

Paulo Berlanga Neto

Proposta de Software de Orientação  
Geográfica para os Jogos Olímpicos  
Rio de Janeiro 2016

XV Curso de Especialização em Geoprocessamento  
2014



UFMG  
Instituto de Geociências  
Departamento de Cartografia  
Av. Antônio Carlos, 6627 – Pampulha  
Belo Horizonte  
cartog@igc.ufmg.br

**PAULO BERLANGA NETO**

**PROPOSTA DE SOFTWARE DE  
ORIENTAÇÃO GEOGRÁFICA PARA  
OS JOGOS OLÍMPICOS RIO DE JANEIRO 2016**

Monografia apresentada como requisito parcial à obtenção do grau de Especialista em Geoprocessamento. Curso de Especialização em Geoprocessamento. Departamento de Cartografia. Instituto de Geociências. Universidade Federal de Minas Gerais.

Orientador (a): Maria Márcia Magela Machado

Co-orientador (a): Amanda Alves dos Santos

**BELO HORIZONTE**

**2014**

B514p  
2014

Berlanga Neto, Paulo.

Proposta de software de orientação geográfica para os jogos olímpicos Rio de Janeiro 2016 [manuscrito] / Paulo Berlanga Neto. – 2014.

37 f., enc. : il. color.

Orientador: Maria Márcia Magela Machado.

Co-orientador: Amanda Alves dos Santos

Monografia (especialização) - Universidade Federal de Minas Gerais, Instituto de Geociências, 2014.

Bibliografia: f. 37.

1. Geoprocessamento. 2. Geotecnologia. 3. Jogos olímpicos 2016.  
I. Machado, Maria Márcia Magela. II. Santos, Amanda Alves dos. III.  
Universidade Federal de Minas Gerais, Instituto de Geociências. IV.  
Título.

CDU: 528

Aluno (a) Paulo Berlanga Neto

Monografia defendida e aprovada em cumprimento ao requisito exigido para obtenção do título de Especialista em Geoprocessamento, em 27 de novembro de 2014, pela Banca Examinadora constituída pelos professores:

---

Profa. Dra. Maria Márcia Magela Machado

---

Profa. Dra. Úrsula Ruchkys de Azevedo

## AGRADECIMENTOS

Agradeço primeiramente a Deus, que sempre me auxilia nos objetivos pessoais e me tranqüiliza nos momentos agitados.

Agradeço a toda a minha família, em especial meus pais, pelo apoio e incentivo fornecido ao longo de todos estes anos de triunfais aventuras.

Agradeço a Profa. Dra. Maria Márcia Magela Machado juntamente com a co-orientadora Amanda Alves do Santos pelas excelentes orientações providas.

Agradeço aos grandes amigos, amigas e ex-namoradas, que de alguma forma me incentivaram a mudar para Belo Horizonte e realizar esta Especialização, e também aos colegas de trabalho que aqui encontrei, destacando o grande amigo Alesander pela indicação de livros e referências para a realização deste estudo.

E por fim, assim como nos trabalhos de graduação, faço um agradecimento especial ao meu querido e saudoso avô José Toniollo (*in memorian*), pela sua humildade, transmissão de princípios e ensinamentos de vida.

“Não siga a multidão, trilhe seu próprio caminho...”

Margaret Thatcher

## RESUMO

Nos últimos anos, o Brasil tem ganhado grande destaque no cenário mundial, não apenas pelo fortalecimento de sua economia e pelos seus já conhecidos vastos recursos naturais, mas também pela organização em seqüência de dois dos maiores eventos esportivos do planeta; a Copa do Mundo FIFA 2014 e os Jogos Olímpicos de 2016. A cidade do Rio de Janeiro será sede da Edição XXXI dos Jogos Olímpicos de Verão, que ocorrerá entre os dias 05 e 21 de agosto de 2016, promovendo as diversas disputas esportivas olímpicas entre os melhores atletas do planeta, e despertando o interesse de grande parcela da população mundial. Para isso, as organizações e comitês responsáveis, estão viabilizando uma série de obras em instalações, acomodações e corredores de mobilidade, com o intuito de aprimorar a infraestrutura na cidade. Juntamente com esta infraestrutura, serão necessárias informações de diversas naturezas para orientação dos turistas interessados em acompanhar os jogos pessoalmente. Diante deste cenário, este trabalho propôs a utilização de meios tecnológicos hoje existentes, para reunir e apresentar estas informações num sistema baseado em mapas disponibilizado para plataformas móveis, simplificando a orientação dos turistas e seus deslocamentos aos pontos de interesse na cidade, como locais de disputas esportivas, vila olímpica, parques olímpicos, pontos turísticos, aeroportos, entre outros. A facilidade para obtenção de coordenadas a partir de um *smartphone* ou *tablet*, tornou essa tarefa de orientação bastante prática e instantânea, e mostrou capacidade para poupar consideráveis parcelas de trabalho dos voluntários da organização, contribuindo ao mesmo tempo com a sustentabilidade, dada a redução da necessidade de guias e impressos informativos.

## SUMÁRIO

	<b>Pág.</b>
LISTA DE FIGURAS .....	ix
LISTA DE TABELAS .....	x
LISTA DE SIGLAS E ABREVIATURAS .....	xi
CAPÍTULO 1 – INTRODUÇÃO.....	01
CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA .....	04
2.1 – Geotecnologias e abordagens técnicas .....	04
2.2 – Plataforma <i>JBoss OpenShift</i> e o conceito de <i>Cloud Computing</i> .....	08
2.3 – Banco de dados <i>PostgreSQL</i> e a extensão geográfica <i>PostGIS</i> .....	09
2.4 – <i>RESTful Webservices</i> e o formato de representação <i>GeoJSON</i> .....	11
2.5 – Serviços integrados ao <i>GoogleMaps Javascript API</i> .....	15
CAPÍTULO 3 – CARACTERIZAÇÃO DA ÁREA DE ESTUDO .....	17
CAPÍTULO 4 – MATERIAIS E MÉTODOS .....	20
4.1 – Especificação de desenvolvimento do <i>software</i> .....	20
4.2 – Levantamento de informações e coleta de dados .....	27
CAPÍTULO 5 – RESULTADOS E CONCLUSÕES .....	29
5.1 – Usabilidade do <i>software</i> .....	30
CAPÍTULO 6 – CONSIDERAÇÕES FINAIS .....	36
REFERÊNCIAS BIBLIOGRÁFICAS .....	37

## LISTA DE FIGURAS

	Pág.
Figura 2.1 – Estrutura de um SIG moderno .....	05
Figura 2.2 – Cobertura de satélites no sistema de navegação GPS.....	06
Figura 2.3 – Modelo ilustrado de geolocalização por triangulação de antenas .....	07
Figura 2.4 – Representação ilustrativa do conceito de computação em nuvem .....	08
Figura 2.5 – Parte dos registros armazenados pela tabela <i>SPATIAL_REF_SYS</i> .....	10
Figura 2.6 – Esquema de conversação com requisição e resposta .....	11
Figura 2.7 – Obtenção de recursos com a operação <i>GET</i> .....	12
Figura 2.8 – Comparativo entre XML e JSON .....	13
Figura 2.9 – Imagem de satélite editada pelo <i>GoogleMaps</i> com marcador .....	15
Figura 2.10 – Exemplo de vetorização de rotas entre pontos .....	16
Figura 3.1 – Mapa da cidade do Rio de Janeiro .....	17
Figura 3.2 – Regiões e locais de competição .....	18
Figura 4.1 – Fluxograma da estrutura geral da metodologia .....	20
Figura 4.2 – Composição das tecnologias do <i>software</i> distribuído .....	22
Figura 4.3 – Representação de comunicação ao ambiente remoto via SSH .....	22
Figura 4.4 – Console e comando de ativação da extensão <i>PostGIS</i> .....	23
Figura 4.5 – Console e comando de criação da tabela <i>location</i> .....	23
Figura 4.6 – Console e <i>script</i> de inserção em tabela .....	23
Figura 4.7 – Representação <i>request/response</i> do serviço de dados .....	24
Figura 4.8 – Apresentação do marcador de geolocalização do usuário .....	25
Figura 4.9 – Aplicação de <i>layouts</i> fluídos .....	26
Figura 5.1 – Tela inicial do aplicativo .....	30
Figura 5.2 – Marcadores temáticos por categoria .....	31
Figura 5.3 – Marcador de localização do destino escolhido .....	32
Figura 5.4 – Exemplo de rota traçada entre pontos A e B .....	33
Figura 5.5 – Informações textuais de rota por carro .....	34
Figura 5.6 – Informações textuais de rota por transporte público .....	35

## LISTA DE TABELAS

	<b>Pág.</b>
Tabela 2.1 – Tipos de geometria em representação WKT .....	10
Tabela 2.2 – Exemplos de funções espaciais .....	11
Tabela 2.3 – Geometrias padrão OGC com a notação <i>GeoJSON</i> .....	14
Tabela 4.1 – Amostra de locais públicos para análise de acessibilidade .....	27
Tabela 4.2 – Categorias e destinos disponíveis no <i>software</i> .....	28

## LISTA DE SIGLAS E ABREVIATURAS

API – Application Programming Interface  
COI – Comitê Olímpico Internacional  
CSS – Cascading Style Sheets  
DOD – Departamento de Defesa dos Estados Unidos  
GeoJSON – Geographical Javascript Object Notation  
GIS – Geographic Information Systems  
GPS – Global Positioning System  
HTML – HyperText Markup Language  
HTTP – Hypertext Transfer Protocol  
IBC – International Broadcast Centre  
IBGE – Instituto Brasileiro de Geografia e Estatística  
IDE – Integrated Development Environment  
IDHM – Índice de Desenvolvimento Humano Municipal  
IP – Internet Protocol  
ISO – International Organization for Standardization  
JSON – Javascript Object Notation  
JVM – Java Virtual Machine  
MPC – Main Press Centre  
OGC – Open Geospatial Consortium  
PaaS – Platform as a Service  
PL – Procedural Language  
PNUD – Programa das Nações Unidas para o Desenvolvimento  
REST – Representational State Transfer  
SGBDOR – Sistema Gerenciador de Banco de Dados Objeto Relacional  
SIG – Sistemas de Informação Geográfica  
SQL – Structured Query Language  
SRID – Spatial Reference System Identifier  
SSH – Secure Shell  
URI – Uniform Resource Identifier  
URL – Uniform Resource Locator  
UTM – Universal Transversa de Mercator  
W3C – World Wide Web Consortium  
WGS84 – World Geodetic System  
WKT – Well-Known Text  
XML – Extensible Markup Language

# CAPÍTULO 1

## INTRODUÇÃO

Os Jogos Olímpicos tem sua origem datada em 776 a.C em Olímpia, na Grécia Antiga, onde a cada quatro anos, eram realizados festivais religiosos e atléticos com grande número de participantes, que disputavam entre si provas de corridas pedestres, corridas eqüestres, lutas e pentatlo. As edições desse período são conhecidas no campo da história, como Jogos Olímpicos da Antiguidade, e só permitiam a disputa entre homens (SCHNAPP, 1996 apud CHIÉS, 2006), dos quais, os vencedores eram contemplados com coroas de folhas de oliva.

Após vários séculos e um longo período de inatividade desses jogos, a organização foi retomada no ano de 1896, em Atenas, capital da Grécia, graças ao empenho do pedagogo e historiador Pierre de Frédy, conhecido como Barão de Coubertin. Este visionário francês fundou no ano de 1894 o Comitê Olímpico Internacional (COI) e foi o grande mentor do movimento olímpico, idealizando o renascimento dos jogos da era antiga. Essa edição, também conhecida como a edição I dos Jogos Olímpicos da Era Moderna, envolveu atletas de quatorze nações diferentes disputando entre si nove modalidades esportivas, e distribuiu aos melhores competidores medalhas de ouro, prata e bronze.

Depois do seu ressurgimento neste formato bem definido, os Jogos Olímpicos de Verão passaram por uma enorme evolução no decorrer do século XX, sendo disputados em diversas cidades a cada quatro anos. A quantidade de nações participantes foi crescendo exponencialmente, bem como a de modalidades em disputa. Além disso, as mulheres, com a conquista de seu espaço na sociedade e no campo esportivo, ajudaram a diversificar os níveis de competição e juntamente com os homens ascenderem ainda mais o número de atletas e medalhas distribuídas.

Diante de toda proporção e dimensão atingida, os Jogos Olímpicos se espalharam ao redor do mundo e despertaram o interesse de vários países a sediá-los. Para cada edição a ser realizada, as nações interessadas apresentam cidades como candidatas a organizá-los e numa votação dividida em etapas é definida a escolha da próxima cidade sede.

No dia 2 de outubro de 2009, o Comitê Olímpico Internacional (COI) anunciou como sede da Edição XXXI dos Jogos Olímpicos a cidade do Rio de Janeiro, após a mesma vencer a cidade de Madri na etapa final de votação com um número absoluto de 66 votos contra 32 da concorrente, definindo assim, esta como a primeira edição a ser realizada no continente da América do Sul. Ao todo, mais de 100 mil pessoas já estão e estarão envolvidas diretamente na organização do evento, incluindo 45 mil voluntários. Além delas, outros milhões de brasileiros serão impactados de alguma forma, seja na cidade sede ou no país. A expectativa é de que participem nos Jogos Olímpicos de 2016, cerca de 10.500 atletas de 204 diferentes nações, disputando entre si as 28 modalidades olímpicas nas 35 diferentes instalações de competição. E o número de pessoas ainda torna-se imensurável quando pensamos na quantidade de apaixonados pelo esporte, que irão acompanhar as disputas pelos mais variados modos de transmissão ao vivo em todo o globo.

Tendo em vista toda esta magnitude, a organização Rio 2016, juntamente com o governo e os comitês responsáveis, está viabilizando através de repasses financeiros uma série de construções e melhorias para a cidade, a fim de aprimorar sua infraestrutura. Uma boa recepção e acomodação dos atletas, turistas e imprensa mundial, a organização das instalações esportivas e vias de acesso até elas, e outros aspectos urbanísticos como a segurança e transporte dependem diretamente desse aprimoramento.

Como forma de apoio a esta infraestrutura, serão necessárias informações dos mais variados tipos para instrução dos turistas que estarão presentes para acompanhar os jogos, mas que na sua maioria não conhecem a cidade do Rio de Janeiro ou não sabem como se locomover dentro dela. Neste contexto, e em conjunto com os meios tecnológicos hoje existentes, este trabalho tem como objetivo geral: Desenvolver um *software* de orientação geográfica para atender a Edição XXXI dos Jogos Olímpicos de Verão.

Especificamente, objetiva-se:

Pesquisar e espacializar os principais elementos geográficos da cidade do Rio de Janeiro, envolvidos durante a organização dos Jogos Olímpicos de 2016;

Desenvolver um *software* de orientação geográfica para dispositivos móveis, baseado em mapas e na geolocalização de seus usuários;

Apontar aos usuários do *software*, os elementos geográficos de interesse em comum no mapa da cidade do Rio de Janeiro, como parques olímpicos, vila olímpica, aeroportos, terminais rodoviários, e outros pontos de grande concentração de pessoas, como unidades de saúde emergenciais, delegacias, principais hotéis, e pontos turísticos da cidade;

Oferecer aos usuários do *software*, rotas de acesso aos pontos de interesse pré-definidos a partir de sua posição geográfica, considerando nos mapas, três tipos de deslocamento: locomoção a pé, locomoção por direção de automóvel, ou locomoção por meio de transportes públicos;

Informar os usuários através de conteúdo textual, os passos pela qual se dá o acesso ao destino escolhido, bem como sua distância e o tempo aproximado de cada deslocamento.

Para a concepção desses objetivos o trabalho se estrutura da seguinte forma: A introdução, no primeiro capítulo, apresenta uma breve referência ao histórico dos Jogos Olímpicos e sua evolução, os objetivos do trabalho e a apresentação dos capítulos desta monografia. O segundo capítulo aborda o referencial teórico fornecendo embasamentos e detalhes técnicos do estudo realizado, bem como as tecnologias, linguagens e *softwares* envolvidos na construção do aplicativo proposto. Em seguida, no terceiro capítulo é apresentada a caracterização da área de estudo, com mapas e informações geográficas da cidade sede do Rio de Janeiro. No quarto capítulo são apresentados os materiais utilizados e os procedimentos metodológicos realizados na concepção dos resultados. O quinto capítulo apresenta os resultados obtidos com o *software* desenvolvido juntamente com suas telas e instruções de usabilidade. E por fim, no sexto capítulo são expostas as considerações finais com a análise crítica deste trabalho e reflexões sobre a sua utilização.

## CAPÍTULO 2

### FUNDAMENTAÇÃO TEÓRICA

#### 2.1 Geotecnologias e Abordagens Técnicas

A idealização e a construção do *software* proposto teve como embasamento primordial o estudo das principais geotecnologias atualmente existentes, combinado aos vastos recursos de linguagens, padrões e API's de programação largamente utilizados pelo mercado de tecnologia da informação, os quais são em geral, mantidos e aprimorados por empresas e comunidades de estudo no campo da ciência da computação.

As geotecnologias são um conjunto de técnicas que tem como função coletar, processar, analisar e oferecer informações com referência geográfica (LONGLEY et al., 2013). Neste contexto de tecnologias, se encaixam itens dos quais podemos destacar os sistemas de informações geográficas (SIG), a cartografia digital, o sensoriamento remoto, a topografia automatizada e o sistema de posicionamento global (GPS).

Segundo Bucher (2012), um SIG é formado pela combinação de sistemas de *hardware*, *software*, informações espaciais, procedimentos computacionais e recursos humanos; e é usado para viabilizar a análise, gestão e representação do espaço e dos fenômenos que nele ocorrem. Os SIG's vem há anos sofrendo uma constante evolução arquitetural e de suas funcionalidades, graças a *internet* e as padronizações impostas pelas organizações *Open Geospatial Consortium* (OGC) e *International Organization for Standardization* (ISO) (LONGLEY et al., 2013).

A figura 2.1 ilustra a estrutura de um SIG moderno, que se integra a variados tipos de plataformas por meio de redes e administra diversos tipos de informação, como imagens digitais e bancos de dados.

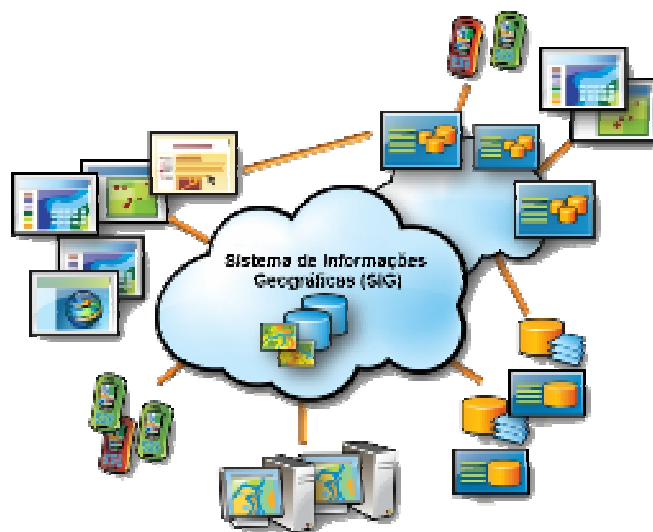


Figura 2.1 – Estrutura de um SIG moderno

Dentro do contexto dos SIG's atuais, é possível ainda trabalhar com recursos de geolocalização. A geolocalização é o processo pelo qual pode ser estabelecido o local exato ou aproximado de um determinado objeto espacial na superfície terrestre, através da atribuição de coordenadas (LONGLEY et al., 2013). De acordo com Keppelen et al. (2013),

“a geolocalização é uma informação expressa em coordenadas geográficas, que similarmente a um ponto do plano cartesiano, são compostas de duas partes obrigatoriamente: latitude e longitude. Há também uma terceira informação (opcional) que pode compor uma coordenada geográfica: a altitude (similar ao z do plano cartesiano).” (KEPPELEN et al., 2013)

Atualmente, esse conceito tem grande importância devido à natureza contextual de sua informação, cada dia mais presente em redes sociais, jogos e em navegação por mapas (KEPPELEN et al., 2013). Uma das maneiras mais precisas de reconhecimento da geolocalização e da navegação por coordenadas geográficas se dá por meio do sistema de posicionamento global (GPS). O desenvolvimento deste sistema foi iniciado pelo Departamento de Defesa dos Estados Unidos (DOD) em 1973, e tinha como objetivo oferecer a posição instantânea, a velocidade e o horário de um ponto qualquer sobre a superfície terrestre ou bem próxima a ela, num referencial tridimensional (LETHAM, 1996).

O sistema de navegação GPS entrou em operação no início da década de 90 e é baseado em uma constelação de vinte e quatro satélites projetados, de forma que pelo menos quatro dos mesmos, estejam sempre visíveis no plano horizontal do observador, em qualquer momento e lugar do mundo, considerando os movimentos do globo terrestre (BLITZKOW, 1995). A figura abaixo ilustra esta abordagem (Fig. 2.2).

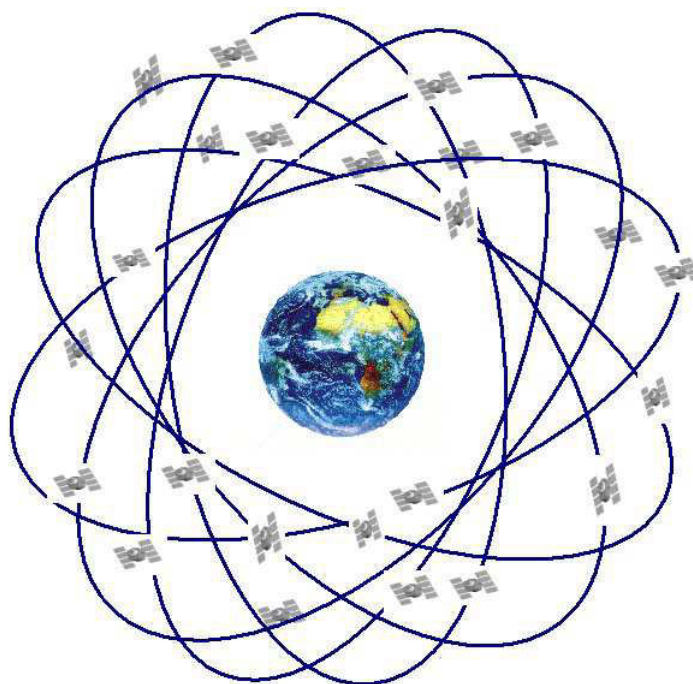


Figura 2.2 – Cobertura de satélites no sistema de navegação GPS

Para o uso do recurso de navegação, e em complemento à cobertura de satélites em órbita, são necessários os aparelhos receptores. Os receptores são capazes de captar os sinais eletromagnéticos oriundos dos satélites para determinação de suas próprias coordenadas, e oferecer navegações baseadas em seqüências de cálculos. Nos dias atuais, os receptores podem ser encontrados numa variedade de formatos, estando presentes em dispositivos dedicados ou integrados a relógios, carros e outros objetos.

Os computadores e dispositivos móveis modernos como *smartphones* e *tablets*, também dão suporte ao reconhecimento das coordenadas geográficas pelo GPS, através de seus respectivos sistemas operacionais e em conjunto com seus navegadores *web*, também conhecidos como *browsers*.

Segundo Keppelen et al. (2013), atualmente o GPS oferece a maior precisão para a obtenção da geolocalização, porém existem, além dele, outras três possibilidades para fazer este reconhecimento. São elas: a informação por *Wi-Fi*, a informação por antenas de celular, e o recurso de *IP lookup*.

Um dispositivo conectado a uma rede *Wi-Fi* é capaz de reconhecer a geolocalização a partir de serviços *online* que informam a localização previamente cadastrada do roteador em que o aparelho está conectado. Já pela abordagem das antenas de celular, um dispositivo móvel conectado a um serviço de celular, pode realizar o reconhecimento pela triangulação das antenas mais próximas, tentando definir assim uma área comum. Esse método é menos preciso que os anteriores, e trabalha com margens de erro maiores, geralmente resultando na delimitação de uma área. A figura a seguir apresenta esse modelo ilustrado (Fig. 2.3).



Figura 2.3 – Modelo ilustrado de geolocalização por triangulação de antenas

No recurso de *IP lookup*, quando um dispositivo está conectado à *internet*, a geolocalização pode tentar ser obtida a partir de serviços de pesquisa de endereços de *Internet Protocol* (IP). Esse método é comumente utilizado pelos computadores de localização fixa, conectados por meio de conexão a cabo (KEPPELEN et al., 2013).

## 2.2 Plataforma *JBoss OpenShift* e o conceito de *Cloud Computing*

O *JBoss OpenShift* é uma plataforma de serviços (PaaS) que possibilita de modo simples e prático a hospedagem de projetos em servidores remotos, utilizando assim, recursos de *hardware* e memória de máquinas remotas. Essa abordagem de serviço está inserida num conceito amplamente difundido no campo da informática, chamado de computação em nuvem, ou no inglês *Cloud Computing*, e utiliza a *internet* como caminho para estabelecer conexões e executar requisições em servidores remotos. De acordo com Taurion (2009), “*Cloud Computing* é um conjunto de recursos como capacidade de processamento, armazenamento, conectividade, plataformas, aplicações e serviços disponibilizados na *internet*”. Esse autor aponta também que há anos a computação em nuvem vem se apresentando como o cerne de um movimento de profundas transformações do mundo da tecnologia.

A figura 2.4 ilustra a abordagem de *Cloud Computing*, apresentando tipos diversos de dispositivos, tais como computadores fixos, *smartphones* e *tablets*, se comunicando com serviços e aplicações da nuvem, que por sua vez oferece capacidade de armazenamento e processamento sustentados por uma infraestrutura de máquinas servidoras.

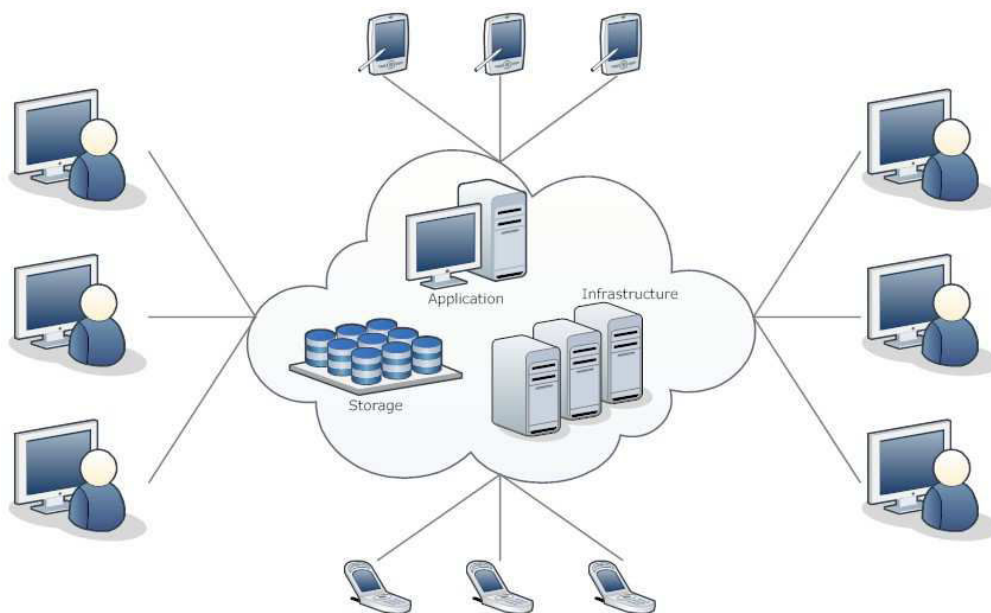


Figura 2.4 – Representação ilustrativa do conceito de computação em nuvem

### 2.3 Banco de dados *PostgreSQL* e a extensão geográfica *PostGIS*

O *PostgreSQL* é um sistema gerenciador de banco de dados objeto relacional (SGBDOR), desenvolvido como projeto de código aberto, que contempla recursos triviais e avançados como: armazenamento de dados, consultas complexas, integridade transacional, indexação, controle de concorrência multi-versão, suporte ao modelo híbrido objeto-relacional, gatilhos e visões; além de oferecer suporte as linguagens procedurais *PL/pgSQL*, *PL/Python*, *PL/Java*, *PL/Perl*.

Um banco de dados *PostgreSQL*, por sua natureza, tem capacidade de armazenar e manipular um amplo conjunto de tipos de dados. Porém, para que o mesmo suporte tipos de dados georreferenciados, é necessário que se ative a extensão geográfica *PostGIS*. A extensão *PostGIS* é um complemento gratuito de código fonte livre e permite que objetos *GIS* sejam armazenados e manipulados, implementando operadores, funções, métodos de indexação e catálogo de metadados espaciais, com compatibilidade aos padrões OGC e *SQL/MM Spatial*.

A partir de sua versão 1.5, ao ativar o complemento *PostGIS*, dois objetos de metadados passam a ser criados automaticamente na definição de cada banco de dados: a *view GEOGRAPHY\_COLUMNS* e a tabela *SPATIAL\_REF\_SYS*.

A *view GEOGRAPHY\_COLUMNS* é usada como referência interna no banco de dados, a fim de rastrear as tabelas que possuam colunas de tipos geográficos. Por padrão, esta *view* não apresenta nenhum registro no ato de sua seleção.

Já a tabela *SPATIAL\_REF\_SYS* é responsável por armazenar os identificadores numéricos e descrições textuais compatíveis com a tabela de dados do padrão OGC, e lista mais de 3000 sistemas de referência espacial conhecidos. Desta maneira, é possível realizar, dentro do sistema gerenciador, diferentes combinações de projeções e *datums*, necessários para cálculos de precisão em todas as áreas do planeta. A figura seguinte apresenta uma parte dos registros contidos na tabela *SPATIAL\_REF\_SYS*, onde se observa na linha em destaque, o registro referente ao *datum* padrão de uso mundial *World Geodetic System* (WGS84), com seu identificador SRID número 4326 (Fig. 2.5).

srid [PK] integer	auth_name character var	auth_srid integer	srtxt character varying(2048)
4315	EPSG	4315	GEOGCS["Conakry 1905", DATUM["Conakry_1905", SPHEROID["Clarke 1880 (IGN)", 637
4316	EPSG	4316	GEOGCS["Dealul Piscului 1930", DATUM["Dealul_Piscului_1930", SPHEROID["Intern
4317	EPSG	4317	GEOGCS["Dealul Piscului 1970", DATUM["Dealul_Piscului_1970", SPHEROID["Krasso
4318	EPSG	4318	GEOGCS["NGN", DATUM["National_Geodetic_Network", SPHEROID["WGS 84", 6378137, 29
4319	EPSG	4319	GEOGCS["KUDAMS", DATUM["Kuwait_Utility", SPHEROID["GRS 1980", 6378137, 298.2572
4322	EPSG	4322	GEOGCS["WGS 72", DATUM["WGS_1972", SPHEROID["WGS 72", 6378135, 298.26, AUTHORITY
4324	EPSG	4324	GEOGCS["WGS 72BE", DATUM["WGS_1972_Transit_Broadcast_Ephemeris", SPHEROID["WG
4326	EPSG	4326	GEOGCS["WGS 84", DATUM["WGS_1984", SPHEROID["WGS 84", 6378137, 298.257223563, AU
4414	EPSG	4414	PROJCS["NAD83(HARN) / Guam Map Grid", GEOGCS["NAD83(HARN)", DATUM["NAD83_High
4415	EPSG	4415	PROJCS["Katanga 1955 / Katanga Lambert", GEOGCS["Katanga 1955", DATUM["Katang
4417	EPSG	4417	PROJCS["Pulkovo 1942(83) / 3-degree Gauss-Kruger zone 7", GEOGCS["Pulkovo 19

Figura 2.5 – Parte dos registros armazenados pela tabela *SPATIAL\_REF\_SYS*

Para viabilizar a representação dos objetos geográficos espaciais, o *PostGIS* oferece suporte ao formato *Well-Known Text* (WKT), que inclui informações sobre o tipo do objeto e as coordenadas que formam sua geometria. A tabela a seguir apresenta os principais tipos de feições existentes, através da representação WKT (Tab. 2.1).








Geometria		Well Known Text (WKT)
POINT		POINT(10 20)
MULTIPOINT		MULTIPOINT(10 20, 15 15, 20 15)
LINestring		LINestring (10 30, 15 15, 25 40)
MULTILINestring		MULTILINestring ((40 40, 30 30), (15 15, 9 9))
POLYGON		POLYGON ((10 10, 40 10, 40 30, 50 30, 50 50, 30 50, 30 40, 10 40, 10 10))
MULTIPOLYGON		MULTIPOLYGON (((40 30, 50 30, 50 50, 30 50, 30 40, 40 40, 40 30)), ((10 10, 40 10, 40 30, 30 30, 30 40, 10 40, 10 10)))
COLLECTION		GEOMETRYCOLLECTION(POINT(10,20), LINestring(40 40, 30 30), POLYGON((15 15, 30 15, 30 30, 15 30, 15 15)))

Tabela 2.1 – Tipos de geometria em representação WKT

Conforme citado anteriormente, o complemento do *PostGIS* habilita uma série de funções extras para tipos de dados espaciais, que equivalem às operações de agregação e junção num banco de dados relacional. Elas são baseadas em relacionamentos espaciais, tais como: determinação de topologia entre dois objetos, aritmética de polígonos, cálculo de área, entre outras funções. A tabela seguinte apresenta exemplos das principais funções espaciais, juntamente de suas respectivas sintaxes e descrições (Tab. 2.2).

Função/Sintaxe	Descrição
ST_DISJOINT(obj1, obj2)	Analisa se dois objetos possuem pontos em comum.
ST_INTERSECTS(obj1, obj2)	Analisa se dois objetos possuem alguma intersecção.
ST_WITHIN(obj1, obj2)	Analisa se um obj1 está completamente dentro de obj2.
ST_CROSSES(obj1, obj2)	Analisa se dois objetos se cruzam.

Tabela 2.2 – Exemplos de funções espaciais

## 2.4 RESTful Webservices e o formato de representação GeoJSON

Segundo Fielding (2000), principal co-autor da especificação do protocolo *Hypertext Transfer Protocol* (HTTP), o *Representational State Transfer* (REST) pode ser definido como um estilo arquitetural para sistemas distribuídos multimídia. O *REST* tem como característica, uma arquitetura cliente-servidor de natureza *stateless*, ou seja, o servidor não mantém contextos de suas transações com o cliente, expandindo assim sua escalabilidade.

A figura abaixo ilustra o esquema de conversação cliente-servidor numa rede, por meio de requisições e respostas providas pela hierarquia de protocolos (Fig. 2.6).

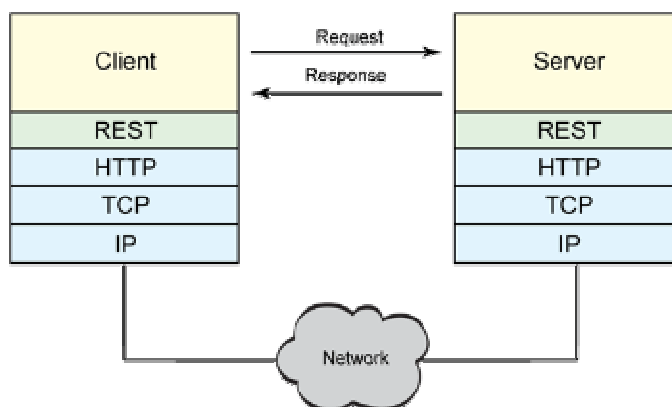


Figura 2.6 – Esquema de conversação com requisição e resposta

A partir do paradigma arquitetônico *REST*, são construídos os *webservices RESTful*. Conforme definição da *World Wide Web Consortium* – W3C (2000), um *webservice* é um sistema de *software* desenvolvido para suportar interoperabilidade entre máquinas sobre uma rede, o qual pode apresentar uma interface que o descreve.

Uma das principais características dos *webservices RESTful*, é que eles utilizam em suas operações, o conjunto de métodos do protocolo de comunicação HTTP, onde dentre eles, os principais são: *GET*, *POST*, *PUT* e *DELETE*. Quando um desses métodos é invocado por uma requisição, um código de *status* é embutido no contexto de resposta da mesma, sinalizando então sucesso ou falha na operação.

A figura seguinte ilustra uma operação com método *GET* sob o protocolo HTTP. No exemplo, uma requisição é realizada a partir de um dispositivo móvel e conduzida pela rede; o servidor a processa e devolve ao cliente o recurso desejado juntamente com o código de *status* 200, sinalizando sucesso na operação (Fig 2.7).

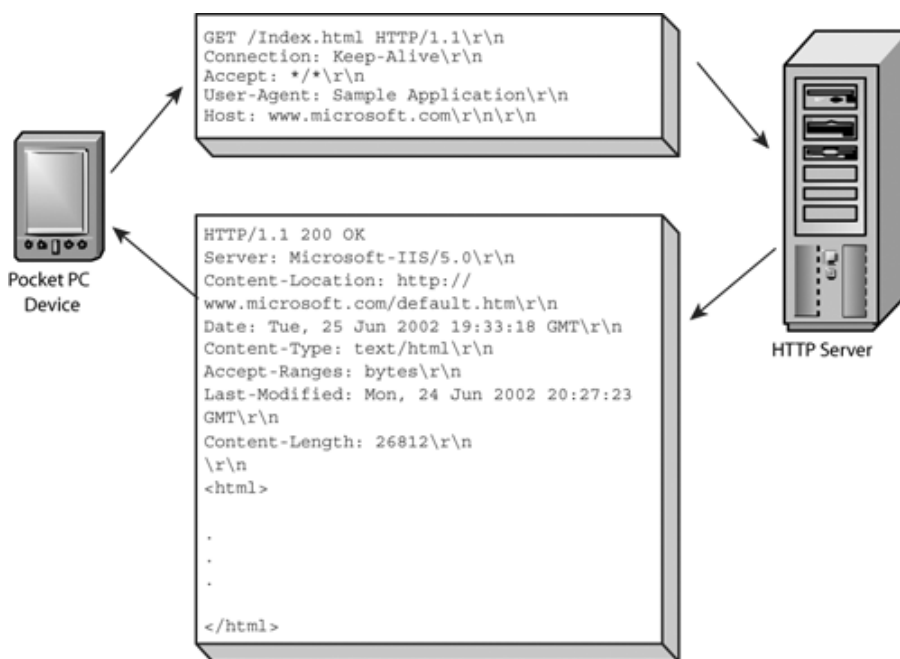


Figura 2.7 – Obtenção de recursos com a operação *GET*

As respostas de um *webservice RESTful* são oferecidas em diferentes formatos, onde os principais são: *Text Plain*, *Extensible Markup Language (XML)* e *Javascript Object Notation (JSON)*. Segundo Dincer (2013), a utilização da notação JSON está se tornando cada vez maior para a transferência de dados via *web*, por ser uma abordagem mais simples e leve em relação ao formato tradicional XML.

A figura 2.8 ilustra em caráter comparativo, as abordagens de XML e JSON para representação dos dados de uma entidade de exemplo.

XML	JSON
<pre> ...&gt; &lt;name&gt;Barry &amp; Associates, Inc.&lt;/name&gt; &lt;phone&gt;612-321-8156&lt;/phone&gt; &lt;street1&gt;14597 Summit Shores Dr&lt;/street1&gt; &lt;street2&gt;&lt;/street2&gt; &lt;city&gt;Burnsville&lt;/city&gt; &lt;state&gt;MN&lt;/state&gt; &lt;postalcode&gt;55306&lt;/postalcode&gt; &lt;country&gt;United States&lt;/country&gt; &lt; ... </pre>	<pre> {   "name"      : "Barry &amp; Associates, Inc.",   "phone"    : "612-321-8156",   "street1"  : "14597 Summit Shores Dr",   "street2"  : "",   "city"     : "Burnsville",   "state"    : "MN",   "postalcode" : "55306",   "country"  : "United States" } </pre>

Figura 2.8 – Comparativo entre XML e JSON

Atualmente, uma das muitas notações derivadas do JSON é o *GeoJSON*. Ela foi criada e especificada em 2008, por grupos de trabalho que pretendiam representar dados geográficos no formato JSON, mas não tinham um padrão definido. Hoje, apesar deste formato ainda estar em processo de aceitação para fazer parte dos padrões OGC, o mesmo já é suportado por múltiplos *softwares* e API's de programação, dentre os quais podemos destacar: *OpenLayers*, *MapServer*, *GeoServer*, *PostGIS*, *GoogleMaps Javascript API*, entre outros.

Dincer (2013), explica que, as geometrias representadas pelo *GeoJSON* são baseadas nos valores de coordenadas de latitude e longitude em graus decimais, e o *datum* padrão considerado é o WGS84. A tabela a seguir apresenta os principais tipos de formatos geométricos e suas respectivas representações nesta notação (Tab. 2.3).








Type	Examples	
Point		<pre>{   "type": "Point",   "coordinates": [30, 10] }</pre>
LineString		<pre>{   "type": "LineString",   "coordinates": [     [30, 10], [10, 30], [40, 40]   ] }</pre>
Polygon		<pre>{   "type": "Polygon",   "coordinates": [     [[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]   ] }</pre>
		<pre>{   "type": "Polygon",   "coordinates": [     [[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]],     [[20, 30], [35, 35], [30, 20], [20, 30]]   ] }</pre>
MultiPoint		<pre>{   "type": "MultiPoint",   "coordinates": [     [10, 40], [40, 30], [20, 20], [30, 10]   ] }</pre>
MultiLineString		<pre>{   "type": "MultiLineString",   "coordinates": [     [[10, 10], [20, 20], [10, 40]],     [[40, 40], [30, 30], [40, 20], [30, 10]]   ] }</pre>
MultiPolygon		<pre>{   "type": "MultiPolygon",   "coordinates": [     [       [[30, 20], [45, 40], [10, 40], [30, 20]]     ],     [       [[15, 5], [40, 10], [10, 20], [5, 10], [15, 5]]     ]   ] }</pre>
		<pre>{   "type": "MultiPolygon",   "coordinates": [     [       [[40, 40], [20, 45], [45, 30], [40, 40]]     ],     [       [[20, 35], [10, 30], [10, 10], [30, 5], [45, 20], [20, 35]],       [[30, 20], [20, 15], [20, 25], [30, 20]]     ]   ] }</pre>

Tabela 2.3 – Geometrias padrão OGC com a notação *GeoJSON*

## 2.5 Serviços integrados ao *GoogleMaps Javascript API*

Atualmente, uma das maneiras mais práticas de se desenvolver *softwares* é com a utilização de *Application Programming Interfaces* (API's). Uma API consiste num conjunto de rotinas prontas para utilização de outras aplicações. Através destas interfaces, resultados rápidos e satisfatórios podem ser obtidos com poucas linhas de código de programação efetivamente escritas, uma vez que a maior parte das funcionalidades desejadas, ou todas elas, já estão encapsuladas nas API's.

A *GoogleMaps Javascript API*, apresenta uma gama de funcionalidades e recursos, que podem ser facilmente invocados e utilizados através da linguagem de programação interpretada *Javascript*. De acordo com Stefanov (2012), o *Javascript* é uma linguagem de *script client-side* projetada inicialmente para rodar em navegadores *web*, mas que nos dias atuais se faz presente em variadas plataformas.

Em conjunto com modernas interfaces gráficas e enormes bases de dados pré-coletados em campo pelos equipamentos geotecnológicos da empresa *Google*, a *GoogleMaps Javascript API* possibilita a renderização e a manipulação de *layers* e objetos vetorizados sobre mapas e imagens de satélite do planeta terrestre. Itens como marcadores, rotas e geometrias padrões do pacote de representação OGC, podem ser programados para serem inseridos e alterados em tempo de execução de *software*. A figura a seguir apresenta como exemplo, uma imagem de satélite editada pelo *GoogleMaps*, e acompanhada de um marcador sobre o continente europeu (Fig. 2.9).



Figura 2.9 – Imagem de satélite editada pelo *GoogleMaps* com marcador

De acordo com Dincer (2013), um dos serviços favoritos entre os usuários do *GoogleMaps*, é o de obtenção de direções e instruções, chamado *Google Directions Service*. O *Google Directions* é um serviço que se integra a *GoogleMaps Javascript API*, capaz de oferecer rotas de acesso entre dois ou mais pontos definidos sobre o mapa. Isso é viável, graças à recuperação de informações a partir de ricas bases de dados mantidas pela *Google*. Entre as principais informações, estão direções de trânsito nas ruas, pontes e túneis, bem como itinerários de transporte público e informações sobre o tráfego em tempo real. Para a definição correta de suas rotas, o serviço ainda considera o meio de locomoção do usuário, podendo ser por caminhada, direção própria de automóveis, deslocamento com transporte público, ou por bicicleta em alguns locais de conhecimento específico.

Outro serviço que pode ser combinado junto ao *GoogleMaps Javascript API* é o recurso de geocodificação do *Google Geocoding Service*. A geocodificação é o processo de conversão de endereços de ruas para latitude e longitude, ou para algum sistema universal de coordenadas similar (LONGLEY et al., 2013). Esta abordagem viabiliza a busca de rotas pelo usuário, bastando a ele informar ao serviço, os endereços de origem e destino desejados, para que o serviço os converta em coordenadas e lhe retorne com a rota definida.

A figura seguinte apresenta como exemplo, três rotas de caminhada entre dois endereços informados como parâmetros, sugeridas por estes serviços integrados (Fig. 2.10).

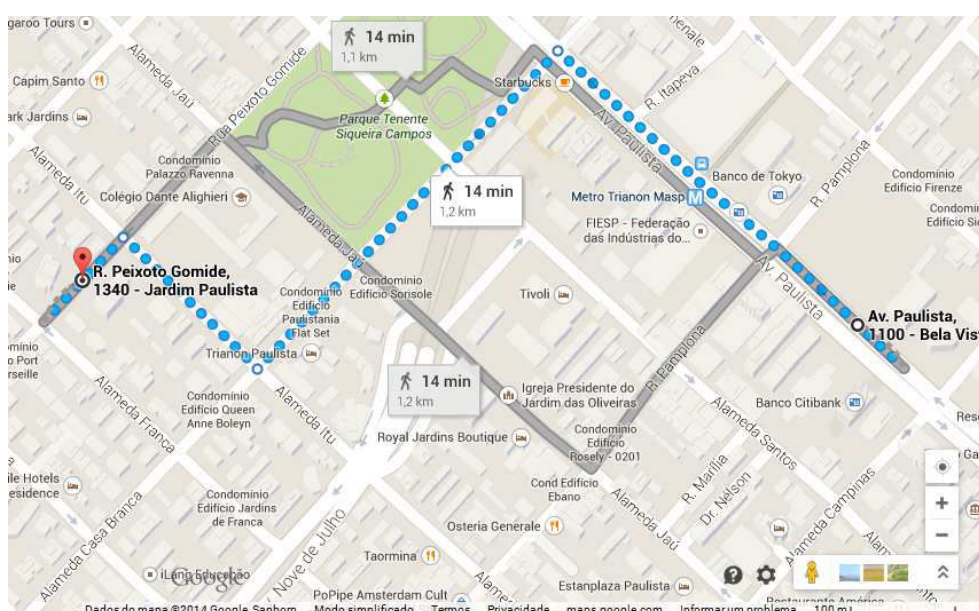


Figura 2.10 – Exemplo de vetorização de rotas entre pontos

### CAPÍTULO 3

## CARACTERIZAÇÃO DA ÁREA DE ESTUDO

A área objeto desse estudo é o perímetro territorial da cidade do Rio de Janeiro, no estado do Rio de Janeiro, região Sudeste do Brasil. Seus limites geográficos dados em projeção UTM e *datum* WGS84 são compostos pelas coordenadas 620.347 a 7.485.946 e 698.267 a 7.444.538. A figura abaixo apresenta o mapa de limite municipal projetado sobre uma imagem de satélite da cidade (Fig. 3.1).

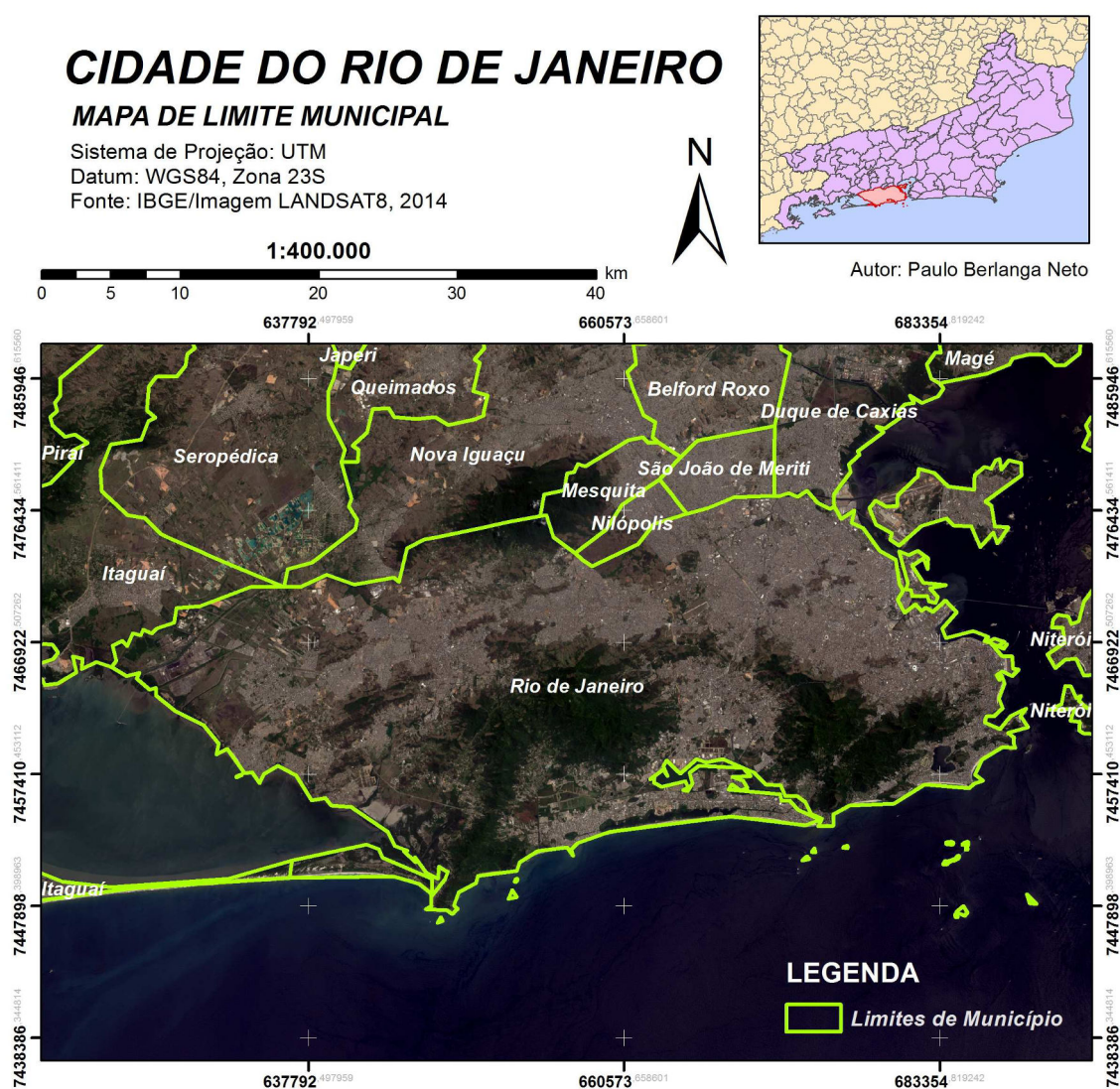


Figura 3.1 – Mapa da cidade do Rio de Janeiro

A cidade do Rio de Janeiro possui atualmente mais de 6 milhões de habitantes distribuídos numa área de cerca de 1200km<sup>2</sup> e apresenta um IDHM de 0.799, sendo um índice considerado alto pelo Programa das Nações Unidas para o Desenvolvimento (PNUD).

Para a realização dos Jogos Olímpicos, a organização Rio 2016 dividirá seus locais de competição entre quatro zonas: Copacabana, Barra, Maracanã e Deodoro. A figura abaixo apresenta a concentração dos pontos de disputa e suas respectivas regiões (Fig 3.2).

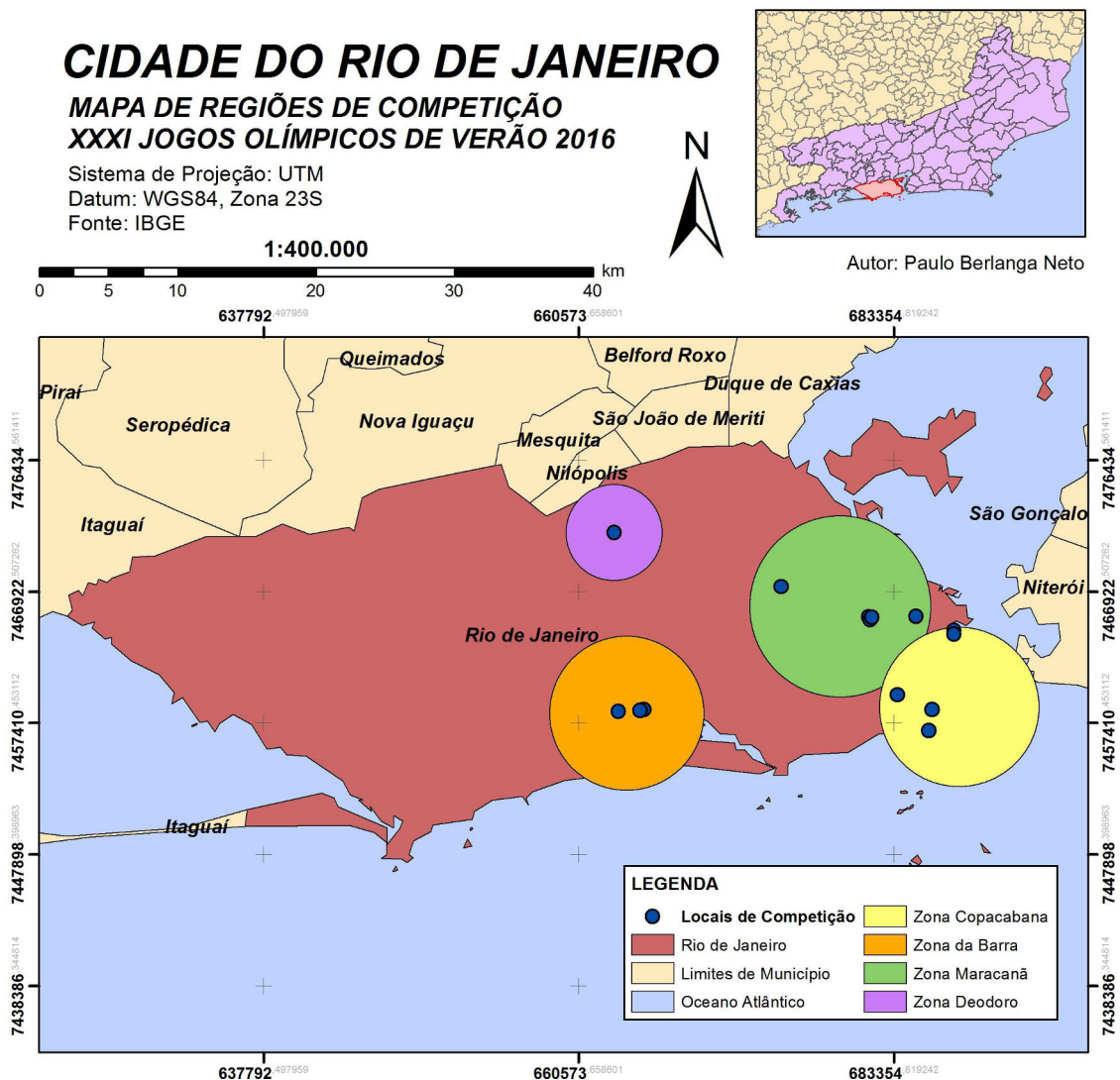


Figura 3.2 – Regiões e locais de competição

Copacabana é um dos bairros mais famosos da cidade e se encontra na Zona Sul, possuindo como cartão postal a sua praia com extensão de quatro quilômetros. Esta região apresenta uma série de montanhas e pontos turísticos mundialmente conhecidos, como o Pão de Açúcar e o Corcovado. As instalações na região incluem o Parque do Flamengo, o Estádio da Lagoa, a Marina da Glória e o Forte de Copacabana, além da Arena de Vôlei de Praia, construída temporariamente nas areias da praia. Nelas, serão disputadas, nos Jogos Olímpicos de Verão, oito competições: marcha atlética, ciclismo de estrada, maratonas aquáticas, triatlo, vela, vôlei de praia, remo e canoagem de velocidade.

O bairro da Barra da Tijuca será o maior ponto de concentração para os jogos. Ele está situado na Zona Oeste dessa cidade numa região cercada por lagoas. A região da Barra acomodará quatorze instalações onde serão realizadas competições de vinte e um esportes olímpicos: boxe, tênis de mesa, *badminton*, levantamento de peso, ginástica artística, ginástica rítmica, ginástica de trampolim, ciclismo de pista, saltos ornamentais, pólo aquático, natação, nado sincronizado, basquetebol, judô, *taekwondo*, luta greco-romana, luta estilo livre, handebol, esgrima, golfe e tênis. A Vila Olímpica e Paralímpica, o Parque Olímpico da Barra, o Riocentro, o Centro Internacional de Transmissões IBC/MPC, e a Vila de Mídia da Barra também ficam localizados nessa região.

A região do Maracanã abrange os bairros Maracanã e Engenho de Dentro na Zona Norte da cidade, e também o bairro Cidade Nova, no Centro. É uma região com predominância de classe média alta, e que em alguns pontos ainda conserva características urbanas do século XIX. Esta região é famosa por abrigar o Complexo Esportivo do Maracanã e o Estádio Olímpico João Havelange. Nela, serão disputadas seis modalidades olímpicas: atletismo, atletismo de maratona, tiro com arco, pólo aquático, futebol e voleibol.

Em Deodoro, situado na Zona Oeste da cidade, está acontecendo a reforma e o aprimoramento das instalações utilizadas nos Jogos Pan-americanos Rio 2007 com o objetivo de atender também aos Jogos Olímpicos. Esta região é cercada por espaços verdes e será palco de onze competições olímpicas: hipismo com saltos, hipismo adestramento, concurso completo de equitação, ciclismo *mountain bike*, ciclismo BMX, pentatlo moderno, tiro esportivo, canoagem *slalom*, hóquei sobre grama, esgrima e *rugby*.

## CAPÍTULO 4

### MATERIAIS E MÉTODOS

O fluxograma abaixo apresenta a estrutura geral da metodologia utilizada para atingir o objetivo deste trabalho. Nos tópicos desse capítulo, serão abordados de maneira detalhada os métodos utilizados (Fig. 4.1).

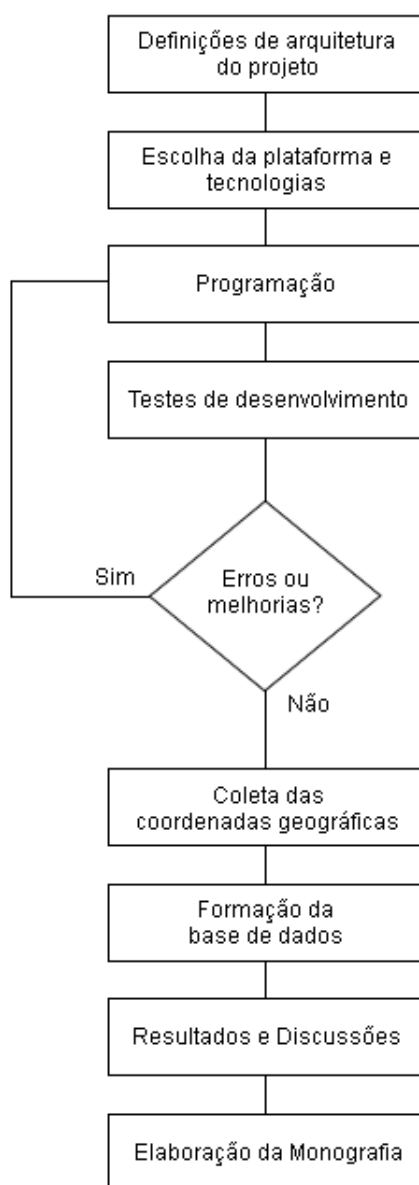


Figura 4.1 - Fluxograma da estrutura geral da metodologia.

#### 4.1 Especificação de desenvolvimento do *software*

O ambiente de desenvolvimento para o *software* proposto foi preparado na IDE *Eclipse*, onde foram trabalhadas as diversas tecnologias, *frameworks* e linguagens de programação utilizadas, e teve como base de execução a plataforma *JAVA*.

Uma vez que a arquitetura do *software* foi baseada no modelo de aplicação distribuída cliente-servidor, podemos dividir a apresentação das tecnologias utilizadas também nesses dois lados distintos: *back-end* e *front-end*.

A escolha e configuração das tecnologias de natureza *back-end* foram o ponto inicial da preparação na IDE, onde através da instalação do *plugin JBoss OpenShift Tools* da empresa *Red Hat JBoss*, foi utilizada a plataforma de serviços *JBoss OpenShift*. No plano gratuito desta plataforma são oferecidas de maneira limitada, umas séries de tecnologias embarcadas, que possibilitaram assim, uma combinação personalizada das mesmas. As seguintes tecnologias foram escolhidas para o *back-end* do projeto: o *JAVA* e seu ambiente de execução *Java Virtual Machine (JVM)*; o servidor de aplicação *JBoss Application Server*; o repositório de arquivos *GIT*; e uma instância do banco de dados *PostgreSQL*. Todas elas compõem o lado servidor do projeto e foram instaladas sem demandar intervenções complexas, uma vez que as configurações básicas vêm definidas na própria plataforma por padrão.

As tecnologias de natureza *front-end* foram escolhidas para trabalhar sobre os navegadores *web* e formam o lado cliente do projeto, podendo ser executadas a partir de diversos dispositivos como computadores fixos, *notebooks*, *smartphones* e *tablets*. A escolha das mesmas foi com a linguagem para estruturação de páginas *web HTML5*; as folhas de estilo e formatação *CSS3*; a linguagem interpretada *Javascript* e a biblioteca *cross-browser JQuery*.

A figura a seguir ilustra uma visão arquitetural do *software* no modelo de aplicação distribuída e as respectivas tecnologias escolhidas para compor seu *back-end* e *front-end* (Fig. 4.2).



Figura 4.2 – Composição das tecnologias do *software* distribuído

A comunicação entre a máquina cliente, a partir da qual foi desenvolvido o projeto, e a plataforma servidora do *JBoss OpenShift* baseada na computação em nuvem, foi estabelecida pela *internet* com a utilização de *Secure Shell* (SSH), permitindo conexões, manutenções e execuções de comandos no ambiente remoto. Como complemento a este protocolo, foi utilizada uma *SSH Key* de criptografia, garantido assim, a segurança na comunicação pela *internet* entre os dois lados. A figura abaixo exibe uma representação da comunicação da máquina cliente ao ambiente remoto via rede por SSH, trafegando dados criptografados na porta de comunicação 22, previamente autorizada no *firewall* (Fig. 4.3).

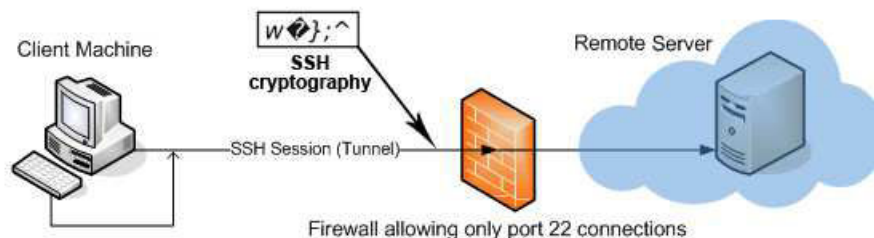


Figura 4.3 – Representação de comunicação ao ambiente remoto via SSH

Conforme citado, a plataforma *JBoss OpenShift* oferece o sistema de gerenciamento de banco de dados objeto relacional *PostgreSQL*, que vem instalado e configurado por padrão, onde um banco de dados também é automaticamente criado. Este banco de dados, porém, não suporta em sua essência os recursos de manipulação e armazenamento de feições e dados geográficos. Para que isso fosse possível, foi necessário ativar a extensão espacial do *PostGIS*. Para ativar essa extensão, foi necessário abrir uma conexão SSH com o servidor de instalação do banco de dados e rodar o comando abaixo ilustrado pela figura 4.4.

```
CREATE EXTENSION postgis;
```

Figura 4.4 – Console e comando de ativação da extensão *PostGIS*

Com a extensão geográfica ativada para o banco de dados, foi elaborada a tabela responsável por armazenar as feições utilizadas no *software*. A mesma foi definida com o nome de *location* e deve armazenar em uma de suas colunas, feições do tipo *POINT*. A figura abaixo apresenta o console com o *script* de criação desta entidade (Fig 4.5).

```
CREATE TABLE location (  
  id INT,  
  name CHARACTER VARYING(40),  
  type CHARACTER VARYING(40),  
  geom GEOGRAPHY (POINT, 4326)  
);
```

Figura 4.5 – Console e comando de criação da tabela *location*

Para alimentar esta tabela e inserir nela as feições de pontos, foi utilizado um comando de inserção combinado a função *ST\_GeogFromText*, permitindo assim, que as mesmas fossem criadas no banco de dados a partir de dados alfanuméricos.

As feições de pontos são fenômenos sem dimensões e compostas apenas por um par de coordenadas geográficas. Dessa forma, ao trabalhar com os *scripts* de banco de dados, bastou reunir as coordenadas de latitude e longitude dos locais de interesse e escrevê-las no formato de graus decimais, dentro da sintaxe da função *ST\_GeogFromText*. A figura abaixo mostra um exemplo do *script* para inserção de um novo registro na tabela *location*, onde o valor para a coluna *geom* representa um ponto georreferenciado que ficou disponível na lista de destinos possíveis do *software* (Fig 4.6).

```
INSERT INTO location (id, name, type, geom)  
VALUES (26, 'Santos Dumont (SDU)', 'Aeroportos',  
ST_GeogFromText('POINT(-22.911144 -43.164876)'));
```

Figura 4.6 – Console e *script* de inserção em tabela

Para recuperar os registros armazenados na tabela *location* do banco de dados, foi disponibilizado uma *Uniform Resource Identifier* (URI) de serviço, acessível a partir de navegadores *web* ou por aplicativos com comandos que implementam a operação *GET* do protocolo de comunicação HTTP. Quando esta operação é acionada pela URI, um *pool* de conexão com o banco de dados faz a seleção de todos os registros armazenados na tabela *location*, e os retorna na notação de feições geográficas *GeoJSON*. A figura a seguir apresenta a URI de serviço, que quando requisitada, responde na representação *GeoJSON* com todos os pontos armazenados na tabela (Fig. 4.7).

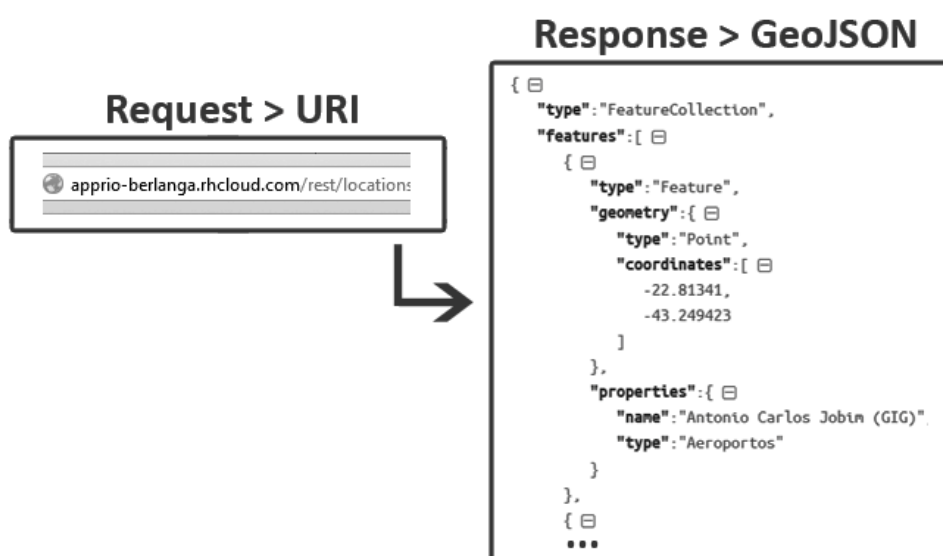


Figura 4.7 – Representação *request/response* do serviço de dados

Todos os pontos recuperados a partir dessa URI formam a lista de destinos disponíveis do *software*, trabalhando em conjunto com o conteúdo de mapas fornecido pela *GoogleMaps Javascript API* e seus serviços integrados para obtenção de geocodificação e rotas. Para utilizar os serviços integrados, foi necessário criar uma *API Key* atrelada a uma conta de *login* do *Google*, que é exigida para que haja um identificador único de qual conta está requisitando os serviços. O plano gratuito tem limitações diárias neste número de requisições, mas foi suficiente para garantir os testes deste trabalho.

Os mapas apresentados trabalham em conjunto com páginas interpretadas na linguagem de estruturação HTML5, que é suportada por grande parte dos navegadores *web* atuais, e traz

como um dos principais atrativos a sua API de reconhecimento da geolocalização. Ao acessar o endereço final de *Uniform Resource Locator* (URL) do *software*, foi programado para que o aplicativo solicite automaticamente através do navegador *web* a geolocalização do usuário para posicioná-lo no mapa, emitindo assim uma pergunta de confirmação. Quando o usuário aceita compartilhar esta informação, o navegador *web* tenta obter as coordenadas geográficas referentes à localização do dispositivo móvel ou computador fixo, utilizando os métodos de reconhecimento citados no capítulo de fundamentação deste trabalho. Com as coordenadas obtidas, um marcador é programaticamente posicionado sobre o mapa, indicando o local atual do usuário sobre o mapa provido pela *GoogleMaps Javascript API*. A figura 4.8 exibe um exemplo ao acessar o endereço URL do aplicativo.

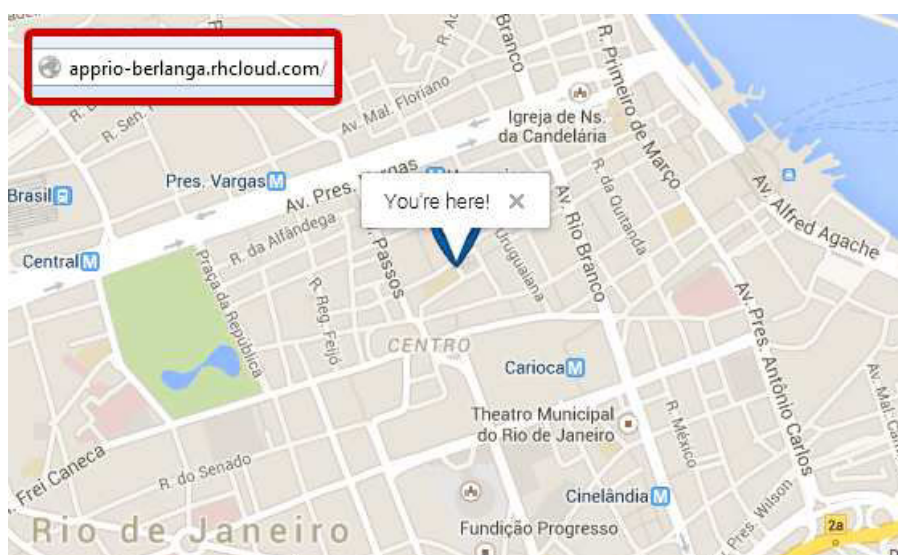


Figura 4.8 – Apresentação do marcador de geolocalização do usuário

Para definir a aparência do *software*, foi estabelecida a utilização dos mesmos moldes de arte criados pela equipe de *design* dos Jogos Olímpicos, de forma a padronizar o *look and feel* do mesmo. E quanto à estruturação dos elementos na página, foi estipulado um *layout* responsivo capaz de se adaptar ao contexto a partir do qual ele está sendo executado.

Para a utilização do conceito de *layouts* fluídos, os tamanhos de elementos chave da página foram definidos com valores percentuais ao invés de tamanhos fixos, garantindo assim, flexibilidade na sua exibição. A figura abaixo apresenta tipos de dispositivo com tamanhos distintos, e exemplifica a necessidade desta flexibilidade, uma vez que todos acessam os recursos pelo mesmo endereço de URL final (Fig. 4.9).

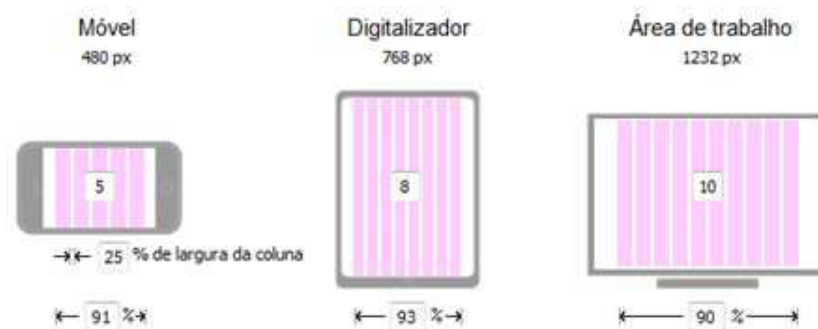


Figura 4.9 – Aplicação de *layouts* fluídos

Após toda a programação com a codificação dos elementos através das tecnologias de *back-end* e *front-end*, foi feita através de conexão SSH, a execução do comando *push*, de forma que o código fonte pudesse ser submetido para o repositório *GIT*, controlando assim as versões dos arquivos envolvidos. Desta forma também, a publicação do aplicativo ficou disponível para acesso em sua URL via *internet*, pois ao receber os arquivos do projeto, a plataforma *JBoss OpenShift* automaticamente se encarrega de implantá-los.

Com o aplicativo implantado, passaram a ser feitos os testes do *software*. A cada teste foi verificada a necessidade de correções ou melhorias no aplicativo e, para cada alteração realizada, foi executado novamente o comando *push*. Este passo foi repetido até que o *software* atingisse a usabilidade e os recursos objetivados sem falhas em sua execução.

## 4.2 Levantamento de informações e coleta de coordenadas geográficas

Para reconhecer os locais de relevância envolvidos na cidade do Rio de Janeiro durante os Jogos Olímpicos, foi feita uma análise das informações providas no *site* oficial da organização Rio 2016 e uma série de pesquisas em *sites* de busca.

A principal fonte para reconhecimento foi obtida a partir da publicação oficial feita pela organização Rio 2016, que apresentou uma amostra de análise de acessibilidade na cidade, listando os principais tipos de locais que julga de interesse público para a população que irá acompanhar os jogos. A tabela a seguir apresenta a amostra coletada (Tab. 4.1).

Serviço	Tipo de local	Local considerado na análise	Fonte de dados
Administração Pública	Cartórios	Aqueles dentro da cidade do Rio de Janeiro	--
	Escritórios de governo	Aqueles no estado e na cidade do Rio de Janeiro	IBDD
Educação	Escolas públicas	Aqueles no estado e na cidade do Rio de Janeiro	IBDD
Saúde	Postos de saúde	Aqueles na cidade do Rio de Janeiro	IBDD
	Hospitais Públicos		
Segurança	Delegacias	Aqueles na cidade do Rio de Janeiro	--
Transporte	Aeroportos	Aeroporto Internacional do Rio de Janeiro (Galeão/Antônio Carlos Jobim)*	Infraero
		Aeroporto do Rio de Janeiro (Santos Dumont)*	Infraero
	Terminais rodoviários	Rodoviária Novo Rio*	TRANSPORTAL
		Rodoviária Campo Grande	--
		Rodoviária Américo Fontenelle	--
	Rodoviária Alvorada	--	
Metrô	Linha 1 (19 estações) e Linha 2 (16 estações)*	METRÔRIO	
Turismo	Pontos Turísticos	Pão de Açúcar*	SETUR
		Cristo Redentor*	SETUR
		Orla Oceânica*	SETUR
		Estádio do Maracanã*	SETUR

Tabela 4.1 – Amostra de locais públicos para análise de acessibilidade

Além de informações extraídas da tabela, foram realizadas novas pesquisas em *sites* de busca, visando complementar os locais considerando seus critérios de relevância. Posteriormente, com todas as categorias e destinos definidos, a geocodificação dos locais foi feita a partir de seus endereços através do serviço *Google Geocoding*, possibilitando assim, a obtenção de suas respectivas coordenadas geográficas.

Ao final, bastou inserir as coordenadas geográficas no banco de dados para que os locais fossem listados para escolha como destinos no *software*. A tabela 4.2 exibe os destinos coletados e disponibilizados para seleção no aplicativo juntamente de suas categorias e marcadores de mapa.

CATEGORIAS	DESTINOS	CATEGORIAS	DESTINOS
 <b>Aerportos</b>	Antonio Carlos Jobim (GIG) Santos Dumont (SDU)	 <b>Terminais Metrô</b>	Estação Ipanema/Osorio L1 Estação Uruguaí L1 Estação Botafogo L2 Estação Pavuna L2
 <b>Delegacias</b>	9a D.P. Catete 13a D.P. Copacabana 20a D.P. Isabel	 <b>Terminais Rodoviários</b>	Novo Rio Américo Fontenele Campo Grande Alvorada
 <b>Hospitais</b>	Hosp. Espanhol do RJ Hosp. Albert Sabin Hosp. São Lucas Hosp. Federal Bonsucesso Hosp. São Vicente de Paulo	 <b>Zona da Barra</b>	Parque Aquático Maria Lenk Arena Olímpica do Rio Rio Centro Vila Olímpica
 <b>Hotéis</b>	Sheraton Rio Hotel Resort Royal Tulip Rio Othon Palace Atlântico Sul Copacabana Rio Windsor Excelsior	 <b>Zona Copacabana</b>	Estádio da Lagoa Arena de Vôlei de Praia Forte de Copacabana Marina da Glória Parque do Flamengo
 <b>Pontos Turísticos</b>	Corcovado/Cristo Redentor Pão de Açúcar Jardim Botânico Cidade do Samba Floresta da Tijuca Museu de Arte Moderna Teatro Municipal Paço Imperial Pedra da Gávea Igreja N. Sra. Candelária	 <b>Zona Deodoro</b>	Arena da Juventude Centro Aquático de Deodoro Centro Olímpico de Hipismo Centro Olímpico de Hóquei Centro Nacional de Tiro Estádio de Deodoro Parque Radical do Rio
 <b>Praias</b>	Barra da Tijuca Urca Copacabana Ipanema Arpoador Leblon Pepê Leme Vermelha Recreio dos Bandeirantes	 <b>Zona Maracanã</b>	Est. Olímpico J. Havelange Samb. Marquês da Sapucaí Estádio do Maracanã Ginásio do Maracanãzinho Pq. Aquático J. Delamare

Tabela 4.2 – Categorias e destinos disponíveis no *software*

## CAPÍTULO 5

### RESULTADOS E CONCLUSÕES

Observando o modo de acesso e as plataformas utilizadas para sua construção, o *software* desenvolvido resultou num aplicativo *web* para dispositivos móveis, simples e prático de ser utilizado. Por ser compatível com a maioria dos navegadores *web*, ele pôde ser caracterizado como uma ferramenta *cross-browser* que apresenta boa estabilidade, sem acusação de falhas graves e erros de execução. Os resultados dos testes de desempenho realizados no aplicativo também apontaram tempos de resposta satisfatórios, trabalhando relativamente bem inclusive em conexões de *internet* com baixa taxa de transferência de dados.

Quanto ao recurso de geolocalização, confirmou-se nos testes a fundamentação de que a obtenção do posicionamento via sistema GPS acusa essa informação com maior precisão, ao ponto em que os outros modelos trabalharam com margens de erros maiores, muitas vezes delimitando grandes áreas.

O serviço de rotas e informações do *Google Directions* integrado pelo *software*, se mostrou bastante preciso nos cálculos e sugestões de rota, demonstrando assim, que o mesmo possui uma grande base de conhecimento sobre a cidade do Rio de Janeiro e arredores, indicando as direções das vias em conformidade com o modelo real. Quanto as rotas por transporte público, notou-se nos testes realizados que, na grande maioria das vezes, foram apontados itinerários de ônibus e metrô corretamente, informando também os horários de partida e as companhias e viações envolvidas. Para as rotas por bicicleta, a base de conhecimento deste serviço ainda se mostrou em fase de aprendizado, deixando de considerar no traçado, rotas como ciclovias e praças, e por este motivo, esta opção de escolha foi desativada no *software*.

## 5.1 Usabilidade do *software*

A usabilidade do *software* se mostrou bastante eficaz, já que através de poucas ações de escolha nas listas apresentadas, o usuário pode ser orientado e chegar ao seu destino através das rotas traçadas e informações textuais.

A figura 5.1 apresenta a tela inicial do aplicativo, exibindo o mapa em *zoom* aproximado e centralizado de acordo com o marcador de geolocalização do usuário. As caixas de seleção acima do mapa permitem a escolha de uma categoria e destino para qual o usuário deseja se locomover.

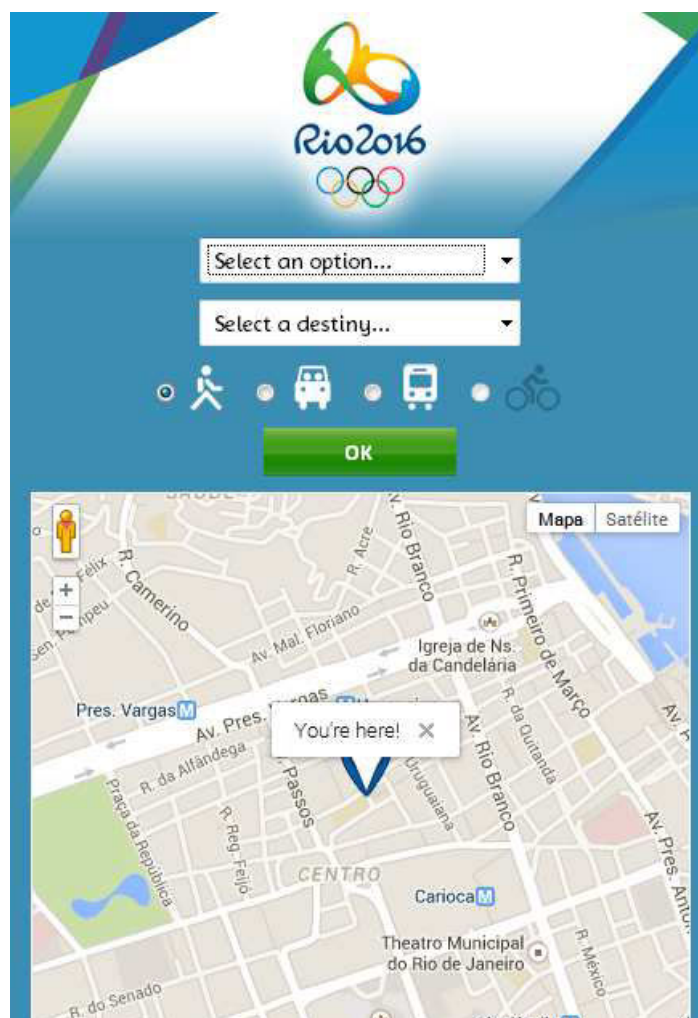


Figura 5.1 – Tela inicial do aplicativo

Ao seleccionar uma determinada categoria, o *software* automaticamente apresenta uma tomada geral sobre o mapa da cidade, configurando um *zoom* mais distante e exibindo marcadores temáticos dos pontos georreferenciados de acordo com a categoria previamente escolhida. A figura 5.2 ilustra esse comportamento, exemplificando com a seleção da categoria “Aeroportos”.

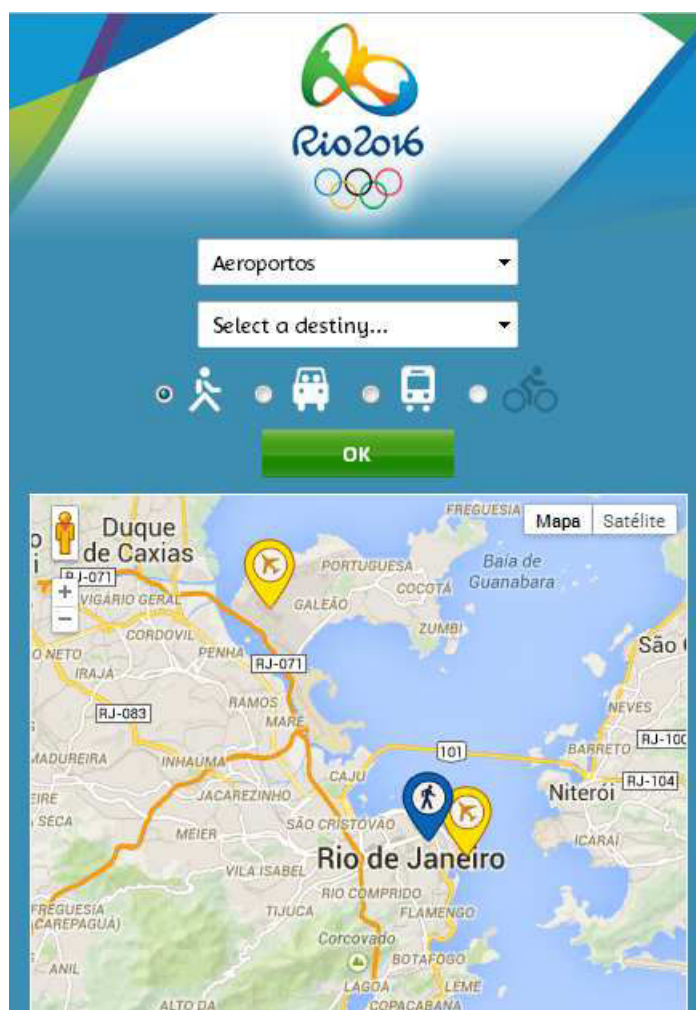


Figura 5.2 – Marcadores temáticos por categoria

Ao escolher um destino específico na segunda caixa de seleção, o *zoom* do mapa então é aproximado e o mesmo centraliza-se automaticamente de acordo com a localização de seu respectivo marcador. A figura 5.3 exhibe esse comportamento, tendo como exemplo o destino no aeroporto “Santos Dumont (SDU)”.

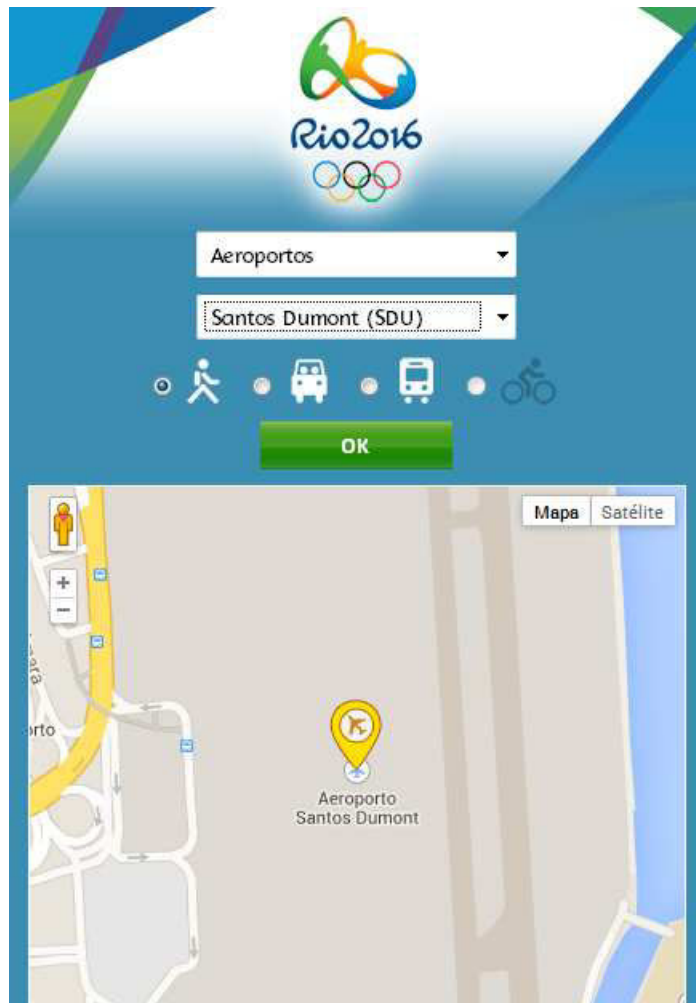


Figura 5.3 – Marcador de localização do destino escolhido

Por fim, para completar sua orientação, o usuário deve selecionar o modo de transporte através do qual deseja se locomover, onde das quatro opções existentes, três ficaram habilitadas. São elas: por caminhada, por direção própria de automóveis, ou por transporte público.

Após a respectiva seleção e clicando em *OK*, o aplicativo traça sobre o mapa a melhor rota entre a posição do usuário representada pelo marcador de origem “A”, e o marcador de destino representado pelo marcador “B”. A figura 5.4 exhibe como exemplo uma rota traçada até o Aeroporto Santos Dumont (SDU), considerando o carro como meio de transporte escolhido.



Figura 5.4 – Exemplo de rota traçada entre pontos A e B

Logo abaixo do mapa, informações textuais também são apresentadas em complemento à rota desenhada, como o tempo estimado de deslocamento e a distância entre os pontos calculada em quilômetros, além de uma sequência de instruções para orientação nos logradouros a serem percorridos. Como exemplo, a figura a seguir ilustra esses detalhes textuais, ainda considerando a rota por carro, de seu ponto de origem até o Aeroporto Santos Dumont (SDU) (Fig. 5.5).

<span style="color: green;">A</span> Instituto de Filosofia e Ciências Sociais, Instituto e Filosofia e Ciências Sociais - Largo São Francisco de Paula, 111-185 - Centro, Rio de Janeiro - RJ, 20051-070, Brasil	
3,1 km - cerca de 7 minutos	
1.	Siga na direção noroeste na Largo São Francisco de Paula em direção à R. Luís de Camões 16 m
<span style="color: blue;">↶</span> 2.	Vire à esquerda na R. Luís de Camões 0,2 km
3.	Pegue a 1ª à esquerda para Av. Passos 82 m
4.	Continue para Praça Tiradentes 0,1 km
<span style="color: blue;">↶</span> 5.	Curva suave à esquerda na R. Gustavo de Lacerda 84 m
<span style="color: blue;">↷</span> 6.	Curva suave à direita na Av. República do Paraguai 0,7 km
7.	Av. República do Paraguai faz uma curva suave à esquerda e se torna Largo da Lapa/R. Visc. de Maranguape 0,1 km
<span style="color: blue;">↶</span> 8.	Vire à esquerda para permanecer na Largo da Lapa/R. Visc. de Maranguape 82 m
9.	Continue para R. Teixeira de Freitas 0,1 km
<span style="color: blue;">↑</span> 10.	Continue em frente para permanecer na R. Teixeira de Freitas 0,1 km
<span style="color: blue;">↶</span> 11.	Vire à esquerda na Av. Beira Mar 1,0 km
<span style="color: blue;">⤴</span> 12.	Pegue a Av. Gen. Justo 72 m
<span style="color: blue;">↷</span> 13.	Pegue a saída em direção a Aeroporto Santos Dumont 47 m
<span style="color: blue;">↶</span> 14.	Mantenha-se à esquerda, siga as indicações para Aeroporto e pegue a Praça Sen. Salgado Filho 0,1 km
<span style="color: blue;">↶</span> 15.	Mantenha-se à esquerda para permanecer em Praça Sen. Salgado Filho 0,2 km
<span style="color: red;">B</span> Aeroporto Santos Dumont (SDU) - Praça Senador Salgado Filho, 702-774 - Centro, Rio de Janeiro - RJ, 20021-340, Brasil	
Dados cartográficos ©2014 Google	

Figura 5.5 – Informações textuais de rota por carro

Em especial, para a oferecer a opção de rota por transporte público, o *software* informa no campo textual sobre as linhas de ônibus e metrô ideais para o destino escolhido, e como se dá o acesso até elas. A figura a seguir, exhibe como exemplo uma escolha de deslocamento do ponto original do usuário até o ponto turístico “Museu de Arte Moderna” (Fig. 5.6).

  
 Rio2016

Pontos Turísticos ▾  
 Museu de Arte Moderna ▾

  
 OK



Versão beta da rota a pé. Seja cuidadoso – Este trajeto pode não ter calçadas ou caminhos de pedestres.

Instituto de Filosofia e Ciências Sociais, Instituto e Filosofia e Ciências Sociais - Largo São Francisco de Paula, 111-185 - Centro, Rio de Janeiro - RJ, 20051-070, Brasil

6,6 km – cerca de 31 minutos

 Ande para BRS 2 - Ouvidor  
 Cerca de 5 minutos

**BRS 2 - Ouvidor**  
 **128** Ônibus em direção a Leblon  
 13:37–14:01 (23 minutos, 5 pontos)  
 Serviço realizado por [Fetranspor](#)

**Avenida Das Nacoes Unidas próximo ao 402**  
 Ande para Avenida Infante Dom Henrique, 609 - Flamengo, Rio de Janeiro - RJ, Brasil  
 Cerca de 3 minutos

 Avenida Infante Dom Henrique, 609 - Flamengo, Rio de Janeiro - RJ, Brasil

Informações das agências locais:  
 Contém mais informações sobre tarifas, horários e avisos de serviços:  
[Fetranspor](#) - 0800 8861000

Dados cartográficos ©2014 Google

Figura 5.6 – Informações textuais de rota por transporte público

## CAPÍTULO 6

### CONSIDERAÇÕES FINAIS

Dada a complexa tarefa de orientação e locomoção dos turistas em grandes centros urbanos, como é o caso da cidade do Rio de Janeiro, o *software* atendeu aos objetivos propostos e se mostrou como uma excelente ferramenta de apoio, podendo atuar como um complemento a outros elementos tradicionais de orientação conhecidos, como sinalizações e placas.

Por ter características de um aplicativo *web* e ser compatível com a maioria das plataformas móveis, o *software* se apresentou como uma fonte de informações prática para consulta do usuário portador de um *smartphone* ou *tablet*, estando implantado e disponível para ser acessado a qualquer momento e em qualquer lugar, bastando apenas que o respectivo dispositivo esteja conectado a *internet*.

A capacidade de traçar rotas sobre o mapa a partir da posição corrente do usuário levando em consideração os três modos de deslocamento possíveis, permite que ele possa fazer a escolha do modo de locomoção mais viável para ir até o destino desejado, pois as estimativas de distância e tempo de deslocamento que o *software* apresentou podem ser informações determinantes para tomar tal decisão.

Por ter apresentado automaticamente os locais de interesse aos turistas, ele também permite economizar tempo e pesquisa do usuário que não conhece sobre os possíveis destinos e atrativos na cidade, uma vez que estas informações já foram coletadas e oferecidas em listas agrupadas por categoria.

Por fim, além dos benefícios citados, com seu caráter informativo e instrutivo, ele se demonstrou capaz de reduzir uma parcela considerável de trabalhos de orientação por parte dos voluntários da organização, contribuindo ao mesmo tempo com a sustentabilidade, já que a necessidade de folhetos, guias e impressos informativos diminuiriam drasticamente com a aquisição desta ferramenta.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Box, D. et. al. Simple Object Access Protocol (SOAP) - W3C 2000. Disponível em: <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>. Acesso em 22 de novembro de 2014;
- BLITZKOW D.; Navstar/GPS: um desafio tornado realidade. In: SIMPÓSIO BRASILEIRO DE GEOPROCESSAMENTO, 3, 1995, Anais, São Paulo, 1995.
- BUCHER. B.; LE BER. F. Innovative Software Development in GIS. Londres: ISTE, 2012. 352p.
- CHIÉS. P. V. “Eis quem surge no estádio: é Atalante!” A história das mulheres nos jogos gregos. *Revista Movimento*, Porto Alegre, v. 12, n. 03, p. 99-121, 2006.
- DINCER. A; URAZ.B. GoogleMaps Javascript API Cookbook. Birmingham: Pack Publishing, 2013. 299p.
- FIELDING. R.; Architectural Styles and the Design of Network-based Software Architectures. 2000. 162p. Tese (Doutorado) – University of California, Irvine, 2000. Disponível em: <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>. Acesso em 20 de novembro de 2014.
- KEPPELEN. G. et.al. Coletânea Front-End. São Paulo: Casa do Código, 2013. 260p.
- LETHAM. L. GPS Made easy: using global positioning systems in the outdoors. Seattle: Mountaineers, 1996.
- LONGLEY. P. A. et al. Sistemas e Ciência da Informação Geográfica. 3ª Ed. Porto Alegre: Bookman, 2013. 540p.
- RICHARDSON. L.; RUBY. S. RESTful Serviços Web. 1ª Ed. Rio de Janeiro: Alta Books, 2007. 336p.
- STEFANOV S.; Javascript Patterns. Sebastopol: O'Reilly Books, 2012.
- TAURION. C.; Cloud Computing: Computação em Nuvem - Transformando o mundo da Tecnologia da Informação. Rio de Janeiro: Brasport, 2009.