50					
Descrição					
Attribute reference: description,					
Schema type: text,					
Schema length: -,					
Html type: Textarea,					
Html width: 50					

Figura 5.3: Atributos descritivos do tema

Para este estudo de caso foram definidos dois atributos descritivos. O primeiro, que recebeu o nome “Linha de ônibus”, permite aos usuários informarem qual a linha de ônibus em que foi realizada a observação, sendo portanto um campo de texto que na tabela será representado por um “*varchar*” (mapeado como string pela ORM). Em seguida foi criado o campo para que o mesmo informe o “*Número aproximado de passageiros*”, sendo aquele do tipo inteiro. Deverá, também, ser coletado por um formulário do tipo “*number*”. Foi ainda incluído um campo para que o usuário informe explicita-

mente quando a observação do fato foi realizado, para esse campo foi utilizado o tipo de entrada “*datepicker*” e, sendo representado como o tipo “*date*” no banco de dados. O quarto atributo foi definido para que os usuários possam inserir uma “*descrição*” adicional à contribuição. Assim sendo que esse atributo apresenta o tipo “*text*”, e é representado como um caixa de texto (“*textarea*”) no formulário.

Em seguida, o administrador pode optar em configurar manualmente a pontuação que cada colaborador deverá receber por determinada ação em um tema (Figura 5.4). Para maior comodidade do usuário, esses valores são previamente preenchidos utilizando os dados incluídos durante a fase de povoamento do banco de dados na tabela *actions*, como foi descrito pela Tabela 4.3 da Seção 4.2.1 do capítulo anterior. Para o exemplo da lotação dos ônibus, foram utilizados os valores padrão.

Collaboration performed	10
Comment performed	5
Evaluated collaboration	1
Extra files sent	2
Performed a Wiki review in a VGI	1
Made a false collaboration	-15
Made a malicious collaboration	-50

Figura 5.4: Configuração da pontuação por ações.

Da mesma forma, o administrador poderá configurar a pontuação necessária para que o usuário modifique o *status* de seu nível de *expertise*. Nesse sentido, deverá informar tanto os pontos mínimos para atingir o nível quanto os pontos máximos (Figura 5.5). Assim como na pontuação por ações, o formulário em questão também apresenta valores de base, que reduzem o esforço do usuário para seu preenchimento. Esses dados também foram introduzidos no banco de dados durante a fase de povoamento, e são descritos na Tabela 4.2 do capítulo anterior. Assim como foi realizado na configuração dos pontos por ação, para esse exemplo foram utilizados a pontuação para progressão do nível de *expertise* como sendo os valores previamente cadastrados na ferramenta.

Malicious Collaborator		-1
Basic Collaborator	0	99
Legal Collaborator	100	249
Master Collaborator	250	499
Experienced Collaborator	500	799
Special Collaborator	800	2147483647

[HIDE GROUPS PROMOTION SETTINGS](#)
[HIDE REPUTATION SETTINGS](#)
[CANCEL](#)
[CREATE](#)

Figura 5.5: Configuração das categorias de *expertise*

Depois de concluídos os passos de criação, é apresentado o administrador a listagem de temas (Figura 5.6). No estado atual o *framework* apresenta dois temas registrados. O primeiro é um tema de exemplo, que é criado no momento em que o banco de dados é povoado, o segundo representa o novo tema criado para esse estudo de caso. Dentre as informações apresentadas na listagem estão o nome do tema, breve descrição, seu ícone e o tipo geográfico.

ThemeRise		PROFILE	USERS	THEMES	CREATE THEME	HOME	LOGOUT
Example	Information about points of interest in the city	🏠	★	DETAILS			
Lotação de ônibus	Mapamento da lotação de ônibus municipais	🏠	★	DETAILS			

Figura 5.6: Listagem dos temas

As informações completas sobre o tema podem ser visualizadas ao clicar na opção *detalhes* (Figura 5.7). No caso do novo tema, são as mesmas citadas anteriormente no processo de criação do tema.

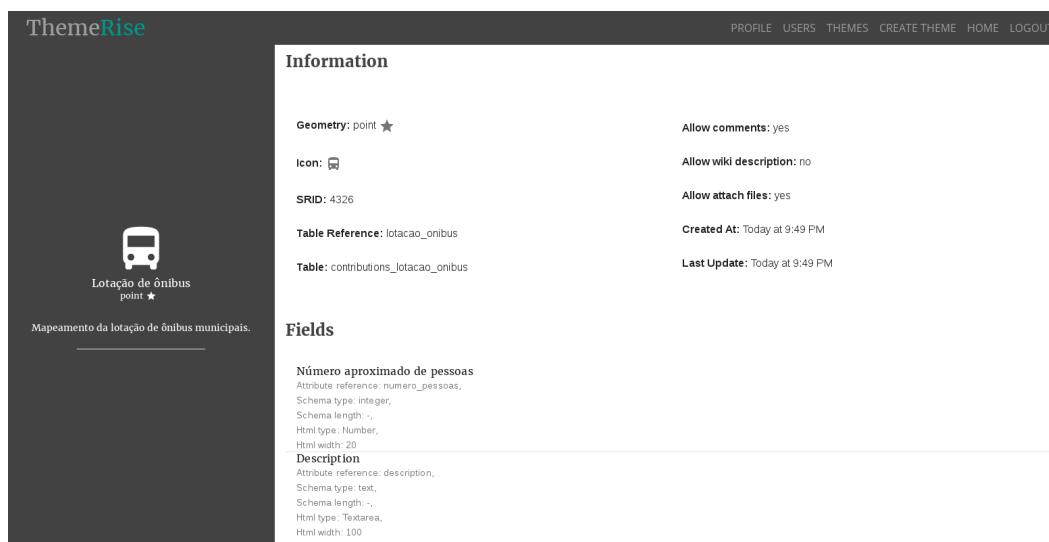


Figura 5.7: Detalhes do tema

O tema torna-se disponível para realização de contribuições por usuários identificados, administradores e, caso seja especificado pelo administrador no momento da criação do tema, usuários anônimos.

Para realizar uma nova contribuição, o usuário deve selecionar a opção contribuição nos comandos situados na região direita inferior da tela. Esses controles estão separados por tipo geográfico e, ao serem acionados, uma barra lateral se expande, exibindo os temas disponíveis para colaboração, como pode ser visto na Figura 5.8.

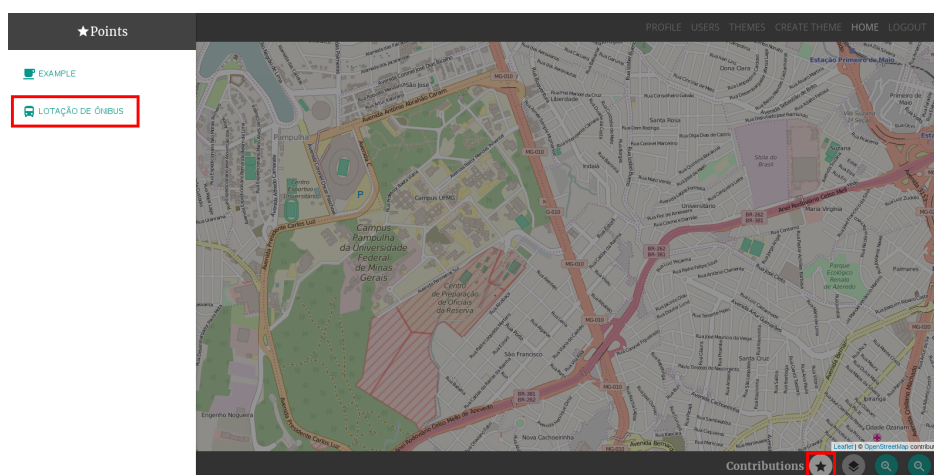


Figura 5.8: Controles de criação de uma nova contribuição

Ao selecionar o tema desejado, o cursor do *mouse* se modifica, indicando o ícone selecionado para o tema no caso de temas com geometria de ponto, ou um quadrado para linhas e polígonos. No caso do exemplo apresentado, à frente do cursor é apre-

sentado o ícone de ônibus na cor azul, como definido no momento da criação do tema. Posteriormente é selecionado um ponto do trajeto da linha de ônibus 5102.

Após clicar no mapa na posição em que deseja inserir uma nova contribuição, a aplicação tem seu fluxo direcionado para a tela contendo o formulário de contribuição, como pode-se ver na Figura 5.9. Para finalizar a contribuição, basta que o usuário preencha o formulário e selecione a opção *create*.

New: Lotação de ônibus

Linha de ônibus 5102	Número aproximado de pessoas 3
Observado em 2016-07-29 08:45:00	Descrição Ponto de partida do itinerário.

✕ CANCEL
CREATE

Figura 5.9: Formulário de nova contribuição

Nesse caso foram preenchidos os dados de uma observação realizada no momento da partida, no início do itinerário, contendo apenas três pessoas às oito horas e quarenta e cinco minutos no período da manhã.

Finalmente, o usuário pode visualizar sua nova contribuição no mapa, assim como suas informações através do clique no ícone ou geometria da contribuição (Figura 5.10).

The screenshot shows the mobile application interface. On the left, a sidebar displays the following information for a new contribution:

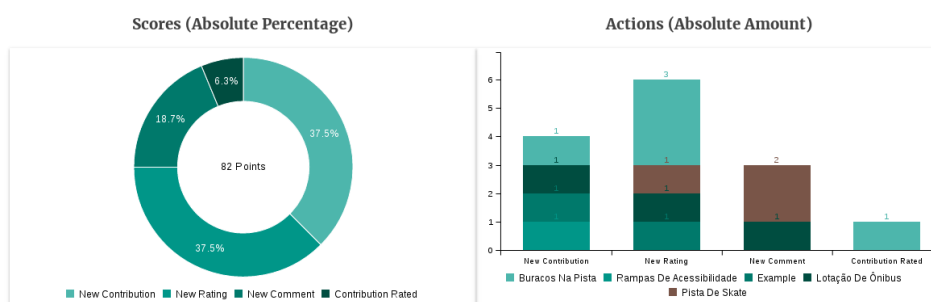
- ADMINISTRATOR USER**
- Today at 3:51 PM
- Linha de ônibus: 5102
- Número aproximado de pessoas: 3
- Observado em: 2016-07-29T11:45:00.000Z
- Descrição: Ponto de partida do itinerário.
- Rate: ★ ★ ★ ★ ★ 0
- New comment: [input field]
- SEND button
- SEE MORE link

The main area shows a map with a blue bus icon and a location pin. The map includes labels for various locations such as 'Departamento de Gestão Ambiental', 'Escola de Música', 'Faculdade de Letras', and 'Editora UFMG'. At the bottom, there is a 'Contributions' section with a star icon and search/magnifying glass icons.

Figura 5.10: Informações sobre a nova contribuição

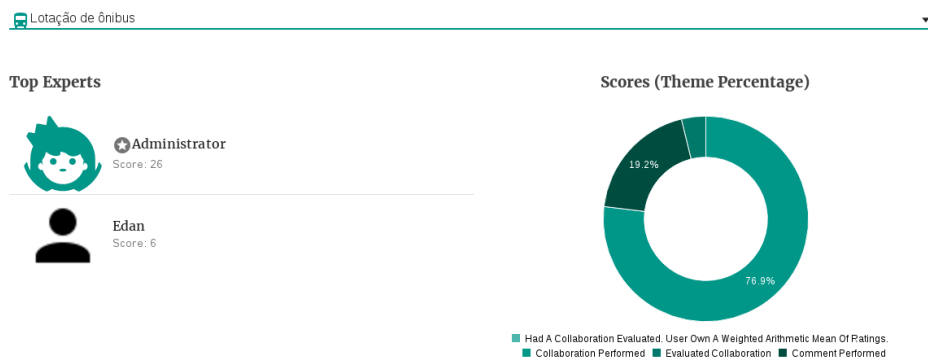
Após algumas interações com a ferramenta, o usuário pode visualizar os pontos obtidos de duas formas: absoluta e por tema. Para o primeiro caso é apresentado um gráfico com o acumulado de pontos obtidos pelo mesmo em todos os temas. Da mesma forma, a ferramenta apresenta um gráfico com a contagem de ações por tema, através de um gráfico de barras. No segundo tipo de *feedback*, o usuário pode escolher um tema específico, em que é mostrado o *ranking* de usuários que mais contribuíram, assim como a porcentagem o peso dos pontos obtidos em cada ação para a composição de seu *score* no respectivo tema.

Statistics



(a) *Feedback* de ações em termos gerais

Themes



(b) *Feedback* de ações para temas específicos

Figura 5.11: *Feedback*

Capítulo 6

Conclusão e Trabalhos Futuros

Com o objetivo de apresentar uma nova abordagem de *frameworks* para aplicações de VGI, o presente trabalho apresentou uma visão geral a respeito do conceito de *crowdsourcing* ativo e dos diferentes tipos aplicações de VGI. Nesse sentido foram apresentados os principais problemas inerentes desse tipo de coleta de dados, e em contrapartida, as abordagens que surgiram para lidar com essas questões. Além disso foram apresentadas as principais abordagens de *frameworks* para o desenvolvimento de aplicações de VGI.

Utilizando-se desse contexto teórico, foi proposta uma nova abordagem de *framework* para VGI, sobre a ideia de múltiplos temas. Os requisitos funcionais e não-funcionais do sistema foram levantados tendo em mente o referencial teórico analisado, sendo posteriormente aplicados no processo de desenvolvimento da aplicação.

O trabalho também realizou uma análise das tecnologias de desenvolvimento de aplicações *web* disponíveis. A partir da avaliação de tais tecnologias, guiadas pelos requisitos não-funcionais levantados previamente, estabeleceu-se então o ambiente de desenvolvimento baseado em NodeJS, Hapi e AngularJS.

Após apresentar detalhes da implementação dos módulos da aplicação, o presente trabalho realizou uma avaliação do *framework* desenvolvido através de um estudo caso sobre mobilidade urbana onde um tema sobre lotação de ônibus foi desenvolvido. Assim sendo, pode-se visualizar o processo de criação de um tema e verificar que o *framework* permite a criação simples de novos temas, sem a necessidade do conhecimento do usuário sobre linguagens de programação e algoritmos.

O presente trabalho apresentou um número ainda reduzido de funções e recursos descritos na literatura, porém alcançou seus objetivos principais, que consistiam na proposta da arquitetura multitemática, e em explorar as implicações preliminares desse tipo de abordagem sobre os processos associados à coleta desse tipo de informação. Des-

sas implicações, pode-se elaborar o princípio de um sistema de *expertise* de usuários, baseando-se no sistema tradicional de reputação e hierarquização dos usuários. Adicionalmente, o presente trabalho foi conscientemente projetado e desenvolvido visando sua contínua evolução, e portanto atingiu o seu objetivo ao elaborar esse alicerce para o debate e inclusão de novas funcionalidades sobre o ponto de vista multitemático.

6.1 Trabalhos futuros

Este trabalho abre portas para o desenvolvimento de diversas extensões, grande parte delas baseadas em funcionalidades tradicionais descritas na literatura sobre o ponto de vista da atual abordagem. Entre essas funcionalidades estão um sistema de anotação de metadados automatizado, a introdução de estratégias de gamificação dos temas, além de estratégias de demanda por colaborações em projetos orientados a missões. Como o presente trabalho já apresenta um sistema de pontuação, o mesmo pode facilmente incorporar um sistema automatizado para a anotação de metadados, similar ao introduzido por de Souza et al. [2013]. Seguindo a mesma linha de raciocínio, através da pontuação dos usuários torna-se viável a introdução de técnicas de gamificação, como aquelas baseadas em alocação e desalocação geográfica [Antoniu & Schlieder, 2014], utilizando como condição de alocação o número de pontos obtidos pelo usuário, por exemplo. Já as estratégias de colaboração orientadas a missões podem utilizar os pontos por ação dos usuários juntamente com a concentração geográfica de suas contribuições para recomendar tarefas [Chen et al., 2014], as quais, por outro lado, devem ter seu arcabouço de requisição igualmente adicionado na ferramenta. Nesse caso o *framework* pode ir além das tarefas orientadas a missões de coleta de dados geográficos, e explorar o que Haklay [2013] chama de *volunteered thinking*, no qual os usuários compartilham sua capacidade cognitiva, implementando um método similar ao *Mechanical Turk* ¹.

Além das expansões baseadas nas funcionalidades convencionais descritas na literatura de aplicações de VGI, é possível explorar o comportamento de técnicas automáticas de avaliação da qualidade desenvolvidas para outros ambientes colaborativos, como fóruns de perguntas [Dalip, 2015] e a *Wikipedia* [Hasan Dalip et al., 2009].

Outra extensão que também pode ser considerada é a utilização de redes sociais para a autenticação do usuário, e a inserção de vínculos sociais, que adicionam mais informação para melhor inferir a pontuação sobre ações do usuário, tendo em vista que aqueles que possuem vínculo podem adotar estratégias para o acúmulo de pontos

¹Mechanical Turk - <https://www.mturk.com/mturk/welcome>

através da validação constante entre “amigos”. Além disso, isso permitiria o estudo da *expertise* em subgrafos, identificando grupos de especialistas, e a mineração de novos voluntários.

Por fim, pode-se também expandir o trabalho para o contexto de *crowdsensing*, como definido por Mateveli et al. [2015], através de uma adaptação da aplicação *Web* para *mobile*. Uma vez que os contratos das funções da API são bem definidos, essa expansão apenas implicaria no desenvolvimento dessa aplicação, consumindo os *end-points* já desenvolvidos na API. Assim sendo, o ambiente *mobile* em si apresenta várias possibilidades para expansão, uma vez que possui à sua disposição diversos sensores que podem ser utilizados para a coleta de outros tipos de mídias, como imagens e vídeos, assim como dados, entre eles, o nível de ruído [Vellozo et al., 2013] e leituras do acelerômetro. A precisão dos equipamentos *mobiles* têm se tornado cada dia maiores, o que viabiliza sua utilização em vários projetos de *crowdsourcing/crowdsensing*, como em *citizen science* [Haklay, 2013] onde a acurácia e os métodos de coleta são de extrema importância. A partir da expansão da aplicação proposta para esse contexto, pode-se então dar os primeiros passos para a criação de um *framework* multitemático para coleta de informação geográfica através de *crowdsensing*.

Referências Bibliográficas

- Antoniou, V. & Schlieder, C. (2014). Participation patterns, vgi and gamification. Em *Proceedings of the Seventeenth AGILE Conference on Geographic Information Science, Geogames and Geoplay Workshop, Castellon, Spain*.
- Castelein, W.; Grus, L.; Cromptvoets, J. & Bregt, A. (2010). A characterization of volunteered geographic information.
- Chen, Z.; Fu, R.; Zhao, Z.; Liu, Z.; Xia, L.; Chen, L.; Cheng, P.; Cao, C. C.; Tong, Y. & Zhang, C. J. (2014). gmission: A general spatial crowdsourcing platform. *Proceedings of the VLDB Endowment*, 7(13):1629--1632.
- Coleman, D. J.; Georgiadou, Y.; Labonte, J. et al. (2009). Volunteered geographic information: The nature and motivation of producers. *International Journal of Spatial Data Infrastructures Research*, 4(1):332–358.
- Committee, F. G. D. et al. (1997). Framework introduction and guide. *Washington, DC: Federal Geographic Data Committee*.
- Craglia, M. (2007). Volunteered geographic information and spatial data infrastructure: when do parallel lines converge? *Workshop on Volunteered Geographic Information*.
- Dalip, D. H. (2015). *Uma abordagem multi-visão para a estimativa automática da qualidade de conteúdo colaborativo na Web 2.0*. Tese de doutorado.
- Davis Jr, A. C.; Vellozo, S. H. & Pinheiro, B. M. (2013). A framework for web and mobile volunteered geographic information applications. *XIV Brazilian Symposium on GeoInformatics (GeoInfo 2013)*.
- de Souza, W.; Lisboa-Filho, J.; de Sousa Câmara, J.; Filho, J. & de Paiva Oliveira, A. (2014). Clickonmap: A framework to develop volunteered geographic information systems with dynamic metadata. Em Murgante, B.; Misra, S.; Rocha, A.; Torre, C.;

- Rocha, J.; Falcão, M.; Taniar, D.; Apduhan, B. & Gervasi, O., editores, *Computational Science and Its Applications – ICCSA 2014*, volume 8580 of *Lecture Notes in Computer Science*, pp. 532–546. Springer International Publishing.
- de Souza, W. D.; Lisboa Filho, J.; Vidal Filho, J. N. & Câmara, J. H. (2013). DM4VGI: A template with dynamic metadata for documenting and validating the quality of volunteered geographic information. *XIV Brazilian Symposium on GeoInformatics (GeoInfo 2013)*.
- Drew, C. H. (2003). Transparency—considerations for PPGIS research and development. *URISA Journal*, 15(1):73–78.
- Elwood, S. (2006). Beyond cooptation or resistance: Urban spatial politics, community organizations, and GIS-based spatial narratives. *Annals of the Association of American Geographers*, 96(2):323–341.
- Elwood, S.; Goodchild, M. F. & Sui, D. (2012). Researching volunteered geographic information: Spatial data, geographic research, and new social practice. *Annals of the Association of American Geographers*, 102(3):571–590.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. Tese de doutorado, University of California, Irvine.
- Flanagin, A. J. & Metzger, M. J. (2008). The credibility of volunteered geographic information. *GeoJournal*, 72(3-4):137–148.
- Foth, M.; Choi, J. H.-j. & Satchell, C. (2011). Urban informatics. Em *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pp. 1–8. ACM.
- Furtado, V.; Ayres, L.; de Oliveira, M.; Vasconcelos, E.; Caminha, C.; D’Orleans, J. & Belchior, M. (2010). Collective intelligence in law enforcement - The Wikicrimes System. *Inf. Sci.*, 180(1):4–17. ISSN 0020-0255.
- GISPro (2007). The gispro interview with osm founder steve coast. *GIS Professional*, (18):20–23.
- Goodchild, M. (2009). Neogeography and the nature of geographic expertise. *J. Locat. Based Serv.*, 3(2):82–96. ISSN 1748-9725.
- Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geograph. *GeoJournal*, pp. 211–221.

- Goodchild, M. F. & Li, L. (2012). Assuring the quality of volunteered geographic information. *Spatial Statistics*, 1(0):110–120.
- Grey, F. (2009). Viewpoint: The age of citizen cyberscience. *Cern Courier*, 29.
- Haklay, M. (2010). How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and planning B: Planning and design*, 37(4):682--703.
- (2013). *Citizen Science and Volunteered Geographic Information: Overview and Typology of Participation*, pp. 105--122. Springer Netherlands, Dordrecht.
- Hasan Dalip, D.; André Gonçalves, M.; Cristo, M. & Calado, P. (2009). Automatic quality assessment of content created collaboratively by web communities: A case study of wikipedia. Em *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '09, pp. 295--304, New York, NY, USA. ACM.
- Johnson, P. A. & Sieber, R. E. (2013). Situating the adoption of VGI by government. Em Sui, D.; Elwood, S. & Goodchild, M., editores, *Crowdsourcing Geographic Knowledge*, pp. 65–81. Springer Netherlands.
- Lin, W. (2013). When Web 2.0 meets public participation GIS (PPGIS): VGI and spaces of participatory mapping in China. Em Sui, D.; Elwood, S. & Goodchild, M., editores, *Crowdsourcing Geographic Knowledge*, pp. 83–103. Springer Netherlands.
- Martella, R.; Kray, C. & Clementini, E. (2015). *A Gamification Framework for Volunteered Geographic Information*, pp. 73--89. Springer International Publishing, Cham.
- Mateveli, V. G.; Machado, G. N.; Moro, M. M. & Davis Jr, A. C. (2015). Taxonomia e desafios de recomendação para coleta de dados geográficos por cidadãos. *XXX Simpósio Brasileiro de Bancos de Dados (SBBDD 2015)*.
- Maué, P. (2007). Reputation as a tool to ensure validity of VGI. *Workshop on Volunteered Geographic Information*.
- Okolloh, O. (2009). Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. *Participatory learning and action*, 59(1):65–70.
- O'Reilly, T. (2005). What is web 2.0: Design patterns and business models for the next generation of software.
- Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49.

- Richardson, W. (2009). *Blogs, Wikis, Podcasts, and Other Powerful Web Tools for Classrooms*. SAGE Publications. ISBN 9781412959711.
- Sheppard, S. A. (2012). wq: A modular framework for collecting, storing, and utilizing experiential VGI. Em *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, GEOCROWD '12, pp. 62–69, New York, NY, USA. ACM.
- Silva, C. T. J. & Davis Jr, A. C. (2008). Um framework para coleta e filtragem de dados geográficos fornecidos voluntariamente. *X Brazilian Symposium on GeoInformatics (GeoInfo 2008)*.
- Soares, F. H.; Pinheiro, B. M. & Davis Jr, A. C. (2014). Mapeamento participativo de opiniões sobre o uso de dinheiro público. *XV Brazilian Symposium on GeoInformatics (GeoInfo 2014)*.
- Sui, D.; Goodchild, M. & Elwood, S. (2013). Volunteered geographic information, the exaflood, and the growing digital divide. Em Elwood, S.; Goodchild, M. F. & Sui, D., editores, *Crowdsourcing Geographic Knowledge*, pp. 1–12. Springer Netherlands.
- Tobler, W. R. (1970). A computer movie simulating urban growth in the Detroit region. *Economic geography*, pp. 234–240.
- Tulloch, D. L. (2007). Many, many maps: Empowerment and online participatory mapping. *First Monday*, 12(2).
- Turner, A. (2006). *Introduction to neogeography*. "O'Reilly Media, Inc."
- Vellozo, S. H.; Pinheiro, B. M. & Davis Jr, A. C. (2013). Strepitus: um aplicativo para coleta colaborativa de dados sobre ruído urbano. *IV Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*.
- Zheng, Y.; Capra, L.; Wolfson, O. & Yang, H. (2014). Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38.