

**MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE  
COMPUTAÇÃO**

**Ten ARTHUR FERNANDES ARAUJO**

**Ten MAIARA BARROSO CARDOSO REINALDO**

**INSTALAÇÃO E CONFIGURAÇÃO DE SERVIÇOS WEB  
PARA DISPONIBILIZAÇÃO DE DADOS ESPACIAIS EM  
OPERAÇÕES MILITARES**

**Rio de Janeiro**

**2015**

**INSTITUTO MILITAR DE ENGENHARIA**

**Ten ARTHUR FERNANDES ARAUJO**

**Ten MAIARA BARROSO CARDOSO REINALDO**

**INSTALAÇÃO E CONFIGURAÇÃO DE SERVIÇOS WEB PARA  
DISPONIBILIZAÇÃO DE DADOS ESPACIAIS EM OPERAÇÕES  
MILITARES**

Trabalho apresentado ao Curso de Graduação em Engenharia de Computação do Instituto Militar de Engenharia, como Verificação Final do Projeto de Fim de Curso.

Orientador: Ivanildo Barbosa

**Rio de Janeiro**

**2015**

c2015

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro – RJ      CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade dos autores e do orientador.

005.8	Araujo, Arthur Fernandes
A663i	Instalação e configuração de serviços web para disponibilização de dados espaciais em operações militares / Arthur Fernandes Araujo, Maiara Barroso Cardoso Reinaldo; orientados por Ivanildo Barbosa – Rio de Janeiro: Instituto Militar de Engenharia, 2015.
	73p. : il.
	Projeto de Fim de Curso (PFC) – Instituto Militar de Engenharia, Rio de Janeiro, 2015
	1. Curso de Engenharia de Computação – Projeto de Fim de Curso. 2. Segurança da informação. 3. Dados espaciais. I. Oliveira, Matheus Vanzan Pimentel de. II. Barbosa, Ivanildo. III. Título. IV. Instituto Militar de Engenharia.

**INSTITUTO MILITAR DE ENGENHARIA**

**ARTHUR FERNANDES ARAUJO  
MAIARA BARROSO CARDOSO REINALDO**

**INSTALAÇÃO E CONFIGURAÇÃO DE SERVIÇOS WEB PARA DISPONIBILIZAÇÃO  
DE DADOS ESPACIAIS EM OPERAÇÕES MILITARES**

Trabalho apresentado ao Curso de Graduação em Engenharia de Computação do Instituto Militar de Engenharia, como Verificação Final do Projeto de Fim de Curso.

Orientador: Ivanildo Barbosa

Aprovado em 13 de outubro de 2015 pela seguinte Banca Examinadora:

---

Ivanildo Barbosa – D.C., do IME

---

Ricardo Choren Noya – D.C., do IME

---

Humberto Henriques de Arruda – M.Sc., do IME

Rio de Janeiro

2015

# SUMÁRIO

<b>LISTA DE ILUSTRAÇÕES.....</b>	<b>7</b>
<b>LISTA DE TABELAS.....</b>	<b>9</b>
<b>LISTA DE SIGLAS.....</b>	<b>10</b>
<b>1 INTRODUÇÃO .....</b>	<b>13</b>
1.1 Motivação .....	13
1.2 Objetivo .....	14
1.3 Metodologia.....	15
1.4 Estrutura do Texto .....	17
<b>2 CONHECIMENTOS BÁSICOS .....</b>	<b>19</b>
2.1 Serviços <i>web</i> .....	19
2.2 <i>Web Map Service</i> (WMS) .....	20
2.3 GeoServer .....	22
2.4 PostGIS .....	23
2.5 Openlayers .....	24
2.6 Bootstrap.....	25
<b>3 CENÁRIO DE APLICAÇÃO .....</b>	<b>27</b>
<b>4 PROJETO DA APLICAÇÃO .....</b>	<b>29</b>
4.1 Requisitos do <i>software</i> .....	29
4.2 Arquitetura do <i>software</i> .....	30
4.2.1 Arquitetura da aplicação do usuário .....	31
4.2.2 Arquitetura da aplicação do administrador .....	33
4.3 Diagrama de classes.....	35
4.4 Persistência dos dados .....	36
4.5 Projeto da interface .....	38
4.5.1 Aplicação do usuário .....	38
<b>5 DESENVOLVIMENTO DA SOLUÇÃO.....</b>	<b>41</b>
5.1 Configuração do ambiente .....	41
5.1.1 Instalação.....	42
5.1.2 Estilos do GeoServer .....	46
5.2 Desenvolvimento da aplicação do usuário.....	47
5.2.1 Desenvolvimento da interface .....	47

5.2.2 Identificação do usuário .....	52
5.2.3 Exibição do mapa .....	53
5.2.4 Filtro de autenticação .....	54
5.3 Desenvolvimento da aplicação do administrador .....	55
5.3.1 Interface com o usuário .....	55
<b>6 INTERAÇÃO COM OUTRAS APLICAÇÕES.....</b>	<b>58</b>
<b>7 APLICAÇÃO DO CENÁRIO.....</b>	<b>60</b>
7.1 Instalação das aplicações .....	60
7.2 Criação do cenário .....	64
7.3 Utilização da aplicação principal .....	66
<b>8 CONCLUSÃO .....</b>	<b>70</b>
<b>9 REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>72</b>

## LISTA DE ILUSTRAÇÕES

FIG. 4.1- Arquitetura da aplicação do usuário .....	32
FIG. 4.2 - Fluxograma de funcionamento da aplicação do usuário .....	33
FIG. 4.3 - Arquitetura da aplicação do administrador.....	34
FIG. 4.4 - Diagrama de classes do domínio das aplicações .....	35
FIG. 4.5 - Mapeamento objeto relacional do banco de dados da aplicação .....	37
FIG. 4.6 - Projeto da interface com o usuário para dispositivos <i>desktop</i> .....	38
FIG. 4.7 – Projeto da interface com o usuário para dispositivos móveis.....	39
FIG. 5.1 - Configuração do banco de dados espaciais no GeoServer .....	44
FIG. 5.2 - Organização do diretório do instalador do sistema .....	45
FIG. 5.3 - Interface da página de <i>login</i> da aplicação do usuário em um tamanho de tela para plataforma <i>desktop</i> .....	48
FIG. 5.4 - Interface da aplicação do usuário em um tamanho de tela para plataforma <i>desktop</i> , com nenhuma camada sendo exibida e a lista de camadas minimizada em um botão na parte superior e esquerda da tela .....	49
FIG. 5.5 - Interface da aplicação do usuário em um tamanho de tela para plataforma <i>desktop</i> , com duas camada e suas feições sendo exibidas, e a lista de camadas maximizada.....	49
FIG. 5.6 - Interface da aplicação do usuário em um tamanho de tela para plataforma móvel, com nenhuma camada sendo exibida e a lista de camadas minimizada em um botão na parte superior e esquerda da tela .....	50
FIG. 5.7 - Interface da aplicação do usuário em um tamanho de tela para plataforma móvel, com uma camada e uma feição da camada sendo exibidas, e a lista de camadas minimizada em um botão na parte superior e esquerda da tela.....	50
FIG. 5.8 - Tela inicial da aplicação do administrador .....	55
FIG. 5.9 - Configuração das camadas .....	56
FIG. 5.10 - Configuração dos níveis de acesso .....	56
FIG. 5.11 - Configuração dos usuários.....	57
FIG. 6.1 - Configuração do servidor WMS no QGIS .....	58
FIG. 6.2 - Visualização de camadas presentes no sistema pelo QGIS .....	59
FIG. 7.1 - Visualização das camadas adicionadas ao PostGIS na ferramenta de administração do GeoServer .....	62
FIG. 7.2 – Trecho de configuração da Bounding Box da página de publicação da camada na ferramenta de visualização do GeoServer .....	62

FIG. 7.3 – Trecho de configuração dos estilos da página de publicação da camada na ferramenta de visualização do GeoServer .....	63
FIG. 7.4 - Página com as camadas publicadas no GeoServer .....	63
FIG. 7.5 - Utilização da aplicação do administrador para adição de uma camada .....	64
FIG. 7.6 - Utilização da aplicação do administrador para adição de um nível de acesso .....	65
FIG. 7.7 - Utilização da aplicação do administrador para adição de um usuário.....	65
FIG. 7.8 - Visualização da página de <i>login</i> em uma plataforma <i>desktop</i> .....	66
FIG. 7.9 - Visualização da página de <i>login</i> em uma plataforma móvel .....	67
FIG. 7.10 - Visualização de camada e feições de um elemento dessa camada na aplicação principal.....	68
FIG. 7.11 - Visualização da localização do usuário e informações acerca dessa localização na aplicação principal.....	68
FIG. 7.12- Visualização da aplicação principal em tela cheia e com a barra lateral minimizada em uma plataforma móvel .....	69

## **LISTA DE TABELAS**

TAB. 2.1 - Operações WMS e suas principais características ..... 21

## **LISTA DE SIGLAS**

CSS – *Cascading Style Sheets*

GeoRSS – *Geography Rich Site Summary*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

HTTPS – *HyperText Transfer Protocol Secure*

JDBC – *Java Database Connectivity*

JSP – *JavaServer Pages*

OGC – *Open Geospatial Consortium*

ORM – *Object-relational Mapping*

PFC – Projeto Final de Curso

SGBD – Sistema de Gerenciamento de Banco de Dados

URL – *Universal Resource Locator*

WAR – *Web Application Archive*

WFS – *Web Feature Service*

WMS – *Web Map Service*

## **RESUMO**

O avanço da tecnologia permitiu a utilização de ferramentas computacionais para armazenar e disponibilizar informações geográficas. Com isso, permitiu-se o desenvolvimento de aplicações para visualização de dados geográficos, onde dados geoespaciais adicionais estão associados a uma localização. Essas aplicações podem ser utilizadas para as mais variadas finalidades, como no auxílio ao planejamento e condução de operações militares. Tendo em vista a importância do uso de informações geográficas e as vantagens obtidas ao se utilizar as tecnologias apropriadas para a manipulação e visualização de dados geoespaciais, esse projeto visa a instalação e configuração de um serviço *web* para disponibilização de dados espaciais em operações militares. Os requisitos para o servidor de mapas a ser customizado incluem: dividir os usuários em categorias de acordo com as permissões de acesso que estes possuam, permitir ao usuário visualizar e interagir com mapas, oferecer ao usuário opções de visualização de outras camadas no mapa, permitir o acesso à aplicação em qualquer navegador e possuir interface que seja compatível com plataformas desktop e plataformas móveis, entre outras funcionalidades. O servidor de mapas pode ser aplicado como forma de apoio à decisão em operações militares, permitindo controle de acesso de acordo com o cargo ocupado ou alguma outra característica específica do usuário. Por ser uma aplicação web, pode ser utilizada por meio de plataformas móveis em campos de batalha. Além disso, existe a possibilidade de criação de superposições táticas ou camadas digitais georreferenciadas que podem ser adicionadas aos dados geoespaciais da aplicação.

## **ABSTRACT**

Advances in technology allowed the use of computational tools to store and provide geographic information. Having that, it was then possible to develop applications to visualize geographic data, where additional geospatial data can be associated to one location. These applications can be utilized for the most variable purposes, as a support for planning and conducting military operations, for example. In view of the importance of the use of geographic information and the advantages obtained when the appropriate technologies are utilized to manipulate and visualize geospatial data, this project aims to install and to set up a web service to provide spatial data in military operations. The requirements to the maps server to be customized include: split the users into categories according to the access permissions that they have, allow the users to visualize and interact with maps, offer the user options to visualize other layers in the map, allow the access to the application in any browser and have an interface that is compatible with desktop platforms and mobile platforms, among other features. The maps server can be applied as a way of decision support in military operations, allowing the access control according to the occupied position or any other characteristic of the user. Once it is a web application, it can be used through mobile platforms in battlefields. Besides that, there is the possibility of creating tactic superpositions or georeferenced digital layers that can be added to the geospatial data of the application.

# 1 INTRODUÇÃO

A utilização de mapas pelo homem para visualização de dados espaciais data desde os tempos pré-históricos. Na Idade Moderna, a importância dos mapas cresceu significativamente e foi fator essencial para o progresso das grandes navegações. Nesse período, a confecção dos mapas já estava associada a tecnologias como a bússola e o astrolábio. Ainda na Idade Moderna, a Cartografia contou com um período de grande crescimento conhecido como a “Idade de ouro da Cartografia”, no qual a cartografia produzida era de primeira categoria e quando grandes cartógrafos se destacaram, entre eles Mercator, hoje considerado o pai da cartografia moderna (ARAÚJO; CARVALHO, 2008).

O surgimento dos computadores e satélites revolucionou a produção e a disponibilização de mapas. Estes deixaram de ser confeccionados apenas em papel e passaram a ser produzidos por meio de *softwares* e armazenados em bancos de dados.

Atualmente existem diversas formas de disponibilização de dados geoespaciais, inclusive via *web*, que permitem o desenvolvimento de aplicações específicas de visualização de mapas. Ou seja, os dados geográficos não são apenas disponibilizados para visualização, mas também são oferecidos serviços que permitem utilizar esses dados em aplicações com funcionalidades específicas desejadas. O desenvolvimento de serviços web destinados ao compartilhamento de dados geoespaciais tem agilizado o acesso à informação geográfica, com diferentes níveis de interatividade, de acordo com os interesses da instituição que disponibiliza esses dados.

## 1.1 Motivação

Quando a Cartografia passou a utilizar todas as tecnologias modernas de aquisição, processamento, visualização e disponibilização de dados espaciais, surgiu a necessidade de aliar o trabalho do engenheiro cartógrafo ao do engenheiro de computação ou outros profissionais da área. Nesse contexto, o acesso mais rápido a volumes crescentes de dados, gerados com precisão cada vez maior, potencializa os resultados do emprego desses dados em aplicações cada vez mais variadas.

Sabendo da importância da evolução tecnológica e seu impacto no combate, o Exército Brasileiro passou a desenvolver soluções para difundir e ampliar o uso da Geoinformação nas atividades desenvolvidas na Força Terrestre, sejam elas operacionais, logísticas ou administrativas (EME, 2014). Para tanto, é necessário que a geoinformação produzida chegue de forma rápida e precisa aos seus usuários.

Uma das formas de disponibilizar dados espaciais é por meio do *Web Map Service* (WMS), especificado pelo *OpenGIS Consortium* (OGC). Em linhas gerais, um servidor WMS processa requisições HTTP e retorna um *bitmap* renderizado a partir de consultas espaciais a dados associados ao serviço. Com isso, um navegador web é suficiente para consultas mais simples, dispensando a instalação de aplicativos próprios de geoprocessamento, de interface complexa para a grande maioria dos usuários.

## 1.2 Objetivo

Tendo em vista a importância do uso de dados geográficos e as vantagens obtidas ao se utilizar as tecnologias apropriadas para a manipulação e visualização desses dados, esse projeto visou à instalação e à configuração de serviços *web* para disponibilização de dados espaciais em operações militares.

Para obter esse resultado, foram estabelecidos os seguintes objetivos secundários:

- a) Instalar e configurar soluções livres para disponibilização de serviços *Web Map Service* (WMS) os quais disponibilizam os dados espaciais. Os serviços WMS são serviços *web* especificados pelo *Open Geospatial Consortium* (OGC) que foram descritos no desenvolvimento do trabalho.
- b) Projetar a interface com o usuário de forma a obter uma aplicação responsiva que pudesse ser utilizada tanto em dispositivos móveis como em dispositivos *desktop*.
- c) Criar uma política de acesso de usuários, visando à integridade dos dados e à segurança de informações sensíveis.

A aplicação desenvolvida não é exclusivamente de emprego militar, mas o emprego em operações militares foi um objetivo desse projeto, uma vez que a aplicação foi desenvolvida tendo como foco satisfazer aos seguintes quesitos:

- a) A usabilidade por usuários de diferentes perfis.

- b) A restrição de acesso à aplicação, obedecendo a uma política de acesso.
- c) A portabilidade para emprego em dispositivos móveis (em campo) e em *desktop* (gabinete).

### **1.3 Metodologia**

Como primeiro passo na realização do projeto foi feito um estudo acerca do que são serviços *web* focando naqueles voltados para disponibilização de dados geoespaciais. Dentre os serviços estudados, o WMS recebeu maior atenção por ter sido o principal serviço empregado no projeto. Foi pesquisado o que é o WMS e quais operações ele disponibiliza.

Como foi necessária a utilização de um servidor WMS para realização do trabalho, o servidor WMS GeoServer foi escolhido por ser um servidor certificado e de alto desempenho, e por facilitar o uso e suporte para padrões abertos, permitindo que informações geoespaciais sejam compartilhadas rapidamente e de maneira interoperável (OSGeo, 2015b). Sendo assim, foi realizado um estudo visando entender como utilizar o GeoServer no contexto deste trabalho. Após isso, foi abordada a ferramenta PostGIS, uma extensão espacial gratuita do Sistema de Gerenciamento de Banco de Dados (SGBD) PostgreSQL, que foi utilizada para armazenamento dos dados espaciais que foram disponibilizados pelo serviço WMS do GeoServer. Os dados que compõem o cenário utilizado para validar esse trabalho foram o resultado de um PFC da Seção de Engenharia Cartográfica do IME (DELGADO *et al*, 2015), e esses dados foram armazenados em um banco de dados no PostGIS para serem utilizados por essa aplicação.

Além disso, buscando uma solução no lado do cliente que permitisse a interação do usuário com a aplicação, foi utilizada a ferramenta OpenLayers, uma biblioteca de código aberto que possui interação com qualquer serviço WMS (como o que é fornecido pelo GeoServer), permitindo a construção de uma imagem a ser apresentada para o cliente a partir dos dados fornecidos pelo serviço WMS.

Após essas etapas, foram levantados os requisitos da aplicação a ser desenvolvida (abordados no desenvolvimento do projeto), desde o cadastro do usuário até a manipulação do mapa e visualização de diferentes camadas de acordo com o nível de acesso, onde cada

camada contém informações geográficas específicas como locais de interesse para uma operação, hospitais, delegacias, etc.

Após o levantamento dos requisitos foi planejada a arquitetura da aplicação tendo em vista aspectos de segurança e desempenho, especificando todos os componentes do sistema, seus papéis e como eles se comunicam.

Com a arquitetura da aplicação elaborada, foi possível iniciar a configuração do GeoServer e do PostGIS criando *scripts* em Shell Script automatizados para permitir a instalação e configuração desses serviços em qualquer máquina que utilize o sistema operacional Ubuntu com versão 14.04. Esses *scripts* não foram anexados ao presente relatório, e sim ao material digital entregue ao final do projeto.

Para validação dessa aplicação foi utilizado um cenário específico no contexto dos Jogos Olímpicos do Rio de 2016, no entanto, a aplicação desenvolvida permite a configuração de qualquer cenário por meio de um usuário administrador que tenha acesso à máquina que hospeda a aplicação. Para permitir essa configuração de cenários foi desenvolvida uma segunda aplicação que permite ao administrador adicionar, editar e deletar as camadas, os níveis de acesso e os usuários relacionados ao cenário.

Paralelamente à atividade de configuração do GeoServer e do PostGIS, foi elaborado o projeto da aplicação e o projeto da interface com usuário.

O projeto da aplicação teve como objetivo identificar as principais classes utilizadas para atender aos requisitos levantados e as relações entre essas classes. Para tal, foi elaborado o diagrama de classes do domínio das aplicações para melhor entendimento e visualização dos atributos e comportamentos necessários a cada classe.

No projeto da interface com o usuário, tendo em vista que a aplicação foi projetada para ser utilizada tanto em dispositivos móveis como em dispositivos *desktop*, foram desenvolvidos modelos da interface a ser implementada para esses dois tipos de dispositivos de forma a atender os requisitos e garantir a fácil utilização das funcionalidades.

Após isso foi iniciada a fase da implementação da aplicação principal, onde foram criadas as classes definidas no projeto da aplicação, os controladores e gerentes dessas classes, a base de dados que foi relacionada ao serviço WMS e utilizada pela aplicação, e foi feita a comunicação entre o cliente e o servidor WMS (o GeoServer) garantindo controle de acesso, ou seja, permitindo que o cliente acesse apenas os recursos que está autorizado a acessar. Nessa fase, iniciou-se também o desenvolvimento da interface com o usuário utilizando o *framework* Bootstrap versão 3.0 (abordado no desenvolvimento do trabalho) para

desenvolvimento responsivo, tendo como base os modelos desenvolvidos na fase de projeto da interface com o usuário. Para implementação da interação do usuário com a aplicação foi utilizado código em Javascript utilizando a biblioteca OpenLayers para traduzir as interações do usuário com os dados espaciais em requisições ao servidor WMS.

Após isso, foi elaborada a segunda aplicação para criação e edição de cenários a serem visualizados na aplicação principal. Essa aplicação buscou prover uma forma simples de configuração dos cenários pelo administrador do sistema. Tal aplicação também foi desenvolvida com base no *framework* responsivo Bootstrap e adicionalmente utilizou-se a biblioteca em Javascript, jQuery, para prover funcionalidades adicionais de interatividade.

Por fim, os scripts que foram desenvolvidos para configuração dos serviços foram agrupados em um pacote único para prover ao administrador do sistema um software único para instalação e configuração de todos os pacotes necessários ao seu funcionamento, e uma forma simples de iniciar, parar e reiniciar o sistema.

## 1.4 Estrutura do Texto

No segundo capítulo do trabalho foram relatados os conhecimentos básicos necessário para o desenvolvimento da aplicação. Entre os assuntos abordados nesse capítulo, estão: serviços *web*, com uma explicação do que são serviços *web*, quais os órgãos reguladores e quais serviços *web* aplicados à disponibilização de dados geoespaciais; o serviço *web* WMS, explicando no que consiste esse serviço e suas principais operações; e as ferramentas Geoserver, PostGIS, OpenLayers e Bootstrap, explicando as características dessas ferramentas que viabilizam a sua utilização nesse projeto.

No terceiro capítulo, foi explicado o cenário que foi utilizado para validação da aplicação.

No quarto capítulo, foi explanado o projeto das duas aplicações desenvolvidas, o que incluiu: os requisitos levantados da aplicação a ser desenvolvida, em que são especificados todos os requisitos funcionais e não-funcionais do *software*; a arquitetura do software; o diagrama de classes do domínio da aplicação; a persistência dos dados da aplicação; e o projeto da interface.

No quinto capítulo, foi explanado o desenvolvimento da solução, onde foram explicados aspectos sobre a configuração do ambiente onde as duas aplicações iriam funcionar e o desenvolvimento dessas aplicações.

No sexto capítulo, foi abordada a utilização da aplicação como servidor de mapas por outras aplicações amplamente utilizadas de Sistemas de Informação Geográfica, exemplificando essa utilização com a configuração do QGIS.

No sétimo capítulo, foi apresentado o resultado do trabalho com uma demonstração de utilização de todo o sistema, identificando os usos do administrador e do usuário. E, por fim, uma conclusão com os objetivos, resultados e contribuições do trabalho.

## 2 CONHECIMENTOS BÁSICOS

Para desenvolvimento do software foi necessário o aprendizado de alguns assuntos relacionados ao tema e das ferramentas que foram utilizadas. Nesse capítulo constam todos os aspectos estudados acerca desses assuntos e ferramentas.

### 2.1 Serviços *web*

Serviço *web* é uma forma de implementação de Arquitetura Orientada a Serviços (*Service Oriented Architecture – SOA*), em que uma aplicação é descrita de tal forma a permitir a composição de serviços (aplicações integram funcionalidades de outras aplicações). Muitas empresas disponibilizam suas competências por meio de uma coleção de serviços *web*.

Serviços *web* utilizam protocolos baseados em XML (*eXtensible Markup Language*) para garantir total independência de Sistemas Operacionais e Linguagens de Programação. Os protocolos utilizados mais significativos são o SOAP (*Simple Object Access Protocol*) e o WSDL (*Web Service Description Language*) sobre HTTP (*Hypertext Transfer Protocol*) para realizar comunicação.

Um serviço encapsula uma funcionalidade de uma aplicação e torna-a disponível por meio de uma interface. Além disso, serviços são normalmente autônomos e heterogêneos, mantendo um baixo acoplamento com outros serviços e aplicações.

O objetivo dos serviços *web* é a comunicação de aplicações através da internet e são padronizados pelas instituições W3C (*World Wide Web Consortium*) e OASIS (*Organization for the Advancement of Structured Information Standards*). Segundo a W3C, um serviço *web* é definido como um sistema de software desenvolvido para suportar interoperabilidade entre máquinas em uma rede (DAMASCENO, 2015).

No âmbito dos dados geoespaciais, o OGC idealizou uma arquitetura de *software* para acesso distribuído a dados geoespaciais e recursos de geoprocessamento em geral, o que inclui um modelo para serviços de acesso, isto é, os serviços *web* geoespaciais. Existem mais de 30

padrões OGC relacionados a serviços web e seus componentes, entre eles o *Web Map Service* (WMS).

## 2.2 *Web Map Service (WMS)*

O WMS é um padrão apresentado pelo OGC que permite implementar uma interface HTTP simples para solicitar imagens de mapas georreferenciados de um ou mais bancos de dados geoespaciais distribuídos. Uma requisição WMS define, entre outros parâmetros, as camadas geográficas e a área de interesse a ser processada. A resposta à requisição pode ser uma ou mais imagens georreferenciadas (retornadas em formato JPEG, PNG, etc) que pode ser exibida em um navegador *web* com respostas formatadas em XML. A interface também suporta que seja especificada a opacidade das imagens de forma que camadas de múltiplos servidores possam ser combinadas ou não. Um benefício do WMS é que o cliente pode requisitar imagens de múltiplos servidores WMS e combinar essas imagens em uma única vista. O padrão WMS garante que essas imagens possam ser sobrepostas da forma como elas seriam na realidade (OGC, 2015).

Na Tabela 2.1 são especificadas as operações que podem ser realizadas com o serviço WMS, seus objetivos e o resultado obtido.

**TAB. 2.1 - Operações WMS e suas principais características.**

Operação	Objetivo	Resultado
<b>GetCapabilities</b>	Recuperar metadados de um serviço, o que inclui as operações suportadas e os valores aceitáveis dos parâmetros de requisição.	Arquivo XML contendo informações relativas às características do serviço disponibilizado.
<b>GetMap</b>	Recuperar uma imagem de mapa para uma área e conteúdo específicos.	Imagen no formato especificado na requisição.
<b>GetFeatureInfo (opcional)</b>	Recuperar dados básicos, incluindo geometria e valores de atributos, para a localização de um pixel no mapa.	Pode vir nos formatos: TEXT, HTML e JSON. Contém informações adicionais da camada.

Na operação GetCapabilities existem os seguintes parâmetros a serem especificados:

- Service: consiste no nome do serviço, nesse caso tem como valor WMS.
- Version: no qual deve-se especificar a versão do serviço WMS desejada. Os possíveis valores são 1.0.0, 1.1.0, 1.1.1 e 1.3.
- Request: consiste no nome da operação, nesse caso o valor para esse parâmetro deve ser GetCapabilities.

A resposta à requisição da operação GetCapabilities é um documento Capabilities XML que descreve o serviço WMS. Esse documento contém três seções:

- Service: contém os metadados do serviço, tais como o nome do serviço, as palavras-chave e informações de contato da organização que opera o servidor.
- Request: descreve as operações que o serviço WMS provê, informando o formato dos parâmetros e dos valores retornados para cada operação.
- Layer: lista com os sistemas de coordenadas e as camadas disponíveis.

A requisição à operação GetMap também demanda a passagem de parâmetros, entre os quais pode-se citar:

- Service, version e request: semelhante ao GetCapabilites.
- Layers: lista das camadas a serem exibidas no mapa.
- Styles: estilos de renderização a ser utilizado nas camadas.

- Src: Sistema de Referência Espacial utilizado para o mapa.
- Width e height: comprimento e altura do mapa, em pixels.

A operação GetFeatureInfo é aplicável apenas em camadas de informação que apresentam como verdadeiro o atributo *queryable*. A requisição a essa operação requer a passagem dos valores (x, y) do pixel de uma imagem retornada pelo serviço WMS e retorna informações específicas acerca desse ponto da imagem, como por exemplo informações de nome, contato e endereço de um hotel em uma camada de hotéis.

## 2.3 GeoServer

O GeoServer consiste em um servidor WMS de código aberto escrito em Java que permite aos usuários compartilhar e editar dados geoespaciais. É projetado para permitir interoperabilidade, publicando dados de qualquer fonte de dados espaciais importante que utilize padrões abertos. O GeoServer é um servidor de WMS (*Web Map Service*), WFS (*Web Feature Service*) e WCS (*Web Coverage Service*), de alta performance e certificado pelo teste de conformidade CITE do OGC (OSGeo, 2015c).

A utilização do GeoServer é feita através de uma ferramenta de administração via *web*, não sendo necessário lidar com arquivos de configuração grandes e complicados.

Para utilizar o GeoServer é necessário ter o Java Runtime Environment (JRE) instalado no sistema, pois o GeoServer requer um ambiente Java 7, além de um *servlet container* como o Tomcat, o Jetty ou o Glassfish por exemplo (OSGeo, 2015c).

O GeoServer é capaz de:

- Operar com diversas fontes de dados, dentre elas o PostGIS utilizado nesse trabalho;
- Exportar para formatos de visualização padronizados como o KML, GML e Shapefile;
- Se conectar a mapas baseados em serviços web como OpenLayers, e serviços próprios como o Google Maps e Bing Maps.

Outra característica importante do Geoserver que contribui para a eficiência no atendimento às suas requisições é a presença de um cache integrado de forma transparente

com o serviço de WMS do GeoServer, o GeoWebCache que pode ser ativado nas suas configurações (OSGeo, 2015d).

Para aplicações que necessitam se comunicar com o GeoServer de modo a configurar aspectos das camadas, isso não poderia ser feito de forma automatizada acessando sua interface *web*. Para esse uso específico o GeoServer provê uma API REST (OSGeo, 2015a) capaz de configurar todos os seus serviços por meio de requisições HTTP utilizando essa API.

No contexto desse trabalho essa API teve grande relevância na configuração de uma biblioteca de estilos para as camadas, tema abordado adiante nesse trabalho.

## 2.4 PostGIS

O PostGIS é uma extensão espacial do *software* de gerenciamento de banco de dados objeto-relacional PostgreSQL adicionando suporte espacial. Isso é feito por meio da definição de novos tipos de dados, operações sobre esses novos tipos e indexação espacial para aumentar a eficiência das consultas. Um banco de dados espaciais é capaz de armazenar e manipular objetos espaciais como qualquer outro objeto no banco de dados (PostGIS, 2015).

O PostGIS segue a especificação “*Simple Features for SQL*” especificada pelo OGC (*Open Geospatial Consortium*). Por ser construído em cima do PostgreSQL, o PostGIS automaticamente herda importantes características e implementação de diversos padrões. Além disso, diversas características presentes no PostgreSQL o tornam uma escolha importante para construção por extensão de um software de gerenciamento de banco de dados espaciais em relação a outros *softwares* como o MySQL por exemplo. Dentre essas características estão presentes (BOUNDLESS, 2015):

- Confiabilidade e integridade transacional comprovadas (ACID).
- Completo suporte a padrões SQL.
- Extensão de tipos e funções.
- Sem limites para o tamanho das colunas (permite suportar grandes objetos geográficos).
- Facilidade em customizar funções.

Diversas aplicações e serviços web possuem suporte ao PostGIS e permitem sua utilização como *backend* de dados. Entre essas aplicações estão o QGIS, que é um software de código aberto de um Sistema de Informações Geográficas (SIG), e o GeoServer, citado nesse trabalho (BOUNDLESS, 2015).

## 2.5 Openlayers

O OpenLayers (OL) é uma biblioteca de código aberto escrita em Javascript executada do lado do cliente que, portanto, funciona na máquina local do usuário, particularmente no seu *browser*. A biblioteca OL pode ser utilizada para exibição ou disponibilização de dados geográficos na Internet, pois fornece uma API para construção de aplicações geográficas *web* similares ao *Google Maps* e ao *Bing Maps*.

O OL facilita a obtenção das informações geográficas do servidor de mapas, pois abstrai todas as requisições de mapa que devem ser feitas ao servidor, montando essas requisições no formato correto e com os parâmetros necessários. Para isso, basta definir o servidor WMS que possui as informações de uma dada camada como fonte dessa camada no OL. Dessa forma, o OL gera automaticamente essas requisições de acordo com as demandas provenientes da interação do usuário com o mapa. Isto é, se o usuário aplica um zoom no mapa, por exemplo, o OL faz as requisições necessárias para obter as informações da nova imagem a ser gerada.

Um dos pontos fortes do OL consiste na sua conformidade com os padrões do OGC, permitindo que ele seja capaz de trabalhar com a maioria dos serviços de dados espaciais. Isso permite à aplicação cliente:

- Se conectar a serviços web providos por WMS, WFS e GeoRSS;
- Adicionar dados vetoriais em formatos como GeoJSON e GML;
- A possibilidade de organizar esses dados em camadas.

Além dessa flexibilidade nos tipos de dados espaciais, o OL permite construir aplicações interativas provendo diversos controles ao usuário. Esses controles permitem ao usuário explorar o mapa construído e os dados espaciais disponibilizados, e vão desde controles simples de navegação como zoom, até controles complexos que permitem a construção de animações. Além disso, parte desses controles possuem foco em aplicações móveis,

explorando características específicas desses aparelhos para realizar a navegação (PACKTUB, 2013).

Como foi explicado anteriormente, o OL é executado do lado do cliente, ou seja, na aplicação web em que o OL é empregado ele atua como o cliente que solicita os dados de mapa ao servidor de mapas. O OL aceita múltiplos servidores de mapa que podem ser adicionados ao mapa como camadas, ou seja, cada servidor fornece os dados de uma certa camada ao mapa. As camadas são adicionadas ao mapa uma em cima da outra, e os servidores de mapa como Google Maps e Bing Maps podem ser utilizados como camadas bases, ou seja, o mapa que será exibido ao fundo das outras camadas (KOMANNA, 2013).

Para adicionar um mapa à aplicação *web* utiliza-se a classe Map do OL que possui alguns atributos onde os principais são: as camadas do mapa; o elemento da página destinado ao mapa; e a visada do mapa (o centro e zoom aplicados no mapa para exibição na tela). Nas camadas são definidas a fonte das informações geográficas dessa camada, ou seja, o servidor de mapas que contém essas informações, o estilo correspondente, a opacidade da camada, dentre outras informações.

## 2.6 Bootstrap

O Bootstrap consiste em uma coleção de ferramentas gratuitas e de código aberto de *front-end* para criação de aplicações *web*, desenvolvido originalmente com o nome de Twitter Blueprint por Mark Otto e Jacob Thornton no Twitter. Como nesse projeto foi escolhida uma interface de navegação em *web browsers*, a fim de simplificar o uso da aplicação de disponibilização de dados espaciais desenvolvida por usuários leigos, a ferramenta Bootstrap foi bastante útil, uma vez que facilita a construção dessa interface e não apresenta nenhum problema de incompatibilidade com o Openlayers e com as requisições WMS, além de várias outras vantagens.

Uma das vantagens na utilização desse *framework* advém da sua compatibilidade com todos os navegadores modernos como Google Chrome, Firefox, Internet Explorer, Opera e Safari.

Além disso, desde a sua versão 2.0, o Bootstrap suporta o design *web* responsivo, permitindo que o *layout* de páginas *web* se ajuste dinamicamente de acordo com o dispositivo que está sendo utilizado (*desktop, tablet, smart phone*).

Para que o desenvolvedor possa utilizar o Bootstrap, é disponibilizado um conjunto de folhas de estilo (arquivos .css, *Cascading Style Sheets*), que possuem definições para formatar os principais elementos de uma página em HTML, além de elementos da interface com o usuário, desde um simples botão até opções mais avançadas como listas de navegação. Opcionalmente o desenvolvedor pode utilizar *plugins* em jQuery disponibilizados pelo Bootstrap para fornecer elementos adicionais à sua interface, como por exemplo animações e recursos para auto-completar formulários.

O Bootstrap permite que o desenvolvedor da interface se concentre nos aspectos principais das funcionalidades a serem implementadas utilizando seus *templates* e classes disponibilizados para permitir um desenvolvimento mais rápido de aplicações *web* (TUTORIAL REPUBLIC, 2015).

### **3 CENÁRIO DE APLICAÇÃO**

O cenário foi definido tendo como objetivo realizar a segurança de comitivas na locomoção entre as principais locações (locais de atrações, hotéis, locais das competições, etc) durante as Olimpíadas de 2016 no Rio de Janeiro. A fonte dos dados empregados no cenário foi o PFC (DELGADO *et al*, 2015) também realizado no IME.

As camadas disponíveis nesse cenário são:

- Locais de atrações diversas e de atrações previstas pelo Comitê Organizador Rio 2016;
- Bairros;
- Locais de competições;
- Locais para lanches e refeições;
- Locais de interesse (pontos turísticos);
- Paradas de metrô;
- Paradas de ônibus;
- Paradas de trem;
- Unidades dos bombeiros e delegacias;
- Hotéis.

O cenário foi definido com três níveis de acesso, além do administrador que tem acesso à máquina que hospeda o servidor:

- Estratégico;
- Tático;
- Operacional.

Da maneira como a aplicação é definida, os usuários são cadastrados pelo administrador em seus respectivos grupos e recebem um *login* e senha para o acesso à aplicação. Para que se possa validar a propriedade da aplicação de permitir acesso a diferentes camadas de dados geoespaciais por diferentes grupos, foram estabelecidas as seguintes regras de acesso:

- Estratégico: acesso ilimitado a todas as camadas.
- Tático: acesso a bairros, hotéis, locais de competição, locais de atrações e locais de interesse.

- Operacional: acesso a paradas de metrô, ônibus e trem, unidades dos bombeiros e delegacias.

## **4 PROJETO DA APLICAÇÃO**

O Projeto da Aplicação incluiu diversas atividades, dentre elas, inicialmente, a atividade de levantamento dos requisitos da aplicação, depois o projeto da arquitetura das aplicações desenvolvidas levando em conta os componentes utilizados, a modelagem das classes e , por fim, o projeto da interface da aplicação do usuário e da aplicação do administrador.

### **4.1 Requisitos do *software***

O produto a ser entregue ao final do projeto possui requisitos funcionais e não funcionais que foram levantados durante e após a realização da pesquisa sobre o tema e as tecnologias disponíveis. Esses requisitos levam em conta o uso de dois atores com papéis diferenciados: o administrador da aplicação e o usuário das informações espaciais disponibilizadas. Os requisitos funcionais levantados, os quais estão diretamente relacionados com as funcionalidades oferecidas pelo *software*, foram:

1. Permitir que o administrador cadastre o usuário para utilizar a aplicação em um determinado nível de acesso.
2. Permitir que o administrador cadastre as camadas que cada nível de acesso possui autorização para acessar, e informações relativas às camadas e feições de cada camada.
3. Permitir que o usuário entre na sua conta para utilizar a aplicação.
4. Separar os usuários de acordo com as permissões que possuam para utilizar a aplicação.
5. Apresentar ao usuário um mapa de uma determinada localização, com as camadas que este pode ter acesso.
6. Permitir que o usuário possa manipular o mapa com operações de ampliação, diminuição e navegação.
7. Permitir que informações disponíveis no banco de dados sejam disponibilizadas ao usuário quando este interagir com os elementos das camadas, como por exemplo ao clicar em um hotel obter informações específicas sobre esse hotel como nome e endereço.

Os requisitos não funcionais identificados foram:

1. Segurança: O software deve garantir que toda comunicação com o usuário e o acesso aos dados seja feito de forma segura, uma parte dessa tarefa foi feita nesse projeto e outra parte por (OLIVEIRA; SILVA, 2015).
2. Usabilidade: O software deve ser de fácil utilização sem a necessidade de manual, seguindo padrões utilizados por outros softwares de dados espaciais como o *Google Maps* e o *QGIS*, com os quais o usuário já possui familiaridade. Para atingir esse objetivo foi determinado que o usuário deve efetuar no máximo 4 interações com a aplicação para realizar uma determinada ação.
3. Portabilidade: O cliente pode acessar a aplicação em qualquer navegador que possua suporte a Javascript, e, além disso, a interação com o usuário deve aproveitar as características do ambiente em que o acesso ocorre. Ou seja, para o desenvolvimento da interface foram consideradas características peculiares de plataformas móveis e plataformas desktop. Além disso, os dados da aplicação também podem ser acessados por softwares de geoprocessamento compatíveis com o padrão OGC, como o QGIS, retratado posteriormente na seção 6.

## **4.2 Arquitetura do software**

O produto final desse trabalho contou com duas aplicações, a do usuário e a do administrador. A aplicação do usuário foi desenvolvida para visualização e interação do usuário com o mapa, de acordo com as permissões inerentes ao perfil cadastrado. A aplicação do administrador foi desenvolvida para permitir ao administrador do sistema configurar o cenário a ser empregado na aplicação do usuário. O administrador pode definir os usuários que têm acesso à aplicação do usuário, os níveis de acesso que são utilizados para alojar esses usuários e as camadas que fazem parte do cenário.

Ambas as aplicações foram instaladas em um Servidor de Aplicação, o GlassFish, juntamente com o GeoServer. Além desses, os outros componentes da arquitetura são os bancos de dados, o espacial (PostGIS) e o da aplicação, e as interfaces de cada uma das aplicações.

#### **4.2.1 Arquitetura da aplicação do usuário**

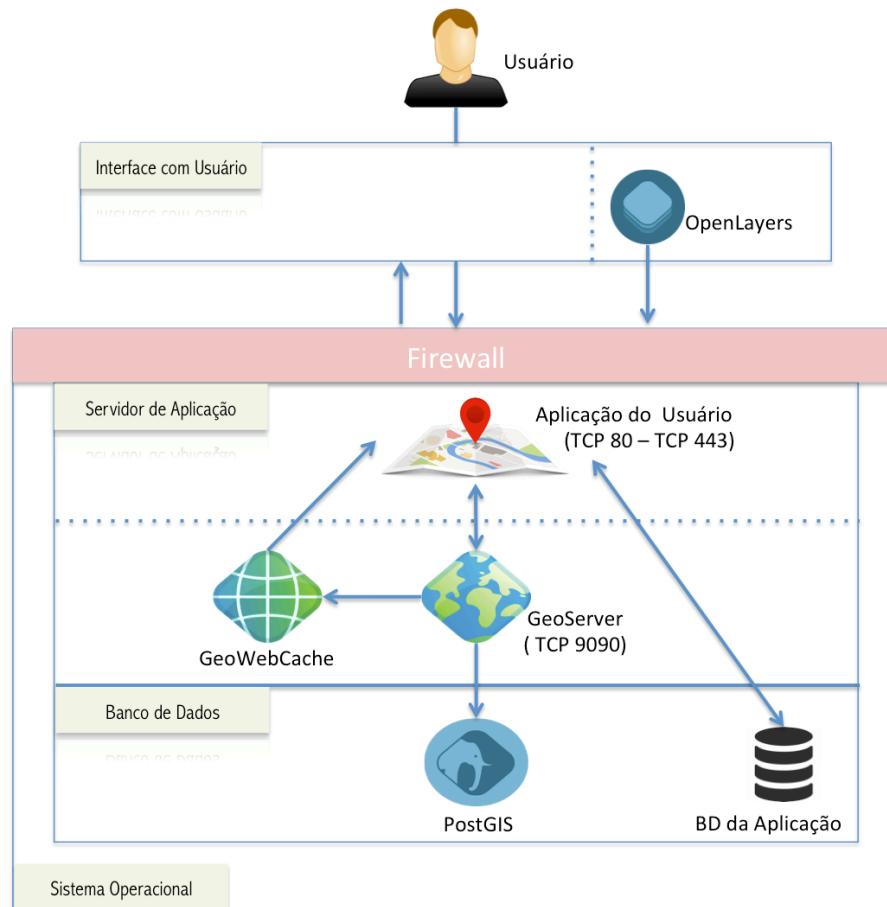
Os elementos presentes na arquitetura da aplicação do usuário são: a interface dessa aplicação, que possui integrada a ela o OpenLayers; o Servidor de Aplicação, contendo a aplicação do usuário (que contém o controlador de acesso) e o GeoServer (e o GeoWebCache) instalados; o PostGIS, que contém os dados espaciais; o banco de dados da aplicação, com os dados do cenário; e o *Firewall*.

Devido à política de acesso inerente à aplicação, foi criado um controlador na aplicação do usuário para garantir o controle do acesso à aplicação. Após o cliente tentar se conectar à aplicação e ser identificado, este é encaminhado para a página contendo o mapa. Para realizar essa identificação do usuário, a aplicação precisa acessar o banco de dados da aplicação para verificar a existência do usuário no cenário criado pelo administrador, e também para ter acesso ao nível de acesso e às camadas que o usuário tem acesso.

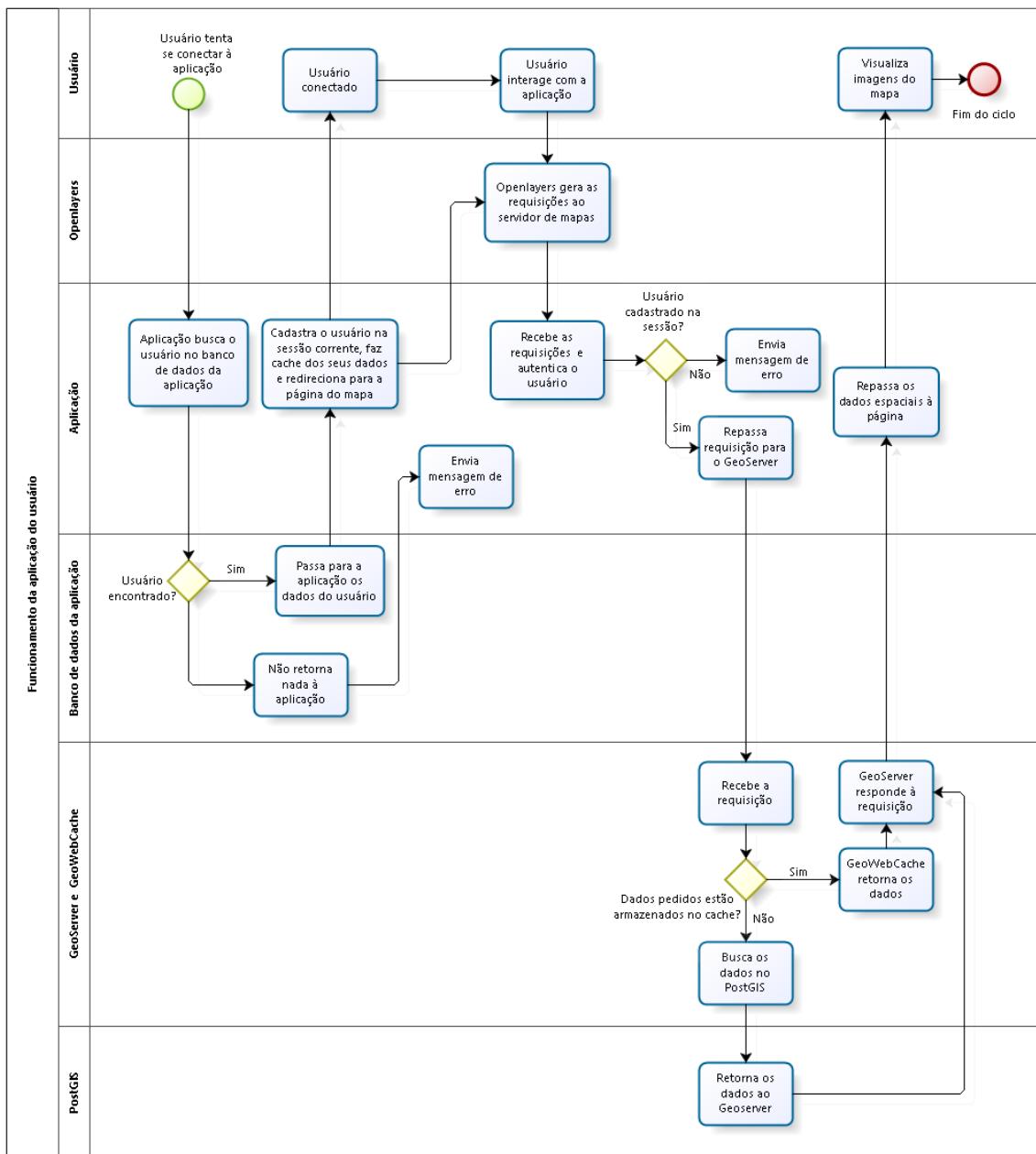
Para facilitar a exibição do mapa, optou-se pela utilização do OpenLayers no desenvolvimento da interface. No entanto, para a aplicação do usuário as requisições que o OL gera não foram repassadas diretamente ao GeoServer. Foi criado um controlador da aplicação que funciona como proxy reverso do GeoServer. Para isso, o controlador foi cadastrado como provedor do serviço WMS nas camadas do OL, de forma que esse controlador passou a receber as requisições que seriam enviadas ao GeoServer. Ao receber a requisição do OL, o controlador realiza mais algumas verificações em relação às permissões de acesso do usuário, que são explicadas detalhadamente na seção sobre o desenvolvimento da solução, e então repassa essa requisição ao GeoServer. Ao receber as requisições da aplicação do usuário, o GeoServer realiza as buscas necessárias no banco de dados do PostGIS para obter os dados referentes à requisição e, obtidos esses dados, envia uma resposta ao controlador. O GeoServer também pode utilizar o GeoWebCache para responder à requisição de forma mais rápida. A explicação da configuração do PostGIS no GeoServer encontra-se posteriormente na seção de desenvolvimento da solução, explicitando como o GeoServer configura sua comunicação com o PostGIS.

Além dos elementos já citados, o sistema também possui um *Firewall* que está fora do escopo desse trabalho, pois é parte do trabalho de um outro Projeto de Final de Curso (OLIVEIRA; SILVA, 2015) já citado anteriormente. Dentre as múltiplas funcionalidades desse *Firewall* descritas no trabalho citado, uma propriedade das mais relevantes no contexto

da aplicação do usuário é bloquear o acesso direto ao GeoServer, bloqueando a porta TCP 9090 na qual é executada essa aplicação. A aplicação do usuário que é executada em portas diferentes: porta TCP 80 (HTTP) e TCP 443 (HTTPS). A Figura 4.1 ilustra a arquitetura da aplicação do usuário e a Figura 4.2, o fluxograma que representa graficamente o funcionamento do sistema do ponto de vista do usuário, da forma como foi explicado acima.



**FIG. 4.1- Arquitetura da aplicação do usuário**



**FIG. 4.2 - Fluxograma de funcionamento da aplicação do usuário**

#### 4.2.2 Arquitetura da aplicação do administrador

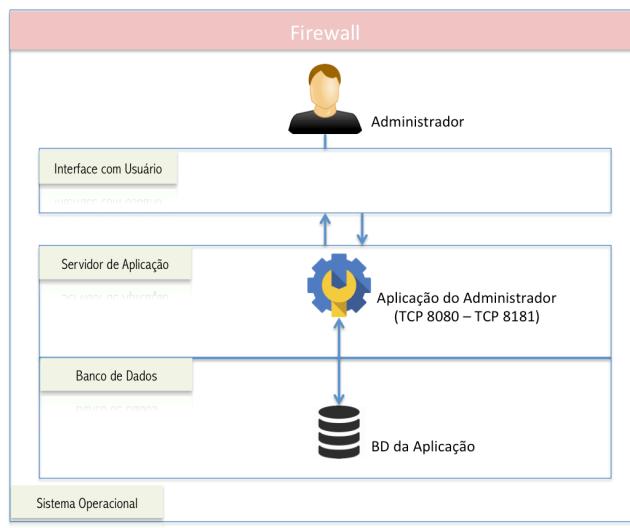
Os elementos presentes na arquitetura da aplicação do administrador são: a interface dessa aplicação; o Servidor de Aplicação contendo a aplicação do administrador (a aplicação do

usuário e o GeoServer usam o mesmo servidor); e o banco de dados da aplicação, o qual armazena os dados do cenário.

Um elemento fundamental na arquitetura dessa aplicação foi o banco de dados da aplicação, uma vez que o principal objetivo dessa aplicação foi permitir ao administrador editar esses dados para modelar o cenário da maneira que lhe fosse conveniente. Logo, praticamente todas as ações do administrador na interface implicam em alguma modificação nesse banco de dados.

Uma vez que o PostGIS é uma extensão do PostgreSQL, optou-se por utilizar o PostgreSQL como sistema de gerenciamento de banco de dados para o banco de dados da aplicação, utilizando o mesmo *driver* e pacotes de instalação para se comunicar tanto com o banco de dados espaciais quanto com o banco de dados da aplicação.

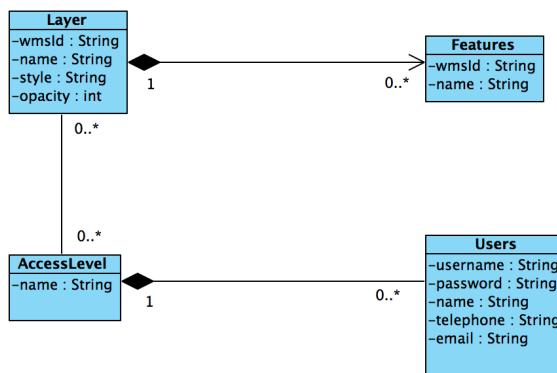
Outro aspecto relevante levantado é que o administrador do sistema tem acesso físico à máquina que hospeda o sistema. Dessa forma a aplicação do administrador funciona em uma porta TCP diferente da aplicação do usuário (8080 para HTTP e 8181 para HTTPS), de forma que o *firewall* bloqueie qualquer tráfego externo direcionado a essa porta.



**FIG. 4.3 - Arquitetura da aplicação do administrador**

### 4.3 Diagrama de classes

Com base nos requisitos do software e com o uso por parte do administrador e do usuário, foi elaborado um diagrama de classes com as classes que armazenam informações específicas do domínio dessa solução e que são necessárias para configuração do cenário. Existem 4 classes principais, são elas: *User*, *AccessLevel*, *Layer* e *Features*. O diagrama de classes é apresentado na Figura 4.4 abaixo.



**FIG. 4.4 - Diagrama de classes do domínio das aplicações**

A classe *User* representa um usuário do sistema e possui os atributos: *username*, que é o seu identificador único no sistema; *password*, que corresponde à sua senha utilizada para autenticação; *name*, um atributo opcional que indica o nome completo do usuário; *telephone*, outro atributo opcional que indica um telefone de contato; *email*, outro atributo opcional que indica um email de contato.

A classe *AccessLevel* representa um nível de acesso do sistema. Usuários só podem existir em um determinado nível de acesso, de forma que se o nível for apagado todos os seus usuários também o são. Possui como atributos: *name*, que é o seu identificador no sistema e representa o seu nome no sistema.

A classe *Layers* representa uma camada com os parâmetros pertinentes à sua visualização pelo usuário final. Os níveis de acesso podem acessar muitas camadas, e uma camada pode ser acessada por diversos níveis de acesso. Possui como atributos: *wmsId*, que é o seu identificador no sistema e também é o seu identificador no GeoServer, de forma que a camada possa ser referenciada pelo OpenLayers; *name*, que corresponde ao nome da camada

apresentado na interface do usuário; *style* que corresponde ao nome do estilo declarado no GeoServer (a descrição completa do que representa um estilo é apresentada mais adiante na seção 5.1.2); e *opacity*, que representa opacidade da camada variando de 0 (totalmente transparente) a 1 (totalmente opaco), um parâmetro utilizado quando há necessidade de sobreposição de camadas.

Toda camada pode possuir um conjunto de atributos associado a ela. A classe *Features* representa um atributo de uma camada, sendo que um atributo pertence a apenas uma camada e não pode existir sem que esteja associado à uma camada. Possui como atributos: *wmsId*; que é o seu identificador no sistema juntamente com a sua camada, e também é um identificador utilizado para referenciar esse recurso no GeoServer nas requisições do tipo GetFeatureInfo realizadas pelo OpenLayers; e *name*, o nome da feição visualizado pelo usuário.

As duas aplicações que foram desenvolvidas utilizam o mesmo banco. A aplicação do usuário utiliza para sua autenticação, usando o nome e a senha do usuário para identificá-lo, e para saber as camadas a que o usuário tem acesso. A aplicação do administrador usa todas as classes, uma vez que o administrador pode adicionar, editar e deletar camadas (que estejam disponíveis no banco de dados espaciais), suas características e estilo, níveis de acesso e usuários.

## 4.4 Persistência dos dados

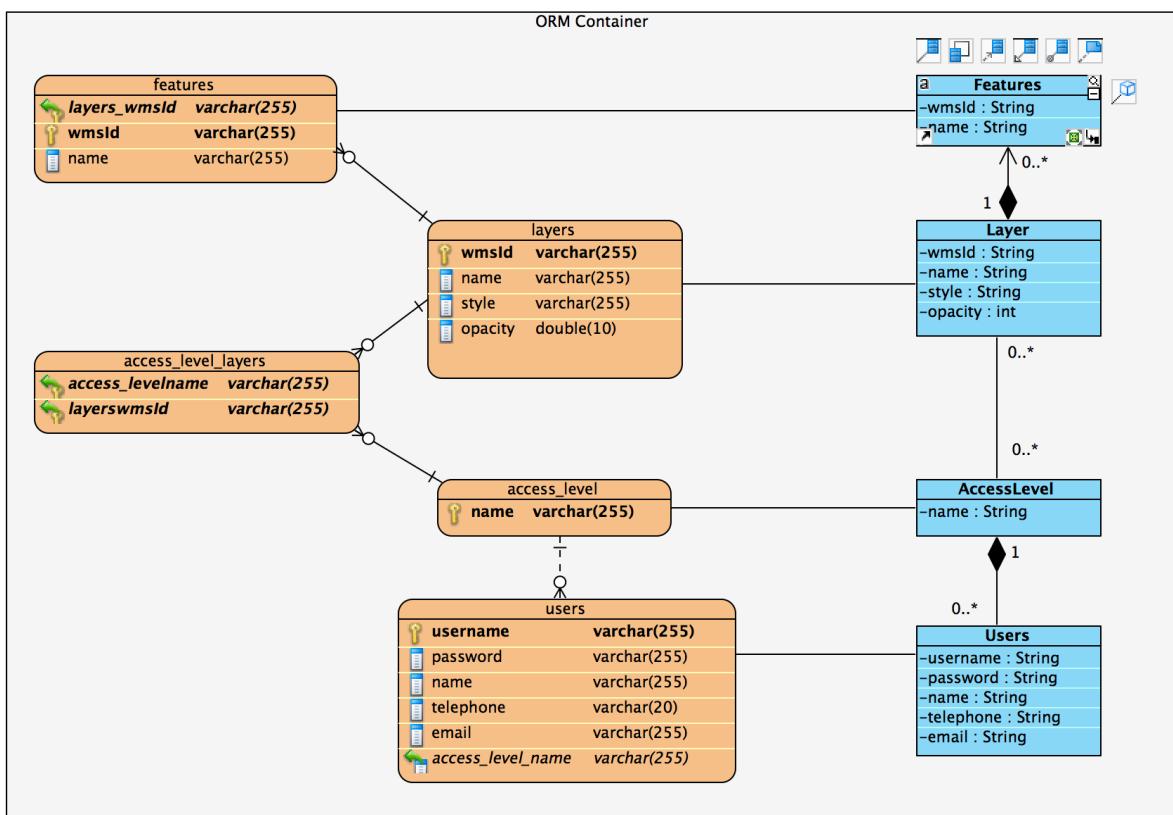
Para armazenar os dados do sistema em um banco de dados foram utilizados dois tipos de banco de dados. Um dos bancos de dados destina-se ao armazenamento dos dados espaciais e para tal foi utilizado o PostGIS. Esse banco de dados espaciais contém as informações geográficas necessárias para exibição das camadas e funciona como *back-end* de dados do GeoServer, como explicado anteriormente na seção 2.3. O outro banco de dados armazena as informações referentes ao cenário criado pelo administrador e é manipulado principalmente pela aplicação do administrador. Este último é referenciado ao longo deste relatório como banco de dados da aplicação.

Como foi utilizado o paradigma de orientação a objetos para o desenvolvimento das aplicações, para persistência dos dados foi utilizada uma ferramenta de mapeamento objeto-relacional (ORM), que facilita consideravelmente a tarefa de “transformar” objetos em

registros e registros em objetos. O Hibernate foi a ferramenta escolhida para realizar esse mapeamento. Trata-se de uma ferramenta ORM de código aberto e líder de mercado, que foi a inspiração para a especificação Java Persistence API (JPA) (CAELUM, 2015). A implementação da ferramenta utilizada foi a do Hibernate 4.

Com o Hibernate, o desenvolvedor não precisa se preocupar na codificação de consultas SQL e no respectivo código JDBC responsável por trabalhar com elas, pois o Hibernate abstrai o seu código SQL e toda a camada JDBC, gerando um código SQL que serve para um determinado banco de dados para o qual possua um *driver*, uma vez que cada banco de dados fala um “dialeto” diferente dessa linguagem. Dessa forma, torna-se possível trocar de banco de dados sem ter que alterar o código Java, já que isso fica como responsabilidade da ferramenta (CAELUM, 2015).

No entanto esse mapeamento deve refletir características da associação entre as classes tal como descrito acima. Para fazer o mapeamento utilizou-se um diagrama de mapeamento objeto relacional apresentado na Figura 4.5 abaixo, no qual as classes do domínio (em azul) são mapeadas nas entidades (em laranja) do banco de dados relacional, que correspondem às tabelas que serão criadas.



**FIG. 4.5 - Mapeamento objeto relacional do banco de dados da aplicação**

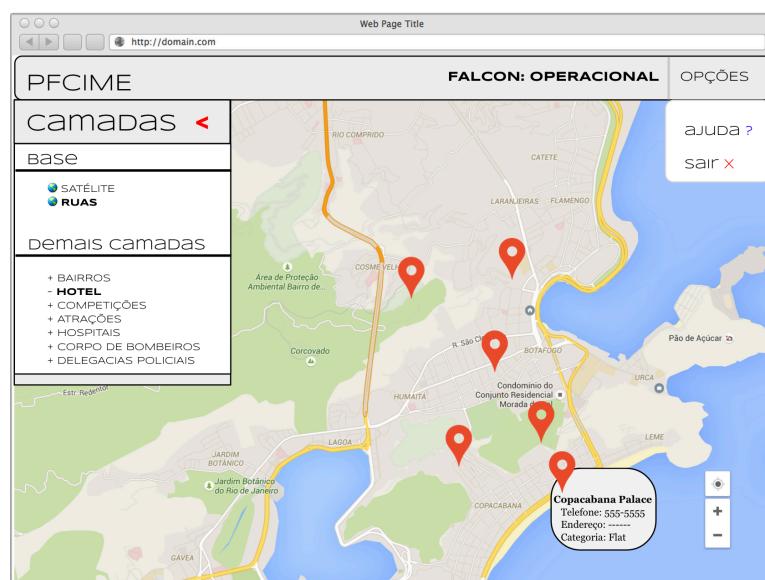
## 4.5 Projeto da interface

A interface das duas aplicações desenvolvidas foram feitas de forma a atender todos os requisitos do *software*. As duas contam com uma interface responsiva, o que foi feito por meio da utilização do *framework* Bootstrap. O Bootstrap permitiu por meio de seu *framework* a criação de aplicações com suporte a diversos tipos de navegadores e aos mais variados dispositivos e tamanhos de tela.

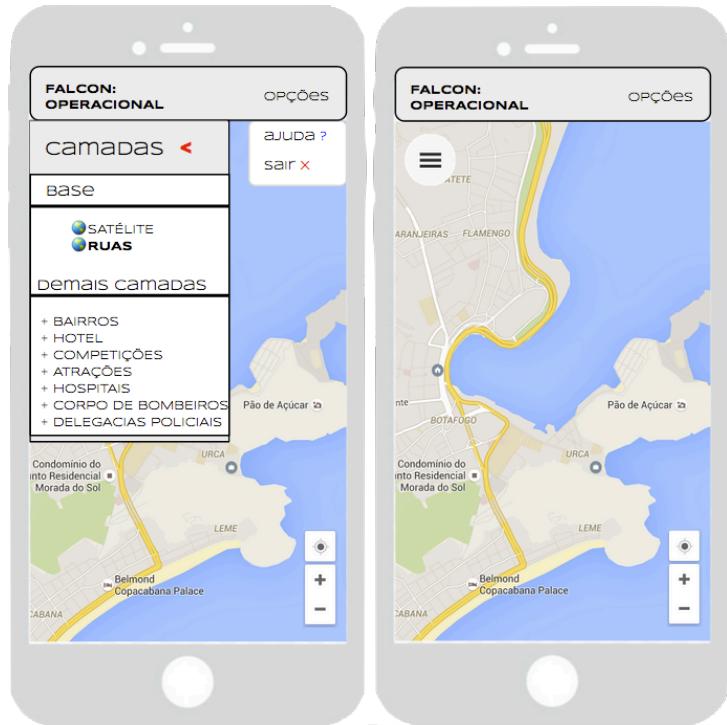
Além disso, para desenvolvimento subsequente da interface foram utilizadas as seguintes linguagens: HTML, para construção da página; JSP, para criação de páginas *web* geradas dinamicamente; CSS, para formatação do *layout* da página em conjunto com o Bootstrap; Javascript para modelar interações na página em conjunto com a biblioteca jQuery.

### 4.5.1 Aplicação do usuário

No projeto da interface da aplicação do usuário foram definidos dois modelos de interface com o usuário: um modelo para plataformas móveis e outro modelo para plataformas *desktop*. Esses modelos são apresentados nas Figuras 4.6 e 4.7:



**FIG. 4.6 - Projeto da interface com o usuário para dispositivos *desktop***



**FIG. 4.7 – Projeto da interface com o usuário para dispositivos móveis**

Pode-se observar nas figuras 4.6 e 4.7 que ambos os modelos possuem elementos semelhantes, mas que foram adequados à tela de cada dispositivo. Essa responsividade faz parte dos requisitos da aplicação e, no projeto da interface, foi definida a utilização do framework responsivo Bootstrap para a construção da interface com essa característica. A vantagem em utilizar o Bootstrap foi que este permitiu que a aplicação fosse desenvolvida com um mesmo código tanto para dispositivos móveis como para dispositivos *desktop*.

Dentre os elementos presentes na interface pode-se destacar a presença de um menu de navegação na parte superior que é utilizado para exibição do nome do usuário seguido do nível de acesso correspondente, no exemplo acima “Falcon” é o nome do usuário e “Operacional” é o nível de acesso correspondente. O menu de navegação exibe também na versão *desktop* o título da aplicação definido como “PFCIME”, nesse caso. Esse menu também é utilizado para exibição da aba “Opções” que é utilizada para sair da conta e obter ajuda acerca da utilização do programa. A opção “Ajuda” leva a uma breve explicação em uma janela de diálogo sobre como o usuário deve interagir.

A seguir, na barra lateral esquerda pode-se observar um painel para seleção de camadas. Existem as camadas base que não são cadastradas pelo administrador e são obtidas por serviços como o *Google Maps*, *Open Street Map* e o *Bing Maps*. O usuário pode escolher qual

camada base deseja utilizar para visualização do mapa, no exemplo acima a camada selecionada é a de “ruas”, fornecida pelo serviço *Google Maps*, que exibe informações de navegação. Uma outra possibilidade de camada base é uma exibição via satélite do mapa.

Abaixo da seleção de camada base, o usuário pode selecionar as camadas que foram cadastradas pelo administrador do sistema para o seu nível de permissão. Essas camadas são adicionados em cima da camada base, portanto, mais de uma camada dessas pode estar presente no mapa a qualquer momento, bastando clicar no nome da camada quando este estiver vermelho para adicioná-la ao mapa, e clicar no nome quanto este estiver verde para retirá-la.

Para não prejudicar a navegação do usuário ele pode minimizar o painel de camadas a qualquer momento, bastando selecionar a aba com nome “Camadas”, tanto para minimizar como para restaurar o painel.

Pode-se observar na Figura 4.3 a presença de pontos de marcação indicando hotéis presentes na região selecionada, exibidos sobre a camada base de ruas. Ao clicar em um desses pontos obtém-se informações adicionais acerca do elemento selecionado, sendo que tais informações variam de acordo com cada camada. Para a camada “hotel” pode-se observar informações como nome do hotel, telefone, endereço e categoria, algo que não seria informado ao selecionar um elemento da camada bairros, por exemplo.

## 5 DESENVOLVIMENTO DA SOLUÇÃO

Para o desenvolvimento de ambas as aplicações, a tecnologia escolhida foi Java para Web, com Servlets e JSP, uma decisão conjunta com os autores do PFC (OLIVEIRA; SILVA, 2015). A vantagem de utilizar essas tecnologias é que o GeoServer também as utiliza para prover os seus serviços, portanto os requisitos existentes para o funcionamento do GeoServer são os mesmos para as aplicações desenvolvidas.

Como servidor de aplicação foi utilizado o Glassfish 4.1, ambas as aplicações foram empacotadas em arquivos de extensão WAR e instaladas no Glassfish. O arquivo WAR do GeoServer também foi instalado no GlassFish, que então desempacota o arquivo *web* e automaticamente monta e executa o GeoServer.

Como dito anteriormente, visando facilitar o processo de instalação e configuração foram desenvolvidas ferramentas para auxiliar o administrador nesse processo, utilizando a linguagem *Shell Script*. O uso dessas ferramentas é apresentado na seção 7.1, mas os seus objetivos são apresentados a seguir na seção 5.1.

### 5.1 Configuração do ambiente

Pode-se observar que o sistema depende de uma série de componentes e tecnologias que devem interagir de forma sinérgica para que o sistema funcione corretamente. Diversas aplicações e tecnologias devem ser instaladas na máquina que hospeda o sistema, além disso, diversas configurações devem ser realizadas para que o sistema funcione corretamente.

Visando facilitar o processo de instalação e configuração por parte do administrador do sistema, além da aplicação do administrador foram desenvolvidas diversas aplicações de linha de comando que buscam automatizar diversas tarefas que tomariam tempo do administrador e que requerem conhecimento muito específico sobre as tecnologias empregadas. Essas ferramentas foram desenvolvidas especificamente para sistemas Linux Ubuntu, e testadas

apenas na versão 14.04, de forma que outras versões podem funcionar corretamente mas não foram testadas e nem foram foco do desenvolvimento das ferramentas.

O uso dessas ferramentas é detalhado mais adiante nesse trabalho, no entanto os conhecimentos necessários para o seu desenvolvimento são relatados abaixo, de forma a permitir que o sistema possa ser configurado em outros sistemas operacionais no futuro.

### 5.1.1 Instalação

Como se pode observar na arquitetura das aplicações do usuário e do administrador, as tecnologias necessárias para implantação do sistema são: GeoServer, PostGIS, Servidor de Aplicação, Aplicação do Administrador, Aplicação do Usuário e um Sistema Operacional suportando todas elas.

No entanto as aplicações e o GeoServer têm como dependência a instalação de uma *Java Virtual Machine (JVM)* para executar as aplicações. O GeoServer apresenta problemas de incompatibilidade com algumas versões do Java, de forma que a versão da *JVM* a ser instalada deve ser escolhida com cuidado pois o sistema pode apresentar falhas de natureza desconhecida ocasionadas por um mau funcionamento do GeoServer.

A versão do GeoServer utilizada no projeto foi a 2.8.0, que foi testada para versão do Java 7, e que apresenta alguns problemas com a versão mais recente do Java, que é o Java 8 (OSGeo, 2015b).

Portanto, a versão escolhida da *JVM* a ser instalada corresponde ao Java 7, e a versão que foi instalada corresponde à implementação da Oracle dessa *JVM*, o *Java Runtime Environment (JRE)* versão 1.7u80. As aplicações do usuário e do administrador foram desenvolvidas levando isso em consideração, de forma a não utilizar certos recursos presentes na versão mais recente do Java, que podem causar problemas de implantação e execução no Servidor de Aplicação.

A versão do Glassfish também possui dependência da versão do Java, de forma que a versão 4.1 possui compatibilidade com o Java 7. Qualquer outro Servidor de Aplicação pode ser utilizado contanto que satisfaça essa condição (Tomcat ou Jetty, por exemplo), no entanto as ferramentas desenvolvidas para configuração visam configurar um servidor do tipo Glassfish.

As configurações que foram necessárias no Glassfish dizem respeito a assinalar as portas corretas a cada aplicação instalada nele. Configurando-se qualquer outro servidor dessa forma permite que ele substitua o Glassfish sem prejudicar o funcionamento da aplicação. Deve-se garantir que a aplicação do usuário funcione na porta TCP 80 (HTTP) e TCP 443 (HTTPS), que a aplicação do administrador funcione na porta TCP 8080 (HTTP) e TCP 8181 (HTTPS) e que o GeoServer funcione na porta 9090 (HTTP).

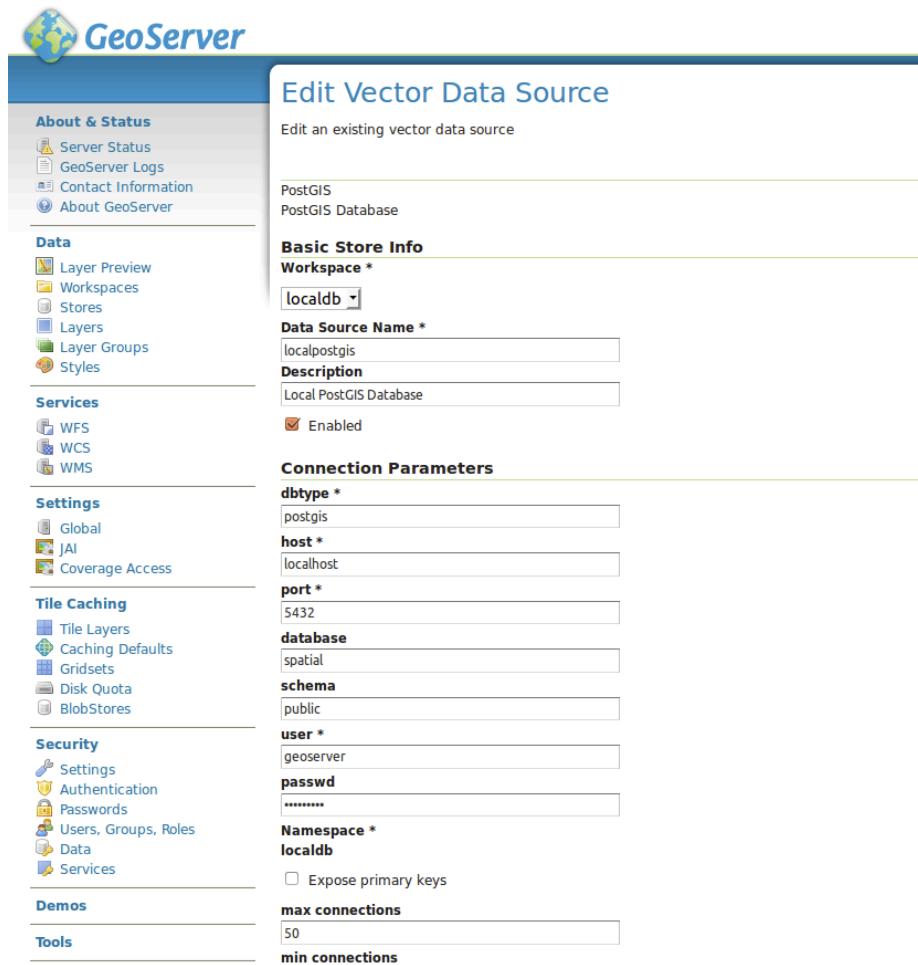
O Glassfish deve conter no seu diretório de bibliotecas os *drivers* do PostgreSQL, uma vez que as duas aplicações utilizam esses *drivers* para se comunicar com o banco de dados da aplicação e não podem conter esses *drivers* nos seus arquivos de implantação uma vez que a duplicação dessa informação causa um problema de *memory leak* no Glassfish.

A versão do PostgreSQL instalada é a versão 9.3, e a versão do PostGIS é a versão 2.1. Versões anteriores do PostgreSQL possuem modos diferentes de configuração do banco de dados, de tal forma que versões anteriores à 9.0 do PostgreSQL podem não funcionar com os *scripts* de configuração desenvolvidos.

Os *scripts* de configuração do banco de dados visam a criação de dois bancos de dados com os nomes de *spatial* e *application*, que respectivamente correspondem ao banco de dados espaciais e ao da aplicação, além da criação de *roles* para acesso ao banco, de forma que as aplicações do administrador e do usuário não acessem o banco de dados espacial.

Para que as aplicações do administrador e do usuário funcionem corretamente deve-se assegurar que o banco de dados *application* contenha *roles* com os nomes de *admin\_application* e *user\_application*, com senhas especificadas no instalador. Essa condição é necessária para que essas aplicações acessem esse banco de dados, do contrário todas as requisições a serviços que busquem informações no banco de dados irão apresentar um código de erro HTTP 500.

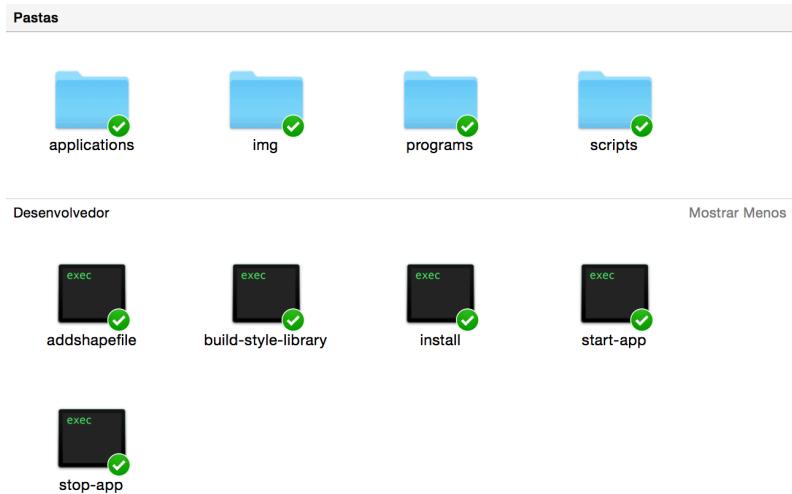
Da mesma forma, para que o GeoServer acesse o PostGIS deve-se especificar um *role* com o nome *geoserver* para o banco de dados *spatial*. A versão do GeoServer contida no instalador já apresenta um GeoServer configurado para referenciar corretamente o banco de dados *spatial*. Essa configuração pode ser visualizada ou reconfigurada na opção “*Stores*” do menu de configuração do GeoServer, como apresentado abaixo:



**FIG. 5.1 - Configuração do banco de dados espaciais no GeoServer**

Todos os aspectos mencionados anteriormente nessa seção foram considerados para criação do instalador, de forma que o administrador não precisa se preocupar com nenhum desses aspectos.

O administrador do sistema, ao copiar o instalador do sistema para sua máquina, observa a organização do diretório ilustrada na Figura 5.2.



**FIG. 5.2 - Organização do diretório do instalador do sistema**

O administrador deve se preocupar apenas com os 5 arquivos executáveis e com a pasta “*img*” cujo objetivo é explicado na próxima seção.

Os *script install* tem como finalidade executar toda instalação das bibliotecas e aplicações necessárias para o correto funcionamento do sistema. Ao término da instalação, terão sido instalados a *JRE 1.7u80*, o *Geoserver 2.8.0* referenciando o banco de dados espaciais no *PostGIS*, o *Glassfish 4.1*, o *PostgreSQL 9.3* e o *PostGIS 2.1*.

O *script start-app* tem por objetivo iniciar ou reiniciar o servidor de aplicação, ao passo que *stop-app* deve interrompe-lo.

O *script build-style-library* foi apresentado anteriormente e é utilizado em conjunto com o diretório “*img*”. É essencial que eles estejam no mesmo diretório.

Por fim o *script addshapefile* procura auxiliar o administrador no processo de adicionar dados espaciais ao banco de dados. Vale ressaltar que existem outras ferramentas para realizar essa tarefa, como o pgAdmin e o QGIS, que convertem arquivos *shapefile* em tabelas no banco de dados PostGIS ou PosgreSQL. Com o *script addshapefile* é possível identificar o caminho do arquivo de extensão *shapefile* (.shp) e o nome de tabela desejado a ser criado no banco de dados. Para identificar o caminho do arquivo *shapefile* basta utilizar a opção “*-f*”, seguida do caminho do arquivo. Para especificar o nome da tabela, basta utilizar a opção “*-t*”, seguida do nome da tabela que deseja-se criar. Caso esse parâmetro não seja especificado, o nome do arquivo sem o caminho de diretório e sem extensão é utilizado. Por exemplo o comando: “*./addshapefile -f /Dados/hoteis.shp*”, adiciona os dados do arquivo numa tabela com o nome *hoteis*. Também são disponibilizadas as opções “*--append*” e “*--drop*”, que

alteram o modo como os dados serão adicionados. Na primeira, os dados são adicionados a uma tabela já existente, o que pode ser utilizado na geração de dados dinâmicos por exemplo, que inserem dados espaciais no banco de dados ao longo do tempo. Na segunda uma tabela existente é destruída e uma nova é criada para se adicionar os arquivos.

### 5.1.2 Estilos do GeoServer

Como já foi citado anteriormente, as camadas têm em sua configuração um estilo associado, o qual define a simbologia adotada para representação da camada. O Estilo da camada é um parâmetro opcional da requisição GetMap do serviço WMS para selecionar o recurso presente no servidor de mapas com o nome passado como parâmetro. Não é uma especificação do OGC, que um servidor WMS possua suporte a estilos. Servidores que possuem suporte para tal são chamados de “*SLD enabled*”, uma vez que SLD (*Style Layer Descriptor*) é justamente a extensão dos arquivos de estilo especificada pelo OGC. O GeoServer é um servidor “*SLD enabled*” que permite configurar os estilos das camadas por meio de arquivos SLD. Esses arquivos possuem formato XML e definem uma linguagem de marcação para criação de estilos para as camadas: informações que serão utilizadas pelo servidor de mapas para decorar as informações geográficas.

No entanto, não é possível adicionar imagens diretamente ao GeoServer de forma a utilizá-las como estilos. Isso foi um problema identificado no cenário utilizado, uma vez que existem diversas camadas pontuais que necessitam ser substituídas por imagens.

Os arquivos de definição de estilo (SLD) são capazes de referenciar recursos externos por meio de URL’s, no entanto o sistema não deve depender de conexão com Internet para o seu correto funcionamento.

Não foi encontrada uma solução disponível de importação presente no GeoServer, nem em um dos *plug-ins* disponíveis no mercado para configurar os estilos dessa forma. A solução criada para esse problema envolveu a utilização da API REST do GeoServer e alocação das imagens no próprio GeoServer para disponibilização das URL’s.

Essa solução consistiu na criação de uma pasta “*img*” no instalador, no mesmo diretório do *script* “*build-style-library*”. Esse comando permite instalar no diretório de dados do GeoServer que está implantado no servidor, as imagens que deseja-se transformar em estilos

disponíveis (com extensão svg, png, jpeg ou jpg), e envia comandos definidos na API REST para inserir e atualizar estilos, gerando os arquivos SLD necessários automaticamente. A pasta “img” pode ser organizada em diretórios aninhados, de forma que os recursos gerados irão apresentar um separador *underline* na sua identificação, permitindo uma melhor organização e importação de imagens e bibliotecas de imagens pré-existentes.

O instalador conta com a biblioteca de imagens de extensão SVG do *software* QGIS. Arquivos SVG são especificados pelo W3C (W3C, 2011). Além dessa biblioteca de imagens o QGIS também pode ser customizado com Adobe Illustrator, GIMP, etc.

Dessa forma, esse *script* permite construir uma biblioteca de estilos baseada na que está presente no QGIS, e que pode ser facilmente estendida.

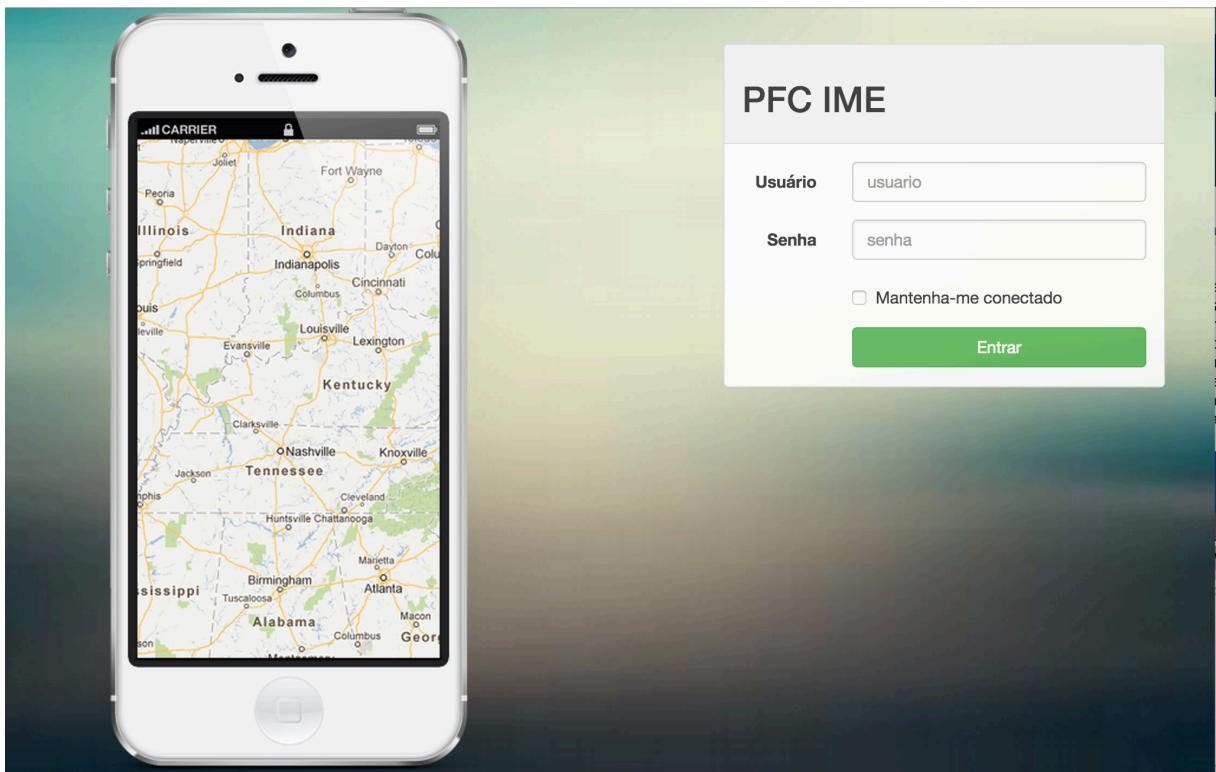
## 5.2 Desenvolvimento da aplicação do usuário

Para a aplicação do usuário foram criadas duas páginas web, uma de *login*, onde o usuário se conecta à aplicação, e outra de acesso ao mapa, que é a página principal da aplicação. Para a página de *login* foi criado apenas um controlador para lidar com as requisições dessa página, visando principalmente realizar a verificação do usuário. Para a página contendo o mapa, foram criados três controladores e um filtro. O primeiro controlador lida com as requisições WMS. O segundo serve para informar à página o nome do usuário e o seu nível de acesso, para que a página possa exibir essas informações na barra de navegação, como se pode observar no projeto da interface. O terceiro controlador foi criado para listar as camadas que o usuário pode visualizar na barra lateral à esquerda da página. Por fim, o filtro foi criado para realizar uma autenticação do usuário em cada requisição.

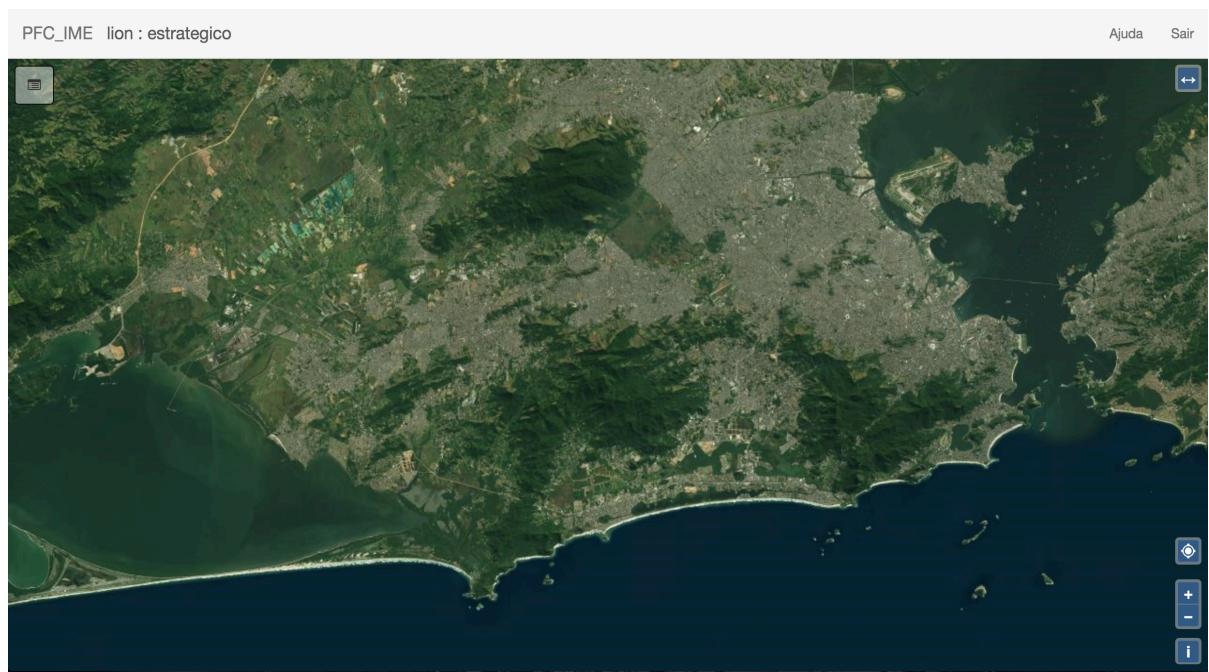
### 5.2.1 Desenvolvimento da interface

A interface da aplicação do usuário foi desenvolvida de acordo com o que foi definido no projeto da interface, contando, porém, com algumas características adicionais que agregaram

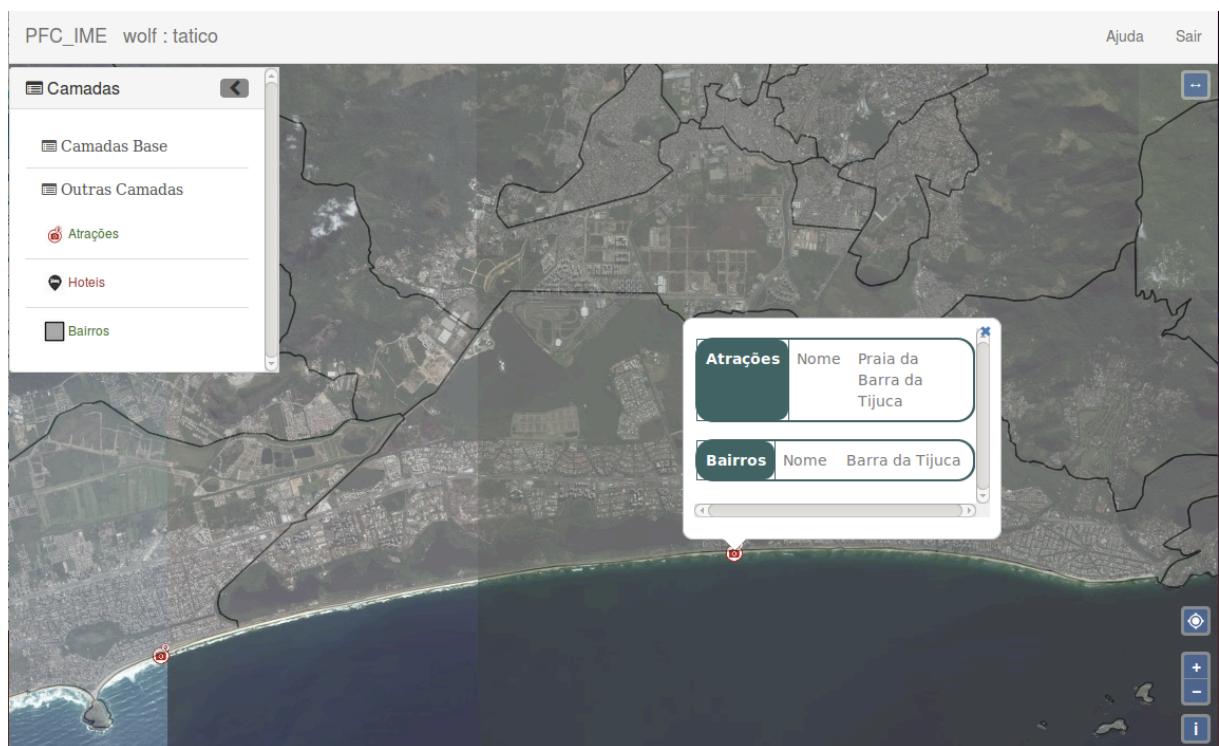
mais funcionalidade à aplicação. A Figura 5.3 ilustra a página de *login* da aplicação. As Figuras 5.4 a 5.7 ilustram a interface desenvolvida para a página de mapa da aplicação em dois possíveis tamanhos de tela, um para plataforma móvel e outro para plataforma *desktop*.



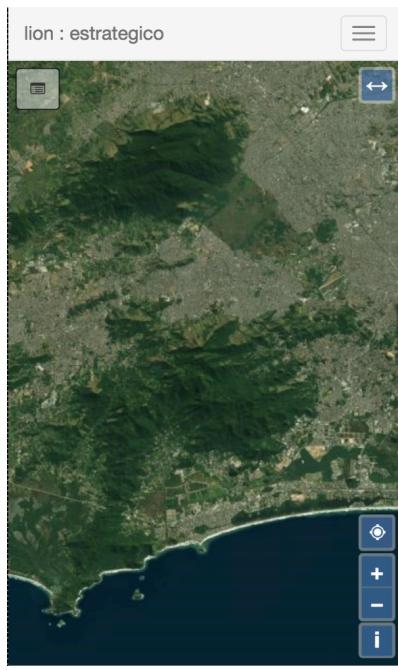
**FIG. 5.3 - Interface da página de *login* da aplicação do usuário em um tamanho de tela para plataforma *desktop***



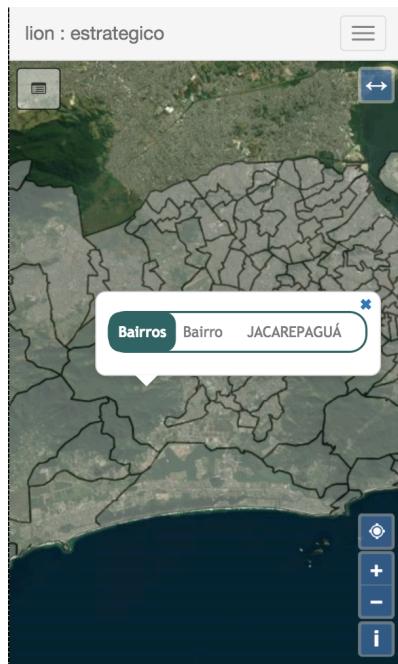
**FIG. 5.4 - Interface da aplicação do usuário em um tamanho de tela para plataforma desktop, com nenhuma camada sendo exibida e a lista de camadas minimizada em um botão na parte superior e esquerda da tela**



**FIG. 5.5 - Interface da aplicação do usuário em um tamanho de tela para plataforma desktop, com duas camada e suas feições sendo exibidas, e a lista de camadas maximizada**



**FIG. 5.6 - Interface da aplicação do usuário em um tamanho de tela para plataforma móvel, com nenhuma camada sendo exibida e a lista de camadas minimizada em um botão na parte superior e esquerda da tela**



**FIG. 5.7 - Interface da aplicação do usuário em um tamanho de tela para plataforma móvel, com uma camada e uma feição da camada sendo exibidas, e a lista de camadas minimizada em um botão na parte superior e esquerda da tela**

Quanto à página de *login*, esta conta apenas com os espaços para o usuário entrar com seu nome de usuário e senha e o botão para este se conectar à aplicação.

Dentre os elementos presentes na interface pode-se destacar a presença do menu de navegação como descrito no projeto da interface. No exemplo acima “PFC\_IME” é o nome da aplicação, “lion” é o nome do usuário e “estratégico” é o nível de acesso correspondente. Uma diferença em relação ao projeto da interface foi que as opções de “Ajuda” e “Sair” foram colocadas diretamente no menu de navegação para um tamanho maior de tela, somente sendo condensadas em um sub-menu em tamanhos de tela menores.

A seguir, na barra lateral esquerda pode-se observar um painel para seleção de camadas. No desenvolvimento da interface, foi adicionada uma legenda ao lado de cada camada com o estilo dessa camada, exceto para as camadas base, que são camadas obtidas pelo *Bing Maps*. Somente são exibidas para o usuário as camadas que foram cadastradas pelo administrador do sistema para o nível de acesso ao qual o usuário pertence.

Para não prejudicar a navegação do usuário ele pode minimizar o painel de camadas a qualquer momento, bastando selecionar a aba com nome “Camadas”, tanto para minimizar como para restaurar o painel.

Como foi definido no projeto da interface, quando uma camada está sendo exibida é possível visualizar informações de um elemento dessa camada, através de um clique em cima desse elemento. Lembrando que essas informações são atributos pré-definidos pelo administrador. Se o usuário clicar em um ponto do mapa onde existem elementos de mais de uma camada, são exibidas as informações de apenas um elemento por camada. Caso, para uma mesma camada, muitos elementos estejam aglutinados em um mesmo ponto, devido ao zoom aplicado ser pequeno, então um elemento é escolhido a critério do GeoServer para ter suas feições exibidas. Se o usuário quiser obter informações de um hotel específico, basta aplicar mais zoom ao mapa até que o hotel possa ser visualizado separadamente (dentro do limite máximo de zoom fornecido pelo Geoserver), e então clicar em cima desse hotel. Os dados dos elementos das camadas são exibidos em tabelas contendo três colunas (sendo uma tabela por camada que possua feições): uma coluna para a camada, identificando a camada a qual o elemento pertence; outra para os nomes das feições e outra para a informação atrelada à feição. Essas características variam de acordo com a camada que está sendo exibida. No exemplo acima, as camadas “Bairros” e “Atrações” estão sendo exibidas no mapa e uma seleção feita no mapa fez com que aparecesse uma pop-up com uma tabela contendo as informações do bairro e da atração contidos naquele ponto.

Os botões na lateral direita do mapa, constituem alguns controles que foram criados para proporcionar funcionalidades adicionais ao usuário. De cima para baixo, tem-se: um botão para acionar a exibição da aplicação em tela cheia, o que pode ser bastante útil quando se trata de um dispositivo móvel; um botão para localizar o usuário no mapa, no caso de o dispositivo que o usuário estiver utilizando possuir GPS; botões para aumentar ou diminuir o zoom no mapa; e, por fim, um botão para exibir as informações do serviço utilizado para prover as camadas base que, na aplicação desenvolvida, foi o *Bing Maps*.

Outro elemento adicionado à interface foi uma barra de progresso que aparece na parte inferior da tela sempre que uma nova camada é carregada pela primeira vez no mapa, indicando o avanço do carregamento da imagem. Isso é particularmente útil quando a conexão com o servidor de mapas está muito lenta ou quando se trata de uma camada com muitas informações, logo, que demora mais a carregar, permitindo ao usuário acompanhar o andamento do processo e não simplesmente inferir que a demora é um problema de conectividade.

### 5.2.2 Identificação do usuário

Como o acesso à aplicação do usuário é restrito aos usuários cadastrados pelo administrador (como definido nos requisitos da aplicação), tornou-se necessário realizar uma identificação do usuário antes que ele tenha acesso à página da aplicação contendo o mapa. Então, inicialmente o usuário tenta se conectar à aplicação informando seu nome de usuário e senha na página de *login*. As requisições provenientes da página de *login* passam por um controlador, o qual recebeu o nome de *LoginServlet* no projeto da aplicação.

O controlador *LoginServlet*, ao receber um requisição HTTP do tipo POST, busca pelos parâmetros de nome e senha que foram preenchidos na página de *login* pelo usuário. Se esses valores forem vazios, o usuário é direcionado para a página de *login* novamente, para preencher os campos deixados em branco. Caso contrário, o controlador inicialmente busca no *UserManager* um nome de usuário igual ao que foi preenchido na página. Se o *UserManager* encontrar o usuário, e a senha cadastrada for igual a senha preenchida na página, então o controlador cria uma sessão para o usuário e armazena nos atributos dessa sessão o nome do usuário e os identificadores das camadas a que ele tem acesso. Vale ressaltar que o

UserManager precisa acessar o banco de dados para acessar as informações dos usuários. Dessa forma, adicionar na sessão o nome do usuário e as camadas que ele pode acessar funciona como uma forma de *cache* das informações do usuário. Com isso, o controlador que lida com as requisições de mapa pode verificar a autenticidade do usuário e se ele tem acesso a camada que está sendo requisitada sem a necessidade de acesso ao banco de dados, já que a sessão armazena essas informações, aumentando o desempenho do sistema, uma vez que uma interação do usuário com o mapa gera um grande número de requisições WMS. A sessão permanece aberta por trinta minutos ou até que o usuário se desconecte da aplicação.

### 5.2.3 Exibição do mapa

Após se conectar à aplicação, o usuário é redirecionado para a página de visualização do mapa. Para exibição do mapa e das camadas, o OL realiza requisições WMS que são direcionadas ao controlador da aplicação chamado *WMSProxyServlet*. Esse controlador é responsável por checar as camadas que podem ser acessadas pelas requisições WMS, utilizando-se do *cache* das informações do usuário armazenadas no momento em que este se conecta. Ao receber a requisição, o *WMSProxyServlet* inicialmente nega todas as requisições que não são do tipo WMS, enviando um código de erro HTTP 403, que identifica uma requisição proibida. Em seguida, busca na sessão da requisição o atributo contendo os identificadores das camadas que o usuário tem acesso. Se todas a(s) camada(s) na requisição forem identificadas entre as camadas que o usuário tem acesso, então o controlador repassa a requisição para o GeoServer. Caso contrário, se existir alguma camada a qual o usuário não tem acesso, o controlador envia o código de erro HTTP 401 identificando acesso não autorizado. No entanto, caso seja feita uma requisição WMS GetCapabilities à aplicação, esta responderá, caso o usuário seja autenticado, todas as camadas que estão disponíveis no Geoserver, incluindo as que o usuário não tem acesso. Essa requisição não pôde ser limitada apenas ao administrador, pois dessa forma os dados da aplicação não poderiam ser visualizados em softwares de geoprocessamento, o que foi um dos objetivos desse trabalho.

Outro controlador criado para a página com o mapa foi o *MapServlet*. Esse controlador basicamente acessa o banco de dados da aplicação, por meio do *UserManager*, para obter o nome do usuário e o seu nível de acesso, e então repassar esses valores para a página gerada

dinamicamente. Isso foi necessário para exibir esses dois valores na barra de navegação da página principal, como mostrado na interface da aplicação. Esse controlador apenas é acionado na inicialização da página, ou seja, o acesso ao banco de dados da aplicação é realizado apenas uma vez.

O outro controlador feito para a página principal foi o *ListLayerServlet*, com função semelhante ao anterior de inicializar elementos da página. Esse controlador foi criado para listar as camadas que o usuário pode acessar na barra lateral à esquerda da página principal. Para isso, o controlador acessa o banco de dados da aplicação, por meio do *UserManager*, para obter as camadas a que o usuário tem acesso, e então encapsula essas informações no formato JSON (*Javascript Object Notation*) e envia como resposta. A página principal obtém essas informações de forma assíncrona e as utiliza para configurar no Openlayers todos os aspectos das camadas. Vale ressaltar que o acesso ao banco de dados foi feito através do *UserManager*, pois este permite encontrar o usuário cujo nome consta na sessão e obter os dados desse usuário, como o nível de acesso e as camada a que tem acesso.

### 5.2.4 Filtro de autenticação

Com relação à autenticação do usuário, foi utilizado na aplicação um filtro para interceptar as requisições do tipo WMS, o qual foi chamado no projeto de *AuthenticationFilter*. Esse filtro foi criado com o objetivo de autenticar o usuário em todas as requisições WMS antes que elas cheguem ao controlador *WMSProxyServlet*. Ao receber a requisição, o filtro primeiramente verifica se o atributo da sessão destinado ao nome do usuário possui um valor. Se possui, significa que esse nome de usuário foi configurado no momento em que o usuário se conectou. Com isso, o filtro repassa a requisição para o controlador. Se não possui, o filtro verifica se a requisição possui um cabeçalho de autenticação do usuário, o que ocorre no caso de a requisição utilizar o método *Basic HTTP Authentication*. Requisições desse tipo são realizadas por softwares de geoprocessamento, como o QGIS, que é abordado adiante na seção 6. Se possuir o cabeçalho, então o filtro identifica o nome de usuário e senha nesse cabeçalho, cadastra esses valores na sessão corrente e repassa a requisição para o controlador. Se não possuir, então envia um código de erro HTTP 401 de acesso não autorizado.

## 5.3 Desenvolvimento da aplicação do administrador

Essa aplicação foi criada com o objetivo de fornecer uma interface para que o administrador do software desenvolvido possa criar ou modificar o cenário que é visualizado na aplicação do usuário. Essa aplicação não foi o objetivo principal desse trabalho, mas sem ela o administrador precisaria trabalhar diretamente com o banco de dados, o que não seria o ideal, pois passaria a ser um requisito do administrador saber trabalhar com bancos de dados relacionais.

Para essa aplicação foi criada apenas uma página *web*. Essa página foi dividida em três grandes elementos que permitem operações de inclusão, modificação, busca e remoção (operações *CRUD*) de camadas, níveis de acesso e usuários respectivamente.

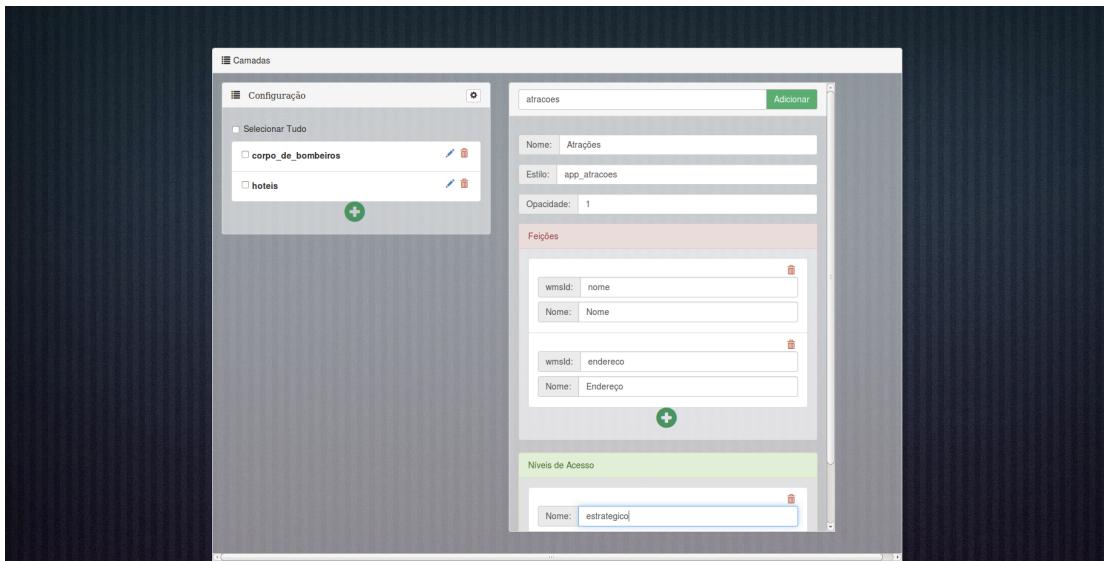
### 5.3.1 Interface com o usuário

A tela inicial da aplicação apresenta todos os três elementos colapsados, da forma como é apresentado na Figura 5.8, abaixo:

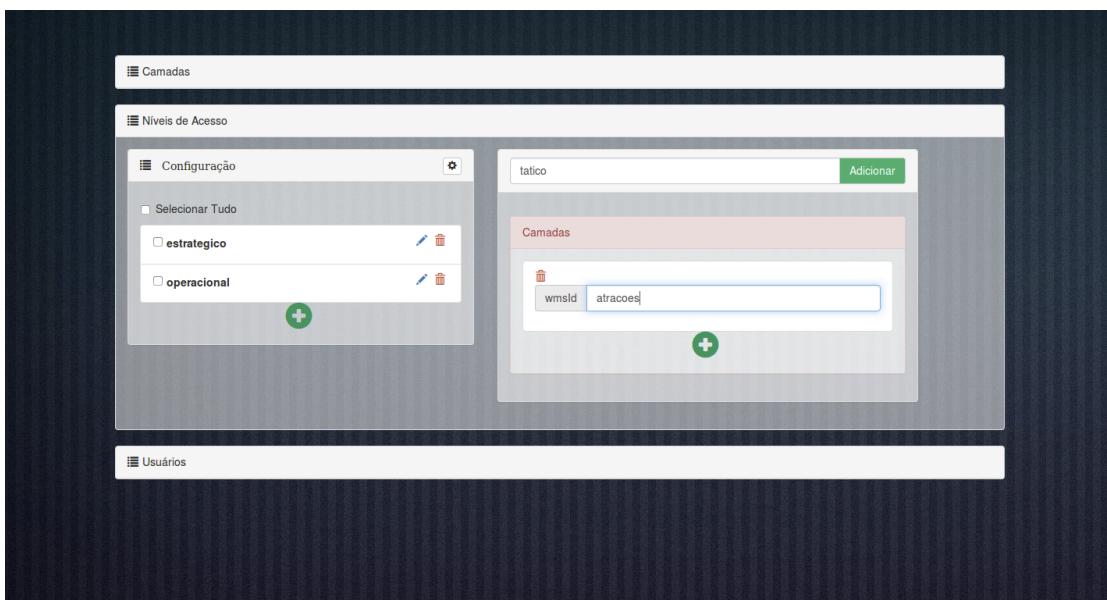


**FIG. 5.8 - Tela inicial da aplicação do administrador**

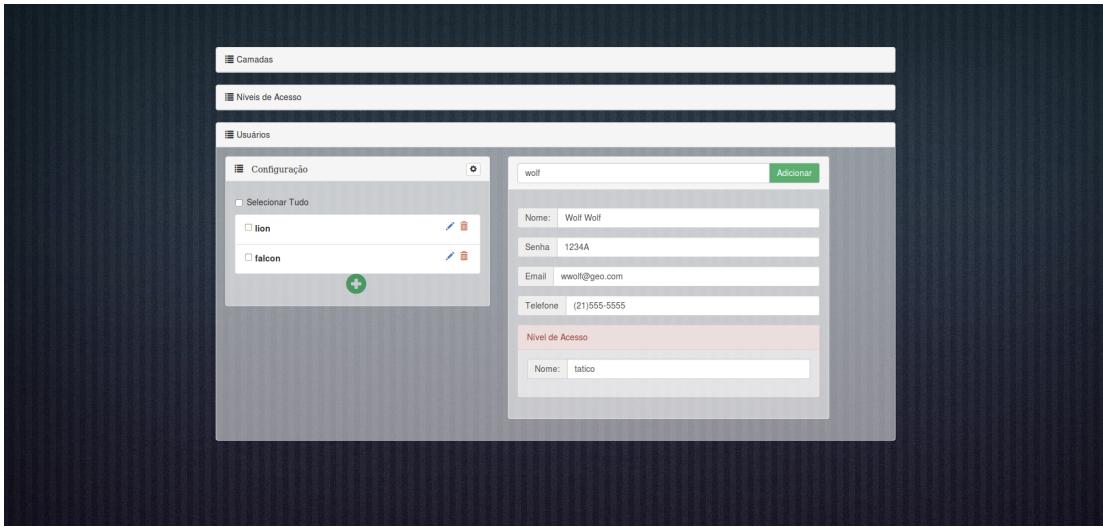
O administrador pode então selecionar uma das três colunas com a qual deseja interagir e realizar uma das operações citadas. O resultado para cada uma das seleções é apresentado nas figuras 5.9 a 5.11:



**FIG. 5.9 - Configuração das camadas**



**FIG. 5.10 - Configuração dos níveis de acesso**



**FIG. 5.11 - Configuração dos usuários**

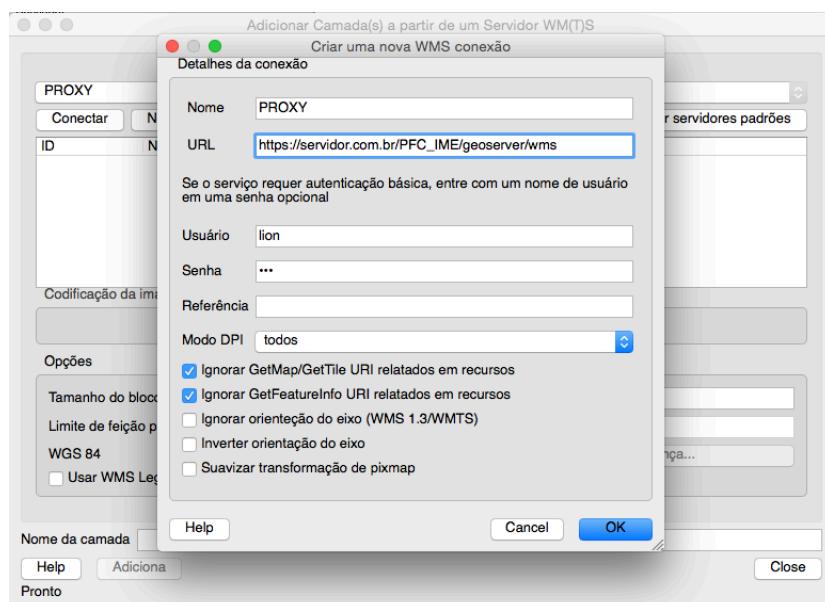
A interface utiliza símbolos convencionais para indicar as operações de adição (símbolo positivo), remoção (símbolo vermelho de uma lata de lixo), seleção (seleção do pincel ao lado de cada camada ou seleção da camada). Quando um elemento é selecionado ele pode ser salvo, ao passo que ao selecionar o símbolo positivo apenas a operação escrita adicionar será visualizada.

Como a aplicação do administrador não é o foco desse trabalho, maiores detalhes sobre sua implementação foram deixados de fora desse trabalho, optando-se por oferecer maior detalhamento da aplicação do usuário.

## 6 INTERAÇÃO COM OUTRAS APLICAÇÕES

Uma das aplicações mais utilizadas em geoprocessamento é o QGIS (também conhecido como Quantum GIS), devido às suas inúmeras funcionalidades de edição, análise e visualização de dados espaciais (QGIS, 2015). Esses dados podem estar armazenados como arquivos, bancos de dados espaciais ou podem ser recuperados a partir da conexão e requisições a servidores WMS.

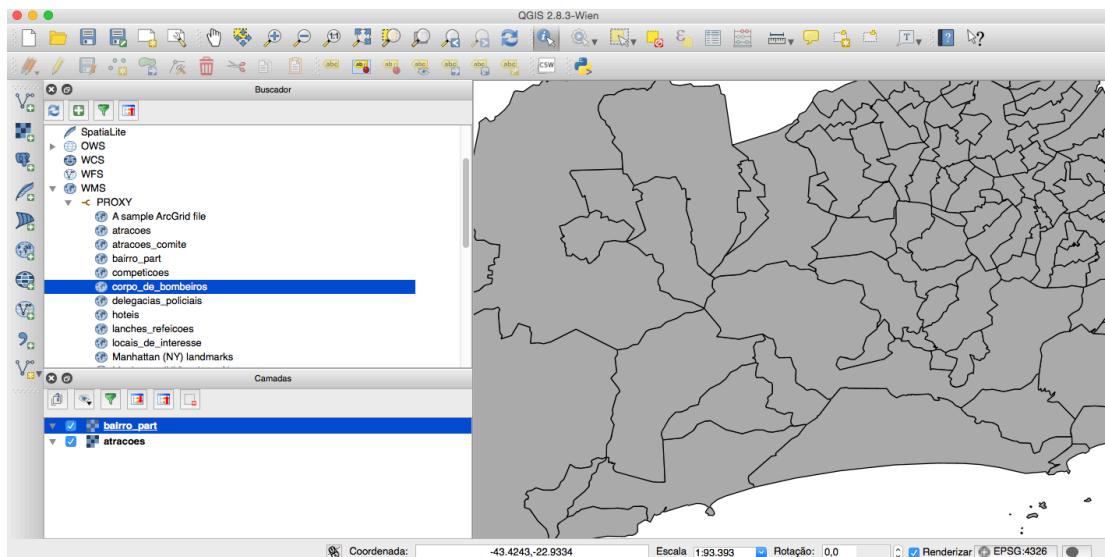
Por meio da extensão do seu método de autenticação, a aplicação do usuário é capaz de fornecer dados espaciais ao QGIS da mesma forma que o faz para o OpenLayers. Dessa forma, o QGIS necessita apenas cadastrar a URL “/geoserver/wms”, relativa ao caminho onde se encontra o servidor da aplicação, os parâmetros de usuário e senha (de um usuário cadastrado no sistema), e desmarcar as duas primeiras opções como apresentado na Figura 6.1, abaixo. Essa opção é utilizada, pois o GeoServer envia o seu endereço DNS de localização na requisição GetCapabilities do WMS. Dessa forma requisições subsequentes podem se comunicar com esse endereço, e é justamente esse comportamento que o QGIS adota. Caso isso ocorra o serviço não funcionará corretamente pois a porta 9090 onde o GeoServer é executado é bloqueada pelo *firewall*.



**FIG. 6.1 - Configuração do servidor WMS no QGIS**

No exemplo acima o caminho onde se encontra a aplicação do usuário é <https://servidor.com.br>, de tal forma que o servidor WMS se encontra no caminho <https://servidor.com.br/geoserver/wms>. É importante notar que o serviço não funcionará para requisições do tipo HTTP, uma vez que o modo como o QGIS opera ao preencher o usuário e senha é utilizando um método de autenticação conhecido como *Basic HTTP Authentication*. Nesse método é enviado num cabeçalho com nome “*Authorization*”, uma cadeia de caracteres iniciada pela palavra “*Basic* ”, seguida da codificação em Base64 do usuário e da senha (W3C, 1996). O detalhamento desse protocolo e da codificação foge ao escopo desse trabalho, no entanto, o aspecto importante a ser notado é o fato de as informações de nome e senha do usuário serem enviadas no cabeçalho da requisição. Para se prevenir de um possível furto dessas informações que a princípio são confidenciais, deve-se utilizar o protocolo HTTPS que criptografa os cabeçalhos da requisição juntamente com o *payload*, garantindo a segurança do sistema, apesar do fato da criptografia das informações reduzir o desempenho do sistema.

O exemplo de um resultado de exibição de camadas pelo QGIS é apresentado na Figura 6.2 abaixo, para a camada de bairros do Rio de Janeiro utilizada no cenário.



**FIG. 6.2 - Visualização de camadas presentes no sistema pelo QGIS**

## 7 APLICAÇÃO DO CENÁRIO

O objetivo desse *software* inicialmente foi apenas a construção da aplicação do usuário, mas, ao longo do seu desenvolvimento, julgou-se necessário a construção das outras partes relacionadas ao administrador.

Após o desenvolvimento da solução, obteve-se como produto final desse projeto um pacote de *softwares* completo, que:

- pode ser instalado e inicializado em uma máquina Ubuntu 14.04, por meio de um administrador;
- provê a esse administrador uma aplicação para alterar o cenário da aplicação do usuário e ferramentas adicionais para auxilia-lo no processo de configuração; e
- provê ao usuário uma aplicação de disponibilização de informações geográficas, que consiste na instalação e configuração de um serviço WMS.

Dessa forma, esse capítulo tem por objetivo demonstrar como ocorre a utilização do software, desde a instalação, configuração e criação do cenário por parte do administrador, até a utilização da aplicação principal por parte do usuário. Ressaltando que a utilização da aplicação do usuário é feita através de qualquer navegador com suporte a Javascript, seja em plataformas *desktop* ou em plataformas móveis, sendo que a aplicação foi testada nos navegadores *Google Chrome*, *Mozilla Firefox* e *Safari*, nas versões 46.0.2490.71, 40.0.3 e 9.0, respectivamente.

### 7.1 Instalação das aplicações

A instalação consiste em alguns passos que devem ser executados pelo administrador da aplicação, no sentido de preparar uma máquina para hospedar as aplicações. Para isso, foi construído um instalador, já explanado no desenvolvimento da solução, contendo os arquivos que o administrador precisa executar para realizar as seguintes tarefas: a instalação de todos

os *softwares* que a máquina precisa ter para hospedar as aplicações, sem a necessidade de acesso à internet; a inicialização e configuração do servidor de aplicação, com as aplicações implementadas nele; a adição de arquivos shapefile no banco de dados espacial, para posterior publicação no GeoServer; e a criação da biblioteca de estilos, que vai cadastrar no GeoServer todas os estilos que poderão ser utilizados para representação das camadas.

Na pasta do instalador, está contido um executável chamado *install* que realiza a tarefa de instalação dos softwares na máquina. A primeira ação do administrador será executar o *install*. Em seguida, o administrador deve executar um outro arquivo chamado *start-app*, que realiza a tarefa de inicializar o domínio do servidor da aplicação, com o GeoServer e aplicações já implementadas e configuradas.

Após isso, o administrador vai adicionar os *shapefiles* contendo informações de camadas no PostGIS. Para isso, o administrador deve executar o arquivo *addshapefile* passando os seguintes parâmetros no formato já especificado anteriormente nesse relatório. Por exemplo, para adicionar o arquivo *shapefile* com a camada de hotéis chamado “hoteis.shp” que se encontra na pasta “dados”, o administrador realizaria o seguinte comando:

```
# ./addshapefile -f dados/hoteis.shp -t Hotéis
```

Como uma das ações realizadas na instalação é a configuração do PostGIS no GeoServer, quando o administrador adiciona os *shapefiles* ao PostGIS, ele já pode visualizar as camadas adicionadas na ferramenta de administração via *web* do GeoServer. Para permitir que a aplicação tenha acesso a essas camadas, o administrador precisa publicar essas camadas por meio da própria ferramenta. Para isso, ele deve acessar o GeoServer, por meio da porta 9090 do localhost, selecionar a opção “Layers” (na barra lateral) > “New resource” e então selecionar o banco de dados local do PostGIS, que se chama “localpostgis”. Com isso, o administrador tem acesso a página ilustrada na Figura 7.1, onde aparecem todas as camadas adicionadas por ele e a opção “publish” ao lado de cada uma. Para cada camada, o administrador vai selecionar a opção “publish”, o que vai encaminhá-lo para uma página onde as únicas modificações que ele precisa fazer antes de publicar a camada estão ilustradas nas Figuras 7.2 e 7.3. Nessa página, o administrador precisa: na seção “Bounding Boxes”, ilustrada na Figura 7.2, selecionar a opção “Compute from native bounds”; e, na seção “WMS Settings”, ilustrada na Figura 7.3, selecionar a opção “Queryable” (necessário para que a aplicação do usuário possa obter informações das feições da camada por meio de requisições

WMS GetFeatureInfo), selecionar o estilo *default* da camada (cuja prévia pode ser visualizada logo abaixo) e selecionar, dentre os estilos disponíveis, os que poderão ser utilizados pela camada. Com isso, basta ir até o final da página e selecionar “publish”. Após publicadas todas as camadas o administrador visualizará a página da Figura 7.4, ao selecionar a opção “Layers”.

The screenshot shows the GeoServer interface with the title 'New Layer'. On the left, there's a sidebar with 'About & Status' (Server Status, GeoServer Logs, About Information, About GeoServer), 'Data' (Layer Preview, Workspaces, Stores, Layers, Layer Groups, Styles), 'Services' (WFS, WCS, WMS), 'Settings' (Global, JAI, Coverage Access), 'Tile Caching' (Tile Layers, Caching Defaults, Gridsets, Disk Quota, BlobStores). The main area is titled 'New Layer' with the sub-instruction 'Add a new layer'. It contains a message: 'You can create a new feature type by manually configuring the attribute names and types. [Create new feature type...](#). On databases you can also create a new feature type by configuring a native SQL statement. [Configure new SQL view...](#). Here is a list of resources contained in the store 'localpostgis'. Click on the layer you wish to configure.' Below this is a table with columns 'Published', 'Layer name', and 'Action'. The table lists 12 items:

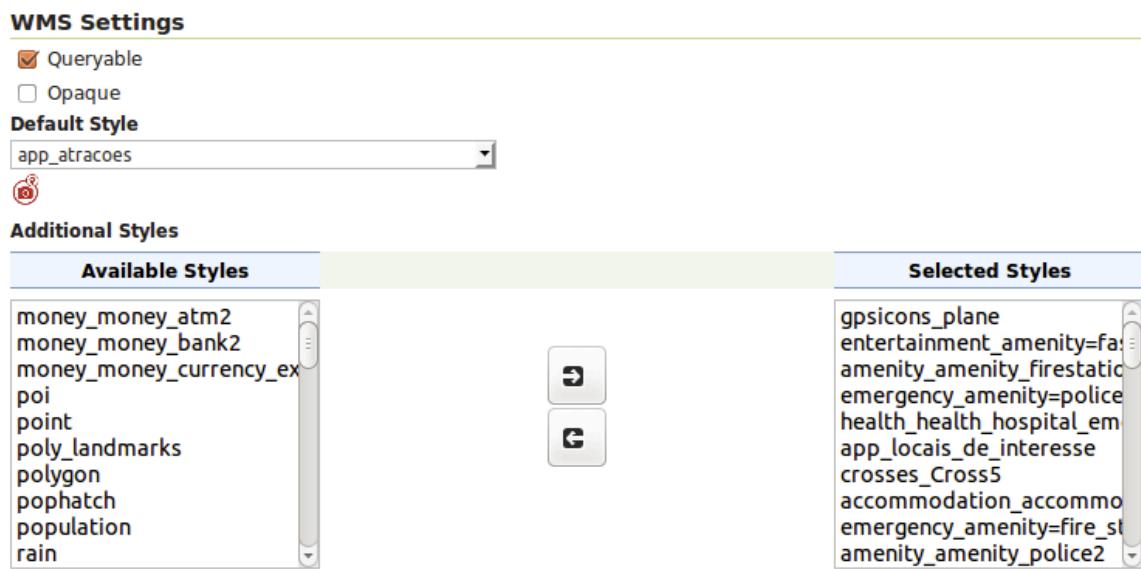
Published	Layer name	Action
	atracoes	Publish
	atracoes_comite	Publish
	competicoes	Publish
	corpo_de_bombeiros	Publish
	delegacias_policiais	Publish
	hoteis	Publish
	lanches_refeicoes	Publish
	locais_de_interesse	Publish
	paradas_metro	Publish
	paradas_onibus	Publish
	paradas_trens	Publish
	rio_bairros	Publish

**FIG. 7.1 - Visualização das camadas adicionadas ao PostGIS na ferramenta de administração do GeoServer**

This screenshot shows the 'Feature Type Details' section of the GeoServer configuration. It includes sections for 'Bounding Boxes' (Native Bounding Box with Min X: -43.46084594726, Min Y: -23.02740097045, Max X: -43.14984512329, Max Y: -22.83777618408; Lat/Lon Bounding Box with the same coordinates), 'Curved geometries control' (checkbox unchecked), 'Linearization tolerance' (text input field empty), and 'Feature Type Details' (table with columns: Property, Type, Nullable, Min/Max Occurrences). The table lists seven properties:

Property	Type	Nullable	Min/Max Occurrences
nome	String	true	0/1
bairro	String	true	0/1
telefone	String	true	0/1
latitude	BigDecimal	true	0/1
longitude	BigDecimal	true	0/1
endereco	String	true	0/1
geom	Point	true	0/1

**FIG. 7.2 – Trecho de configuração da Bounding Box da página de publicação da camada na ferramenta de visualização do GeoServer**



**FIG. 7.3 – Trecho de configuração dos estilos da página de publicação da camada na ferramenta de visualização do GeoServer**

The screenshot shows the 'Layer Preview' page of the GeoServer interface. It lists 12 published layers in a table:

Type	Name	Title	Common Formats	All Formats
localdb	corpo_de_bombeiros	corpo_de_bombeiros	OpenLayers KML GML	Select one
localdb	atracoes	atracoes	OpenLayers KML GML	Select one
localdb	hoteis	hoteis	OpenLayers KML GML	Select one
localdb	lanches_refeicoes	lanches_refeicoes	OpenLayers KML GML	Select one
localdb	atracoes_comite	atracoes_comite	OpenLayers KML GML	Select one
localdb	competicoes	competicoes	OpenLayers KML GML	Select one
localdb	delegacias_policiais	delegacias_policiais	OpenLayers KML GML	Select one
localdb	locais_de_interesse	locais_de_interesse	OpenLayers KML GML	Select one
localdb	paradas_metro	paradas_metro	OpenLayers KML GML	Select one
localdb	paradas_onibus	paradas_onibus	OpenLayers KML GML	Select one
localdb	paradas_trens	paradas_trens	OpenLayers KML GML	Select one
localdb	rio_bairros	rio_bairros	OpenLayers KML GML	Select one

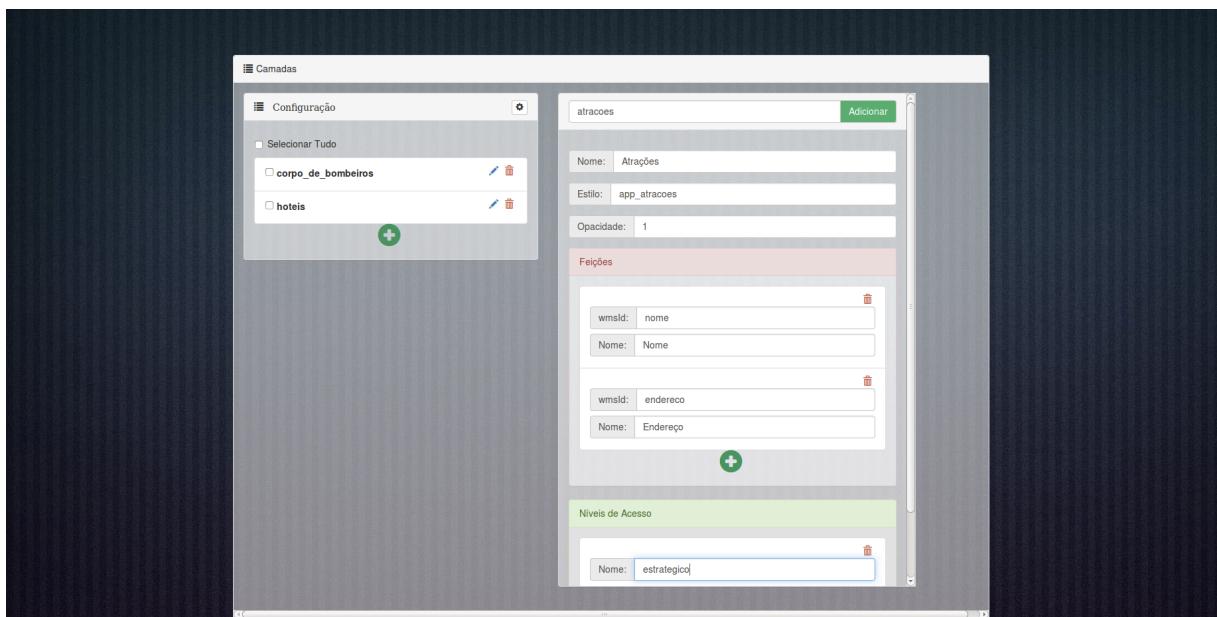
**FIG. 7.4 - Página com as camadas publicadas no GeoServer**

Por fim, para criação da biblioteca de estilos, o administrador precisa apenas executar o arquivo chamado “build-style-library”. Após isso, o administrador pode passar para a próxima etapa de criação dos cenários.

## 7.2 Criação do cenário

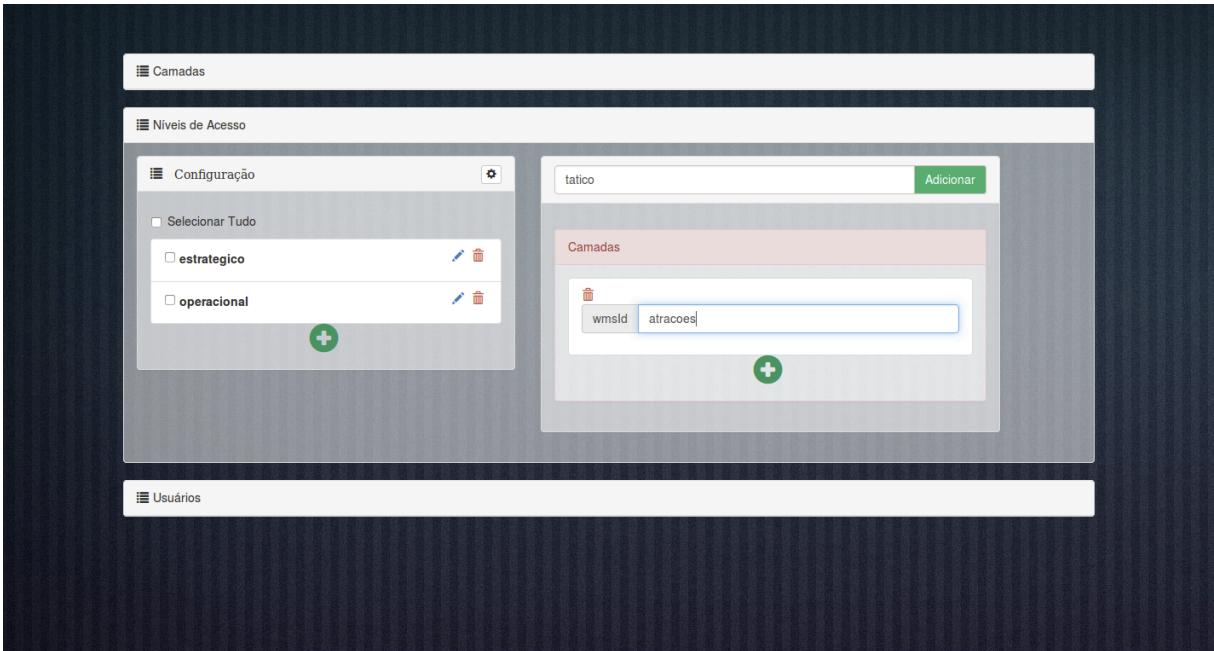
Para criação ou alteração do cenário, o administrador utiliza a aplicação do administrador criada com esse propósito. As figuras abaixo ilustram exemplos de utilização da aplicação.

A Figura 7.5 ilustra a adição de uma camada ao cenário. No exemplo da figura está sendo adicionada a camada identificada no GeoServer por “atrações”. São atribuídos à camada um nome, um estilo (pode não definir nada, que será utilizado o *default*) e uma opacidade. Podendo também adicionar os atributos que se deseja exibir para a camada (estas são exibidas na ordem em que são configuradas), que no exemplo são o nome e o endereço, e adicionar a camada a um nível de acesso, que no exemplo foi o nível estratégico (a camada pode ser adicionada a mais de um nível de acesso).



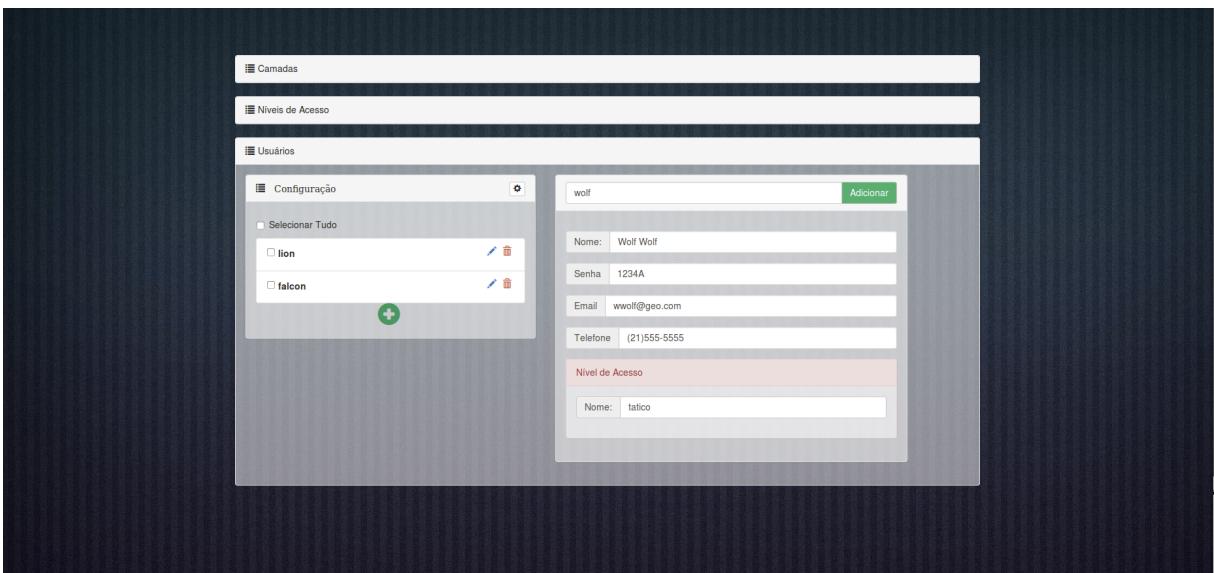
**FIG. 7.5 - Utilização da aplicação do administrador para adição de uma camada**

A Figura 7.6 ilustra a adição de um nível de acesso ao cenário. No exemplo da figura, o nível de acesso adicionado chama-se “tático” e a ele foi adicionada a camada atrações.



**FIG. 7.6 - Utilização da aplicação do administrador para adição de um nível de acesso**

A Figura 7.7 ilustra a adição de um usuário ao cenário. No exemplo da figura, o identificador do usuário é “wolf”. São atribuídos ao usuário um nome, uma senha, um e-mail e um telefone, sendo os dois últimos opcionais. Além disso, ao usuário deve ser atribuído o nível de acesso a que ele pertence. No exemplo o usuário foi adicionado ao nível “tático”.

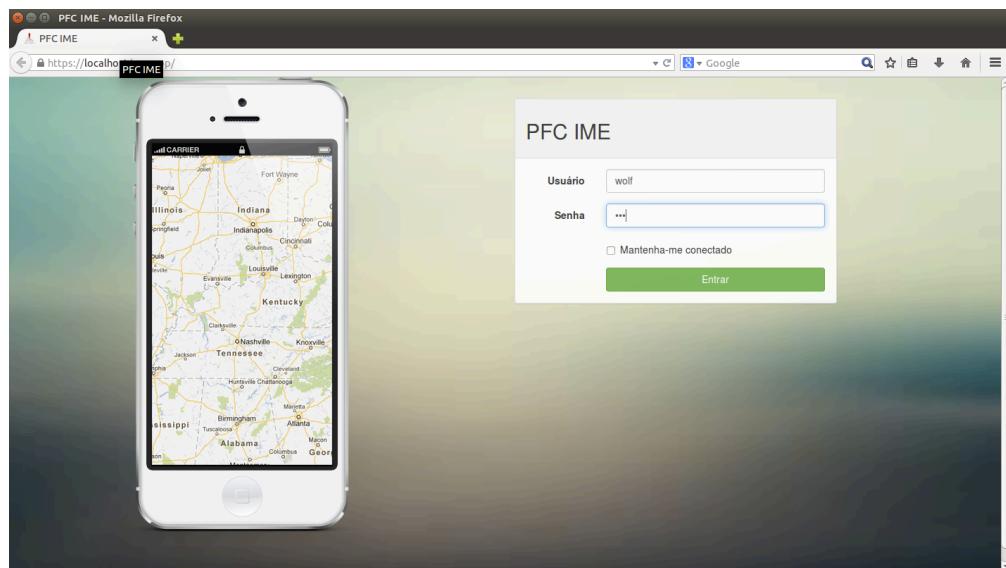


**FIG. 7.7 - Utilização da aplicação do administrador para adição de um usuário**

Após a criação do cenário, a aplicação principal fica pronta para ser utilizada pelos usuários cadastrados.

### 7.3 Utilização da aplicação principal

Na aplicação principal, o usuário inicialmente é direcionado para a página de *login*, ilustrada nas Figuras 7.8 e 7.9. Nessa página, ele simplesmente vai preencher seu identificador e senha e clicar em “Entrar”.



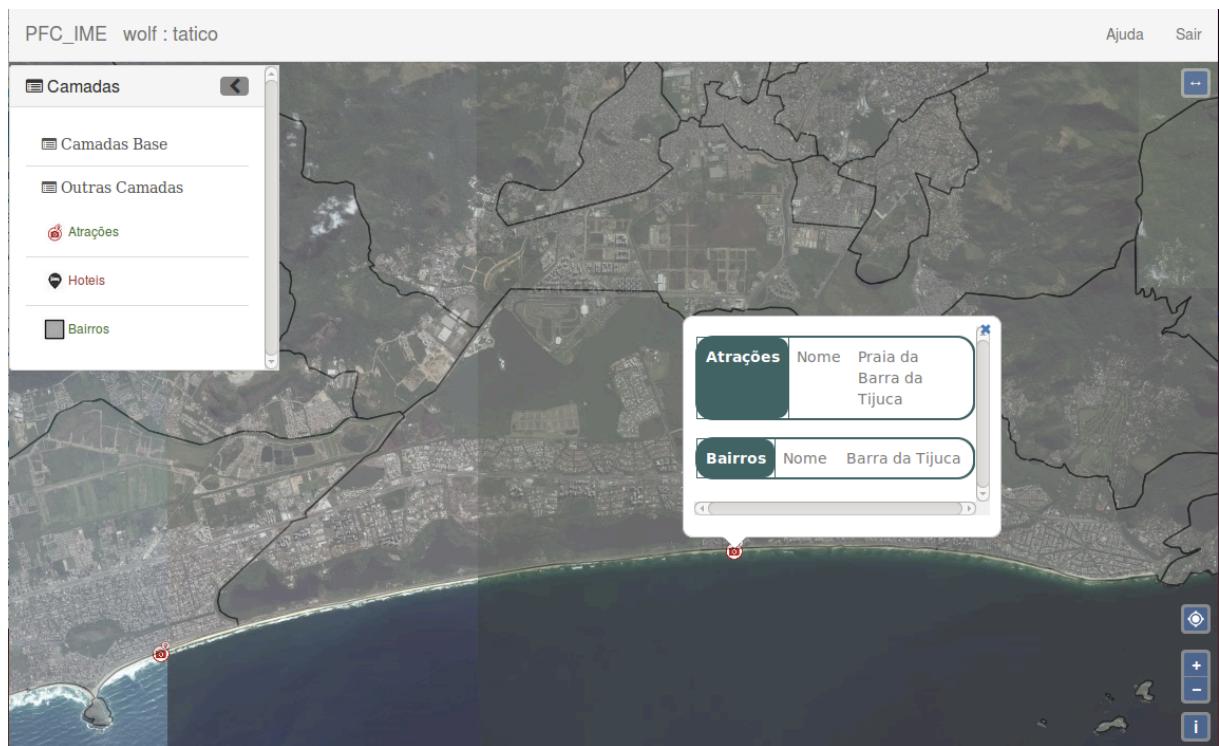
**FIG. 7.8 - Visualização da página de *login* em uma plataforma *desktop***



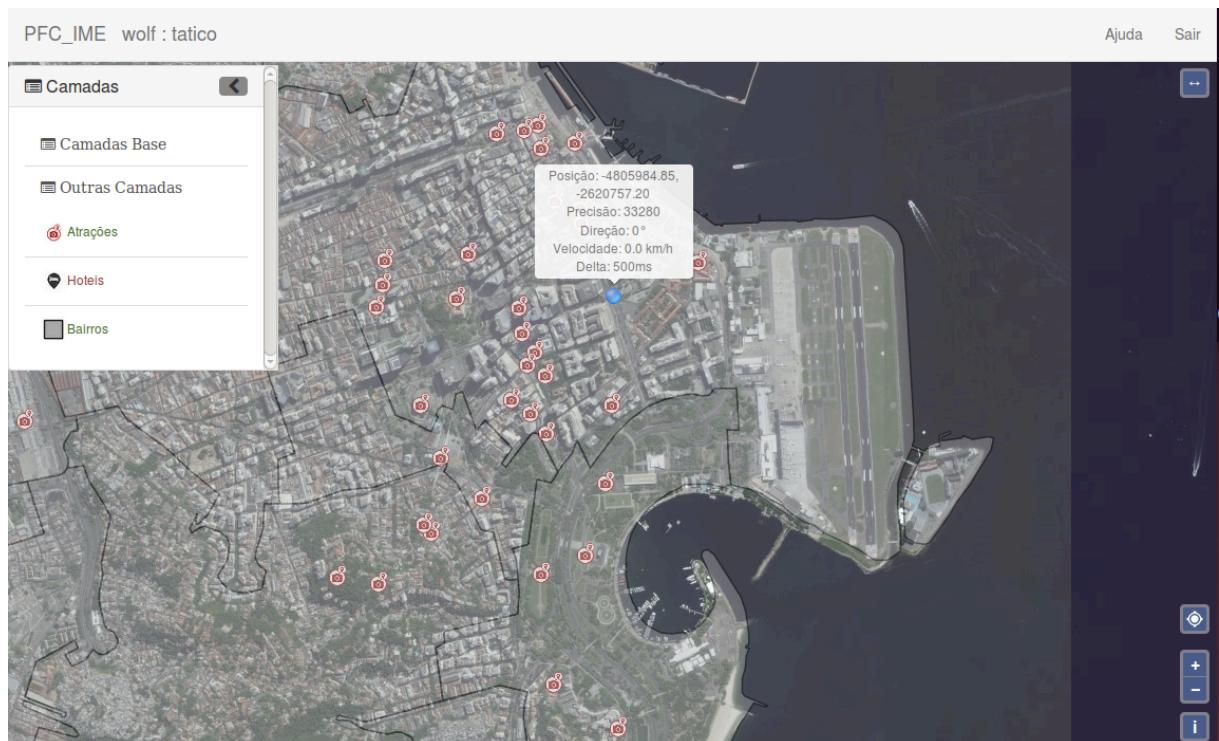
**FIG. 7.9 - Visualização da página de *login* em uma plataforma móvel**

Após conectar à aplicação, o usuário pode interagir com o mapa de diversas formas. Alguns exemplos de interação serão demonstrados a seguir.

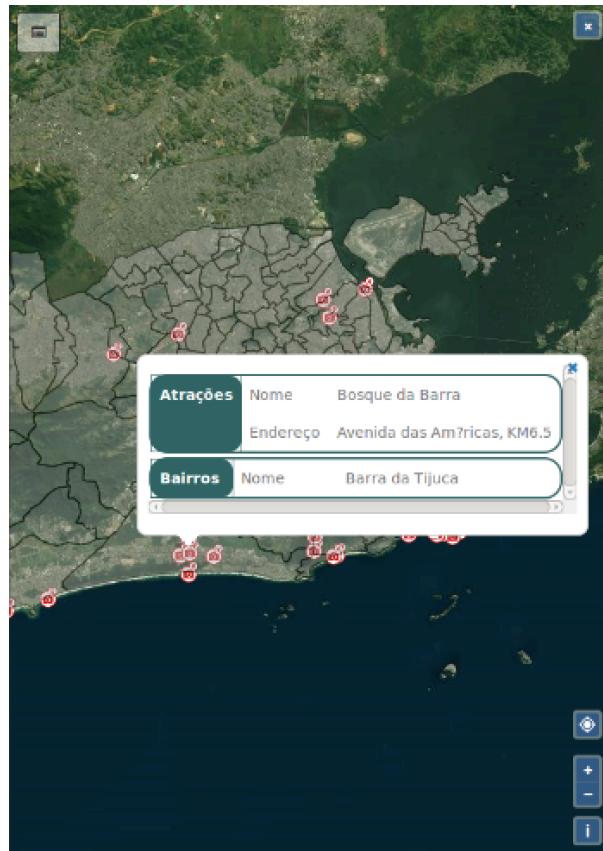
O usuário pode exibir uma camada e clicar em um dos elementos dessa camada para visualizar suas informações, como ilustrado na Figura 7.10. Como ilustrado na Figura 7.11, o usuário pode clicar no botão de localização, já mencionado anteriormente, e ter sua localização marcada no mapa, podendo também visualizar informações a respeito da sua localização, como latitude, longitude e velocidade, ao passar o *mouse* em cima da marcação. Por fim, o usuário pode minimizar a barra lateral das camadas e colocar a página em tela cheia (ao selecionar o botão de tela cheia já descrito anteriormente), para melhor visualização do mapa, como ilustrado na Figura 7.12.



**FIG. 7.10 - Visualização de camada e feições de um elemento dessa camada na aplicação principal**



**FIG. 7.11 - Visualização da localização do usuário e informações acerca dessa localização na aplicação principal**



**FIG. 7.12- Visualização da aplicação principal em tela cheia e com a barra lateral minimizada em uma plataforma móvel**

## 8 CONCLUSÃO

O emprego da geoinformação é um fator fundamental em operações militares, seja para auxiliar à tomada de decisão de um comandante, para apoiar o deslocamento estratégico de meios, ou para estabelecimento de medidas de segurança locais. O crescimento da importância da geoinformação levou ao surgimento de organizações que se responsabilizam por estabelecer padrões para que dados geoespaciais estejam disponíveis por meio de serviços *web*, permitindo o desenvolvimento de aplicações *web* para fins específicos que se utilizam desses dados.

Nesse contexto, esse trabalho teve por objetivo a instalação e a configuração de serviços *web* para a disponibilização de dados espaciais em operações militares, tendo como foco os quesitos de usabilidade por diferentes perfis, o controle de acesso e a portabilidade. Contudo, além da aplicação que atende a esse objetivo de maneira completa, que foi denominada aplicação do usuário, foi desenvolvido um *software* completo que permite instalar e configurar essa aplicação em um sistema operacional Ubuntu 14.04, além de prover um conjunto de ferramentas para permitir ao administrador do sistema configurar de maneira mais rápida o cenário, abstraindo diversos conceitos que não precisam ser de conhecimento do administrador.

Além disso, para possibilitar que trabalhos futuros possam configurar essa solução em outros sistemas operacionais, foram documentadas todas as aplicações, bibliotecas e dependências necessárias para se instalar e configurar o sistema.

Como contribuições adicionais desse projeto, pode-se citar: a possibilidade de acesso às informações geográficas por diversos dispositivos, permitindo que em uma operação militar, um comandante possa definir um cenário e compartilhar as informações com os diversos níveis operacionais, garantindo que as informações mais relevantes a cada nível sejam acessadas por estes; e a criação dinâmica de uma biblioteca extensível de estilos no GeoServer a partir de imagens, um procedimento concebido nesse projeto e que é de extrema importância para camadas pontuais, onde o estilo que representa a camada deve permitir associar intuitivamente o nome da camada com a imagem exibida.

Como sugestões de trabalhos futuros, sugere-se a criação de procedimentos de instalação para outros sistemas operacionais, como outras versões do Linux ou o Windows, e a adição de mais funcionalidades à aplicação do usuário, como permitir ao usuário desenhar figuras no mapa ou importar imagens do que ele visualiza. Outra sugestão, seria a identificação dos usuário com a sua localização no servidor, para prover uma camada de dados dinâmica contendo as localizações de todos os usuários conectados ao sistema.

## 9 REFERÊNCIAS BIBLIOGRÁFICAS

- ARAÚJO, P. C.; CARVALHO, E. A. **História da Cartografia**. 2008. Disponível em: <[http://www.ead.uepb.edu.br/ava/arquivos/cursos/geografia/leituras\\_cartograficas/Le\\_Ca\\_A01\\_J\\_GR\\_260508.pdf](http://www.ead.uepb.edu.br/ava/arquivos/cursos/geografia/leituras_cartograficas/Le_Ca_A01_J_GR_260508.pdf)> Acesso em: 15 de março de 2015.
- BENERS-LEE, T.; FIELDING, R.; FRYSTYK, H. **Hypertext Transfer Protocol**. 1996. Disponível em: <<http://www.w3.org/Protocols/HTTP/1.0/spec.html>> Acesso em: 25 de agosto de 2015.
- BOUNDLESS. **Introdução ao PostGIS**. Disponível em: <<http://workshops.boundlessgeo.com/postgis-intro/introduction.html>> Acesso em: 5 de abril de 2015.
- CAELUM. **Apostila Java para Desenvolvimento Web**. Disponível em: <<http://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/#14-2-java-persistence-api-e-framework-orm>> Acesso em: 10 de setembro de 2015.
- DAMASCENO, J. C. **Introdução à Composição de Serviços Web**. Disponível em: <<http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/minicurso/mc8.pdf>> Acesso em: 17 de março de 2015.
- DELGADO, R. E.; JÚNIOR, E. J. S.; MAIOR, R. S. **Construção de Conjunto de Dados Geoespaciais Vetoriais para Apoio ao Turista nas Olimpíadas Rio 2016 a Partir de Dados Abertos**. 2015. Projeto Final de Curso (Engenharia Cartográfica) – Instituto Militar de Engenharia, Rio de Janeiro, 2015.
- EME. **Manual de Campanha EB20-MC10.209 GEOINFORMAÇÃO**. 1<sup>a</sup> edição, 2014.
- KOMANNA, K. **Implementation of a GeoServer Application for GIS Data Distribution and Manipulation**. 2013. Master's thesis - Stockholm University, Department of Physical Geography and Quaternary Geology, 2013. Disponível em: <<https://www.diva-portal.org/smash/get/diva2:640096/FULLTEXT01.pdf>> Acesso em: 10 de setembro de 2015.

OLIVEIRA, M. V. P. de; SILVA, M. de A. V. M. **Protocolo de Segurança para Serviços Web de Disponibilização de Dados Espaciais em Operações Militares**. 2015. Projeto Final de Curso (Engenharia de Computação) – Instituto Militar de Engenharia, Rio de Janeiro, 2015.

OPEN GEOSPATIAL CONSORTIUM. **Web Map Service Overview**. Disponível em: <<http://www.opengeospatial.org/standards/wms>> Acesso em: 18 de março de 2015.

OSGeo. **Manual do Usuário do Geoserver: Configuração API REST**. Disponível em: <<http://docs.geoserver.org/2.6.5/user/rest/api/index.html>> Acesso em: 30 de março de 2015.

OSGeo. **Manual do Usuário do Geoserver: Considerações do Java**. Disponível em: <<http://docs.geoserver.org/latest/en/user/production/java.html>> Acesso em: 28 de agosto de 2015.

OSGeo. **Manual do Usuário do Geoserver: Introdução**. Disponível em: <<http://docs.geoserver.org/stable/en/user/introduction/overview.html>> Acesso em 20 de março de 2015.

OSGeo. **Manual do Usuário do Geoserver: Usando GeoWebCache**. Disponível em: <<http://docs.geoserver.org/latest/en/user/geowebcache/using.html>> Acesso em: 30 de março de 2015.

PACKTPUB. **What is OpenLayers?** 2013. Disponível em: <<https://www.packtpub.com/books/content/what-openlayers>> Acesso em: 6 de abril de 2015.

PostGIS. Disponível em: <<http://postgis.net>> Acesso em: 3 de abril de 2015.

QGIS. **Features of QGIS**. Disponível em: <<http://www.qgis.org/en/site/about/features.html>> Acesso em: 25 de agosto de 2015.

TUTORIAL REPUBLIC. **Bootstrap Introduction**. Disponível em <<http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-introduction.php>> Acesso em: 20 de junho de 2015.

W3C. **Scalable Vector Graphics (SVG) 1.1**. 2<sup>a</sup> edição. 2011. Disponível em: <<http://www.w3.org/TR/SVG/>> Acesso em: 20 de agosto de 2015.