



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Arquitetura para Coleta de Opinião sobre Serviços
Públicos em um Sistema de Informação Geográfica
Móvel com Participação Popular**

Breno Diogo de Carvalho Camargos

Brasília
2015



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Arquitetura para Coleta de Opinião sobre Serviços Públicos em um Sistema de Informação Geográfica Móvel com Participação Popular

Breno Diogo de Carvalho Camargos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Engenharia da Computação

Orientadora
Prof.^a Dr.^a Maristela Terto de Holanda

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Engenharia da Computação

Coordenador: Prof. Dr. Ricardo Zelenovsky

Banca examinadora composta por:

Prof.^a Dr.^a Maristela Terto de Holanda (Orientadora) — CIC/UnB
Prof. Ms. Henrique Pereira de Freitas Filho — IFG
Prof. Dra. Aletéia Patrícia Favacho de Araújo — CIC/UnB

CIP — Catalogação Internacional na Publicação

Camargos, Breno Diogo de Carvalho.

Arquitetura para Coleta de Opinião sobre Serviços Públicos em um Sistema de Informação Geográfica Móvel com Participação Popular / Breno Diogo de Carvalho Camargos. Brasília : UnB, 2015.

139 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. Sistema de Informação Geográfica com Participação Popular - SIGPP, 2. Gestão de Serviços Públicos, 3. Arquitetura para Opiniões Georreferenciadas.

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedico à minha família e aos meus amigos.

Dedico especialmente aos meus avós que tiveram grande importância na formação do meu caráter.

Coming together is a beginning; keeping together is progress; working together is success.

–Henry Ford

Agradecimentos

Agradeço à Deus.

Agradeço aos meus pais que me apoiam incondicionalmente em todos os aspectos da minha vida desde o dia em que nasci.

Agradeço aos meus irmãos que sempre têm tanto a ensinar.

Agradeço à Dielen que teve paciência e soube me motivar nos momentos que precisei.

Agradeço aos meus amigos que estando ao meu lado compartilharam altos e baixos, dividindo o peso dos momentos difíceis e multiplicando a alegria dos momentos felizes.

Agradeço aos meus professores que tanto me ensinaram nesses últimos anos. Agradeço em especial à Dr.^a Maristela que me orientou pacientemente e me ajudou com qualquer problema que surgira ao longo do desenvolvimento deste trabalho.

Agradeço à Universidade de Brasília que me deu a oportunidade de estudar em uma das melhores universidades do país.

Resumo

Uma ferramenta importante para a gestão de qualquer serviço público é a opinião dos seus usuários. Por meio da opinião dos cidadãos que estão no ambiente do serviço público, o gestor pode tomar decisões administrativas que resultem de forma direta na melhoria dos serviços. O rápido avanço do Sistema de Posicionamento Global - GPS e das tecnologias incluídas na Web 2.0, seguidos pela maior acessibilidade a essas tecnologias pelos cidadãos por meio de *smartphones* pode ser utilizado para incluir a população, que está presente nos ambientes dos estabelecimentos públicos, no processo administrativo desses locais. O chamado Sistema de Informação Geográfica com Participação Popular – SIGPP é caracterizado pela criação de informações com características geográficas direcionadas à entidade governamental.

No Brasil, os gestores dos serviços públicos ainda não dispõem de dados georreferenciados contendo a opinião dos usuários dos serviços públicos. Nesse contexto, esta monografia propõe e valida, através de um estudo de caso, uma arquitetura para obter opiniões georreferenciadas dos usuários de qualquer serviço público no Brasil, e disponibilizar tais opiniões de uma maneira que facilite a gestão desses serviços. A proposta inclui a obtenção da opinião de usuários de algum serviço público através de um *smartphone*, a validação da localização do usuário através das coordenadas geográficas de seu dispositivo móvel e a visualização dessa opinião pelo gestor em um mapa. O mapa contendo as opiniões cadastradas irão auxiliar as decisões administrativas do gestor do serviço público.

Palavras-chave: Sistema de Informação Geográfica com Participação Popular - SIGPP, Gestão de Serviços Públicos, Arquitetura para Opiniões Georreferenciadas.

Abstract

An important tool to the management of any public service is the feedback of its users. Based on the opinion of the citizens that are inside the public service's environment, the manager is able to make administrative choices that impact directly on the improvement of the services. The quick advance of the Global Position System - GPS and the technologies embedded on the Web 2.0, followed by the bigger accessibility to those technologies by the citizens through smartphones may be used to include the population, that are present on the public establishments' environment, into the administrative process of those places. The denominated Public Participation Geographic Information System - PPGIS is characterized by the creation of information with geographic features directed to the government entity.

In Brazil, the public services' managers still lack georeferenced data with the opinion of the public services' users. In this context, this monograph proposes and validates, through a case study, one architecture to acquire georeferenced feedback from users of any public service in Brazil, and provide such feedback in a way that improves the management of these services. The proposal includes the acquirement of the opinion of public service's users through a smartphone, the validation of the users' location through the geographic coordinates of his mobile device and the preview of this feedback by the manager in a map. The map containing the opinions will assist the administrative decisions of the public service's manager.

Keywords: Public Participation Geographic Information System - PPGIS, Public Services Management, Architecture for Georeferenced Opinions.

Sumário

1	Introdução	1
1.1	Objetivo	2
1.1.1	Objetivos Específicos	2
1.2	Estrutura do Trabalho	2
2	Sistema de Informação Geográfica com Participação Popular	3
2.1	Sistema de Informação Geográfica	3
2.1.1	Bancos de Dados Geográficos	6
2.2	Sistema de Informação Geográfica Voluntária	8
2.3	Sistema de Informação Geográfica Móvel	10
2.3.1	Computação Móvel	10
2.4	Sistema de Informação Geográfica com Participação Popular	13
2.5	Sistema de Informação Geográfica Livre	14
2.5.1	<i>Software</i> Livre	15
3	Arquitetura Proposta	17
3.1	Cenário de Execução	17
3.2	Detalhamento da Arquitetura	18
3.2.1	Interface Móvel	19
3.2.2	Interface Web	21
3.2.3	Comunicação de Dados	22
3.2.4	Camada de Dados	23
3.2.5	Sistema Gerenciador de Banco de Dados	24
4	Estudo de Caso	26
4.1	Visão Geral	26
4.1.1	ConsultaOpinio	26
4.1.2	Interface Web	29
4.1.3	Banco de Dados Central	30
4.1.4	Algoritmos para Comunicação de Dados	31
4.2	Testes Realizados	33
5	Conclusão	41
5.1	Trabalhos Futuros	42
	Referências	43

A	<i>Scripts</i> Fonte Utilizados no Banco de Dados	48
B	Códigos Fonte Utilizados nas Interfaces	51

Lista de Figuras

2.1	Sistema de Informação Geográfica - Abstração	4
2.2	Sistema de Informação Geográfica - Camadas	5
2.3	Sistema de Informação Geográfica - Vetores e Rasters	6
2.4	Modelo OMT-G - Geo-objetos	7
2.5	Modelo OMT-G - Relacionamentos	8
2.6	Banco de Dados Geográficos	8
2.7	Topologia da Computação Móvel	11
3.1	Ambiente de Execução	17
3.2	Funcionamento da Arquitetura	18
3.3	Abstração da Arquitetura	19
3.4	Fluxograma da Interface Móvel	20
3.5	Fluxograma da Interface Web	21
3.6	Modelo do Banco de Dados Central	23
3.7	Solicitações ao Sistema Gerenciador de Banco de Dados	24
4.1	Fluxograma do Aplicativo - Autenticação	27
4.2	Fluxograma do Aplicativo - Mapa	28
4.3	Fluxograma do Aplicativo - Perguntas	28
4.4	Fluxograma da Página Web - Mapa	29
4.5	Fluxograma da Página Web - Tabela	30
4.6	Modelo do Banco de Dados Central	31
4.7	Interface Móvel - Validação da Localização	32
4.8	Interface Web - Mapa	35
4.9	Funcionalidade de Cadastro e Autenticação	36
4.10	Funcionalidade de Seleção do Local	37
4.11	Funcionalidade de Cadastro de Opinião	38
4.12	Funcionalidade de Visualização do Mapa pelo Gestor	39
4.13	Funcionalidade de Visualização da Tabela pelo Gestor	39

Lista de Tabelas

4.1	Escolas Cadastradas no Banco de Dados Central.	34
4.2	Hospitais Cadastradas no Banco de Dados Central.	34

Lista de Apêndices

A.1	<i>Script</i> de Criação do Banco de Dados Central	48
A.2	<i>Script</i> para Cadastro das Perguntas no Banco de Dados Central	49
B.1	Interface do Gestor	51
B.2	Verificação da Existência de Usuário	54
B.3	Página Requisitada na Verificação do Usuário	55
B.4	Consulta aos Estabelecimentos Cadastrados	55
B.5	Página Requisitada para Descobrir os Estabelecimentos Cadastrados	55
B.6	Validação das Coordenadas do Usuário	56
B.7	Consulta às Perguntas Cadastradas no Banco	57
B.8	Página Requisitada para Descobrir as Perguntas Cadastradas	57
B.9	Cadastro da Avaliação do Usuário	57
B.10	Página Requisitada para Cadastro da Avaliação do Usuário	58
B.11	Página Responsável por Listar Todas as Avaliações dos Usuários para o Local Selecionado	59

Capítulo 1

Introdução

A *World Wide Web* evoluiu de uma enciclopédia unidirecional de informação para um meio mais dinâmico, interativo e participativo, cunhado de Web 2.0 [2]. Essa evolução coincidiu com um refinamento dos padrões de troca de dados geográficos além de tecnologias de base, particularmente, o Sistema de Posicionamento Global – GPS (*Global Position System*) [30]. As tecnologias capazes de lidar com dados espaciais deixaram de ser de alto custo e restritas às instituições formais e passaram a ser mais acessíveis ao cidadão comum [49]. O termo Sistema de Informação Geográfica – SIG (*Geographic Information System*) descreve o uso das tecnologias espaciais disponíveis para facilitar a gestão ou análise de fenômenos espaciais [40, 47].

Por outro lado, no começo dos anos 2000 os *smartphones* passaram a se inserir cada vez mais na realidade dos brasileiros [16, 44]. Os novos celulares aumentaram a capacidade do usuário de criar conteúdo e interagir com outros usuários em tempo real, além de permitirem o uso de GPS e outras tecnologias emergentes [7]. Os *smartphones* também permitiram o georreferenciamento, isto é, a associação de alguma informação a uma localização no espaço, geralmente por meio das coordenadas de latitude e longitude [34].

O Sistema de Informação Geográfica com Participação Popular – SIGPP (*Public Participation Geographic Information System*) possui a particularidade de descrever como tecnologias SIG podem dar suporte à participação pública com o objetivo de incluir população local ou marginalizada no processo de decisão e planejamento de suas áreas de vida [32, 55]. E ainda, no Brasil, os gestores dos serviços públicos não dispõem de dados georreferenciados contendo a opinião dos usuários dos serviços públicos.

Nesse contexto, este trabalho propõe uma arquitetura para obter dados georreferenciados dos usuários de serviços públicos no Brasil, e disponibilizar tais dados de uma maneira que auxilie na gestão desses serviços. Os dados coletados serão armazenados em um banco de dados central e poderão ser visualizados por um gestor em um computador. Esse acesso será feito por uma interface web, que apresentará ao gestor um mapa com os serviços que ele administra. No mapa haverá indicadores visuais para facilitar a gestão dos serviços. Assim, o sistema proposto irá funcionar em dois ambientes diferentes:

- Ambiente do usuário: para o usuário será desenvolvido um aplicativo capaz de utilizar a localização fornecida pelo GPS do *smartphone* para mostrar os locais próximos que podem ser avaliados;

- Ambiente do gestor: para o gestor será desenvolvida uma interface web capaz de apresentar em um mapa os locais cadastrados e a possibilidade de visualizar as opiniões sobre cada um desses locais.

A validação desse sistema é feita em um estudo de caso e os resultados também são expostos neste documento.

1.1 Objetivo

O objetivo geral desta monografia é propor e implementar um sistema capaz de obter dados sobre opinião dos usuários dos serviços públicos e apresentá-los ao gestor desses serviços. Os cidadãos que utilizam qualquer serviço público e desejem registrar sua opinião sobre como foram atendidos poderão instalar um aplicativo em seu *smartphone*, e através dele registrar sua opinião acerca do serviço em questão.

1.1.1 Objetivos Específicos

Os objetivos específicos necessários para o desenvolvimento deste trabalho são:

- Definir o banco de dados central, que armazenará os dados relacionados com os serviços e as suas respectivas avaliações;
- Definir a arquitetura para a aplicação móvel onde o usuário realizará as suas avaliações;
- Definir a arquitetura web, baseado em sistema de informação geográfica, onde será possível a visualização das avaliações pelos gestores.
- Implementar a arquitetura proposta;
- Validar através de testes o sistema implementado.

1.2 Estrutura do Trabalho

Os próximos capítulos desta monografia estão divididos da seguinte maneira:

- Capítulo 2: Fundamentação Teórica, onde são apresentados os conceitos de Sistemas de Informação Geográfica;
- Capítulo 3: Arquitetura Proposta, onde é apresentada de forma detalhada a arquitetura desenvolvida neste trabalho;
- Capítulo 4: Estudo de Caso, onde são apresentados os dados da validação da arquitetura proposta;
- Capítulo 5: Conclusões, onde é feita a análise dos resultados obtidos e são propostos alguns trabalhos futuros.

Capítulo 2

Sistema de Informação Geográfica com Participação Popular

Neste capítulo, a fundamentação teórica para o desenvolvimento deste trabalho é apresentada. Inicialmente, na Seção 2.1, são definidos os Sistemas de Informação Geográfica e seus principais componentes. Na Seção 2.2 os Sistemas de Informação Geográfica Voluntária são explicados e em seguida, na Seção 2.3, os Sistemas de Informação Geográfica Móvel são elucidados. A Seção 2.4 define os Sistemas de Informação Geográfica com Participação Popular. Por fim, na Seção 2.5, é abordada a importância do *software* livre no contexto deste projeto.

2.1 Sistema de Informação Geográfica

O termo Sistema de Informação Geográfica, em sua definição mais abrangente, se refere a qualquer sistema de informação digital que possua dados de alguma forma referenciados geograficamente [29]. Em sua definição mais técnica, SIG é frequentemente utilizado para sistemas compostos pela entrada, armazenamento, gerenciamento, análise e saída de dados geográficos [29, 40, 47]. A seguir, algumas definições de SIG¹.

Conjunto poderoso de ferramentas para coletar, armazenar, recuperar, transformar e visualizar dados sobre o mundo real.

–Burrough, 1986 [9]

Um sistema de suporte à decisão que integra dados referenciados espacialmente num ambiente de respostas aos problemas.

–Cowen, 1988 [17]

Um banco de dados indexados espacialmente, sobre o qual opera um conjunto de procedimentos para responder a consultas sobre entidades espaciais.

–Smith, 1987 [57]

¹As definições originais estão em inglês e foram adicionadas em tradução livre.

Sistemas de Informações Geográficas são sistemas computacionais usados para armazenar e manipular informações geográficas.

–Aronoff, 1989 [4]

Uma tecnologia de informação que armazena, analisa e apresenta ambos dados espaciais e dados não espaciais.

–Parker, 1988 [51]

Um sistema para obter, armazenar, examinar, manipular, analisar e apresentar dados que são espacialmente referenciados à Terra.

–DoE, 1987 [21]

Como todo sistema de informação, o SIG combina um banco de dados e vários procedimentos ou algoritmos que operam sobre esses dados [29]. Um modelo genérico de um SIG está na Figura 2.1. Como pode ser observado, à esquerda há vários bancos de dados com informações geográficas armazenadas e à direita há um *software* SIG responsável por manipular esses dados.



Figura 2.1: Sistema de Informação Geográfica - Abstração. Adaptado de [45].

Por causa da natureza dos dados em questão, a entrada e a saída dos dados são frequentemente mais elaboradas do que em outros sistemas de informação, dependendo da capacidade de processamento para dados geográficos. Historicamente, o desenvolvimento de SIG foi de certa forma limitado pela disponibilidade de equipamentos adequados [8]. No entanto, o crescimento exponencial da capacidade de processamento dos computadores atuais possibilitou que até mesmo dispositivos portáteis como notebooks, *tablets* e *smartphones* fossem capazes de exibir mapas e lidar com informações geográficas.

Como observado, o SIG é visto como uma ferramenta importante para pesquisa científica [1]. Em essência, SIG provê uma perspectiva geográfica da informação. Mapas

disponibilizando a localização geográfica do dado que está sendo analisado podem ser importantes para novas descobertas e interpretações sobre o dado original.

Nos sistemas de informação geográfica, coleções homogêneas de objetos geográficos como ruas, prédios, cidades, distritos são organizados em camadas ou temas. A Figura 2.2 apresenta uma abstração da maneira como os dados são tratados em SIG. Cada camada representa alguma informação sobre uma cidade. De cima para baixo, a primeira camada são pontos que representam os clientes armazenados na base de dados. A segunda camada são linhas que representam as ruas da cidade. As outras camadas representam lotes, a elevação e o uso do terreno da cidade respectivamente. A representação mais próxima do mundo real ocorre quando todas as camadas são visualizadas ao mesmo tempo.

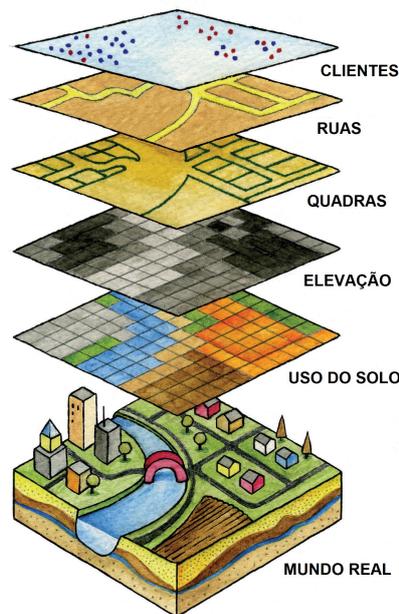


Figura 2.2: Sistema de Informação Geográfica - Camadas. Adaptado de [12].

Muitos *software* oferecem uma grande quantidade de ferramentas para processar camadas geográficas e derivar muitas relações entre os elementos geográficos que elas contêm [40]. Em um SIG é possível representar objetos do mundo real em formatos que podem ser armazenados, manipulados e exibidos em um computador. Os dois formatos mais comuns são vetores e *rasters*, discutidos a seguir:

- Vetores: em um modelo de vetores o mundo real pode ser representado e modelado em elementos discretos e homogêneos, como representado na parte superior da Figura 2.3 [47]. Os elementos que são representados podem ser pontos, linhas ou polígonos. Por exemplo, casas podem ser pontos, ruas podem ser linhas e lagos podem ser polígonos. Para cada um desses tipos de dados, um código é associado às coordenadas para identificá-los. Pontos são armazenados como um par de coordenadas discretas, linhas como pares de coordenadas e polígonos do mesmo modo que as linhas com as mesmas coordenadas no início e no final. Esse sistema de vetores geralmente é utilizado para representar dados que possuem fronteiras bem definidas, tais como prédios, ruas ou distritos.

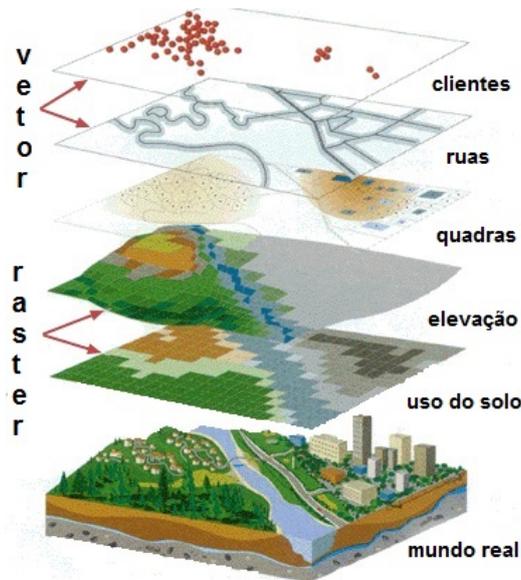


Figura 2.3: Sistema de Informação Geográfica - Vetores e Rasters. Adaptado de [12].

- *Rasters*: em um modelo de *raster* o mundo real é representado continuamente em uma matriz de grade, tal como representado na parte inferior da Figura 2.3. Cada célula na matriz de grade armazena a característica espacial e o atributo do dado. Cada célula do modelo corresponde a uma localização. A localização de cada célula é definida por sua coordenada da linha e da coluna. Geralmente o modelo de *raster* é utilizado para dados que são contínuos tais como temperatura, altitude de terrenos ou poluição [47].

2.1.1 Bancos de Dados Geográficos

Informações geográficas devem ser armazenadas em um banco de dados geográficos. Banco de Dados Geográficos - BDG, também chamado de Banco de Dados Espaciais - BDE, é semelhante a um banco de dados relacional, com a diferença de suportar feições geométricas em suas tabelas [10, 18, 54].

Este tipo de base com geometria oferece a possibilidade de análise e consultas espaciais [54]. É possível calcular nestes casos, por exemplo, áreas, distâncias e centroides, além de realizar a geração de buffers e outras operações entre as geometrias [10].

Para a modelagem de bancos de dados geográficos a Técnica de Modelagem de Objetos para Aplicações Geográficas - OMT-G (*Object Modeling Technique for Geographic Applications*) é a mais utilizada [10]. Essa técnica provê primitivas para modelar a geometria e a topologia de dados geográficos, incluindo classes de objetos, relacionamentos convencionais e espaciais, e restrições de integridade espaciais, em concordância com a Infraestrutura Nacional de Dados Espaciais - INDE, "conjunto integrado de tecnologias; políticas; mecanismos e procedimentos de coordenação e monitoramento; padrões e acordos, necessário para facilitar e ordenar a geração, o armazenamento, o acesso, o compartilhamento, a disseminação e o uso dos dados geoespaciais de origem federal, estadual, distrital e municipal". [38, 54].

O modelo OMT-G apresenta um conjunto fixo de alternativas de representação geométrica, usando dois tipos de classes: geo-objeto com geometria e geo-objeto com geometria e topologia. A classe geo-objeto com geometria representa objetos que possuem apenas propriedades geométricas, e é especializada em classes: Ponto, Linha e Polígono. A classe geo-objeto com geometria e topologia representa objetos que possuem, além das propriedades geométricas, propriedades de conectividade topológica, sendo especificamente voltadas para a representação de estruturas em rede, tais como sistemas de abastecimento de água ou fornecimento de energia elétrica [10]. A Figura 2.4 apresenta a representação dessas classes em um diagrama. Como pode ser observado, a geometria fica representada no canto superior esquerdo, onde ponto é representado por uma estrela, linha por um traço e polígono por um quadrado. Para geo-objetos com geometria e topologia, uma linha unidirecional é representada por uma seta para direita, uma linha bidirecional é representada por uma seta que aponta para os dois lados e um nó de rede é representado por um ponto.

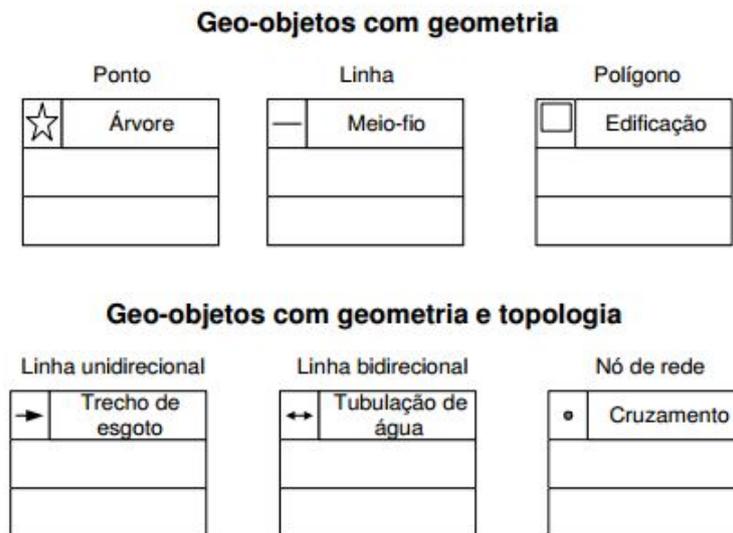


Figura 2.4: Modelo OMT-G - Geo-objetos. Retirado de [18].

O modelo OMT-G representa três tipos de relacionamentos entre suas classes: associações simples, relacionamentos topológicos em rede e relacionamentos espaciais. A discriminação de tais relacionamentos tem o objetivo de definir explicitamente o tipo de interação que ocorre entre as classes. Associações simples representam relacionamentos estruturais entre objetos de classes diferentes, convencionais ou georreferenciadas. Relacionamentos topológicos em rede ocorrem quando é possível derivar automaticamente alguma relação a partir da forma geométrica do objeto cadastrado no banco. Relacionamentos espaciais representam relações topológicas [10].

No modelo OMT-G, associações simples são indicadas por linhas contínuas, enquanto relacionamentos espaciais são indicados por linhas pontilhadas tal como pode-se observar na parte superior da Figura 2.5.

Os relacionamentos topológicos em rede são relacionamentos entre objetos que estão conectados uns com os outros. Relacionamentos de rede são indicados por duas linhas pontilhadas paralelas tal como pode-se observar na parte inferior da Figura 2.5.

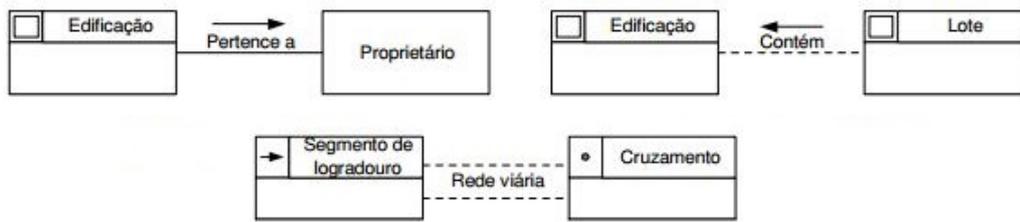


Figura 2.5: Modelo OMT-G - Relacionamentos. Retirado de [18].

Os sistemas gerenciadores de bancos de dados convencionais não suportam a implementação de banco de dados geográficos de forma nativa [56]. Por isso, diversas extensões espaciais que possibilitam trabalhar com esse tipo de informação espacial foram desenvolvidas. Um exemplo do uso dessas extensões é o PostGis, que é a extensão espacial do sistema gerenciador de banco de dados (SGBD) de código aberto PostgreSQL [31, 48, 53]. Dessa maneira, o banco de dados geográficos pode ser representado como na Figura 2.6. Como pode ser observado, o banco de dados geográficos passa a ser a combinação de um SGBD tradicional com uma extensão espacial responsável por lidar com os dados espaciais.

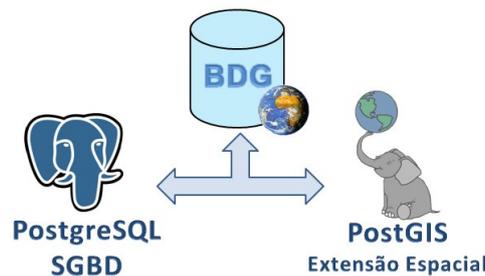


Figura 2.6: Banco de Dados Geográficos. Retirado de [46].

2.2 Sistema de Informação Geográfica Voluntária

Um Sistema de Informação Geográfica Voluntária - SIGV (*Volunteered Geographic Information System*) é o aproveitamento de ferramentas para criar, reunir e disseminar dados geográficos providos voluntariamente por indivíduos [13, 14, 30]. O SIGV se diferencia ainda dos SIG tradicionais por serem dados criados por voluntários geralmente sem treinamento algum [30]. SIGV faz com que a distinção entre “produtor, comunicador e consumidor de informação geográfica” fique cada vez mais confusa, uma vez que uma mesma pessoa tende a assumir os três papéis [28].

A Web 2.0, caracterizada pela capacidade do usuário de criar conteúdo e interagir com conteúdos de outros usuários em tempo real, o georreferenciamento, *geotagging*²,

²Processo de adição de informação geográfica aos metadados de diversas mídias, tais como fotografias e vídeos.

GPS, conexão de banda larga, além de outras tecnologias do começo dos anos 2000 foram os possibilitadores do surgimento do SIGV [30].

Sistemas de Informação Geográfica Voluntária têm o potencial para se tornar uma fonte significativa para entendimento geográfico da superfície terrestre. Possui a vantagem do baixo custo, devido à ação voluntária dos colaboradores, além de poder ser atemporal, com atualizações tão constantes quanto necessária à cada local. Ressalta também que o valor mais importante do SIGV pode estar nas informações acerca de várias localizações remotas geograficamente que não são mostradas pelas grandes mídias, e são catalogadas através dos olhos dos habitantes locais por meio do SIGV.

Focar apenas no aspecto “*informação*” do SIGV acaba por ignorar o rico elemento humano que dirige esse fenômeno [13]. As motivações para os contribuintes do SIGV podem estar relacionadas com a satisfação de ver seu trabalho disponível para outros usuários, além de observar a evolução do trabalho conjunto e o sentimento de realização por ter sido parte do projeto. Abaixo estão algumas razões para os agentes voluntários fazerem contribuições construtivas e também informações falsas. Entre as razões para contribuições positivas destacam-se [8, 13]:

- Altruísmo: contribuir para benefício do próximo;
- Interesse pessoal ou profissional: contribuir como parte de um projeto existente ou parte do trabalho ou projeto pessoal;
- Motivação intelectual: para aumentar suas capacidades técnicas e conhecimento por meio da colaboração;
- Proteção ou melhoramento de um investimento pessoal: quando oferecer uma resposta para um problema compartilhado pode oferecer recompensa ao ter um problema que também o afeta sendo resolvido;
- Recompensa social: a recompensa do sentimento de fazer parte de uma grande comunidade;
- Aumento da reputação pessoal: a oportunidade dos usuários criarem perfis *online* e desenvolverem uma reputação de respeito;
- Orgulho local: quando adicionar informações sobre sua comunidade local pode melhorar o turismo, desenvolvimento da economia, relações públicas ou simplesmente demonstrar que a rua do indivíduo “*aparece no mapa*”.

Quanto aos motivos para contribuições com informação falsa, os principais são [8, 13]:

- Travessura: pessoas travessas ou vândalos com esperança de gerar ceticismo ou confusão ao alterar informações corretas por conteúdo muitas vezes ofensivo ou sem sentido;
- Conluio: indivíduos independentes motivados por crenças em uma determinada comunidade, organização ou causa;
- Malícia e/ou intenção criminal: pessoas com intenção criminal ou maliciosa visando ganhos pessoais.

Devido à sua natureza, o tratamento do dado obtido através de um SIGV deve ser diferente do tratamento do dado obtido, por exemplo, por infraestruturas de dados espaciais, onde os dados são feitos de profissionais para profissionais em ambientes SIG. Em SIGV, deve se ter a consciência de que os dados foram obtidos por usuários sem necessariamente um treinamento formal. Até por isso, nenhum modelo padronizado de dados SIGV surgiu até o momento [41].

Diferentes lições podem ser aprendidas com *wikis* tais como *Wikipedia.org*. Wikipedia originalmente confiava apenas na “*sabedoria das massas*” para avaliar o seu conteúdo. No entanto, começando em dezembro de 2009, Wikipedia vem confiando em um time de editores para auxiliar em “*entradas alertas*” antes de decidir se incorpora ou não o novo conhecimento [63].

2.3 Sistema de Informação Geográfica Móvel

Sistema de Informação Geográfica Móvel - SIG Móvel (*Mobile Geographic Information System*) pode ser definido como uma estrutura que integra hardware e software para acesso aos dados espaciais através de uma rede sem fio, utilizando-se dispositivos móveis [42]. Existem duas principais áreas de aplicação dos SIG Móveis [42]:

- SIG baseado em campo: Com o objetivo de dar suporte à coleta de dados espaciais e descritivos no campo. A obtenção de dados censitários é um exemplo deste tipo de aplicação;
- Serviço Baseado em Localização - LBS (*Location Based Service*): utiliza dispositivos de orientação espacial, tipicamente o GPS, para prover serviços. Serviços de determinação de rotas ou monitoramento de veículos são exemplos dessa aplicação.

Diferentemente dos SIG tradicionais, o SIG Móvel está distribuído em ambientes computacionais. Logo, na modelagem de aplicações SIG alguns problemas requerem atenção especial [62]:

- Armazenagem e recuperação de dados: ferramentas para gerenciamento de dados, capazes de armazenar e manipular dados geográficos;
- Tecnologia de determinação da posição geográfica: em que se insere o problema de precisão da localização, confiabilidade, disponibilidade e cobertura. A tecnologia mais utilizada é o GPS;
- Acesso remoto aos dados: tecnologia de comunicação entre os diferentes dispositivos da rede e protocolos de comunicação.

2.3.1 Computação Móvel

O termo computação móvel é utilizado para descrever a comunicação de dispositivos com redes fixas ou móveis, através de um meio sem fio [19, 24, 39, 66]. A computação móvel permite que o portador do dispositivo móvel possa criar, acessar, processar, armazenar e comunicar informação sem estar limitado a uma determinada localização física [66].

A computação móvel é possibilitada por três fatores: um dispositivo móvel, um sistema (aplicações de *software*) e algum tipo de meio de comunicação. Ela permite que usuários se comuniquem com redes fixas através de um meio sem fio. Com o uso de algum dispositivo equipado com tecnologia para comunicação sem fio tais como *smartphones*, *tablets*, notebooks, entre outros, o usuário pode se comunicar com a rede local e com a internet. Hoje isso é possível através da tecnologia desenvolvida nas áreas de comunicação celular, redes locais sem fio e serviços via satélite.

Para melhor compreensão sobre o ambiente de funcionamento da computação móvel, a topologia do ambiente típico é reproduzida na Figura 2.7. Como pode-se observar, o ambiente possui dispositivos estáticos e dispositivos móveis. Em diferentes momentos utiliza-se comunicação cabeada ou comunicação sem fio, e toda a informação é capaz de fluir por todo o sistema. Esse modelo foi adaptado de [5, 19, 22, 35]. Como pode ser observado na Figura 2.7, os principais componentes são:

- *Hosts* fixos: componentes estáticos da rede de computadores;
- Estações base: responsáveis pela comunicação entre a unidade móvel e os *hosts* fixos. São equipadas com interfaces sem fio para receber e transmitir sinais em uma determinada área geográfica (célula) e em unidades móveis;
- Unidades móveis: dispositivos que se movem no domínio geográfico, tais como telefone celular, notebooks ou *PDA*s. Essas unidades acessam serviços de rede, independente de sua localização ou movimento.

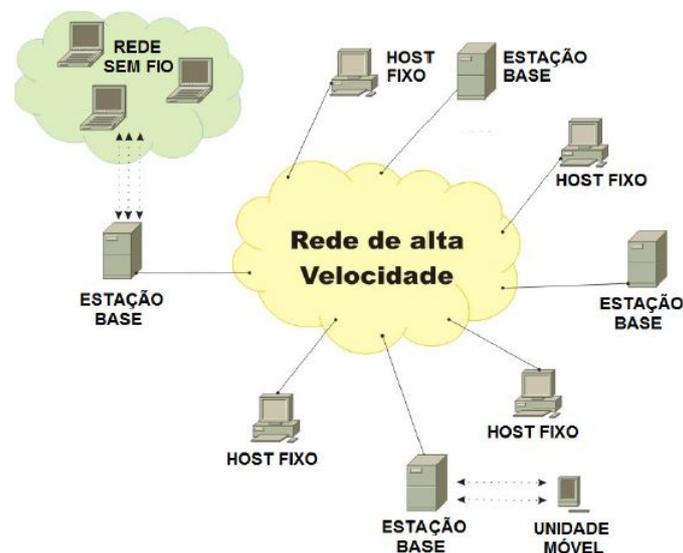


Figura 2.7: Topologia da Computação Móvel.

Características da Computação Móvel

Nesse cenário, nota-se que um dos principais objetivos da computação móvel é a mobilidade dos *smartphones*. Essa mobilidade introduz problemas e desafios que não existem

em ambientes fixos, envolvendo desde a velocidade do canal até a duração da bateria da unidade móvel. Uma característica importante da computação móvel é como os dispositivos se conectam com a informação que está na rede fixa. Segundo Zimmerman [66], existem quatro modos:

- Conectado: essa categoria assume uma conexão contínua de alta velocidade;
- Fracamente conectado: quando há a possibilidade de conexão contínua, porém com baixa velocidade;
- *Batch*: quando o dispositivo móvel não consegue se comunicar continuamente com a informação fixa. Nesse caso, a comunicação é estabelecida periodicamente ou aleatoriamente para trocar ou atualizar informações entre o dispositivo e o ambiente fixo;
- Desconectado: nesse modo não há conexão do dispositivo móvel com a rede de dados fixa. Os dados só chegam até a rede de dados central se inseridos manualmente após serem armazenados no dispositivo móvel.

Limitações

Contudo, a comunicação móvel é mais difícil de se alcançar do que a rede de comunicação cabeada porque o ambiente interage com o sinal sem fio, bloqueando caminhos e introduzindo ecos e ruídos. Como resultado, as conexões sem fio são de menor qualidade do que as conexões com fio: menores larguras de banda, maiores taxas de erro e desconexões mais frequentes [24]. Assim, as principais limitações das redes sem fio estão descritas a seguir:

- Desconexão: a falha de conexão com a rede é uma grande preocupação no desenvolvimento de aplicações para dispositivos móveis. Para lidar com a perda completa da comunicação com a rede é possível que se invista nos recursos da rede para evitar que ocorra desconexão, ou que se invista em sistemas que lidem com as desconexões de forma mais natural e consiga continuar funcionando desconectados pelo maior tempo possível.
- Baixa largura de banda: o desenvolvedor de aplicações para dispositivos móveis deve se preocupar com o consumo de dados pois redes sem fio tendem a oferecer menor largura de banda do que redes cabeadas.
- Grande variação da largura de banda: as aplicações para dispositivos móveis podem enfrentar grandes variações na largura de banda, da ordem de 1 a 4 vezes a magnitude. A aplicação pode lidar com isso sempre assumindo redes de alta velocidade (não funcionando bem quando em redes de baixa velocidade), sempre assumindo redes de baixa velocidade (não aproveitando para aumentar seu desempenho quando em redes de alta velocidade) ou pode se adaptar às diferentes larguras de banda (aumentando a complexidade de desenvolvimento).

2.4 Sistema de Informação Geográfica com Participação Popular

O termo Sistema de Informação Geográfica com Participação Popular - SIGPP (*Public Participation Geographic Information System*) foi concebido para descrever como tecnologias SIG podem dar suporte à participação pública com o objetivo de incluir a população local ou marginalizada no processo de decisão e planejamento de suas áreas de vida [8]. SIGPP difere de SIGV no processo de coleta de dados que é propositado e direcionado à entidade governamental, ao invés de iniciado por cidadãos e voluntários [8].

De acordo com [55], a definição de *público* e a definição de *participação* em SIGPP são de importância fundamental para o entendimento do componente "*participação popular*" do SIGPP.

- Participação: concentra-se na motivação para se utilizar *participação* como uma política de abordagem ao problema. Deve se definir em cada projeto SIGPP ao menos dois elementos principais: qual o propósito maior que a participação no projeto visa atingir e quais as atividades específicas que os indivíduos irão desempenhar rumo a esse propósito;
- Público: como no domínio da participação, para que não haja problemas na implementação do SIGPP, ao menos dois elementos principais devem ser definidos: quem é o grupo de pessoas considerado o público e como serão escolhidos os tomadores de decisão no meio desse grupo. Em SIGPP a decisão de incluir o público não é suficiente. O tipo de público deve ser explicitamente definido baseado nos objetivos e resultados desejados no projeto.

O uso de SIGPP resulta em decisões estratégicas mais socialmente equitativas [8]. O aumento de aplicações SIG na internet criou um ambiente que deveria ser favorável à expansão do SIGPP, mas apesar dos avanços metodológicos de todos os SIG, SIGPP ainda não alcançou um impacto significativo nos resultados de planejamentos ambientais e regionais. Para que esse impacto aumente, as entidades governamentais devem se comprometer mais para envolver o público em processos de planejamento no que diz respeito ao SIGPP [8].

Estudos SIGPP foram implementados para diversas aplicações de planejamento regional e ambiental, incluindo planejamento ambiental para parques e florestas, planejamento para conservação regional, conservação de área marinha e costeira, planejamento de espaço urbano e aberto e para desenvolvimento de turismo. Nesses estudos, observa-se que os participantes de SIGPP tendem a ser usuários com maior familiaridade e experiência na área de planejamento [8]. Isso pode ser visto como um viés positivo porque conhecedores da área tendem a descrever com mais precisão as qualidades dos ambientes. Por exemplo, participantes SIGPP selecionados individualmente por estarem mais familiarizados com regiões de estudo na Nova Zelândia foram mais precisos em identificar vegetação nativa do que moradores selecionados aleatoriamente [8].

Críticos de SIGPP tendem a argumentar que as decisões sobre preservação de meio ambiente são geralmente para um escopo nacional, enquanto a maioria dos SIGPP implementados coletam informação de uma população regional [8]. Os moradores locais,

argumenta-se, são mais sujeitos a desenvolverem interesses no desenvolvimento da economia local e não considerarem na devida proporção a importância nacional da natureza local.

Os estudos SIGPP citados em [8] utilizaram residentes locais com a premissa de que essas pessoas estão mais familiares com o ambiente em questão e, discutivelmente, mais influenciadas pelos resultados do processo de planejamento. Consenso da população local sobre o resultado planejado parece uma condição necessária (porém insuficiente) caso deseje-se que o planejamento futuro não seja indeterminado. E ainda, é importante dar oportunidades de participação no processo para cidadãos não regionais e indivíduos não selecionados aleatoriamente. Um dos argumentos mais fortes em favor do SIGPP é que ele expande o processo de participação aos indivíduos e aos grupos que não participariam em outra situação.

A lenta adoção de métodos SIGPP por instituições governamentais para planejamento de políticas não parece um problema tecnológico mas um reflexo da falta de comprometimento do governo com participação pública e consulta de opinião no geral [8]. A falta de familiaridade com o SIGPP como uma nova metodologia de consulta e preocupações com a validação da informação vinda de fontes externas servem para reforçar a propensão à inércia da entidade governamental. A falta de métodos e modelos padronizados para coletar e integrar dados SIGPP nos processos de decisão aumentam a resistência à adoção do SIGPP.

2.5 Sistema de Informação Geográfica Livre

A variedade dos dados geográficos, vindo de cada vez mais fontes, é um grande desafio para os pesquisadores da área de geoprocessamento [61]. Lida-se hoje com dados produzidos por observação de campo, simulação de modelos situacionais, fotos e vídeos georreferenciados além de uma infinidade de mapas de regiões específicas criados por usuários no mundo inteiro.

A abordagem do SIG Livre - utilização de *software* livre para SIG - pode ser uma maneira de resolver o problema do processamento da grande quantidade de dados geográficos que estão disponíveis hoje. A acessibilidade e a maneira da produção de dados geográficos podem ser definidos através de padrões abertos. A definição de padrões abertos para acesso e produção de dados geográficos pode não ser o suficiente para resolver o problema do processamento dessa grande massa de dados, mas certamente é o primeiro passo [61].

Para lidar com tamanho volume de dados geográficos, os melhores *softwares* e os melhores *hardwares* devem ser utilizados [59]. A abordagem do SIG Livre lida com isso, pois algoritmos abertos podem ser melhorados em conjunto pelos usuários [59]. Parcerias como a *GeoCloud Initiative* que oferecem acesso à Amazon EC2 e instâncias de servidores geoespaciais comerciais e abertos são exemplos de iniciativas que visam melhorar o *hardware* utilizado para o processamento dos dados [61]. O SIG Livre aumenta a capacidade de resolver os problemas gerados com a grande quantidade de dados atuais [60].

Há diversos benefícios conhecidos no uso de *software* livre, tais como economia de custos, independência de licença comercial e utilização de padrões abertos [11, 59, 60]. A escolha das ferramentas utilizadas nesse trabalho foi feita visando utilizar, sempre que viável, tecnologias livres. A seguir o conceito de *software* livre é apresentado.

2.5.1 *Software* Livre

Software livre é aquele sem restrições para cópia, redistribuição, estudo e modificação [23, 64, 65]. A história do *software* livre pode ser dividida em 4 eras, descritas a seguir [3, 58].

Código livre

A primeira era dos programas de computador pode ser descrita como a era do código livre, quando a comunidade de cientistas programadores seguia a tendência de "*dar*" o seu código fonte, na esperança de que alguém pudesse melhorá-lo. Sempre foi um "*código de honra*" implícito na ciência de que todos os resultados das pesquisas e desenvolvimentos relacionados deveriam ser compartilhados com a comunidade científica para que pesquisas mais aprofundadas possam ser desenvolvidas. Ao final da década de 1950, os *softwares* científicos estavam em circulação livre [58].

A era do código customizado

As décadas de 1960 à 1980 foram marcadas por um período em que os *softwares* eram desenvolvidos apenas para o *hardware* onde eram instalados. Comerciantes de computadores como a IBM investiram um dinheiro considerável no desenvolvimento de programas para suas máquinas, mas eram programas únicos e necessários para aqueles equipamentos. No início da década de 1970 a ciência da computação estava ganhando espaço como disciplina acadêmica, com crescente experimentação de sistemas operacionais e linguagens de programação nos laboratórios de pesquisa [3, 58].

Ao final da década de 1970, o UNIX desempenhou um papel chave na quebra do monopólio dos *softwares* por parte das fabricantes de *hardware*. O UNIX é um sistema operacional portátil baseado na então recente linguagem de programação C, desenvolvido no Bell Labs e na época disponibilizado para uso sem custo em pesquisa acadêmica e como código aberto. A universidade UCB (*University of California at Berkeley*) se aproveitou do código aberto do UNIX para fazer grandes avanços e melhorias e chegou a distribuir livremente a versão BSD (*Berkeley Software Distribution*) do Unix juntamente com seu código fonte para incentivar melhorias e correções de bugs [3].

A era do pacote proprietário

As décadas de 1980 à 2000 foram marcadas pelo surgimento dos computadores pessoais e aumento da complexidade dos *softwares*. Diversas empresas especializadas em fabricação de *software* passaram a vender os programas como produtos finais e ao final da década de 80 os custos com a compra de *software* passou a dominar os custos de se possuir um computador pessoal [58].

A era atual

Em 1984 um grande marco do movimento do *software* livre ocorreu com a fundação da Fundação para o Software Livre (FSF) por Richard Stallman. Quando trabalhava como programador no Laboratório de Inteligência Artificial do Instituto de Tecnologia de Massachusetts (*AI Lab of MIT*) Stallman teve problemas com a ineficiência do software de uma impressora e foi impossibilitado de resolver o problema pelo fato do *software* do equipamento ser proprietário. Com o propósito de garantir

ao usuário final a capacidade de alterar de acordo com sua necessidade, rodar o programa livremente e redistribuir cópias sejam pagas ou gratuitas, Richard criou a Fundação para o Software Livre (FSF). Vale ressaltar que o propósito da Fundação para o *Software* Livre não é garantir que se entregue o *software* sem custo ao usuário final, mas entregar o *software* de uma maneira que o usuário possa usá-lo livremente depois de obtê-lo. Nesse sentido, é importante não confundir *software* livre com *software* gratuito [3, 58].

Richard encorajou mais desenvolvedores à criarem *softwares* com código aberto ao mesmo tempo em que ele próprio trabalhava para criar uma plataforma completa de código aberto, incluindo sistema operacional, ferramentas, compiladores, etc. Dois grandes destaques do movimento do *software* livre são o Linux e o Open Office, dois projetos que criaram alternativas de código livre para o núcleo de um sistema operacional e para as aplicações do pacote Microsoft Office respectivamente.

As décadas de 1990, 2000 e até os dias de hoje o código aberto já se tornou uma realidade inclusive para grandes empresas de tecnologia. A popularidade de bases comerciais com PostgreSQL e o domínio de servidores web Apache mostram que os grandes negócios já estão incorporando o *software* livre [58].

No que se refere ao desenvolvimento de SIG, destacam-se algumas iniciativas como o GeoServer, o PostGIS e o OpenStreetMap assim como várias bibliotecas livres para desenvolvimento tanto de SIG web quanto de SIG móvel. O GeoServer é um servidor para compartilhamento de dados espaciais que permite grande flexibilidade na criação e acesso à esses dados [27]. O PostGIS é uma extensão espacial do SGBD PostgreSQL que permite o armazenamento e a manipulação de dados geográficos [31, 53]. O OpenStreetMap é um projeto de mapeamento global com o objetivo de criar um conjunto de mapas que são livres para usar, editar e licenciados sob novos esquemas de direitos autorais. Uma motivação chave para o projeto OpenStreetMap é prover acesso gratuito à informação geográfica [33, 50].

No próximo Capítulo será apresentada a arquitetura proposta. Para melhor exposição da solução algumas tecnologias serão abordadas em detalhes.

Capítulo 3

Arquitetura Proposta

Este capítulo apresenta a arquitetura proposta para o problema motivador deste projeto. O objetivo é obter opiniões dos usuários de serviços públicos através de um sistema capaz de georreferenciar esses dados, e exibi-los ao gestor desses serviços em um mapa. Inicialmente, na Seção 3.1, o cenário de execução do projeto é apresentado. Em seguida, a Seção 3.2 detalha todos os pontos da arquitetura do sistema.

3.1 Cenário de Execução

A arquitetura proposta funciona em torno de um banco de dados central responsável por armazenar informações sobre os estabelecimentos públicos que serão avaliados por seus usuários. As opiniões dos usuários também serão cadastradas nesse banco para serem, posteriormente, obtidas pelo gestor. A visão geral do funcionamento está na Figura 3.1.



Figura 3.1: Ambiente de Execução.

Pode-se observar na Figura 3.1 que os serviços públicos serão avaliados por usuários através de seus dispositivos móveis. O *smartphone* deverá ter o aplicativo responsável por

receber as informações do usuário e transmiti-las para o banco de dados central. Essas informações serão então visualizadas por um gestor, através de um navegador web, de uma maneira que possa auxiliá-lo em suas decisões para melhorar os serviços públicos.

O cenário de execução é um sistema de informação geográfica. As informações cadastradas pelos usuários são associadas à sua posição geográfica e são visualizadas em um mapa. Para isso, os usuários devem ir até o local que eles desejam avaliar de posse de seu *smartphone*. O GPS do *smartphone* localiza onde o usuário está e autoriza que ele avalie apenas o estabelecimento em que se encontra. As avaliações de cada um dos usuários são enviadas para o banco de dados central.

A Figura 3.2 apresenta os passos envolvidos nesse processo. Os passos 1 e 2 consistem da solicitação do usuário para obter os estabelecimentos cadastrados no banco de dados central. Os passos 3 e 4 consistem da obtenção da localização do usuário através do GPS do *smartphone*. Os passos 5 e 6 consistem na obtenção das perguntas cadastradas no banco para o estabelecimento que o usuário selecionou. O passo 7 consiste no envio da avaliação feita pelo usuário para o banco de dados central. Por fim, o passo 8 consiste da visualização pelo gestor das avaliações cadastradas pelos usuários.



Figura 3.2: Funcionamento da Arquitetura.

O gestor visualiza essas informações em uma interface capaz de apontar visualmente quais estabelecimentos precisam de mais atenção imediata. Essa interface apresenta um mapa com todos os estabelecimentos avaliados, e cores sobre esses estabelecimentos. Quanto menores forem as notas distribuídas ao estabelecimento, mais vermelhos os estabelecimentos estão. Da mesma maneira, quanto maiores forem as notas, mais verdes os estabelecimentos estão. De posse das informações enviadas pelos usuários e com o mapa à sua disposição, o gestor pode tomar decisões que irão impactar de forma mais positiva os usuários dos serviços públicos.

3.2 Detalhamento da Arquitetura

A arquitetura abstrata proposta está na Figura 3.3. A interface móvel lida com a obtenção da opinião dos usuários dos serviços públicos. A interface web lida com a

apresentação dos dados ao gestor. Há dados que só são mostrados na interface web. A comunicação de dados é a parte da arquitetura responsável por fornecer os dados necessários para o funcionamento das duas interfaces citadas no formato requisitado. A camada de dados deverá armazenar as opiniões dos usuários dos serviços públicos. O sistema gerenciador de banco de dados cuida da consistência e manipulação dos dados da camada de dados. Nas próximas seções, todos os itens da Figura 3.3 são descritos em detalhes.

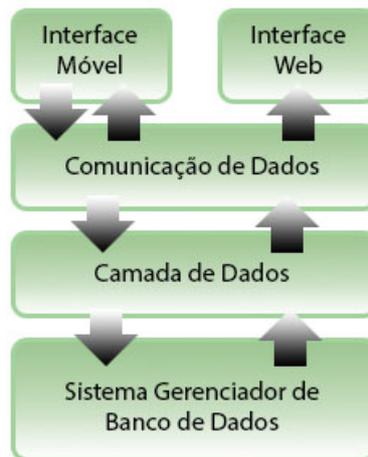


Figura 3.3: Abstração da Arquitetura.

3.2.1 Interface Móvel

A interface móvel é a parte do sistema que irá funcionar no *smartphone* do usuário. Consiste de um aplicativo que deve apresentar um mapa com os estabelecimentos cadastrados para serem avaliados. O usuário irá cadastrar sua opinião, e essa informação irá para a camada de dados (Figura 3.3). O fluxograma do aplicativo está na Figura 3.4. O fluxograma consiste de 3 eventos principais, descritos a seguir.

Autenticação do usuário

Na Figura 3.4, o passo número 1 consiste do usuário abrir o aplicativo em seu *smartphone*. O passo número 2 é a primeira tela que será exibida ao usuário. Nessa tela o usuário deverá fazer *login* ou criar uma conta. Quando o usuário digitar o *login* e a senha o aplicativo irá para o passo 3. Nessa etapa, o aplicativo irá fazer requisições HTTP à comunicação de dados (Figura 3.3) para saber se o usuário está cadastrado no sistema. Uma requisição HTTP é uma solicitação de um determinado arquivo ou página através do protocolo HTTP (*Hypertext Transfer Protocol*) [6]. Detalhes sobre essa requisição estão no Capítulo 4.

Caso o dado obtido da comunicação de dados indique que o usuário não existe, o aplicativo voltará ao passo 2 e o usuário poderá selecionar a opção de cadastrar novo usuário. Caso o usuário exista, o aplicativo passará para o passo 4.

Escolha do estabelecimento

O passo 4 do fluxo do aplicativo (Figura 3.4) consiste em apresentar ao usuário um

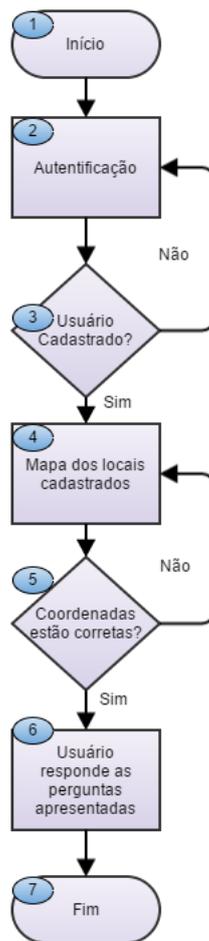


Figura 3.4: Fluxograma da Interface Móvel.

mapa com os ícones dos locais cadastrados. Para isso, o aplicativo fará mais uma requisição HTTP à comunicação de dados (Figura 3.3) para obter a latitude e a longitude dos locais cadastrados no banco de dados, além do nome e do tipo do local (se é uma escola, um hospital, etc.). O usuário irá então clicar no ícone do estabelecimento que ele deseja avaliar.

Ao clicar no ícone do estabelecimento que deseja avaliar, o usuário irá para o passo 5 do fluxo, onde será decidido se ele pode ou não avaliar aquele local. Essa decisão é feita utilizando-se as coordenadas do usuário e as coordenadas do local selecionado. As coordenadas do usuário são obtidas através do GPS do *smartphone* utilizado, e as coordenadas do estabelecimento são obtidas no banco de dados central. Um cálculo é feito para decidir se o usuário está a menos de 100 metros do local que ele selecionou. Caso ele esteja a 100 metros ou mais de distância, ele não será autorizado à avaliar o local e voltará ao passo 4 para poder selecionar outro estabelecimento. Caso ele esteja a menos de 100 metros, ele irá para o passo 6.

Avaliação do estabelecimento

No passo do fluxo do aplicativo (Figura 3.4) será apresentado ao usuário um formulário com perguntas sobre o local que ele selecionou no passo 4. Para isso, o aplicativo

irá fazer outra requisição HTTP à comunicação de dados (Figura 3.3) para obter as perguntas cadastradas no banco de dados para aquele estabelecimento. Por fim, após responder as perguntas o usuário deverá clicar em um botão *Enviar* para ir ao passo 7, onde a avaliação será enviada à comunicação de dados (Figura 3.3) por meio de mais uma requisição HTTP. Todas as requisições HTTP estão descritas na Seção 4.1.4 do Capítulo 4.

3.2.2 Interface Web

A interface web será acessada pelo gestor dos serviços públicos. Esta página web deve apresentar um mapa com os estabelecimentos cadastrados e as informações sobre as avaliações dos usuários desses locais. O fluxograma da interface está representado na Figura 3.5.

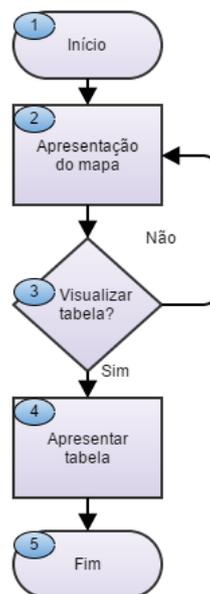


Figura 3.5: Fluxograma da Interface Web.

O passo número 1 do fluxograma (Figura 3.5) é a abertura da página web pelo gestor. No passo número 2 será mostrado para ele um mapa com os ícones dos estabelecimentos cadastrados. Utilizando-se a média das notas de todas as opiniões cadastradas para os estabelecimentos, cada ícone irá aparecer com uma cor diferente, vermelho quando as notas estiverem abaixo da média e verde quando as notas estiverem acima da média. Para isso, a interface web deverá fazer uma requisição HTTP à comunicação de dados (Figura 3.3) para obter dois dados: os estabelecimentos cadastrados (para poderem ser apresentados no mapa) e as opiniões cadastradas pelos usuários para esses estabelecimentos.

O passo 3 acontece quando o gestor clica em algum dos ícones dos estabelecimentos. Em um *popup* será apresentado o valor da média das notas para aquele lugar e a opção de visualizar a tabela com todas as notas. Caso o gestor feche o *popup*, ele voltará ao passo 2 e poderá selecionar outro local. Caso ele clique na opção *visualizar todas as notas*, ele será direcionado para outra página web contendo uma tabela com todas as avaliações

cadastradas para aquele local. O passo 5 é o término da análise dos dados por parte do gestor.

3.2.3 Comunicação de Dados

A comunicação de dados tem como funcionalidade responder as requisições da interface móvel e da interface web. Para isso, na camada de dados é realizada a busca das informações solicitadas para que elas possam ser entregues da maneira desejada pelas interfaces. Para cada interface, o dado a ser entregue é diferente.

Dados entregues à interface web

Para a interface web, os dados entregues pela comunicação de dados são:

- Estabelecimentos a serem avaliados: no passo 2 do fluxograma (Figura 3.5) a interface web necessita da latitude e da longitude dos estabelecimentos cadastrados no banco de dados para poder colocá-los no mapa. Além disso, o tipo de estabelecimento (escola, hospital, etc.) deve ser conhecido para poder escolher qual ícone deve representá-lo;
- Avaliações: no passo 2 do fluxograma (Figura 3.5) a interface web utiliza a média das notas que os usuários avaliaram para cada estabelecimento. E ainda, no passo 4 do fluxograma (Figura 3.5) a interface web recebe todas as avaliações cadastradas pelos usuários.
- Usuários: no passo 4 do fluxograma (Figura 3.5) a interface web precisa dos usuários que cadastraram as opiniões.

Dados entregues à interface móvel

Para a interface móvel, os dados entregues pela comunicação de dados são:

- Usuário: no passo 3 do fluxograma (Figura 3.4) a interface móvel precisa saber se o usuário está cadastrado no banco de dados para poder seguir para a etapa seguinte;
- Estabelecimentos: no passo 4 do fluxograma (Figura 3.4) a interface móvel utiliza a latitude e a longitude dos estabelecimentos cadastrados no banco de dados para poder colocá-los no mapa apresentado ao usuário. Além disso, precisa do tipo de estabelecimento (escola, hospital, etc.) para poder escolher qual ícone deve representá-lo;
- Perguntas: no passo 6 do fluxograma (Figura 3.4) a interface móvel tem as perguntas que estão cadastradas para aquele estabelecimento, para apresentá-las ao usuário.

Dados obtidos da interface móvel

Para a interface móvel, os dados obtidos pela comunicação de dados são:

- Usuário: no passo 3 do fluxograma (Figura 3.4) a interface móvel tem a opção de preencher um formulário de cadastro e enviar os dados do usuário que será incluído no banco de dados;

- Avaliações: no passo 6 do fluxograma (Figura 3.5) a interface móvel deve enviar os dados do usuário juntamente com as respostas da sua avaliação.

Para entregar os dados citados, a comunicação de dados os busca, na camada de dados, e responde às requisições HTTP, descritas na Seção 4.1.4.

3.2.4 Camada de Dados

A camada de dados armazena os dados que serão acessados pela comunicação de dados. Ela deve ser capaz de armazenar 4 itens:

- Estabelecimentos que podem ser avaliados: os nomes dos locais cadastrados, suas latitudes e longitudes e os seus tipos;
- Usuários: usuário e senha para verificar se eles podem cadastrar opinião;
- Perguntas: apresentadas ao usuário no passo 6 do fluxo, apresentado na Figura 3.4;
- Avaliações: as respostas cadastradas pelos usuários para as perguntas.

O modelo do banco de dados proposto está na Figura 3.6. Há três tabelas principais e duas tabelas auxiliares. As entidades principais são: *Usuario*, *Estabelecimento* e *Avaliacao*. As entidades auxiliares são: *tipoEstabelecimento* e *tipoAvaliacao*.

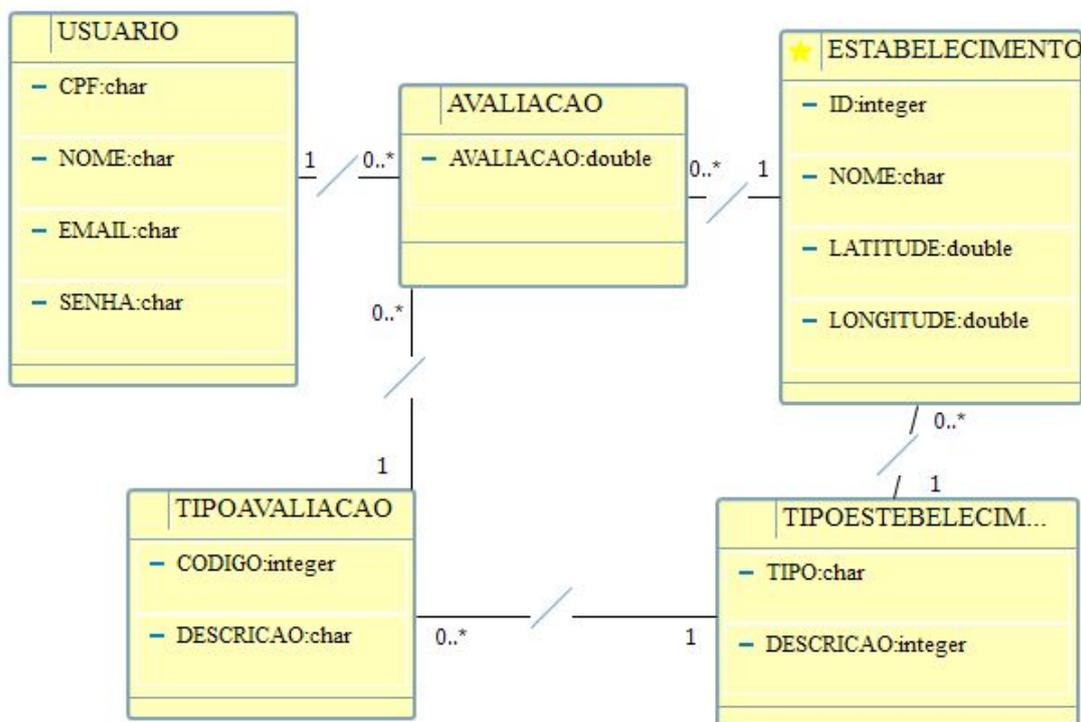


Figura 3.6: Modelo do Banco de Dados Central.

A entidade *Usuario* armazena os dados sobre o usuário no ato do cadastro: CPF, nome, e-mail e senha de acesso. A entidade *Estabelecimento* armazena os dados sobre

os locais que serão avaliados: nome, latitude e longitude. Também armazena o tipo de local, isto é, se é escola, hospital, posto de saúde, etc. A entidade *Avaliacao* armazena a avaliação do usuário sobre o estabelecimento, armazenando a nota para cada pergunta e referenciando o identificador único do estabelecimento e o CPF do usuário.

A entidade *tipoEstabelecimento* se relaciona com a entidade *Estabelecimento* e é responsável por associar o estabelecimento com a descrição de seu tipo (se é escola, hospital, etc.). A entidade *tipoAvaliação* armazena as perguntas e o tipo de estabelecimento a que aquela pergunta se refere. Essa entidade se relaciona com a entidade *tipoEstabelecimento* para indicar o tipo de estabelecimento que aquela pergunta se refere. A entidade *tipoAvaliacao* também se relaciona com a entidade *Avaliacao* para indicar para qual pergunta as avaliações armazenadas se referem.

3.2.5 Sistema Gerenciador de Banco de Dados

O sistema gerenciador de banco de dados - SGBD - dá suporte à camada de dados, mantendo os dados sempre consistentes fisicamente no servidor. O SGBD é responsável por gerenciar o acesso, a manipulação e a organização dos dados. O acesso requisitado pela camada de dados está descrito na Figura 3.7.

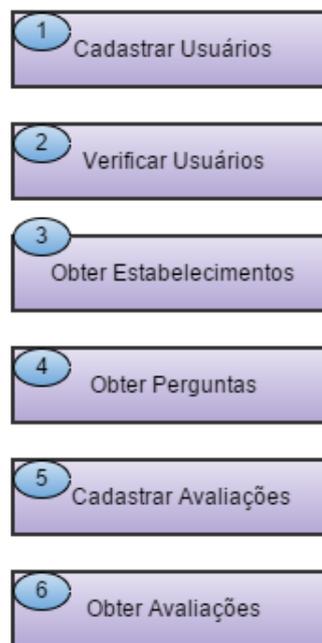


Figura 3.7: Solicitações ao Sistema Gerenciador de Banco de Dados.

Como pode ser observado na Figura 3.7, o SGBD mantém a consistência do banco no momento de cadastrar usuários e avaliações (itens 1 e 5). As consultas ao banco são para verificar se o usuário e a senha existem (item 2), quais os dados dos estabelecimentos cadastrados (item 3), quais as perguntas cadastradas (item 4) e quais foram as avaliações cadastradas pelos usuários (item 6).

O cadastro de usuário é feito através da interface móvel. O próprio usuário deve preencher seus dados com CPF, nome, email, senha e solicitar o cadastro no banco de

dados central. O modelo de dados implementado no SGBD não permite dois usuários ou mais com CPF iguais.

Além disso, o modelo de dados implementado no SGBD não permite que dois estabelecimentos possuam o mesmo ID, já que esse número deve ser um identificado único para as duas interfaces saberem qual o estabelecimento que elas estão lidando.

As perguntas cadastradas no banco possuem o campo CodAvaliacao que o SGBD garante que seja único para cada pergunta. O sistema gerenciador de banco de dados também é responsável por certificar que o campo Tipo na entidade tipoEstabelecimento permaneça único.

Assim, este capítulo especificou a arquitetura para um sistema de coleta de opinião de usuários de serviços públicos. No Capítulo 4 é apresentada a implementação de um estudo de caso baseado na arquitetura proposta, assim como também são discutidos os resultados.

Capítulo 4

Estudo de Caso

O objetivo deste estudo de caso é implementar o sistema utilizando a arquitetura proposta para avaliar sua viabilidade. Portanto, todos os requisitos funcionais do sistema foram desenvolvidos e estão documentados neste capítulo. Primeiramente, na Seção 4.1, são apresentados os detalhes da implementação do sistema de acordo com a arquitetura proposta. Em seguida, na Seção 4.2, são discutidos os testes realizados no sistema desenvolvido.

4.1 Visão Geral

Os módulos do sistema discutidos no Capítulo 3 e ilustrados na Figura 3.3 são detalhados nesta seção. Para implementar a interface móvel foi desenvolvido um aplicativo para *smartphone* com o sistema operacional Android. O Android é um sistema operacional móvel baseado no núcleo Linux e, atualmente, desenvolvido pela empresa de tecnologia Google [20].

Para implementar a interface web foi desenvolvida uma página web em PHP. O PHP é uma linguagem interpretada livre, usada para o desenvolvimento de aplicações no lado do servidor, capazes de gerar conteúdo dinâmico [15].

Para implementar a comunicação de dados e a camada de dados (Figura 3.3) foram desenvolvidos algoritmos capazes de lidar com as requisições discutidas no Capítulo 3. Os algoritmos estão descritos na Seção 4.1.4. O *script* para criação do banco está no Apêndice A.1.

Para o sistema gerenciador de banco de dados foi utilizado o PostgreSQL. O PostgreSQL é um sistema gerenciador de banco de dados objeto relacional gratuito (SGB-DOR), desenvolvido como projeto de código aberto [48].

As próximas seções descrevem detalhadamente cada passo do desenvolvimento dos módulos da Figura 3.3.

4.1.1 ConsultaOpiniao

O aplicativo para Android foi chamado de *ConsultaOpiniao*. Para desenvolvê-lo foi utilizado o ambiente integrado de desenvolvimento Eclipse, juntamente com o *android development tools*, um *plugin* que estende as capacidades do Eclipse para lidar rapidamente com criação de projetos de aplicativos para Android [36]. O ambiente integrado

de desenvolvimento Eclipse é uma ferramenta para desenvolvimento Java, porém suporta várias outras linguagens a partir de *plugins* como C/C++, PHP, ColdFusion, Python, Scala e plataforma Android. O Eclipse foi feito em Java e segue o modelo *open source* de desenvolvimento de software [25].

O fluxograma de funcionamento do aplicativo proposto neste trabalho foi implementado. A tela inicial do aplicativo, juntamente com seu lugar no fluxo, estão na Figura 4.1.

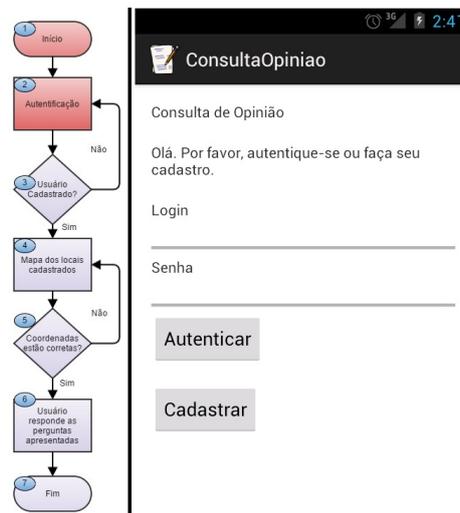


Figura 4.1: Fluxograma do Aplicativo - Autenticação.

Como no capítulo anterior, o passo 1 do fluxograma da interface móvel consiste na abertura do aplicativo pelo usuário em seu *smartphone*. A tela inicial mostrada na Figura 4.1 exige que o usuário preencha os dados de *login* e senha e clique no botão *Autenticar*. Nesse momento, o aplicativo irá requisitar à comunicação de dados que verifique se o usuário existe ou não. Caso o usuário exista, irá seguir para o passo 4 do fluxo. Caso o usuário não exista, irá voltar para o passo 2 do fluxograma. Detalhes sobre a implementação do algoritmo responsável por fazer essa verificação estão na Seção 4.1.4.

Após garantir que o usuário foi autenticado, o aplicativo irá para a tela mostrada na Figura 4.2. Como pode-se observar, o aplicativo apresenta um mapa com ícones que representam os estabelecimentos cadastrados na camada de dados. Há ainda um ícone indicando a localização atual do usuário.

O usuário deve clicar no local que deseja avaliar. Ao identificar o clique no local, o aplicativo irá decidir se o usuário está perto o suficiente do local selecionado para poder avaliá-lo (passo 5 do fluxograma). Em caso negativo, irá apresentar uma mensagem ao usuário explicando que ele não está perto o suficiente do local que deseja avaliar, e voltará ao passo 4 do fluxograma, apresentando novamente a tela da Figura 3.3 para que o usuário possa escolher outro local. Caso o usuário não esteja em nenhum local cadastrado no banco de dados, ele não será capaz de seguir no fluxo.

O aplicativo só seguirá para o passo 6 do fluxo quando o local selecionado pelo usuário estiver a menos de 100 metros das coordenadas do GPS do *smartphone*. Nesse caso, um algoritmo irá cruzar as coordenadas do GPS do *smartphone* com as coordenadas do estabelecimento. Detalhes sobre a implementação do algoritmo responsável por fazer essa verificação estão na Seção 4.1.4.



Figura 4.2: Fluxograma do Aplicativo - Mapa.

O passo 6 do fluxograma consiste na avaliação do estabelecimento selecionado pelo usuário. A Figura 4.3 apresenta a tela que será mostrada ao usuário nessa etapa do fluxo. Como pode ser observado, é apresentada uma tela com o nome do aplicativo, a mensagem "Por favor, responda as perguntas abaixo" e todas as perguntas obtidas no banco. Abaixo do texto de cada pergunta há cinco estrelas. Essas estrelas representam a nota que o usuário deseja cadastrar para a pergunta.

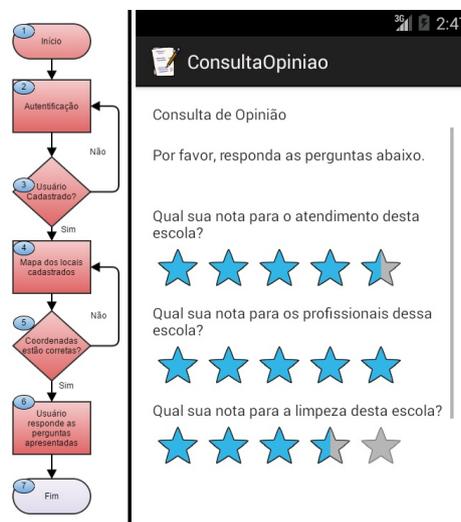


Figura 4.3: Fluxograma do Aplicativo - Perguntas.

Após selecionar notas para as perguntas, o usuário deve apertar o botão *Enviar* para que sua avaliação seja transmitida para a camada de dados. Nesse momento, uma requisição HTTP é feita para cadastrar a opinião do usuário. Nessa requisição as seguintes informações são cadastradas:

- CPF do usuário;

- Identificador do estabelecimento;
- Identificador de cada uma das perguntas;
- Valor selecionado nas estrelas para cada uma das perguntas.

Os detalhes dessa requisição estão descritos na Seção 4.1.4.

4.1.2 Interface Web

Para a implementação da interface web foi criada uma página em PHP chamada *interfacegestor.php*, disponível no Apêndice B.1, que pode ser visualizada na Figura 4.4. Utilizou-se o editor Sublime Text 2 para escrever o *script* PHP [37]. Para o mapa que é exibido nessa página web foi utilizado o Leaflet, uma biblioteca JavaScript de código livre para mapas interativos [43].

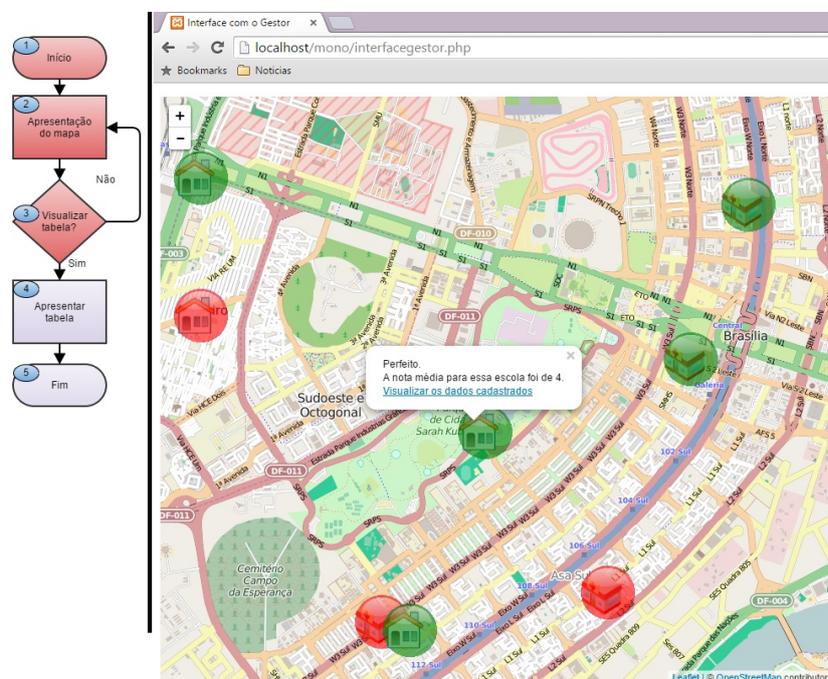


Figura 4.4: Fluxograma da Página Web - Mapa.

O fluxograma de funcionamento da página web proposto no capítulo anterior foi implementado. O primeiro passo do fluxograma consiste na abertura da página web *interfacegestor.php* e o segundo passo é a apresentação do mapa com todos os estabelecimentos cadastrados.

Em cada estabelecimento há um círculo que varia de cor, dependendo da média das avaliações dos usuários para esse local. Caso a média das avaliações seja menor ou igual a 2,5 o círculo é vermelho. Caso essa média seja maior do que 2,5 o círculo é verde. O objetivo desse *feedback* visual é auxiliar o gestor na visualização dos estabelecimentos com maior necessidade de intervenção. A média pode variar de 0 a 5, e o valor 2,5 foi escolhido no estudo de caso para delimitar avaliações boas ou ruins. Em trabalhos

futuros é possível fazer uma análise mais detalhada, atribuindo diferentes tons de cores para diferentes médias.

O terceiro passo do fluxograma acontece ao clicar sobre algum círculo, o que faz com que um *popup* seja exibido. O *popup* pode ser visualizado na Figura 4.4. No *popup* é possível visualizar a nota média para o estabelecimento e um *link* para visualizar os dados completos daquele estabelecimento. Caso o gestor clique no *link* uma nova página é aberta, e então a interface web chega ao passo 4 do fluxograma. Na Figura 4.5 é possível visualizar a tabela que é apresentada no passo 4 do fluxo, após o gestor clicar no *link* da Figura 4.4.



Figura 4.5: Fluxograma da Página Web - Tabela.

Como pode ser observado, o gestor é capaz de visualizar todas as avaliações feitas pelos usuários desse local. A tabela apresenta o nome do estabelecimento que foi avaliado, o nome do usuário que cadastrou a avaliação, o quesito que ele avaliou e a nota que ele cadastrou para o quesito em questão. Em trabalhos futuros é possível adicionar a possibilidade de não exibir os nomes dos usuários que cadastraram as opiniões, caso a visualização destes nomes não seja pertinente ao gestor.

4.1.3 Banco de Dados Central

O banco de dados central é o elemento fundamental para o funcionamento da arquitetura. O PostgreSQL foi utilizado para manter a consistência do banco. Para armazenar todas as informações necessárias o modelo apresentado na Figura 4.6 foi implementado. Esse modelo foi desenvolvido no pgModeler. O pgModeler é uma ferramenta de código aberto voltada para a modelagem de banco de dados PostgreSQL [52]. O *script* gerado pelo pgModeler para criação do banco está reproduzido no Apêndice A.1. Foram criadas 5 entidades:

- Usuario: armazena os dados do usuário, tais como CPF, nome, *e-mail* e a senha informada no momento do cadastro;

- Avaliacao: guarda as avaliações dos usuários. Para isso, armazena o cpf do usuário, o código da pergunta que ele respondeu, o id do estabelecimento que está avaliando e a nota para a pergunta que está sendo respondida;
- Estabelecimento: armazena os dados relativos aos estabelecimentos. Além de um código próprio armazena nome do local, latitude, longitude e um código para indicar o tipo do estabelecimento, que será buscado na tabela TipoEstabelecimento;
- TipoEstabelecimento: responsável por associar um código para cada tipo de estabelecimento (1 - Escola, 2 - Hospital, etc.);
- TipoAvaliacao: armazena as perguntas que serão exibidas ao usuário no momento da avaliação. Para cada pergunta associa também um código próprio e o tipo de estabelecimento ao qual a pergunta se refere.

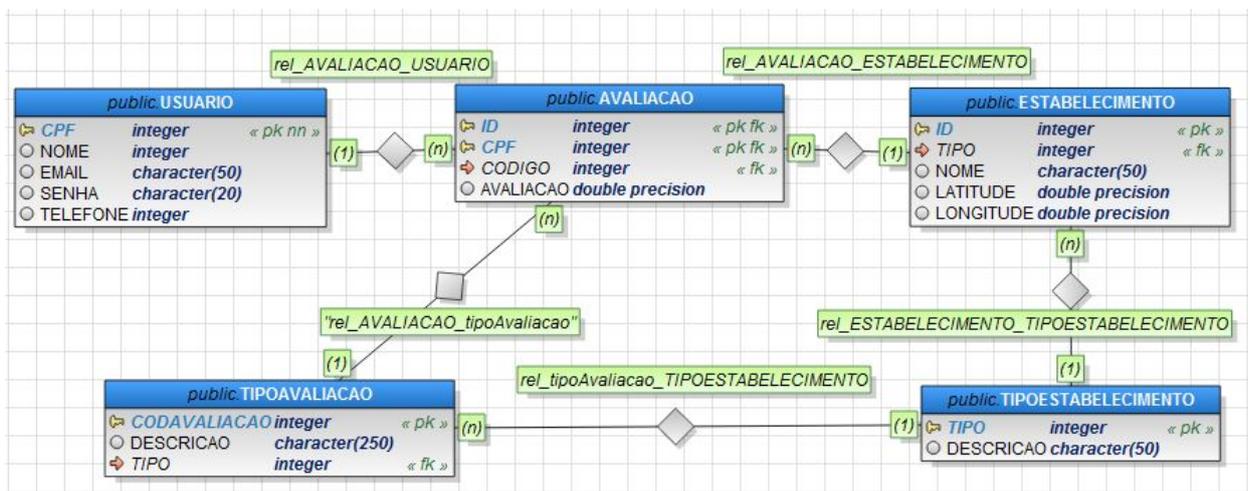


Figura 4.6: Modelo do Banco de Dados Central.

4.1.4 Algoritmos para Comunicação de Dados

A interface móvel e a interface web se comunicam com o banco de dados central através da comunicação de dados e da camada de dados. Os algoritmos utilizados para a comunicação de dados e a camada de dados estão no Apêndice B, e são descritos em detalhes a seguir.

Algoritmo da requisição da interface móvel para a validação do usuário

O passo número 3 do fluxograma de funcionamento do aplicativo (Figura 4.1) consiste na verificação da existência do usuário no banco de dados central. Para tal, o aplicativo *ConsultaOpiniaio* irá executar o *script* descrito no Apêndice B.2. O aplicativo faz uma requisição HTTP a uma página, e segue adiante caso a resposta seja 1. Caso a resposta seja 0 ele apresenta uma mensagem que indica que o usuário não foi encontrado.

A página que é requisitada está no Apêndice B.3. A página utiliza o usuário e a senha digitados, e busca por ocorrências no banco de dados central. Caso exista alguma ocorrência com aquele determinado usuário e senha, retorna o valor 1. Caso não exista, retorna o valor 0.

Algoritmo da requisição da interface móvel para obter os locais

O passo número 4 do fluxograma de funcionamento do aplicativo (Figura 4.2) consiste na apresentação do mapa ao usuário com os locais cadastrados. Para tal, o aplicativo *ConsultaOpinio* irá executar o *script* descrito no Apêndice B.4. O aplicativo faz uma requisição HTTP a uma página para obter a latitude, longitude, tipo, nome e identificador dos estabelecimentos, e utiliza esses dados para preencher o mapa que será apresentado ao usuário.

A página que é requisitada está no Apêndice B.5. A página consulta no banco de dados central, todos os estabelecimentos cadastrados e retorna a latitude, a longitude, o tipo, o nome e o identificador.

Algoritmo da interface móvel para verificar a localização do usuário

O passo número 5 do fluxograma de funcionamento do aplicativo (Figura 4.2) consiste na verificação das coordenadas do usuário. Para tal, o aplicativo *ConsultaOpinio* irá executar o *script* descrito no Apêndice B.6. O aplicativo obtém as coordenadas do usuário através do GPS do *smartphone*, e compara com as coordenadas do local que ele deseja avaliar. Caso a distância do usuário para o estabelecimento seja menor do que 100 metros, o aplicativo permite que o usuário avalie aquele local. Caso contrário, apresenta a mensagem da Figura 4.7.



Figura 4.7: Interface Móvel - Validação da Localização.

Algoritmo da requisição da interface móvel para obter as perguntas

O passo número 6 do fluxograma de funcionamento do aplicativo (Figura 4.2) consiste na avaliação do local selecionado pelo usuário. Para tal, o aplicativo *ConsultaOpiniaio* irá executar o *script* descrito no Apêndice B.7. O aplicativo irá buscar as perguntas cadastradas para aquele estabelecimento por meio de uma requisição HTTP a uma página, e irá apresentá-las ao usuário.

A página requisitada está no Apêndice B.8. A página PHP consulta no banco de dados central todas as perguntas cadastradas para aquele tipo de estabelecimento e as retorna em uma lista.

Algoritmo do envio da opinião cadastrada pela interface móvel

O passo número 7 do fluxograma de funcionamento do aplicativo (Figura 4.3) consiste no envio da opinião do usuário. Para tal, o aplicativo *ConsultaOpiniaio* irá executar o *script* descrito no Apêndice B.9. O aplicativo irá enviar as informações cadastradas através de uma requisição HTTP a uma página.

A página requisitada está no Apêndice B.10. A página obtém os dados do usuário e os cadastra no banco de dados central.

Algoritmo da requisição da interface web para obter dados dos locais

O passo número 2 do fluxograma de funcionamento da interface web (Figura 4.4) consiste na apresentação do mapa ao gestor com os estabelecimentos cadastrados e com um indicativo visual das notas médias para esses estabelecimentos. Para tal, a página web irá executar o *script* descrito no Apêndice B.1. O *script* busca no banco de dados todos os estabelecimentos cadastrados e os adiciona ao mapa. Em seguida, busca no banco de dados as notas médias para cada estabelecimento e adiciona um círculo verde ou vermelho, dependendo da nota.

Algoritmo da requisição da interface web para obter todas as notas

O passo número 4 do fluxograma de funcionamento da interface web (Figura 4.5) consiste na apresentação de todas as avaliações cadastradas para o estabelecimento selecionado. Para tal, a página web irá executar o *script* descrito no Apêndice B.11. O *script* busca no banco de dados central todas as avaliações cadastradas para aquele estabelecimento e apresenta ao gestor uma tabela com o nome do estabelecimento avaliado, o nome do usuário que cadastrou a avaliação e a nota para cada pergunta cadastrada.

4.2 Testes Realizados

Para testar a arquitetura foram cadastrados dois tipos de estabelecimentos no banco de dados: escolas e hospitais. No total foram cadastradas seis escolas e cinco hospitais. As escolas cadastradas foram o Centro de Ensino Médio Asa Norte, Centro de Ensino

Médio Elefante Branco, Centro Educacional 01 do Cruzeiro, Centro Educacional 02 do Cruzeiro Novo, Centro de Ensino Fundamental 04 de Brasília e Centro de Ensino Medio Paulo Freire. As localizações geográficas dessas escolas estão na Tabela 4.1.

Tabela 4.1: Escolas Cadastradas no Banco de Dados Central.

Nome da Escola	[Latitude,Longitude]
Centro de Ensino Médio Asa Norte	[-15.767938,-47.874228]
Centro de Ensino Médio Elefante Branco	[-15.808065,-47.909539]
Centro Educacional 01 do Cruzeiro	[-15.7830155,-47.938376]
Centro Educacional 02 do Cruzeiro Novo	[-15.796561,-47.938454]
Centro de Ensino Fundamental 04 de Brasília	[-15.827537,-47.917211]
Centro de Ensino Medio Paulo Freire	[-15.75597,-47.87861]

Os hospitais cadastrados foram o Hospital Geral de Brasília, Hospital Regional da Asa Sul, Hospital de Base do Distrito Federal, Hospital Universitário de Brasília e Hospital Regional da Asa Norte. As localizações geográficas dos hospitais cadastrados estão na Tabela 4.2.

Tabela 4.2: Hospitais Cadastradas no Banco de Dados Central.

Nome do Hospital	[Latitude,Longitude]
Hospital Geral de Brasília	[-15.82669,-47.92002]
Hospital Regional da Asa Sul	[-15.82382,-47.897158]
Hospital de Base do Distrito Federal	[-15.800837,-47.888676]
Hospital Universitário de Brasília	[-15.772453,-47.873429]
Hospital Regional da Asa Norte	[-15.785671,-47.882864]

Para cada tipo de estabelecimento foram cadastradas cinco perguntas para serem respondidas pelos usuários, as quais foram:

- Qual a nota para o atendimento desta/deste escola/hospital?
- Qual a nota para os profissionais desta/deste escola/hospital?
- Qual a nota para a limpeza desta/deste escola/hospital?
- Qual a nota para a organização desta/deste escola/hospital?
- Qual a nota para a pontualidade dos profissionais desta/deste escola/hospital?

O *script* para cadastro de cada uma dessas informações está descrito no Apêndice A.2. Além disso, o banco de dados central ficou em uma máquina com um processador Intel Core i5 de 2,3GHz, 8GB 1333MHz DDR3 de memória RAM e o Mac OS X versão 10.9.5. Nesta máquina foi instalado o servidor local XAMPP para teste das requisições HTTP. O XAMPP é um servidor independente de plataforma, software livre, que consiste

principalmente no servidor web Apache e os interpretadores para linguagens de *script*: PHP e Perl [26].

Para testar a interface móvel o aplicativo *ConsultaOpiniao* foi instalado em um *smartphone* Motorola Moto G com o sistema operacional Android 4.4.4 KitKat, e em um *smartphone* Samsung Galaxy S5 com o sistema operacional Android 5.1 Lollipop. Com esses dispositivos em mãos, todos os estabelecimentos foram visitados e preenchidas algumas opiniões fictícias. Para enviar e receber os dados, o *smartphone* utilizou uma rede de dados 3G.

Para testar a interface web foram utilizados os navegadores Chrome e Mozilla Firefox na mesma máquina utilizada para armazenar o banco de dados central. A Figura 4.8 mostra duas telas da interface web, na parte superior da figura pode-se observar vários estabelecimentos com círculos verdes e alguns com círculos vermelho. Na parte de baixo da Figura 4.8 é possível observar a lista de avaliações cadastradas para a escola Centro de Ensino Medio Asa Norte. Todas as funcionalidades foram testadas com sucesso e estão descritas a seguir.

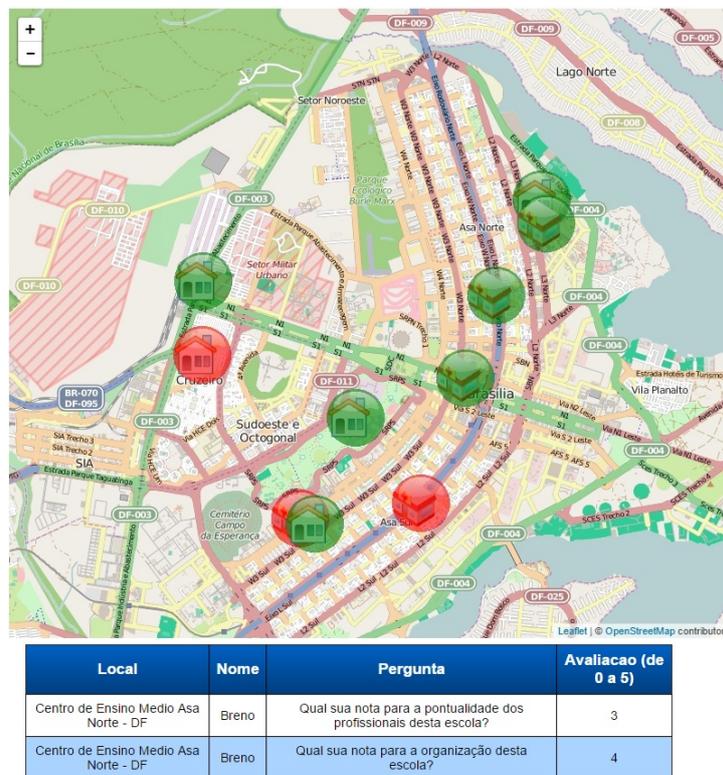


Figura 4.8: Interface Web - Mapa.

Cadastro e autenticação de usuários

O aplicativo *ConsultaOpiniao* permite cadastro e autenticação de usuários. A Figura 4.9 mostra as telas apresentadas ao usuário.

Da esquerda para direita na Figura 4.9, a primeira tela apresenta os campos para cadastro de novo usuário. Esses campos são:

- Nome;

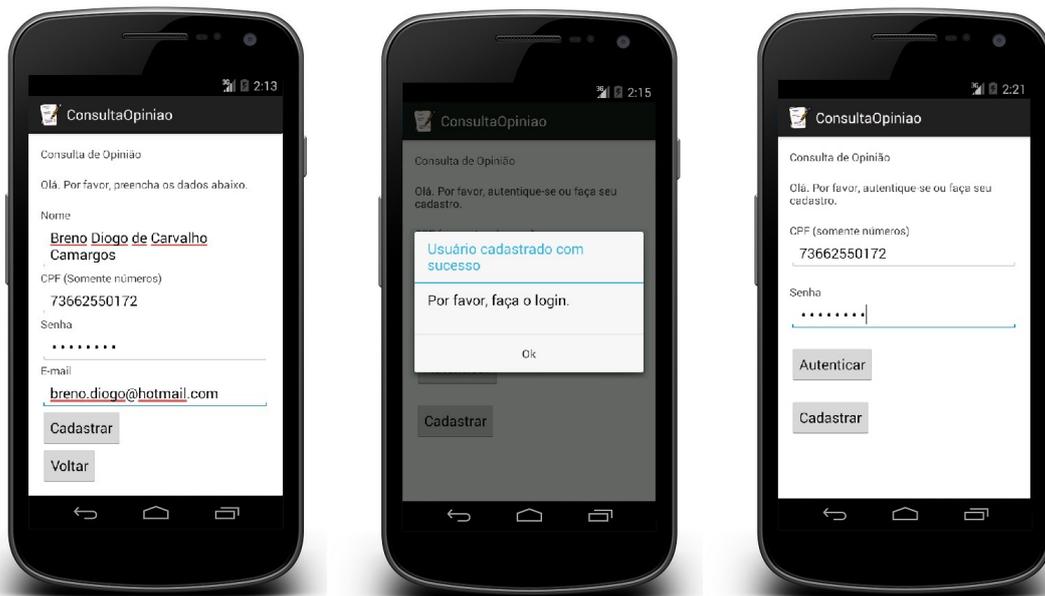


Figura 4.9: Funcionalidade de Cadastro e Autenticação.

- CPF (somente números);
- Senha;
- *E-mail*:

Há ainda dois botões: *Cadastrar* e *Voltar*. A tela central mostra a mensagem de confirmação do cadastro do usuário após este pressionar o botão *Cadastrar*: "Usuário cadastrado com sucesso. Por favor, faça o *login*". Após essa mensagem o usuário irá para a última tela da figura.

A tela mais à direita da figura apresenta o *login* do aplicativo, que solicita ao usuário os seguintes dados:

- CPF (somente números);
- Senha;

Além disso, há dois botões: *Autenticar* e *Cadastrar*.

Seleção do estabelecimento

Após autenticado, o usuário será apresentado a um mapa. Nesse mapa é possível visualizar sua posição atual e todos os estabelecimentos disponíveis para avaliação ao seu redor. O usuário deve então selecionar o local que deseja avaliar.

As coordenadas do GPS do *smartphone* são utilizadas para decidir se o usuário está no local selecionado. O aplicativo *ConsultaOpiniao* permite que apenas usuários

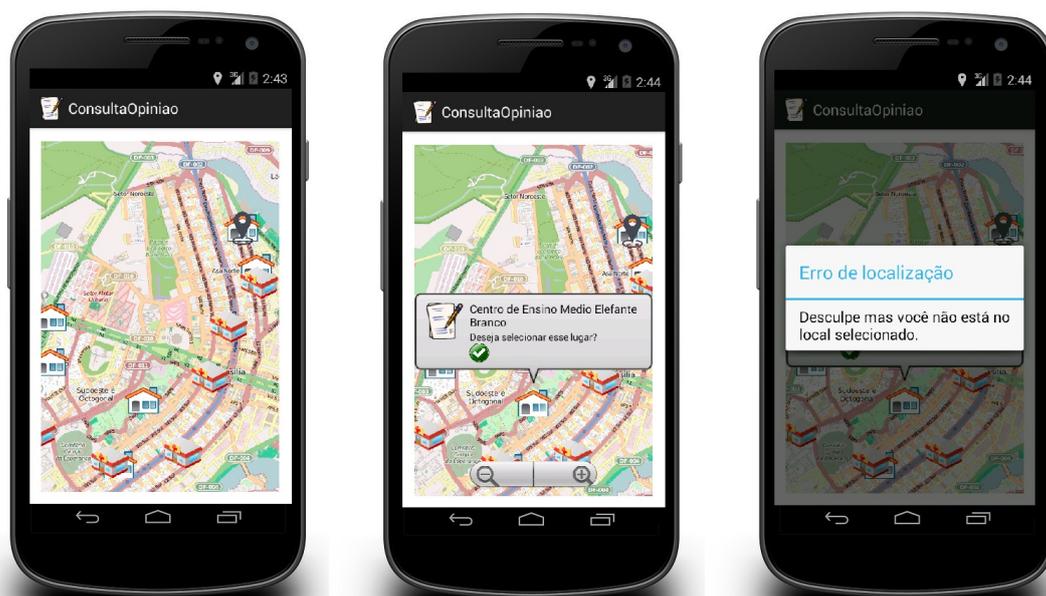


Figura 4.10: Funcionalidade de Seleção do Local.

que estejam geograficamente no estabelecimento selecionado possam avaliá-lo. A Figura 4.10 mostra as telas dessa funcionalidade.

Da esquerda para direita na Figura 4.10, a primeira tela apresenta o mapa que é mostrado ao usuário. Nela é possível visualizar todos os estabelecimentos cadastrados no banco de dados central. A segunda tela da Figura 4.10 mostra a seleção de um local em que o usuário não está. Nesse caso, uma mensagem deve informar que o usuário selecionou um local inválido.

Ainda na Figura 4.10, a terceira tela mostra a mensagem de erro apresentada, já que o usuário não se encontra no local que ele selecionou para avaliar: "Erro de localização. Desculpe mas você não está no local selecionado".

Caso o usuário selecione algum estabelecimento que esteja a menos de 100 metros da posição geográfica do GPS de seu *smartphone*, a próxima funcionalidade deverá ser apresentada: uma tela que permite o cadastro da opinião.

Cadastro de opinião

Caso o usuário esteja no local selecionado, o aplicativo *ConsultaOpinio* deve apresentar perguntas que serão respondidas pelo usuário. Essas perguntas são consultadas no banco de dados central, e foram cadastradas através do *script* que está no Apêndice A.2.

Para o estudo de caso realizado neste trabalho, o número máximo de perguntas que podem ser cadastradas são cinco. Essas perguntas podem ser alteradas no banco de dados central a qualquer momento.

A Figura 4.11 mostra as telas do aplicativo *ConsultaOpinio*, após buscar as perguntas no banco de dados central, e apresentá-las ao usuário.

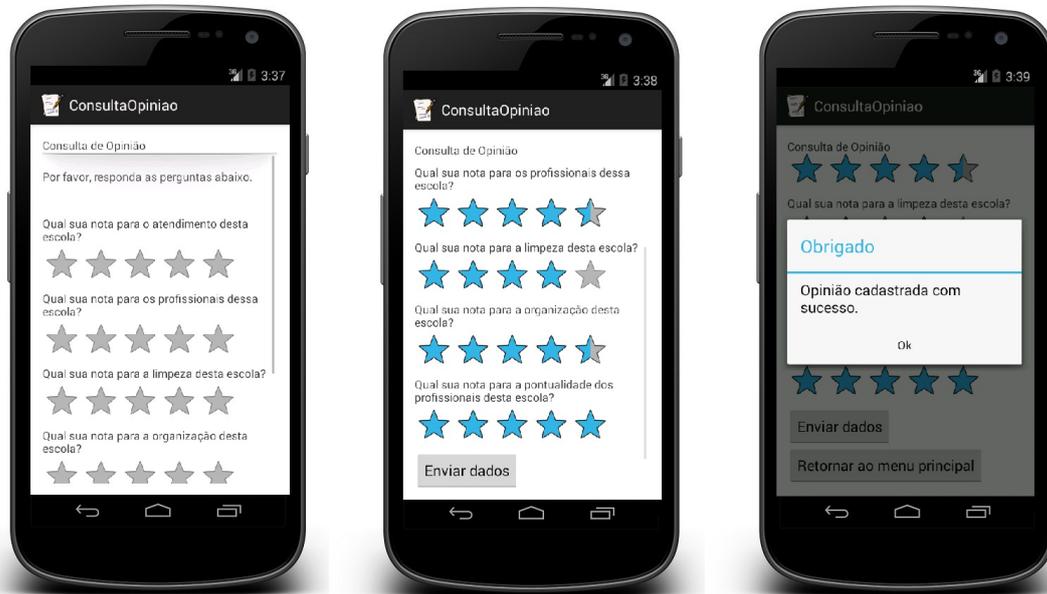


Figura 4.11: Funcionalidade de Cadastro de Opinião.

Da esquerda para direita na Figura 4.11, a primeira tela mostra as perguntas que são apresentadas ao usuário. Como pode ser observado, abaixo de cada pergunta há estrelas que indicam a nota que o usuário pode selecionar. Inicialmente, nenhuma estrela está selecionada.

Na segunda tela da Figura 4.11, as estrelas de cada pergunta estão preenchidas e é possível visualizar o botão *Enviar Dados*. Quando o usuário aperta esse botão um formulário é enviado para o banco de dados.

Por fim, na terceira tela da Figura 4.11, está a mensagem mostrada após clicar no botão *Enviar Dados*: "Obrigado. Opinião cadastrada com sucesso".

Visualização das opiniões em um mapa

O gestor deve ser capaz de visualizar um mapa com indicativos visuais de quais estabelecimentos estão com as melhores notas. A Figura 4.12 mostra essa funcionalidade.

Na tela da esquerda da Figura 4.12 é possível visualizar um mapa com os estabelecimentos cadastrados e círculos verdes e vermelhos sobre eles. Os círculos verdes indicam que o estabelecimento está com notas boas. O círculo vermelho indica que o local está com nota ruim. A tela da direita na Figura 4.12 mostra o *popup* que é exibido após clicar no estabelecimento com círculo vermelho.

No *popup* está a nota média cadastrada para aquela escola e um *link* para poder visualizar todas as opiniões cadastradas para aquele estabelecimento.



Figura 4.12: Funcionalidade de Visualização do Mapa pelo Gestor.

Visualização das opiniões em uma tabela

O gestor deve ser capaz de visualizar todas as opiniões cadastradas para o estabelecimento. A Figura 4.13 mostra essa funcionalidade.

Local	Nome	Pergunta	Avaliacao (de 0 a 5)
Centro de Ensino Fundamental 04 de Brasilia	Breno	Qual sua nota para o atendimento desta escola?	3
Centro de Ensino Fundamental 04 de Brasilia	Pedro	Qual sua nota para o atendimento desta escola?	0.5
Centro de Ensino Fundamental 04 de Brasilia	Pedro	Qual sua nota para os profissionais dessa escola?	5
Centro de Ensino Fundamental 04 de Brasilia	Pedro	Qual sua nota para a limpeza desta escola?	5
Centro de Ensino Fundamental 04 de Brasilia	Pedro	Qual sua nota para a organização desta escola?	0.5
Centro de Ensino Fundamental 04 de Brasilia	Pedro	Qual sua nota para a pontualidade dos profissionais desta escola?	5
Centro de Ensino Fundamental 04 de Brasilia	Joao	Qual sua nota para o atendimento desta escola?	0.5
Centro de Ensino Fundamental 04 de Brasilia	Joao	Qual sua nota para os profissionais dessa escola?	5
Centro de Ensino Fundamental 04 de Brasilia	Joao	Qual sua nota para a limpeza desta escola?	5
Centro de Ensino Fundamental 04 de Brasilia	Joao	Qual sua nota para a organização desta escola?	0
Centro de Ensino Fundamental 04 de Brasilia	Joao	Qual sua nota para a pontualidade dos profissionais desta escola?	5
Centro de Ensino Fundamental 04 de Brasilia	Tiago	Qual sua nota para o atendimento desta escola?	0.5
Centro de Ensino Fundamental 04 de Brasilia	Tiago	Qual sua nota para os profissionais dessa escola?	5
Centro de Ensino Fundamental 04 de Brasilia	Tiago	Qual sua nota para a limpeza desta escola?	4.5
Centro de Ensino Fundamental 04 de Brasilia	Tiago	Qual sua nota para a organização desta escola?	0.5
Centro de Ensino Fundamental 04 de Brasilia	Tiago	Qual sua nota para a pontualidade dos profissionais desta escola?	4
Centro de Ensino Fundamental 04 de Brasilia	Luis	Qual sua nota para o atendimento desta escola?	0.5
Centro de Ensino Fundamental 04 de Brasilia	Luis	Qual sua nota para os profissionais dessa escola?	1.5
Centro de Ensino Fundamental 04 de Brasilia	Luis	Qual sua nota para a limpeza desta escola?	0.5

Figura 4.13: Funcionalidade de Visualização da Tabela pelo Gestor.

A tela apresentada ao gestor consiste de uma tabela que traz o nome do local que está sendo avaliado, do usuário que cadastrou a avaliação, a pergunta respondida e o valor cadastrado para aquela pergunta. É possível extrair algumas informações do banco de dados, tais como:

- Quantos cadastros foram realizados;
- Qual estabelecimento recebeu o maior número de avaliações;
- Qual foi o estabelecimento com melhores avaliações;

- Qual foi o estabelecimento com piores avaliações:

A interface web pode ser evoluída em trabalhos futuros para ser capaz de visualizar esses e outros dados semelhantes.

Capítulo 5

Conclusão

Este documento apresenta uma arquitetura proposta para o desenvolvimento de sistemas de informação geográfica com participação popular, onde é possível coletar e exibir opiniões georreferenciadas de usuários sobre os serviços públicos.

Dessa forma, neste trabalho foi desenvolvido um sistema de informação geográfica com participação popular em que o objetivo é fornecer uma solução para aproximar gestores e usuários dos serviços públicos utilizando tecnologias livres. O número crescente de brasileiros donos de *smartphones* com acesso à internet, além da grande parcela desses dispositivos que utilizam o sistema operacional Android justificam sua escolha para utilização no trabalho.

O sistema possui como público alvo gestores e usuários de serviços públicos. Os usuários estão presentes no dia a dia dos estabelecimentos, e assim acumulam informações importantes sobre o local. Visando melhorar os pontos mais deficientes, os usuários informam os maiores problemas na prestação do serviço. Os gestores visualizam as opiniões fornecidas pelos usuários e decidem quais são os pontos que precisam de intervenção imediata. Para aumentar a confiabilidade das opiniões disponibilizadas ao gestor, são aceitas apenas opiniões cadastradas por usuários que estão fisicamente no local avaliado.

A arquitetura proposta para o sistema conta com um banco de dados central responsável por armazenar todas as opiniões cadastradas pelos usuários. O banco foi definido de forma a ser capaz de armazenar as informações sobre os usuários, os dados sobre os serviços públicos que podem ser avaliados e as avaliações cadastradas pelos usuários.

No lado do usuário, a arquitetura proposta possibilita que a localização geográfica deste seja obtida, e que ele seja autorizado a avaliar apenas o estabelecimento em que se encontra. A localização geográfica do usuário é obtida através do sistema de GPS de seu *smartphone*.

No lado do gestor, a arquitetura foi definida de forma a possibilitar que todos os estabelecimentos sejam exibidos em um mapa associados à cor verde ou à cor vermelha. A cor verde indica que aquele serviço público recebeu boas avaliações de seus usuários ao passo que a cor vermelha indica que as avaliações foram negativas. Também é possível visualizar em uma tabela as opiniões de todos os usuários para cada um dos estabelecimentos.

O estudo de caso foi capaz de verificar a viabilidade do sistema através de testes com dados fictícios. Os dados cadastrados podem ser visualizados corretamente através da interface web desenvolvida.

É importante que os usuários dos serviços públicos sejam incentivados à utilizar o sistema para gerar o maior número de informação possível para o gestor. Quanto mais informação estiver disponível ao gestor, mais embasadas poderão ser suas decisões.

5.1 Trabalhos Futuros

Alguns trabalhos futuros podem ser desenvolvidos, tanto para a interface do usuário móvel quanto para a interface do gestor.

A interface móvel possui um sistema de autenticação que pode ser integrado com redes sociais como o Facebook e o Google+. Com isso, o usuário teria mais facilidade para ser autenticado e continuar com o cadastro de sua opinião. Para obter um alcance ainda maior, o aplicativo pode ser desenvolvido para outros sistemas operacionais além do Android, tal como o iOS. O aplicativo também pode ser alterado para aceitar um número maior de perguntas, uma vez que o máximo possível são cinco. Outra evolução é a adição de outros tipos de estabelecimentos, uma vez que o estudo de caso foi desenvolvido apenas para escolas e hospitais.

A interface web pode apresentar gráficos construídos com base na opinião cadastrada para cada pergunta. Os gráficos são mais uma ferramenta para auxiliar as decisões dos gestores. É possível adicionar a funcionalidade de visualizar as opiniões cadastradas por períodos. Dessa maneira, o gestor pode verificar se as medidas tomadas estão surtindo efeito nas opiniões dos usuários. Também pode-se desenvolver uma interface capaz de cadastrar novos estabelecimentos e novas perguntas no banco de dados central.

Referências

- [1] Kathleen MS Allen, Stanton W Green, and Ezra BW Zubrow. *Interpreting space: GIS and archaeology*. Taylor & Francis London, 1990. 4
- [2] Henrique Antoun. *Web 2.0*. Mauad Editora Ltda, 2014. 1
- [3] Bill Appelbe. The future of open source software. *Journal of Research & Practice in Information Technology*, 35(4), 2003. 15, 16
- [4] Stan Aronoff. Geographic information systems: a management perspective. 1989. 4
- [5] Daniel Barbará. Mobile computing and databases-a survey. *Knowledge and Data Engineering, IEEE Transactions on*, 11(1):108–117, 1999. 11
- [6] Tim Berners-Lee, Roy Fielding, and Henrik Frystyk. Hypertext transfer protocol–http/1.0, 1996. 19
- [7] Maged NK Boulos, Steve Wheeler, Carlos Tavares, and Ray Jones. How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from ecaalyx. *Biomedical engineering online*, 10(1):24, 2011. 1
- [8] Greg Brown. Public participation gis (ppgis) for regional and environmental planning: Reflections on a decade of empirical research. *URISA journal*, 24(2):7–18, 2012. 4, 9, 13, 14
- [9] Peter A Burrough. Principles of geographical information systems for land resources assessment. 1986. 3
- [10] Gilberto CÂMARA. Modelos, linguagens e arquiteturas para bancos de dados geográficos. *São José dos Campos, SP: Instituto Nacional de Pesquisas Espaciais (INPE)*, 1995. 6, 7
- [11] Gilberto Câmara, Lúbia Vinhas, and Ricardo Cartaxo Modesto de Souza. Free and open source gis: will there ever be a geo-linux? In *Geospatial free and open source software in the 21st century*, pages 229–245. Springer, 2012. 14
- [12] Jonathan E Campbell and Michael Shin. Geographic information system basics. 2012. 5, 6
- [13] David J Coleman. Volunteered geographic information in spatial data infrastructure: an early look at opportunities and constraints. In *GSDI 12 World Conference*, 2010. 8, 9

- [14] David J Coleman, Yola Georgiadou, Jeff Labonte, et al. Volunteered geographic information: The nature and motivation of producers. *International Journal of Spatial Data Infrastructures Research*, 4(1):332–358, 2009. 8
- [15] Tim Converse, Joyce Park, and Clark Morgan. *PHP5 and MySQL bible*, volume 147. John Wiley & Sons, 2004. 26
- [16] Gustavo Leuzinger Coutinho. A era dos smartphones: um estudo exploratório sobre o uso dos smartphones no brasil. 2015. 1
- [17] David J Cowen. Gis versus cad versus dbms: What are the differences?. *Photogramm. Eng. Remote Sens.*, 54(11):1551–1555, 1988. 3
- [18] Clodoveu Davis and Gilberto Câmara. Bancos de dados geográficos. *MA Casanova, G. Câmara, CA Davis Jr, L. Vinhas and GR d. Queiroz. Curitiba (PR), Editora MundoGEO*, pages 379–395, 2004. 6, 7, 8
- [19] Henrique Pereira de Freitas Filho. *Arquitetura de Coleta de Dados para Pesquisas de Campo em Ambientes Computacionais Heterogêneos*. PhD thesis, Universidade de Brasília, 2014. 10, 11
- [20] Android Developers. What is android, 2011. 26
- [21] Department of Environment DoE. Handling geographic information. 1987. 4
- [22] Margaret H Dunham and Abdelsalam Helal. Mobile computing and databases: Anything new? *Acm Sigmod Record*, 24(4):5–9, 1995. 11
- [23] Brian Fitzgerald and Nic Suzor. Legal issues for the use of free and open source software in government. *Melb. UL Rev.*, 29:412, 2005. 15
- [24] George H. Forman and John Zahorjan. The challenges of mobile computing. *Computer*, 27(4):38–47, 1994. 10, 12
- [25] The Eclipse Foundation. *Eclipse IDE*. 2015. <http://eclipse.org/>. 27
- [26] Apache Friends. *XAMPP*. 2015. <https://www.apachefriends.org/>. 35
- [27] GeoServer. *GeoServer*. 2015. <http://www.geoserver.org/>. 16
- [28] Michael Goodchild. Neogeography and the nature of geographic expertise. *Journal of location based services*, 3(2):82–96, 2009. 8
- [29] Michael F . Goodchild. *Geographic Information Systems and Environmental Modeling*. 2015. <http://www.geog.ucsb.edu/~good/papers/134.pdf>. 3, 4
- [30] Michael F Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007. 1, 8, 9
- [31] The PostgreSQL Global Development Group. *PostgreSQL*. 2015. <http://www.postgresql.org/>. 8, 16

- [32] Mordechai Haklay and Carolina Tobón. Usability evaluation and ppgis: towards a user-centred design approach. *International Journal of Geographical Information Science*, 17(6):577–592, 2003. 1
- [33] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12–18, 2008. 16
- [34] Linda L Hill. *Georeferencing: The geographic associations of information*. Mit Press, 2009. 1
- [35] Tomasz Imielinski and BR Badrinath. Mobile wireless computing: challenges in data management. *Communications of the ACM*, 37(10):18–28, 1994. 11
- [36] Google Inc. *Google ADT Plugin*. 2015. <http://developer.android.com/tools/sdk/eclipse-adt.html>. 26
- [37] Sublime Inc. *Sublime Text*. 2015. <http://www.sublimetext.com/>. 29
- [38] INDE. *INDE*. 2015. http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2008/Decreto/D6666.htm. 6
- [39] Giani Carla Ito. *Bancos de Dados Móveis: Uma Análise de Soluções Propostas para Gerenciamento de Dados*. 2001. <https://repositorio.ufsc.br/bitstream/handle/123456789/79554/178707.pdf?sequence=1&isAllowed=y>. 10
- [40] HK Kadupitiya. Geographic information system. 2013. 1, 3, 5
- [41] Fritz Kessler. Volunteered geographic information: A bicycling enthusiast perspective. *Cartography and Geographic Information Science*, 38(3):258–268, 2011. 10
- [42] ANDERSON RESENDE LAMAS. *Uma arquitetura para o desenvolvimento de Sistemas de Informação Geográfica Móveis sensíveis ao contexto*. PhD thesis, Universidade Federal de Viçosa, 2009. 10
- [43] Leaflet. *Leaflet*. 2015. <http://www.leafletjs.com/>. 29
- [44] André Lemos. Arte e mídia locativa no brasil. *Comunicação e mobilidade*, page 89, 2009. 1
- [45] Cutlan Consultant Ltda. *Cutlan Consultant*. 2012. http://cutlan.co.tz/gis_rs.htm. 4
- [46] InfoEscola Ltda. *InfoEscola*. 2015. <http://www.infoescola.com/informatica/banco-de-dados-geograficos/>. 8
- [47] David J Maguire. An overview and definition of gis. *Geographical Information Systems: principles and applications*, 1:9–20, 1991. 1, 3, 5, 6
- [48] Bruce Momjian. *PostgreSQL: introduction and concepts*, volume 192. Addison-Wesley New York, 2001. 8, 26

- [49] Nancy J Obermeyer. The evolution of public participation gis. *Cartography and Geographic Information Systems*, 25(2):65–66, 1998. 1
- [50] OpenStreetMap. *OpenStreetMap*. 2015. <http://www.openstreetmap.org/>. 16
- [51] H Dennison Parker. The unique qualities of a geographic information system: a commentary. *Photogrammetric engineering and remote sensing (USA)*, 1988. 4
- [52] pgModeler Project. *PostgreSQL Database Modeler*. 2015. <http://www.pgmodeler.com.br/>. 30
- [53] PostGIS Project Steering Committee (PSC). *PostGIS*. 2015. <http://postgis.net/>. 8, 16
- [54] Philippe Rigaux, Michel Scholl, and Agnes Voisard. *Spatial databases: with application to GIS*. Morgan Kaufmann, 2001. 6
- [55] Marc Schlossberg and Elliot Shuford. Delineating "public" and "participation" in ppgis. 2005. 1, 13
- [56] Rosângela Silva. *Bancos de dados geográficos: uma análise das arquiteturas dual (Spring) e integrada (Oracle Spatial)*. PhD thesis, Universidade de São Paulo, 2002. 8
- [57] Terence R Smith, Sudhakar Menon, Jeffrey L Star, and John E Estes. Requirements and principles for the implementation and construction of large-scale geographic information systems. *International Journal of Geographical Information System*, 1(1):13–31, 1987. 3
- [58] Richard Stallman. *Free software, free society: Selected essays of Richard M. Stallman*. Lulu. com, 2002. 15, 16
- [59] Stefan Steiniger and Andrew JS Hunter. Free and open source gis software for building a spatial data infrastructure. In *Geospatial free and open source software in the 21st century*, pages 247–261. Springer, 2012. 14
- [60] Stefan Steiniger and Andrew JS Hunter. The 2012 free and open source gis software map—a guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems*, 39:136–150, 2013. 14
- [61] Daniel Sui. Opportunities and impediments for open gis. *Transactions in GIS*, 18(1):1–24, 2014. 14
- [62] M Tomko. *Spatial Databases for Mobile GIS Applications*. PhD thesis, TU Delft, Delft University of Technology, 2003. 10
- [63] Wikipedia. *Wikipedia*. 2015. <https://www.wikipedia.org/>. 10
- [64] Sam Williams. *Free as in Freedom [Paperback]: Richard Stallman's Crusade for Free Software*. "O'Reilly Media, Inc.", 2011. 15

- [65] Ming-Wei Wu and Ying-Dar Lin. Open source software development: an overview. *Computer*, 34(6):33–38, 2001. 15
- [66] James Bryan Zimmerman. Mobile computing: Characteristics, business benefits, and the mobile framework. *University of Maryland European Division-Bowie State*, 1999. 10, 12

Apêndice A

Scripts Fonte Utilizados no Banco de Dados

Apêndice A.1: *Script* de Criação do Banco de Dados Central

```
-- Database generated with pgModeler (PostgreSQL Database Modeler).
-- pgModeler version: 0.7.0
-- PostgreSQL version: 9.3
-- Project Site: pgmodeler.com.br
-- Model Author: Breno Diogo de Carvalho Camargos
```

```
SET check_function_bodies = false;
-- ddl-end --
```

```
-- Database creation must be done outside an multicommand file.
-- These commands were put in this file only for convenience.
-- -- object: bdCentral | type: DATABASE --
-- -- DROP DATABASE bdCentral;
-- CREATE DATABASE bdCentral
-- ;
-- -- ddl-end --
--
```

```
-- object: USUARIO | type: TABLE --
-- DROP TABLE USUARIO;
CREATE TABLE USUARIO(
    CPF character(11) NOT NULL,
    NOME character(50),
    EMAIL character(50),
    SENHA character(20),
    CONSTRAINT rpKUsuario PRIMARY KEY (CPF)
```

```
);
-- ddl-end --
```

```
-- object: AVALIACAO | type: TABLE --
-- DROP TABLE AVALIACAO;
```

```
CREATE TABLE AVALIACAO(
    ID integer,
    CPF character(11),
    CODIGO integer,
    AVALIACAO double precision
```

```
);
-- ddl-end --
```

```
-- object: TIPOAVALIACAO | type: TABLE --
-- DROP TABLE TIPOAVALIACAO;
```

```
CREATE TABLE TIPOAVALIACAO(
    CODAVALIACAO integer,
    DESCRICAO character(250),
    TIPO integer,
```

```

        CONSTRAINT rpktIPOAVALIACAO PRIMARY KEY (CODAVALIACAO)
    );
-- ddl-end --
-- object: ESTABELECIMENTO | type: TABLE --
-- DROP TABLE ESTABELECIMENTO;
CREATE TABLE ESTABELECIMENTO(
    ID integer,
    TIPO integer,
    NOME character(50),
    LATITUDE double precision,
    LONGITUDE double precision,
    CONSTRAINT rpkestabelecimento PRIMARY KEY (ID)
);
-- ddl-end --
-- object: TIPOESTABELECIMENTO | type: TABLE --
-- DROP TABLE TIPOESTABELECIMENTO;
CREATE TABLE TIPOESTABELECIMENTO(
    TIPO integer,
    DESCRICAO character(50),
    CONSTRAINT rpktipoestabelecimento PRIMARY KEY (TIPO)
);
-- ddl-end --
-- object: rfkUSUARIO | type: CONSTRAINT --
-- ALTER TABLE AVALIACAO DROP CONSTRAINT rfkUSUARIO;
ALTER TABLE AVALIACAO ADD CONSTRAINT rfkUSUARIO FOREIGN KEY (CPF)
REFERENCES USUARIO (CPF) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: rfkCODIGO | type: CONSTRAINT --
-- ALTER TABLE AVALIACAO DROP CONSTRAINT rfkCODIGO;
ALTER TABLE AVALIACAO ADD CONSTRAINT rfkCODIGO FOREIGN KEY (CODIGO)
REFERENCES TIPOAVALIACAO (CODAVALIACAO) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: rfkESTABELECIMENTO | type: CONSTRAINT --
-- ALTER TABLE AVALIACAO DROP CONSTRAINT rfkestabelecimento;
ALTER TABLE AVALIACAO ADD CONSTRAINT rfkestabelecimento FOREIGN KEY (ID)
REFERENCES ESTABELECIMENTO (ID) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: rfkTipoEstabelecimento | type: CONSTRAINT --
-- ALTER TABLE TIPOAVALIACAO DROP CONSTRAINT rfktipoestabelecimento;
ALTER TABLE TIPOAVALIACAO ADD CONSTRAINT rfktipoestabelecimento FOREIGN KEY (TIPO)
REFERENCES TIPOESTABELECIMENTO (TIPO) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

-- object: rfkTIPOESTABELECIMENTO | type: CONSTRAINT --
-- ALTER TABLE ESTABELECIMENTO DROP CONSTRAINT rfktipoestabelecimento;
ALTER TABLE ESTABELECIMENTO ADD CONSTRAINT rfktipoestabelecimento FOREIGN KEY (TIPO)
REFERENCES TIPOESTABELECIMENTO (TIPO) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
-- ddl-end --

```

Apêndice A.2: *Script* para Cadastro das Perguntas no Banco de Dados Central

```

INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (1, 'Qual sua nota para o atendimento desta escola?', 1);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (2, 'Qual sua nota para os profissionais desta escola?', 1);

```

```

INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (3, 'Qual_sua_nota_para_a_limpeza_destescola?', 1);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (4, 'Qual_sua_nota_para_a_organizacao_destescola?', 1);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (5, 'Qual_sua_nota_para_a_pontualidadedosprofissionais_destescola?', 1);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (6, 'Qual_sua_nota_para_o_atendimento_destehospital?', 2);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (7, 'Qual_sua_nota_para_osprofissionais_destehospital?', 2);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (8, 'Qual_sua_nota_para_a_limpeza_destehospital?', 2);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (9, 'Qual_sua_nota_para_a_organizacao_destehospital?', 2);
INSERT INTO TIPOAVALIACAO (CODAVALIACAO, DESCRICAO, TIPO) VALUES (10, 'Qual_sua_nota_para_a_pontualidadedosprofissionais_destehospital?', 2);

```

Apêndice B

Códigos Fonte Utilizados nas Interfaces

Apêndice B.1: Interface do Gestor

```
<!DOCTYPE html>
<html>
  <head>
    <title>Interface com o Gestor</title>
    <meta name="viewport"
          content="width=device-width, initial-scale=1.0, user-scalable=no">
    <meta charset="UTF-8">
    <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
    <!--[if lte IE 8]>
      <link rel="stylesheet" href="http://code.leafletjs.com/leaflet-0.3.1/leaflet.ie.css" />
    <![endif]-->

    <!--<script src="http://code.leafletjs.com/leaflet-0.3.1/leaflet.js"></script-->
  </head>
</body>

<?php
header('Content-Type: text/html; charset=utf-8');
class estabelecimentos {
  public $id;
  public $name;
  public $latitude;
  public $longitude;

  public $connection;
  public function __construct() {
    $this->connection = pg_connect("host=localhost port=5433 dbname=bdCentral user=
      postgres password=breno123") or print("cant connect");
  }
  public function __destruct() {
  }

  public function retornarPosicao() {
    $result = pg_query($this->connection, "select a.latitude, a.longitude, a.tipo,
      AVG(avaliacao) AS Media, a.id from estabelecimento a, avaliacao b where a.
      id=b.id group by a.latitude, a.longitude, a.tipo, a.id");
    if (!$result) {
      echo "An error occurred.\n";
      exit;
    }
    else {
      $i = 0;
      $i2 = 1;
      while ($row = pg_fetch_row($result)) {
        $posicao[$i] = $row[0];
        $posicao[$i+1] = $row[1];
        $posicao[$i+2] = $row[2];
      }
    }
  }
}
```

```

        $posicao[$i+3] = $row[3];
        $posicao[$i+4] = $row[4];
        $i++;
        $i++;
        $i++;
        $i++;
        $i++;
    }
}

pg_close($this->connection);
return $posicao;
}

public function retornar() {
    $result = pg_query($this->connection, "select a.latitude, a.longitude, a.tipo
        from estabelecimento a");
    if (!$result) {
        echo "An error occurred.\n";
        exit;
    }
    else {
        $i = 0;
        while ($row = pg_fetch_row($result)) {
            $posicao[$i] = $row[0];
            $posicao[$i+1] = $row[1];
            $posicao[$i+2] = $row[2];
            $i++;
            $i++;
            $i++;
        }
    }
}

pg_close($this->connection);
return $posicao;
}

public function update(){
}

public function delete(){
}
}

$meusEstabelecimentos = new estabelecimentos();
$estabelecimento = $meusEstabelecimentos->retornar();
$meusEstabelecimentos = new estabelecimentos();
$posicao = $meusEstabelecimentos->retornarPosicao();
?>
<p id="demo"></p>
<p id="demo2"></p>
<p id="demo3"></p>
<div id="map" style="width: 950px; height: 550px"></div>
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
<script>

    var map = L.map('map').setView(new L.LatLng(-15.809762, -47.887945), 11);
    L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
        attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a>
            contributors'
    }).addTo(map);

//ICONE DA ESCOLA
var escola = L.icon({
    iconUrl: 'iconescol.png',
    shadowUrl: 'iconescol.png',

    iconSize: [50, 64], // size of the icon
    shadowSize: [0, 0], // size of the shadow
    iconAnchor: [22, 94], // point of the icon which will correspond to marker'
        s location
    shadowAnchor: [4, 62], // the same for the shadow

```

```

        popupAnchor: [-3, -76] // point from which the popup should open relative to
            the iconAnchor
    });

//ICONE DO HOSPITAL
var hospital = L.icon({
    iconUrl: 'iconhosp.png',
    shadowUrl: 'iconhosp.png',

    iconSize: [50, 64], // size of the icon
    shadowSize: [0, 0], // size of the shadow
    iconAnchor: [22, 94], // point of the icon which will correspond to marker's
        location
    shadowAnchor: [4, 62], // the same for the shadow
    popupAnchor: [-3, -76] // point from which the popup should open relative to
        the iconAnchor
});

//ICONE CIRCULO VERDE
var greencircle = L.icon({
    iconUrl: 'greencircle.png',
    shadowUrl: 'greencircle.png',

    iconSize: [64, 64], // size of the icon
    shadowSize: [0, 0], // size of the shadow
    iconAnchor: [22, 94], // point of the icon which will correspond to marker's
        location
    shadowAnchor: [4, 62], // the same for the shadow
    popupAnchor: [-3, -76] // point from which the popup should open relative to
        the iconAnchor
});

//ICONE CIRCULO VERMELHO
var redcircle = L.icon({
    iconUrl: 'redcircle.png',
    shadowUrl: 'redcircle.png',

    iconSize: [64, 64], // size of the icon
    shadowSize: [0, 0], // size of the shadow
    iconAnchor: [22, 94], // point of the icon which will correspond to marker's
        location
    shadowAnchor: [4, 62], // the same for the shadow
    popupAnchor: [-3, -76] // point from which the popup should open relative to
        the iconAnchor
});

<?php $contadorLocais = count($estabelecimento) ?>
<?php $contadorPosicao = count($posicao) ?>

var i3 = <?php echo $contadorLocais ?>;
var i2 = <?php echo $contadorPosicao ?>;

var js_locais = <?php echo json_encode($estabelecimento); ?>;
var js_array = <?php echo json_encode($posicao); ?>;

for (var i = 0; i < i3; i++) {
    if(js_locais[i+2] == 1){
        var popup = "Desculpe. <br>ãNo_ãh_notas_cadastradas_para_essa_escola.";
        L.marker([js_locais[i],js_locais[i+1]], {icon:escola}).bindPopup(popup).
            addTo(map);
    } else if (js_locais[i+2] == 2){
        var popup = "Desculpe. <br>ãNo_ãh_notas_cadastradas_para_esse_hospital.";
        L.marker([js_locais[i],js_locais[i+1]], {icon:hospital}).bindPopup(popup).
            addTo(map);
    }
    i++;
    i++;
}

```

```

for (var i = 0; i < i2; i++) {
  if(js_array[i+2] == 1){

    if(js_array[i+3] > 2.5){
      var popup = "Perfeito.␣<br>A␣nota␣é␣mdia␣para␣essa␣escola␣foi␣de␣" +
        js_array[i+3] + ".<br><a␣href=␣'http://localhost/mono/listaravaliacoes.
        php?id=" + js_array[i+4] + "'␣target='_blank'>Visualizar␣os␣dados␣
        cadastrados␣</a>";
      L.marker([js_array[i],js_array[i+1]], {icon:escola}).bindPopup(popup).addTo
        (map);
      L.marker([js_array[i],js_array[i+1]], {icon:greencircle, opacity:0.6}).
        bindPopup(popup).addTo(map);
    } else if(js_array[i+3] <= 2.5) {
      var popup = "çãAteno!<br>A␣nota␣é␣mdia␣para␣essa␣escola␣foi␣de␣" + js_array[
        i+3] + ".<br><a␣href=␣'http://localhost/mono/listaravaliacoes.php?id="
        + js_array[i+4] + "'␣target='_blank'>Visualizar␣os␣dados␣cadastrados␣</
        a>";
      L.marker([js_array[i],js_array[i+1]], {icon:escola}).bindPopup(popup).addTo
        (map);
      L.marker([js_array[i],js_array[i+1]], {icon:redcircle, opacity:0.6}).
        bindPopup(popup).addTo(map);
    } else {
      L.marker([js_array[i],js_array[i+1]], {icon:escola}).addTo(map);
    }
  }
  else if (js_array[i+2] == 2){

    if(js_array[i+3] > 2.5){
      var popup = "Perfeito.␣<br>A␣nota␣é␣mdia␣para␣esse␣hospital␣foi␣de␣" +
        js_array[i+3] + ".<br><a␣href=␣'http://localhost/mono/listaravaliacoes.
        php?id=" + js_array[i+4] + "'␣target='_blank'>Visualizar␣os␣dados␣
        cadastrados␣</a>";
      L.marker([js_array[i],js_array[i+1]], {icon:hospital}).bindPopup(popup).
        addTo(map);
      L.marker([js_array[i],js_array[i+1]], {icon:greencircle, opacity:0.6}).
        bindPopup(popup).addTo(map);
    } else if(js_array[i+3] <= 2.5) {
      var popup = "çãAteno!<br>A␣nota␣é␣mdia␣para␣esse␣hospital␣foi␣de␣" +
        js_array[i+3] + ".<br><a␣href=␣'http://localhost/mono/listaravaliacoes.
        php?id=" + js_array[i+4] + "'␣target='_blank'>Visualizar␣os␣dados␣
        cadastrados␣</a>";
      L.marker([js_array[i],js_array[i+1]], {icon:hospital}).bindPopup(popup).
        addTo(map);
      L.marker([js_array[i],js_array[i+1]], {icon:redcircle, opacity:0.6}).
        bindPopup(popup).addTo(map);
    } else {
      L.marker([js_array[i],js_array[i+1]], {icon:hospital}).addTo(map);
    }
  }
  i++;
  i++;
  i++;
  i++;
}

</script>
</body>
</html>

```

Apêndice B.2: Verificação da Existência de Usuário

```

try{
    respostaRetornada = ConexaoHttpClient.executaHttpGet(urlGet);
    String existeUsuario = respostaRetornada.toString();
}
existeUsuario = existeUsuario.replaceAll("\\s+", "");

```

```

        if (existeUsuario.equals("1")){
            chamamenuprincipal();
        }
    else{
        mensagem("Nao foi possivel encontrar o usuario.",
            resposta2);
    }
}

```

Apêndice B.3: Página Requisitada na Verificação do Usuário

```

<?php
$usuario=$_GET['usuario'];
$senha=$_GET['senha'];
$conexao = pg_connect("host=localhost_port=5433_dbname=bdCentral_user=postgres_
password=breno123") or print("cant connect");
$result = pg_query($conexao,
"SELECT_usuario, senha FROM_USUARIO WHERE_usuario='$_usuario' AND_senha='
$senha'");
echo pg_num_rows($result);
if (pg_num_rows($result) >= 1) {
    echo 1;
    exit;
}
else {
    echo 0;
}
pg_close($conexao);
?>

```

Apêndice B.4: Consulta aos Estabelecimentos Cadastrados

```

public void obterEstabelecimentos(){
    String respostaRetornada = null;
    String url = VariaveisFinais.urlEstabelecimentos;
    try {
        respostaRetornada = ConexaoHttpClient.executaHttpGet(url);

        String resposta = respostaRetornada.toString();
        String resposta2 = resposta;

        resposta = resposta.replaceAll("\\s+", "");

        if (!resposta.contains("erro")){
            String[] estabelecimentos = resposta.split("AAA");
            String latitude = "";
            latitude = estabelecimentos[2];
            estabelecimento = estabelecimentos;
        }
    }
    catch (Exception erro){
    }
}

```

Apêndice B.5: Página Requisitada para Descobrir os Estabelecimentos Cadastrados

```

<?php
class estabelecimento {
    public $id;
    public $name;
    public $latitude;
    public $longitude;

    public $connection;
    public function __construct() {
        $this->connection = pg_connect("host=localhost_port=5433_dbname=bdCentral_user=
postgres_password=breno123") or print("cant connect");
    }
    public function __destruct() {

```

```

}

public function retornarPosicao() {
    $result = pg_query($this->connection, "SELECT latitude, longitude, nome, tipo, id FROM estabelecimento");
    if (!$result) {
        echo "An error occurred.\n";
        exit;
    }
    else {
        $i = 0;
        $i2 = 1;
        while ($row = pg_fetch_row($result)) {
            $posicao[$i] = $row[0];
            $posicao[$i2] = $row[1];
            $posicao[$i2+1] = $row[2];
            $posicao[$i2+2] = $row[3];
            $posicao[$i2+3] = $row[4];
            echo $posicao[$i];
            echo 'AAA';
            echo $posicao[$i2];
            echo 'AAA';
            echo $posicao[$i2+1];
            echo 'AAA';
            echo $posicao[$i2+2];
            echo 'AAA';
            echo $posicao[$i2+3];
            echo 'AAA';
            $i++;
            $i++;
            $i++;
            $i++;
            $i++;
            $i2++;
            $i2++;
            $i2++;
            $i2++;
            $i2++;
        }
    }

    pg_close($this->connection);
    return $posicao;
}

public function update(){
}
public function delete(){
}
}

$estabelecimentos = new estabelecimento();
$posicao = $estabelecimentos->retornarPosicao();
?>

```

Apêndice B.6: Validação das Coordenadas do Usuário

```

public boolean distanciaCorreta(GeoPoint estabelecimento){

    Location locationA = new Location("Estabelecimento");
    locationA.setLatitude(estabelecimento.getLatitude());
    locationA.setLongitude(estabelecimento.getLongitude());
    Location locationB = new Location("Usuario");
    locationB.setLatitude(VariaveisFinais.latitudeModificavel);
    locationB.setLongitude(VariaveisFinais.longitudeModificavel);
    float distanciaUsuarioEstabelecimento = locationA.distanceTo(locationB);
    if(distanciaUsuarioEstabelecimento < 1000){
        return true;
    } else {
        return false;
    }
}

```

```
}
```

Apêndice B.7: Consulta às Perguntas Cadastradas no Banco

```
public void obterPerguntasBanco(){
    String url = VariaveisFinais.urlPerguntas + VariaveisFinais.LOCALSEL;
    try {
        respostaRetornada = ConexaoHttpClient.executaHttpGet(url);
        String resposta = respostaRetornada.toString();

        resposta = resposta.replaceAll("\\s+", " ");

        if (!resposta.contains("erro")){
            if(!resposta.equals("0")){
                perguntas = resposta.split("AAA");
            } else {
                mensagem("Erro ao buscar no banco de dados", "Desculpe,
                mas não há perguntas cadastradas para esse
                estabelecimento.");
                startActivity(new Intent(AtividadeEstabelecimento.this,
                AtividadeOSM.class));
            }
        }
    }
    catch (Exception erro){
    }
}
```

Apêndice B.8: Página Requisitada para Descobrir as Perguntas Cadastradas

```
<?php
$tipo = $_GET['tipo'];
$conexao = pg_connect("host=localhost_port=5433_dbname=bdCentral_user=postgres
password=breno123") or print("cant connect");
$result = pg_query($conexao, "SELECT DESCRICAO, CODAVALIACAO FROM TIPOAVALIACAO WHERE
TIPO=$tipo");
if (!$result) {
    echo "An error occured.\n";
    exit;
}
else {
    $i = 0;
    $i2 = 1;
    if (pg_num_rows($result) == 0) {
        echo 0;
        exit;
    }
    while ($row = pg_fetch_row($result)) {
        $posicao[$i] = $row[0];
        $posicao[$i+1] = $row[1];
        echo $posicao[$i];
        echo 'AAA';
        echo $posicao[$i+1];
        echo 'AAA';
        $i++;
    }
    pg_close($conexao);
}
?>
```

Apêndice B.9: Cadastro da Avaliação do Usuário

```
public void onClick(View v) {
    String enviarResposta;
    try {
        String urlCadastrar = VariaveisFinais.
        urlCadastraPerguntas + VariaveisFinais.
        cpfUsuario + "&id=" + VariaveisFinais.
        idEstabelecimento;
    }
}
```

```

urlCadastrar = urlCadastrar + "&p1=" + perguntas
    [1] + "&p1nota=" + rbNotas[0].getRating();
urlCadastrar = urlCadastrar + "&p2=" + perguntas
    [3] + "&p2nota=" + rbNotas[1].getRating();
urlCadastrar = urlCadastrar + "&p3=" + perguntas
    [5] + "&p3nota=" + rbNotas[2].getRating();
urlCadastrar = urlCadastrar + "&p4=" + perguntas
    [7] + "&p4nota=" + rbNotas[3].getRating();
urlCadastrar = urlCadastrar + "&p5=" + perguntas
    [9] + "&p5nota=" + rbNotas[4].getRating();
enviarResposta = ConexaoHttpClient.executaHttpGet
    (urlCadastrar);
String resposta = enviarResposta.toString();
String erroOcorrido = resposta;
resposta = resposta.replaceAll("\\s+", "");
if (!resposta.contains("erro")){
    if (resposta.equals("1")){
        mensagem("Obrigado", "ãOpinio_cadastrada_
            com_sucesso.");
        startActivity(new Intent(
            AtividadeEstabelecimento.this,
            AtividadeConsultaPrincipal.class));
    } else {
        mensagem("Erro", "Desculpe_mas_ãno_foi_
            ipossvel_cadastrar_sua_opinio_Por_
            favor_tente_novamente.");
        mensagem("Erro", erroOcorrido);
        startActivity(new Intent(
            AtividadeEstabelecimento.this,
            AtividadeOSM.class));
    }
} else {
    mensagem("Erro", erroOcorrido);
    startActivity(new Intent(
        AtividadeEstabelecimento.this,
        AtividadeOSM.class));
}
}
catch (Exception erro){
Log.i("Conexao", "Erro_na_rotina_do_servidor_"+erro);
}
}
}

```

Apêndice B.10: Página Requisitada para Cadastro da Avaliação do Usuário

```

<?php
$cpf=$_GET['cpf'];
$id=$_GET['id'];
$pergunta1=$_GET['p1'];
$pergunta1nota=$_GET['p1nota'];
$pergunta2=$_GET['p2'];
$pergunta2nota=$_GET['p2nota'];
$pergunta3=$_GET['p3'];
$pergunta3nota=$_GET['p3nota'];
$pergunta4=$_GET['p4'];
$pergunta4nota=$_GET['p4nota'];
$pergunta5=$_GET['p5'];
$pergunta5nota=$_GET['p5nota'];
$conexao = pg_connect("host=localhost_port=5433_dbname=bdCentral_user=postgres_
    password=breno123") or print("Nao_foi_possivel_se_conectar_ao_banco_de_dados.");
$result = pg_query($conexao, "INSERT INTO AVALIACAO (ID, CPF, CODIGO, AVALIACAO)
    VALUES ($id, '$cpf', $pergunta1, $pergunta1nota)");
$result = pg_query($conexao, "INSERT INTO AVALIACAO (ID, CPF, CODIGO, AVALIACAO)
    VALUES ($id, '$cpf', $pergunta2, $pergunta2nota)");
$result = pg_query($conexao, "INSERT INTO AVALIACAO (ID, CPF, CODIGO, AVALIACAO)
    VALUES ($id, '$cpf', $pergunta3, $pergunta3nota)");
$result = pg_query($conexao, "INSERT INTO AVALIACAO (ID, CPF, CODIGO, AVALIACAO)
    VALUES ($id, '$cpf', $pergunta4, $pergunta4nota)");
$result = pg_query($conexao, "INSERT INTO AVALIACAO (ID, CPF, CODIGO, AVALIACAO)
    VALUES ($id, '$cpf', $pergunta5, $pergunta5nota)");

```

```

    if($result)
    echo 1;

    pg_close($conexao);
?>

```

Apêndice B.11: Página Responsável por Listar Todas as Avaliações dos Usuários para o Local Selecionado

```

<head>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<?php
    $id=$_GET['id'];
    $conexao = pg_connect("host=localhost_port=5433_dbname=bdCentral_user=postgres_
        password=breno123") or print("cant_connect");
    $result = pg_query($conexao, "select a.nome as NomeEstabelecimento , c.nome as
        NomeUsuarioAvaliacao , d.descricao as Pergunta , b.avaliacao from estabelecimento a
        , avaliacao b , usuario c , tipoavaliacao d where a.id = b.id and c.cpf = b.cpf and
        a.id = $id and d.codavaliacao = b.codigo");
    if (!$result) {
        echo "An error occurred.\n";
        exit;
    }
    else {
        $i = 0;
        $i2 = 1;
        if (pg_num_rows($result) == 0) {
            echo 0;
            exit;
        }
        echo "<div class='CSSTableGenerator'>";
        echo "<table border=1>";
        echo "<tr><td>Local</td><td>Nome</td><td>Pergunta</td><td>Avaliacao(
            de 0 a 5)</td></tr>";

        while ($row = pg_fetch_row($result)) {
            echo "<tr>";
            echo "<td valign=top>".$row[0]. "</td>";
            echo "<td valign=top>".$row[1]. "</td>";
            echo "<td valign=top>".$row[2]. "</td>";
            echo "<td valign=middle>".$row[3]. "</td>";
            echo "</tr>";
        }

        echo "</table>";
        echo "</div>";

        pg_close($conexao);
    }
?>

```