

Universidade Federal dos Vales do Jequitinhonha e Mucuri

Faculdade de Ciências Exatas e Tecnológicas

Departamento de Computação

Bacharelado em Sistemas de Informação

**Desenvolvimento de Algoritmo e Interface para  
Aplicações Roteamento Utilizando Open Source  
Routing Machine**

**Aurélio Antônio de Pádua Gandra**

Diamantina, 26 de junho de 2015

Universidade Federal dos Vales do Jequitinhonha e Mucuri

Faculdade de Ciências Exatas e Tecnológicas

Departamento de Computação

Bacharelado em Sistemas de Informação

# **Desenvolvimento de Algoritmo e Interface para Aplicações Roteamento Utilizando Open Source Routing Machine**

**Aurélio Antônio de Pádua Gandra**

Monografia submetida à Banca Examinadora designada pelo curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito para obtenção do título de Bacharel em Sistemas de Informação

Orientador: Prof. Dr. Alessandro Vivas de Andrade

Diamantina, 26 de junho de 2015

Aurélio Antônio de Pádua Gandra

**Desenvolvimento de Algoritmo e Interface para Aplicações Roteamento  
Utilizando Open Source Routing Machine**

Monografia submetida à Banca Examinadora designada pelo curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito para obtenção do título de Bacharel em Sistemas de Informação

**Banca Examinadora**

---

Prof. Dr. Alessandro Vivas Andrade

Universidade dos Vales do Jequitinhonha e Mucuri

---

Prof. Dr. Luciana Pereira de Assis

Universidade Federal dos Vales do Jequitinhonha e Mucuri

---

Prof. Dr. Cristiano Grijo Pitangui

Universidade Federal dos Vales do Jequitinhonha e Mucuri

Diamantina, 26 de Junho de 2015

# Agradecimento

Agradeço pelo conhecimento adquirido e pelo crescimento que foi obtido através deste estudo. Agradeço à minha família, pelo apoio constante em minhas aventuras.

Agradeço também ao Professor Alessandro Vivas por confiar em meu trabalho e pela grande disposição em ajudar e fazer disto uma realidade. Agradeço a todos da UFVJM pelo apoio e contribuição.

# Resumo

A maneira como os negócios são feitos no mundo atual está em mudança constante devido aos avanços tecnológicos e à globalização. Gestores dependem cada vez mais de dados para melhorar suas operações, e isto está mudando a forma como organizações definem seus processos. A área de Logística é uma das mais afetadas por este fator, o entendimento de cada camada funcional da empresa através do entendimento do processo completo de processamento de dados pode garantir grandes economias, aumento da eficiência e também o preço final ao consumidor. A aplicação do Problema de Roteamento de Veículos é peça chave e pode ser utilizado em organizações de todas as áreas que utilizam rotas.

O objetivo principal deste estudo é de desenvolver um sistema de boa interface utilizando Java como linguagem de programação, a API do OpenStreetMaps para mapeamento e o Open Source Routing Machine como máquina de roteamento. Tem uma abordagem simples e pode ser aplicado em uma grande variedade de problemas de roteamento de veículos. O sistema lê um arquivo texto criado pelo usuário com quantos endereços precisar, calcula a distância entre os pontos e imprime a rota final em um mapa JxMap.

Palavras-chave: Dados GPS, Roteamento de Veículos, OpenStreetMap, Open Source Routing Machine, Logística, JxMap

# Abstract

The way business is done in the world is in constant change due to the globalization and technology. Managers need to rely on data in order to improve business operations, and it is changing the way organizations set their processes. Logistics is one of the most affected areas by this factor, the understand of every layer of its functionality by understanding the whole process of data exchange can save big numbers improving the service efficiency and final cost to the costumer. The application Vehicle Routing Problem is key for it and can be used in any organization using routes.

The main goal of this study is to develop a well interfaced system utilizing Java, OpenStreetMaps API, and Open Source Routing Machine as programming languages, mapping, and routing machines. It has a simple approach and can be used in all kinds of Vehicle Routing applications. The system reads a text file created by the user with as many addresses as needed, calculates the distance between these points and plot a route on JxMap application.

Keywords: GPS data, Vehicle Routing, OpenStreetMap, JxMAP, Logistics, Open Source Routing Machine

# Sumário

Lista de Figuras.....	9
Capitulo 1.....	11
Introdução.....	11
Capítulo 2.....	14
Referencial Teórico.....	14
2.1 Modelagem da Superfície Terrestre.....	14
2.2 Sistemas de Coordenadas Geográficas.....	17
2.3 Sistemas de Coordenadas Esféricas.....	21
2.4 Sistemas de Coordenadas Planimétricas.....	21
2.5 Sistema de Posicionamento Global.....	22
2.6 Tipos de Dados de GPS.....	28
2.7 Formato de dados de GPS.....	30
Capítulo 3.....	40
OpenStreetMap/Open Source Routing Machine.....	40
3.1 OpenStreetMap.....	40
3.2 Google Maps.....	41
3.3 Ferramentas de Comparação de Aplicativos.....	42
3.4 Open Source Routing Machine.....	45
3.5 CloudMade.....	47
3.6 MapQuest.....	47
3.7 Google Directions API.....	48
Capítulo 4.....	50
Ferramentas.....	50
4.1 Eclipse.....	50
4.2 Java.....	51
4.3 Nominatim.....	51
4.4 SwingX-WS.....	53
4.6 JxMapView.....	54
4.5 JxMapKit.....	54
Capitulo 5.....	55
Desenvolvimento.....	55

5.1 Diagrama de Caso de Uso.....	59
5.2 Diagrama de Classes .....	60
5.3 Diagrama de Frequência .....	60
Capítulo 6.....	62
Testes .....	62
Capítulo 7.....	71
Conclusões Finais .....	71
Referências.....	72



# Lista de Figuras

Figura 2 - 1.....	15
Figura 2 - 2.....	15
Figura 2 - 3.....	16
Figura 2 - 5.....	18
Figura 2 - 6.....	19
Figura 2 - 9.....	22
Figura 2 - 11.....	25
Figura 2 - 12.....	26
Figura 2 - 13.....	26
Figura 2 - 14.....	27
Figura 2 - 15.....	29
Figura 2 - 16.....	29
Figura 2 - 17.....	30
Figura 2 - 18.....	31
Figura 2 - 19.....	32
Figura 2 - 20.....	33
Figura 2 - 21.....	34
Figura 2 - 22.....	37
Figura 2 - 23.....	38
Figura 2 - 24.....	39
Figura 3 - 1.....	43
Figura 3 - 2.....	44
Figura 3 - 3.....	45
Figura 3 - 4.....	46
Figura 3 - 5.....	47
Figura 3 - 6.....	48
Figura 3 - 7.....	49
Figura 4 - 1.....	54
Figura 5 - 1.....	55
Figura 5 - 2.....	57
Figura 5 - 3.....	59
Figura 5 - 4.....	60
Figura 5 - 5.....	61
Figura 6 - 1.....	62
Figura 6 - 2.....	63
Figura 6 - 3.....	64
Figura 6 - 4.....	65
Figura 6 - 5.....	66
Figura 6 - 6.....	66

Figura 6 - 7.....	67
Figura 6 - 8.....	67
Figura 6 - 9.....	68
Figura 6 - 10.....	69
Figura 6 - 11.....	69

# Capítulo 1

## Introdução

As organizações que querem prosperar devem mudar continuamente devido às constantes mudanças do ambiente de mercado (Donaldson, 1996). Na tentativa de melhorar o desempenho das empresas, o foco tem mudado nos últimos anos de um nível organizacional para o nível inter-organizacional (Malone and Rockart, 1991). Esse interesse crescente em relações inter-organizacionais pode ser ilustrado por diversas tendências de negócios (Bower-sox and Closs, 1996).

De acordo com den Hengst e Sol (2001), a globalização como tendência de negocio foi criada através de acordos internacionais de troca, como a União Européia, Acordo Geral de Tarifas e Trocas (GATT), e o Acordo Livre de Troca da America do Norte (NAFTA), isto resultou em um mercado mais homogêneo, onde organizações se competem.

Também, através das ultimas décadas, avanços na tecnologia de informação (TI) tiveram um impacto profundo no panorama global e na progressão da globalização. A inovação em TI tem sido um fator majoritário por trás da globalização e é, agora, um componente chave em estratégia de negócios de corporações globais (Lawlor, 2007)

A dimensão e extensão do valor de uma ferramenta de TI depende de vários fatores, incluindo o tipo de TI, práticas de gerenciamento, estrutura organizacional, a concorrência e também o macro ambiente de mercado. (Brynjolfsson, 2002; Cooper, 2000; Dewan and Kraemer 2000)

A importância de TI em logística tem crescido muito na prática de *Just in Time*, manufatura de fluxo contínuo e menor estoque, por empresas. Com esta ênfase em JIT, o impacto da logística vem crescendo juntamente, e reconhecida como fonte de consistência, menor tempo e entregas menos arriscadas (Bardi, 1994).

Segundo Laporte (1991), o Problema de Roteamento de Veículos (VRP) pode ser descrito como um problema de criação de entrega ótima ou coleções de rotas de uma ou várias localidades para outras geograficamente distribuídas. É um aspecto fundamental da Gerência de Distribuição, ele desafia constantemente milhares de empresas e organizações que trabalham com coletas e entregas de bens. Como existe uma variação de condições entre situações, os objetivos e restrições são altamente variáveis. A maioria dos algoritmos e desenvolvimento de softwares nesta área são limitadas em questão de protótipos de problemas (Banhart e Laporte, 2007).

Para a realização do estudo, é necessária a utilização de um serviço de interface de mapas, para que seja possível a extração de dados, como por exemplo conjunto de vértices em um determinado mapa, a distância entre cada um e a possibilidade de uma apresentação gráfica. Em primeira instância, foram apresentados 2 sistemas, o Open StreetMap (OpenStreetMap, 2015) e o Google Maps (Google Maps, 2015) para a possível utilização no estudo.

O Open StreetMaps (OSM), é um mapa editável e gratuito criado em 2004 conhecido como a “solução de mapas mais extensa, disponível ao público do mundo”. A sua comunidade de usuários desenvolvedores trabalha para coletar dados e adicionar e atualizar mapas que são disponíveis para qualquer pessoa sem custo algum. Os dados vem de uma variedade de fontes, tal como gravações de dispositivos GPS (Sistema de Posicionamento Global), imagens gratuitas de satélites e o conhecimento próprio de usuários. Qualquer usuário pode acessar e adicionar novas informações para um mapa e também modificar informações existentes com erros (similar ao sistema do Wikipedia). Os mapas OSM são utilizados como alternativa para empresas comerciais como a Navteq ou Tele Atlas.

Por outro lado, o Google Maps é um serviço de mapas gratuito e já bem estabelecido fornecido pela Google. O website fornece vista da rua, mapas de interiores de locais públicos (shopping centers, aeroportos, estações de trem), planejadores de rotas, localização de empresas, assim como imagens de satélites. Os seus mapas também estão disponíveis em várias plataformas móveis.

O Google Maps é uma aplicação referência no mundo todo para navegação, possui sistemas de rotas para ciclistas, pedestres, motoristas e utilizadores de sistema de transporte público local.

Já com o sistema de mapeamento definido, a próxima etapa é utilizar o sistema de roteamento, que vai coletar as coordenadas dadas pelo usuário e criar um caminho de coordenadas, gerar um arquivo para que seja impresso no mapa. Existem alguns sistemas de roteamento gratuitos disponíveis na web, como o LeafLet, Grasshopper, OSRM, OpenRoute Service.

Com as informações exibidas, o objetivo deste estudo é desenvolver um sistema com linguagem de programação Java, a API OpenStreetMaps e máquina de roteamento Open Source Routing Machine (OSRM). Tem uma abordagem simples e pode ser aplicado em uma grande variedade de problemas de roteamento de veículos. O sistema lê um arquivo texto criado pelo usuário com quantos endereços precisar, calcula a distancia entre os pontos e imprime a rota final em um mapa JxMap.

Este estudo foi dividido em 6 capítulos. O primeiro apresenta conceitos básicos e iniciais sobre o que foi desenvolvido. O capítulo 2 aborda as principais ferramentas desde trabalho, o OpenStreetMap e o Open Source Routing Machine. O terceiro capítulo conceitua outras ferramentas utilizadas. O capítulo 4 mostra o desenvolvimento e funcionamento do algoritmo. O capítulo 5 apresenta os testes e seus resultados e, finalmente, o capítulo 6 apresenta limitações, discussões finais e propostas para próximos estudos relacionados.

# Capítulo 2

## Referencial Teórico

Este capítulo tem como objetivo apresentar os conceitos básicos que serão utilizados durante este estudo bem como suas utilizações.

### 2.1 Modelagem da Superfície Terrestre

#### 2.1.1 Elipsóide de Revolução

Segundo Karney (2011), elipsoide de revolução ou esferoide é uma superfície produzida da rotação de um eixo de um elipse tendo um plano circular perpendicular com o eixo rotacionado. Elipse é um círculo com formato oval onde a distância de  $(c,0)$ ,  $(x,y) + (x,y)$ ,  $(-c,0)$  tem sempre o mesmo valor, como mostra a Figura 2-1.

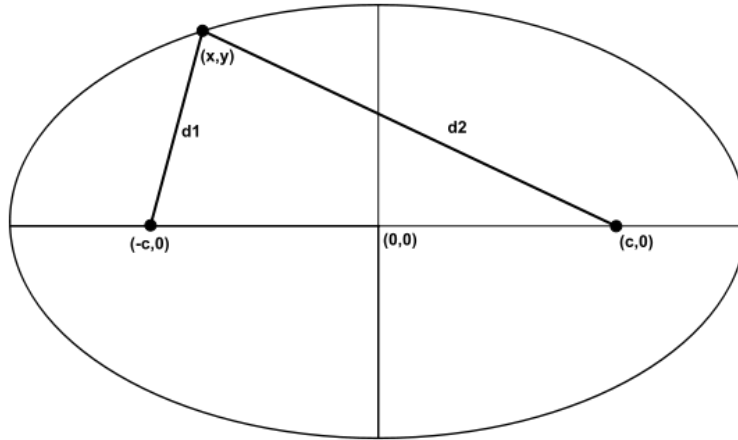


Figura 2 - 1

A Terra é um pouco achatada nos pólos e é mais larga na linha do Equador. A figura geométrica que mais se aproxima da superfície terrestre é uma elipsóide de revolução. (Karney, 2011). A Figura 2-2 exhibe as propriedades de um Elipsóide.

### O elipsóide e as suas propriedades

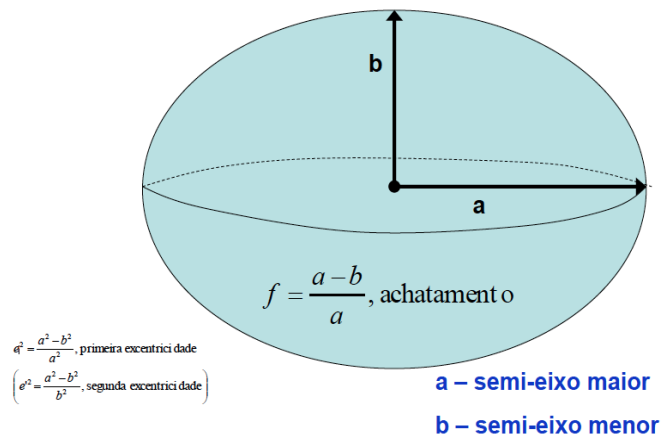


Figura 2 - 2

### 2.1.2 Datum

Datum é definido como qualquer quantidade numérica ou geométrica ou um conjunto de tais quantidades que servem como um ponto de referencia na qual se medem outras quantidades (NIMA, 1996).

Na cartografia e navegação, existem 2 tipos de datums: o datum horizontal e o datum vertical. O primeiro forma a base para a computação da posição horizontal. O segundo fornece a referencia para medir alturas ou profundidades, e pode ser de 2 tipos: Datum Geodésico Vertical, que é a referencia utilizada por pesquisadores para medir alturas de superfícies topográficas, e por cartógrafos para retratá-los. Este não se deve ser confundido com Datum Altimétrico, que são datums verticais por definição (e que não tem componentes horizontais), utilizados para definir alturas e profundidades de superfícies hidrográficas, como profundidade de aguas e apuramentos de pontes.

O Datum Geodésico Vertical vem das expressões matemáticas e já o Datum Altimétrico vem exatamente dos dados de marés (NIMA, 1996). A Figura 2-3 apresenta, na língua inglesa, os datums existentes no mundo.

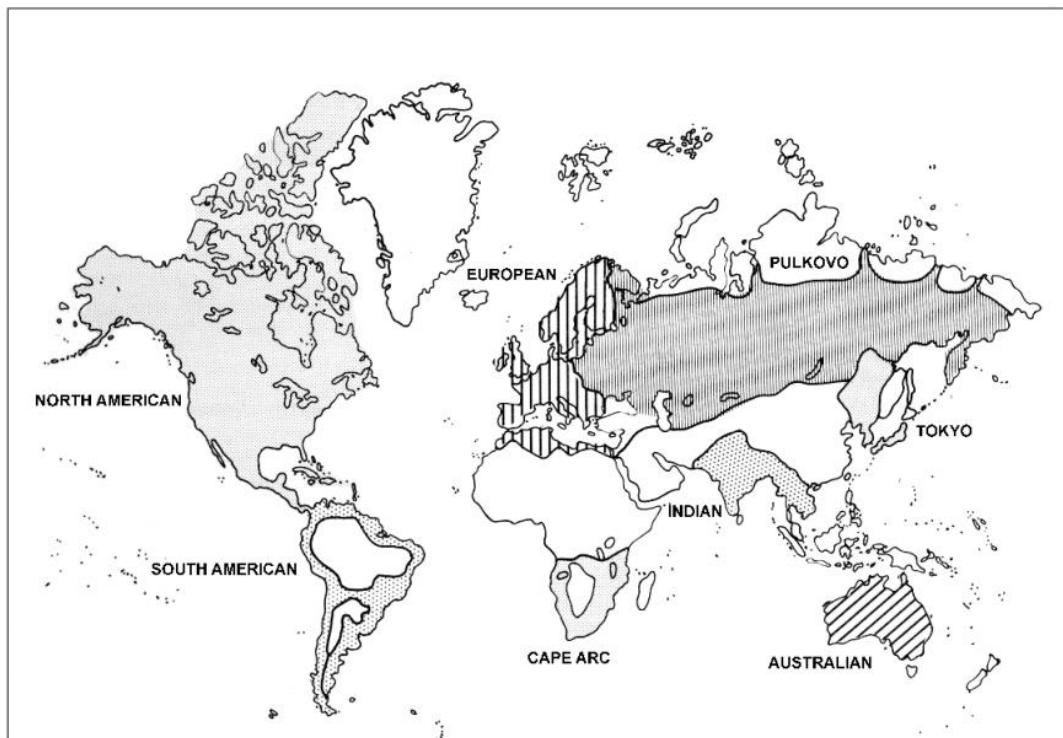


Figura 2 - 3



O Brasil possui 2 datums oficiais: O Córrego Alegre e o SAD 69. O primeiro foi definido em 1961 e pode ser utilizado em áreas terrestres da área do país, entre as 18°S e 27°30'S, também a leste de 54°W entre 15°S e 18°S. O Córrego Alegre referencia o elipsóide internacional de 1924 e o Meridiano Central de Greenwich. O seu nome vem do local da coordenada inicial que tem o nome do Datum. O segundo significa South American Datum (DAT) e a data em que foi definido, 1969, e tem extensão de alguns países da America do Sul incluindo o Brasil, o seu ponto fundamental de referencia é a cidade de Choa/MG na região do triângulo mineiro (Geo Repository, 2011).

## 2.2 Sistemas de Coordenadas Geográficas

Existem algumas maneiras de referenciar coordenadas no elipsóide de revolução, sendo as coordenadas geográficas as mais utilizadas. Para fins de referência o globo terrestre é dividido em uma série de linhas de referência: meridianos e paralelos.

Define-se como rede geográfica o conjunto de linhas imaginárias na direção leste-oeste (meridianos) e na direção norte-sul (paralelos).

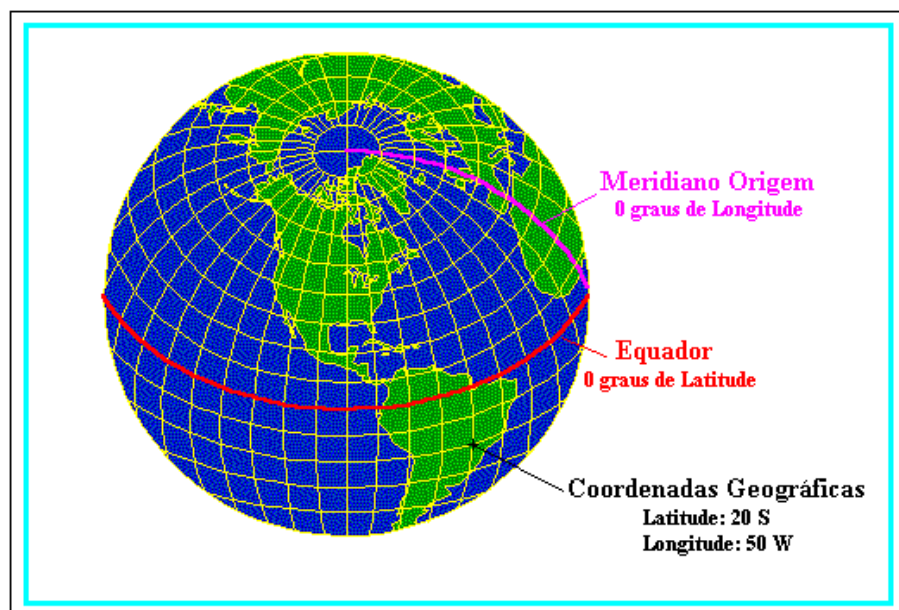


Figura 2 - 4

As coordenadas geográficas são referenciadas por dois ângulos: o latitudinal e o longitudinal. O ângulo latitudinal é em relação à linha imaginária do equador e o ângulo longitudinal é em relação à linha imaginária de Greenwich.

### 2.2.1 Longitude

A longitude pode ser definida como a distância em graus de qualquer ponto da Terra em relação ao Meridiano de Greenwich. Por convenção, o ponto  $0^\circ$  foi estabelecido no Observatório Real em Greenwich, na Inglaterra e é calculado por graus negativos (a oeste do ponto  $0^\circ$ ) indo até  $-180^\circ$ , e graus positivos (a leste do ponto  $0^\circ$ ) indo até  $180^\circ$ . A Figura 2-5 apresenta o conceito de Longitude.

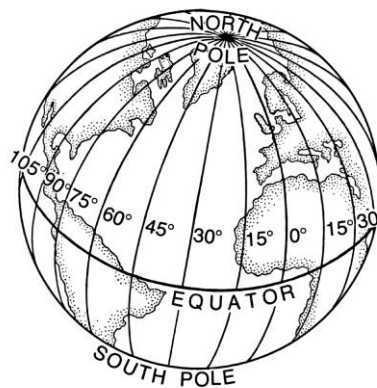


Figura 2 - 5

### 2.2.2 Latitude

Latitude é uma coordenada geográfica que especifica a posição norte-sul de um ponto na superfície da Terra. Latitude é também um ângulo que varia de  $0^\circ$ , do Ponto Médio do Equador até  $+90^\circ$  e  $-90^\circ$ , pólos norte e sul respectivamente. Linhas de latitudes constantes, ou paralelas, marcam círculos horizontais (leste-oeste) paralelos ao ponto equador. A Latitude é utilizada junto com a longitude, com algumas especificações definidas para altitudes, para especificar o

local preciso de recursos da superfície da Terra, constituindo o Sistema de Coordenadas Geográficas, como definido na especificação do ISO 19111 (OGC, 1998).

A atual documentação do ISO 19111 pode ser comprada no site <http://www.iso.org>, mas existem alguns rascunhos disponíveis sem custo em vários locais da web, por exemplo o seguinte no CSIRO.

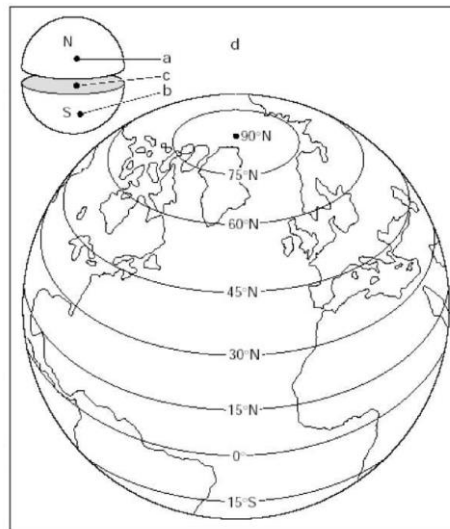


Figura 2 - 6

### 2.2.3 Dados Vetoriais

Os dados vetoriais representam e definem fronteiras de características geográficas, que tem normalmente 3 formas: pontos, linhas e polígonos

Um ponto é definido como um par ordenado  $xy$ , onde  $x$  é a latitude e  $y$  longitude, representando uma localização ou evento no mapa. Uma linha é um segmento de dois ou mais pontos no mapa. Área é um conjunto de pontos  $xy$ , conseqüentemente um conjunto de linhas que se fecham em um espaço fechado (Geobank, 2013).

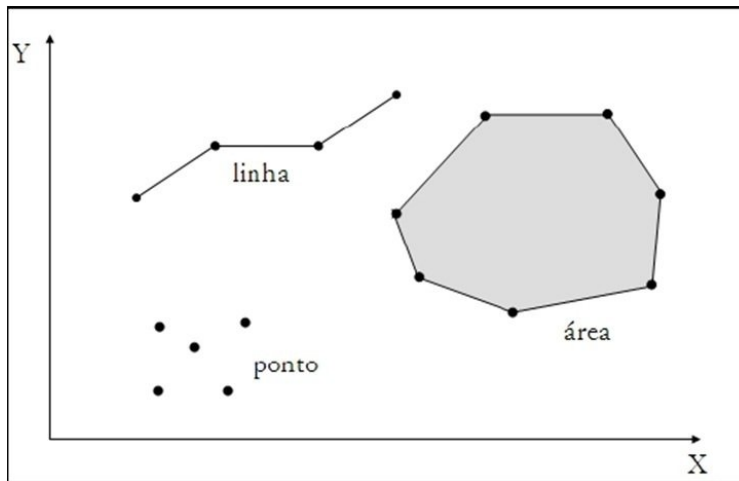


Figura 2 - 7

#### 2.2.4 Dados Geográficos

Segundo GeoRepository (2011), dados geográficos são utilizados para armazenar informações gráficas espaciais, estas são utilizadas para representar a localização de dados vetoriais no mapa. Eles permitem a integração de coordenadas em um espaço comum, como mapas, gps, etc.

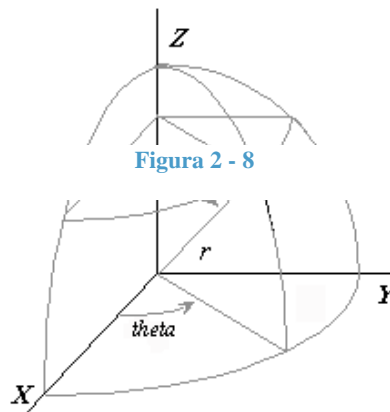
Cada sistema de coordenadas é definido por um framework de medição, no qual pode ser geográfico ou planimétricas e unidades de medida, tipicamente em metros para sistemas de coordenadas projetados ou em graus para latitude/longitude.

O tipo mais comum de sistema de coordenadas é a global ou esférica, mais conhecida como Coordenadas Geográficas, elas utilizam latitude e longitude e são medidas em graus. (Geo Repository, 2011).

## 2.3 Sistemas de Coordenadas Esféricas

Coordenadas esféricas, também conhecidas como coordenadas esféricas polares (Walton, 1967, Afken, 1985), são sistemas de coordenadas curvilíneas que descrevem naturalmente posições em uma esfera. O ângulo azimutal no plano do eixo  $X$  (denota-se para longitude), e o ângulo polar (também conhecido como ângulo zenital ou co-latidade, sendo  $\theta$ , onde  $\phi$  é a latitude) do eixo positivo  $Z$  sendo  $Y$ , e  $r$  é a distância (raio) de um ponto da origem. Todos estas definições são convenções matemáticas.

## 2.4 Sistemas de Coordenadas Planimétricas



A projeção de Mercator (UTM) utilizado em muitos mapas mundiais é uma projeção cilíndrica, o que significa que o globo é circulado por um cilindro imaginário que toca na Linha do Equador e o planeta terra é projetado dentro deste cilindro. (Dutch e Steven, 2000).

Por ser uma criação inspirada por razões militares, há uma grande rejeição de sua utilização por cientistas.

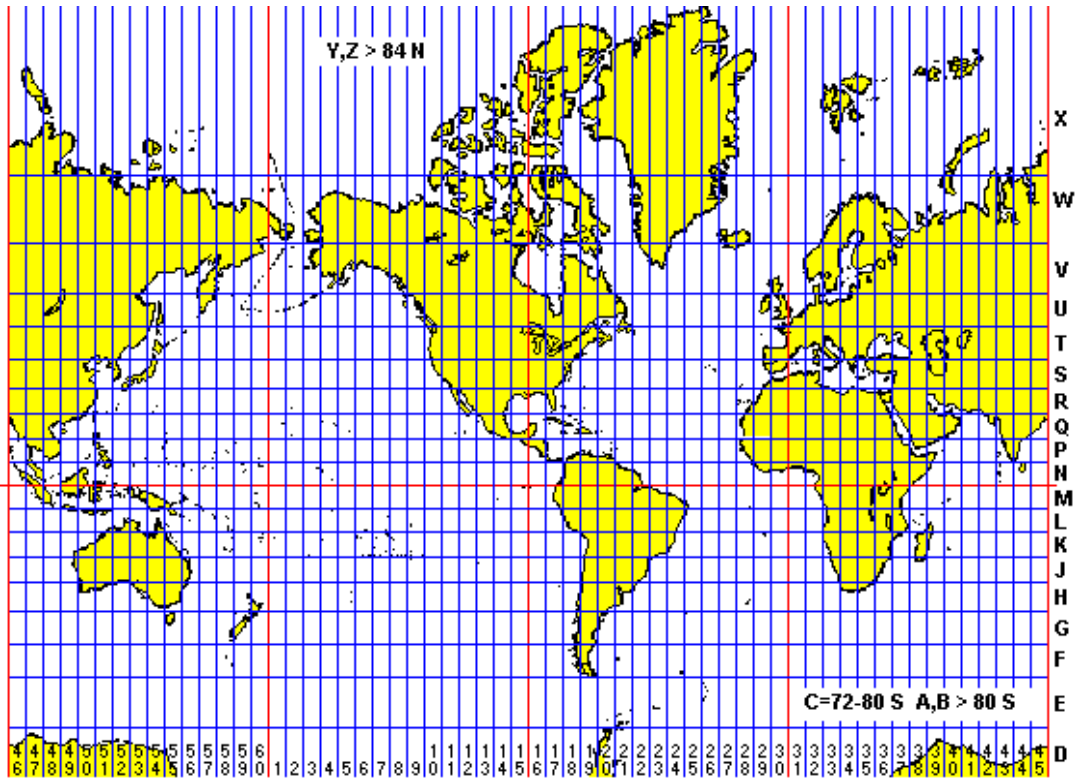


Figura 2 - 9

O mapa acima mostra como o sistema NATO UTM divide a Terra em 60 zonas de longitudes, com 6 graus por zona. Longitudes à Oeste são negativas e à Leste são positivas, O sistema numérico começa na Zona 1 com 180° oeste e caminha para leste. Para encontrar a zona de alguma longitude, adicione 180° (converte a longitude em um numero entre 0-360°), divida por 6 e arredonde para o maior.

Por exemplo, Rio de Janeiro - RJ tem longitude 43° oeste, ou -43°, ao adicionar 180° se tem 137° (Rio de Janeiro está a 137° da Linha Internacional de Data).  $137/6 = 22.83$ , em que se deve ser arredondado, então o resultado é 23. Então o Rio de Janeiro é considerado Zona 23.

## 2.5 Sistema de Posicionamento Global

O Sistema de Posicionamento Global (GPS) é um sistema de navegação baseado em uma rede de 24 satélites em órbita terrestre criado pelo Departamento de Defesa Americano para fins

militares. O sistema é utilizado para identificar coordenadas geográficas em qualquer lugar do mundo independente das condições climáticas.

O primeiro satélite foi lançado em 1978. Em 1980 o Governo Americano decidiu disponibilizar a aplicação para uso civil, o que tornou um mercado interessante, já que seria um serviço sem custo algum.

Receptores de GPS recebem o sinal dos satélites e calculam a posição exata através de triangulação, uma vez que o GPS recebe a localização 2D (longitude, latitude) do usuário, se torna possível o cálculo de outras informações, tais como localização 3D (longitude, latitude, altitude), velocidade, por do sol, nascer do sol, distância de viagem, tempo, entre outras. Hoje em dia, dispositivos GPS representam uma tecnologia fascinante e barata de escanear o planeta. (Beiglbock, 2012).

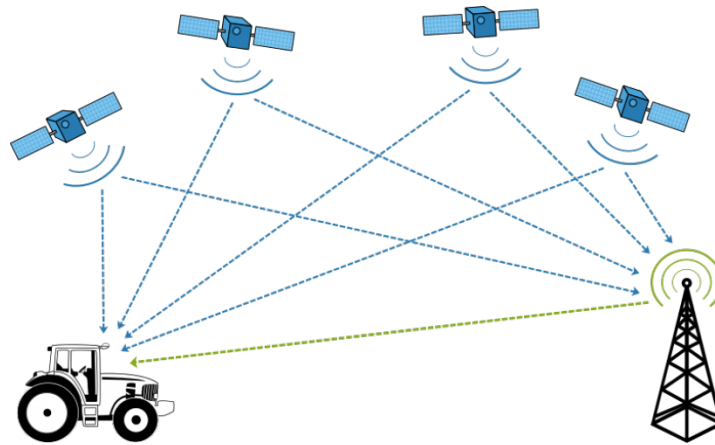


Figura 2 - 10

### 2.5.1 Mapas

Mapa é uma representação visual de uma determinada área em um local plano. O objetivo de um mapa é ilustrar uma área bem detalhada e específica. Existem vários tipos, estáticos, bidimensionais, tridimensionais, dinâmicos e até interativos e podem representar diferentes tipos de informações: política, demográfica, topográfica, econômica, ambiental, rodovias e climáticas (National Geographic, 2008).

Mapas políticos mostram informações geográficas gerais sobre uma determinada área, incluindo informações sobre cidades, limites, estradas, montanhas, rios e costas marítimas. Algumas agências de governo por exemplo a Centro de Pesquisa Geológica dos EUA (USGS) fazem alguns mapas de representação geral. Muitos são mapas topográficos, que mostram mudanças de elevação, eles mostram todas as montanhas e vales existentes em uma área. Isto pode ser muito eficiente para pesquisadores procurando escolher uma rota para engenheiros criarem rodovias ou represas.

Criadores de mapa, ou cartógrafos, tem uma grande desafio na hora de criar a projeção do mapa, que é transferir informações da superficial esférica e colocar no plano em papel. Todo mapa tem algum tipo de distorção. Quanto maior a área, maior a distorção. Características como



tamanho, forma, distância ou escala podem ser medidas com grande precisão no globo, mas quando projetadas em uma superfície plana, muitas dessas podem não ser tao bem representadas.

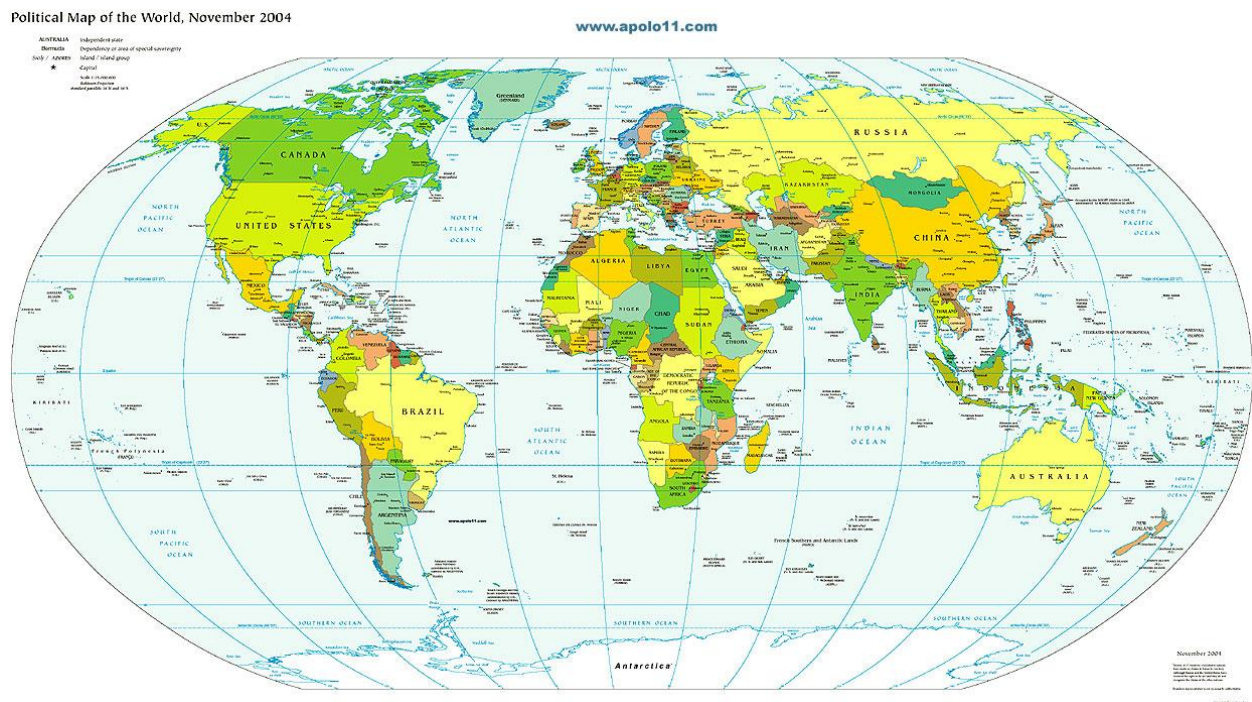


Figura 2 - 11

## 2.5.2 Mapas Digitais

Mapa Digital é a representação de mapa convencional baseado em dados armazenados em um computador para fácil acesso e manutenção. As informações destes mapas digitais são altamente confiáveis devido a quantidade de dados adquiridos por satélites desde os anos 80.

Com o surgimento e expansão de dados em tempo real nos últimos anos, mapas digitais tem ganhado cada vez mais importância e se tornou uma ferramenta essencial para organização de dados espaciais.

Com a utilização de mapas digitais, a produtividade de profissionais da área aumentou muito, fatores como: exatidão de posicionamento, velocidade do mapeamento, padronização assistida, maior integração entre mapas diferentes, melhoria no mapeamento 3d e sua interpretação e a produção de mapas sem papéis (Swanson-Hysell, Nicholas, 2014).

### 2.5.3 Distancia

Distância é uma linha que mostra o quão distantes dois objetos estão. É representado por uma linha reta entre A e B.

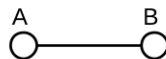


Figura 2 - 12

Em cartografia, a distância entre dois pontos não é necessariamente uma distância em linha reta, é seguido o caminho das estradas, que muito dificilmente são em linha reta. Assim a representação de distância pode ser definida pelo valor numérico entre dois pontos, em linha reta ou não.



Figura 2 - 13

### 2.5.4 Grafos

Grafos são estruturas matemáticas utilizadas para modelar relações entre objetos, são constituídos de vértices ou nos e arestas, que são linhas que conectam os vértices.

Um grafo é um par ordenado  $G=(V,A)$ , incluindo um conjunto de vértices combinado com um conjunto de arestas, que é um subconjunto de 2 elementos de  $V$  (uma aresta se relaciona com dois vértices, e esta relação é representada como um par não ordenado de vértices com a aresta dada).

### 2.5.5 Vértices

Vértices ou nós são a unidade principal para que os grafos sejam formados, um grafo não direcional consiste em um conjunto de vértices e um conjunto de arestas. Em um grafo, um vértice geralmente é representado por um círculo com uma letra número, e uma aresta é representado por uma linha ligando dois vértices (Frederik, Topf e Chilton, 2010).

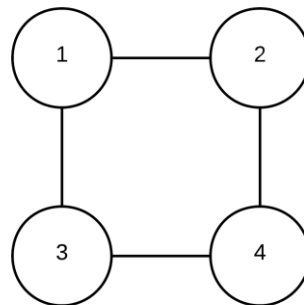


Figura 2 - 14

### 2.5.6 Vertices no OSM

Dentro do espaço de programação do OSM, o vértice é chamado de Nó e representa muito mais que apenas um ponto no mapa. Segundo Frederik, Topf e Chilton (2010) um nó armazena as seguintes informações:

- Data e horário da última modificação,

- Nome do usuário que último editou,
- Número da versão do conjunto de alteração (changeset) com a mudança que foi efetuada
- Nenhuma ou várias tags (etiquetas)

## **2.6 Tipos de Dados de GPS**

Dispositivos de GPS são operados por 3 tipos diferentes de dados;

### **2.6.1 Ponto de Caminho (Trackpoint)**

Caminhos (tracks) são registros de movimento do dispositivo, basicamente consiste em pontos de caminhos (trackpoints), que consistem em pontos de caminho. Um ponto de caminho contém valores de latitude e longitude, e dependendo do tipo do dispositivo, ele pode ter mais valores, como por exemplo velocidade, altitude, etc. Pontos de caminho são normalmente salvos a pequenos intervalos de segundos, mas em alguns dispositivos eles podem ser salvos apenas quando a rota é mudada ou quando o dispositivo vai além do destino. Quando o dispositivo perde sinal ou é desligado, ele irá começar um novo segmento de caminho (Topf e Chilton, 2011).

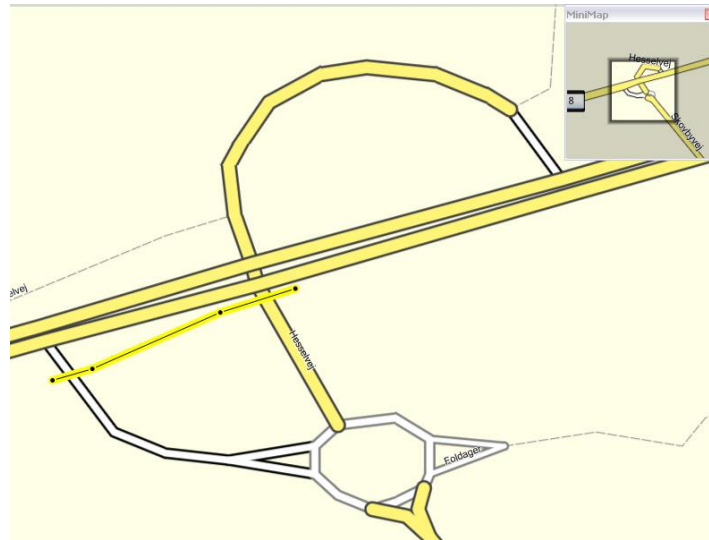


Figura 2 - 15

### 2.6.2 Ponto de Interesse (Waypoints)

Segundo Topografix (2014), pontos de interesse (waypoints) são definidos manualmente pelo usuário, como por exemplo quando colocamos um pino em determinado lugar do mapa. É bem parecido com um segmento de caminho, ele também possui valores latitude e longitude, e talvez velocidade e altitude, mas a diferença é que ele possui um ícone para representá-los e talvez até conteúdo de multimídia.



Figura 2 - 16

### 2.6.3 Rota

A rota é um conjunto de pontos que deve ser visitado em sequência para chegar até um destino (Topografix, 2014).

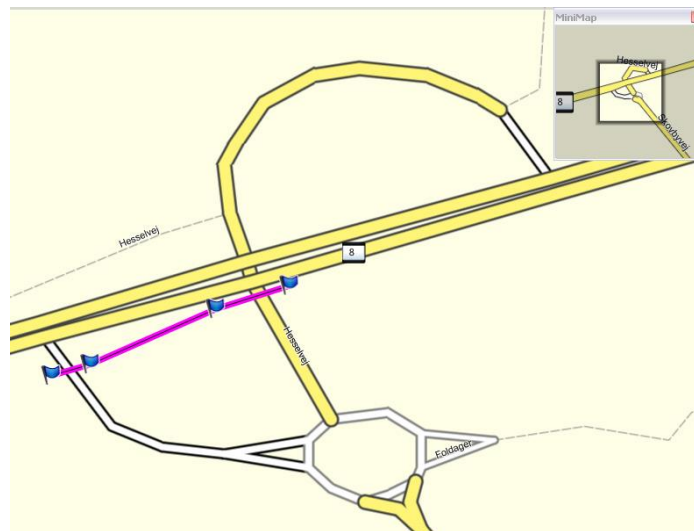


Figura 2 - 17

## 2.7 Formato de dados de GPS

### 2.7.1 XML

Segundo Medina (2014), Extensible Markup Language (XML) é um padrão de formatação de dados. Os documentos XML podem ser facilmente compreendidos e facilita o processo de desenvolvimento de aplicativos.

Ele foi criado pelo World Wide Consortium (WC3, 2001) como uma maneira de contornar as limitações impostas pelo HTML (Connolly, 2003).

Os bancos de dados relacionais não conseguem processar dados independentes do seu conteúdo, assim as demandas de negócios eletrônicos não estavam sendo supridas. Bancos de dados convencionais não conseguem lidar com audio, video ou dados mais complexos, já o XML suporta todos estes tipos de dados sem problemas. Existem várias outras razões de este ser o mais utilizado por programadores, por exemplo alto nível de integração de formatos e bancos de dados, grande internacionalização e escalabilidade, é um formato aberto e é conhecido por estar sempre orientado para o futuro (WC3, 2001).

```
<?xml version="1.0" encoding='UTF-8'?>
-<kml xmlns="https://earth.google.com/kml/2.0">
  -<Response>
    <name>avenida ipiranga1200portoalegrersbrasil</name>
  -<Status>
    <code>200</code>
    <request>geocode</request>
  </Status>
</kml>
```

Figura 2 - 18

## 2.7.2 KML

O Keyhole Markup Language (KML) é uma gramática especializada da linguagem de marcação XML utilizada para codificar representações de dados geográficos para exibição em navegadores de mapas. Ou seja, o formato KML diz ao navegador o que e como exibir os dados. Assim como o XML, o KML utiliza uma estrutura baseada em tags com elementos aninhados e propriedades. Entre os exemplos de dados geógrafos que podem ser representados utilizando a KML estão pontos, linhas e polígonos (de Jesus Junior and Marozzi, 2014).

O arquivo KML especifica uma série de informações (marcações de lugares, imagens, polígonos, modelos 3D, descrição de texto, etc.) para ser exibido em aplicações geospaciais que necessitam deste tipo de linguagem. Todo local sempre tem representação de longitude e

latitude. Existem opções de tornar a visualização KML mais específica com os seguintes códigos: tilt, heading, altitude, que, juntos definem uma visualização de câmera.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <description>Attached to the ground. Intelligently places itself
      at the height of the underlying terrain.</description>
    <Point>
      <coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

Figura 2 - 19

### 2.7.3 GPX

O formato GPX (GPS Exchange File) é um formato padrão XML para troca de dados GPS entre diversos softwares e aplicações de Internet.

Normalmente utilizado junto com XML, o formato GPX pode acomodar extensões de uso especial, por exemplo para armazenar hyperlinks ou referências para registros de som. (Frederik, Topf e Chilton, 2010).



```

<gpx version="1.1" creator="GPSBabel - http://www.gpsbabel.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.topografix.com/GPX/1/1"
  xsi:schemaLocation="http://www.topografix.com/GPX/1/1
    http://www.topografix.com/GPX/1/1/gpx.xsd">
  <wpt lat="49.785581670637695" long="9.26481034667384">
    <time>2007-10-01T11:46:27Z</time>
    <name>001</name>
  </wpt>
  <wpt lat="49.786560191735184" lon="9.265752519584733">
    <time>2007-10-01T11:48:47Z</time>
    <name>002</name>
  </wpt>
</gpx>

```

Figura 2 - 20

De acordo com da Costa (2011) o formato GPX é composto por três tipos de elementos:

- Caminhos;
- Rotas;
- Pontos de interesse.

Pode-se dizer que caminhos e rotas são composto por pontos de interesse e eles possuem informações obrigatórias como latitude e longitude, apesar de conter também informações adicionais como a altitude e o tempo.

## 2.7.4 TCX

Training Center XML (TCX) é um formato de troca de dados, criado em 2007 como parte do produto Garmin's Training Center. O XML é parecido com o GPX já que ele troca trilhas do GPS, mas ele trata as trilhas como uma atividade e não apenas como uma serie de pontos do GPS. O TCX permite em seu padrão transmitir batimentos cardíacos, ritmo da caminhada, ritmo na bicicleta e calorias na trilha detalhada. Ele também fornece os dados resumidos na forma de voltas (Garmin, 2007).

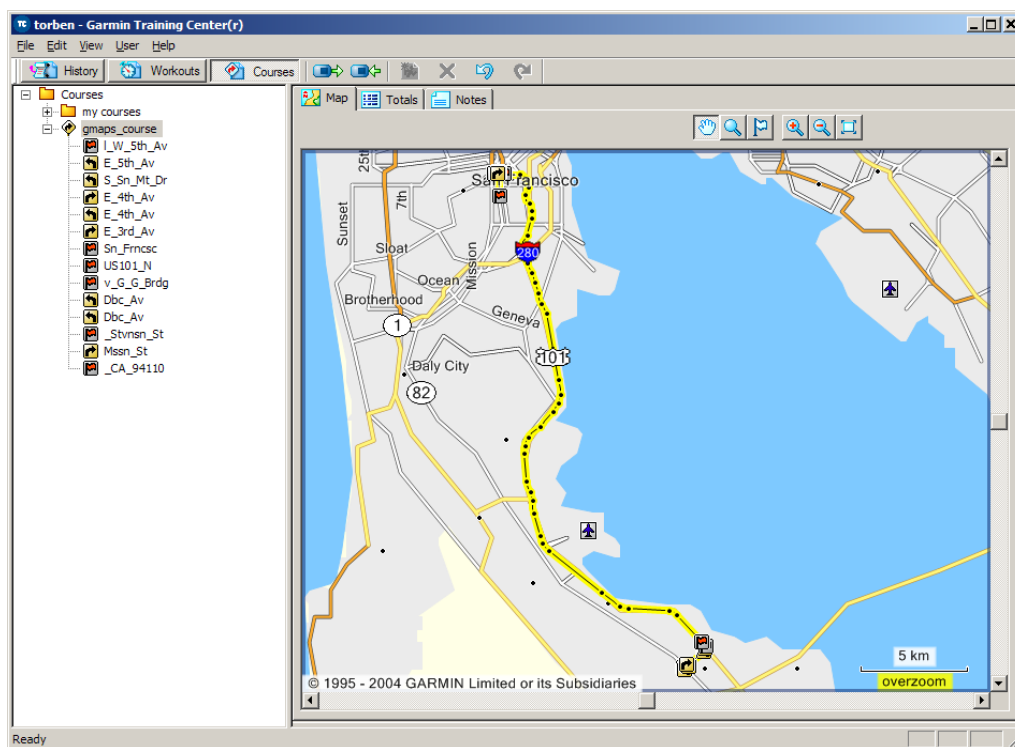


Figura 2 - 21

## 2.7.5 Tipos de Objetos Básicos

Os dois tipos de objetos no OpenStreetMap são os nós (também chamados de pontos, ou vértices) e caminhos. Os atributos que são colocados para descrever o que representam são chamados de tags. Um tag contém uma chave e um valor que normalmente são escritos com um

símbolo de igual entre eles. Além disso, existe um tipo de dados de relação que pode ser utilizado para modelar relações entre objetos. A combinação destes objetos permite a edição de dados do mundo, OpenStreetMap. (Topf e Chilton, 2011).

### **2.7.6 Nós**

Um nó consiste em: Coordenadas geográficas (latitude e longitude, tempo da ultima edição (timestamp), ID e nome de usuário do ultimo editor, numero de versão e o numero do changeset no qual a mudança foi feita. Zero ou mais tags (Topf e Chilton, 2011).

Os nós podem ser conectados para formar caminhos, e então servem de vertices na geometria linear, uma estrada por exemplo, e também para polígonos, praças, rotatórias, lagos.

Outra aplicação muito funcional são os pontos de interesse, os nós mapeam estes pontos, por exemplo, postos de gasolina, hotéis, cinemas, etc...

### **2.7.7 Caminhos**

Caminho é uma lista ordenada de no mínimo dois nos, ele também possui as informações de ultima edição, usuário que editou por ultimo.

Os caminhos sempre vão mapear objetos lineares, como estradas, rios ou linha de metrô, todos os caminhos tem uma direção, mas nem sempre é relevante, já que estradas podem ser mão única ou duas mãos, e, se o primeiro nó é exatamente igual ao último, então pode ser considerada uma area, como por exemplo areas de habitação indígena, lagos.

### **2.7.8 Relações**

Segundo Lopf e Chilton, 2011, relação é a adição mais recente do OpenStreetMaps, é utilizado para fazer conexões entre os diferentes objetos no banco de dados. O servidor não sabe o significado destas conexões, mas ele provê integridade referencial, ou seja, nunca haverá links quebrados entre os objetos.

Os membros de uma relação pode ser nós, caminhos ou outras relações (Topf e Chilton, 2011). A função de cada membro é registrada por um conjunto de strings livres, ou seja, não necessariamente precisa haver uma descrição se não faz sentido ter uma. O mesmo objeto pode ser membro de uma relação mais de uma vez, se pensarmos em uma rota onde há mão e contra mão e o carro tem que passar mais de uma vez para seguir rotas diferentes.

### **2.7.9 Tags**

Segundo Topf e Chilton, 2001, Objetos Básicos podem ter uma quantidade qualquer de tags, uma tag possui uma chave e um valor, no qual ambos podem ser qualquer string de até 255 caracteres. Chaves não podem ser nulas e valores podem ser nulos mas raramente faz sentido e por isso softwares de edição tratam como erro.

Há uma convenção que diz que todo e qualquer usuário é livre para criar suas próprias tags, mas é recomendado que se use as existentes e criar novas somente se não são cobertas por já existentes (Codescu, 2011).

```
<node id="834034642"
lat="53.0871310" lon="8.8091071"
version="7" changeset="6027662"
user="Kerridge" uid="324245"
timestamp="2010-10-13T09:51:39Z">
<tag k="addr:city" v="Bremen" />
<tag k="addr:country" v="DE" />
<tag k="addr:housenumber" v="20" />
<tag k="addr:postcode" v="28215" />
<tag k="addr:street" v="Theodor-Heuss-Allee" />
<tag k="amenity" v="charging_station" />
<tag k="name" v="Elektrotankstelle swb" />
<tag k="note" v="telephone reservation necessary" />
<tag k="opening_hours" v="Mo-Fr 6:00-18:00; Sa off; Su off" />
<tag k="operator" v="swb" />
<tag k="phone" v="+49 421 3593186" />
</node>
```

Figura 2 - 22

### 2.7.10 Changeset

Segundo Smith e Mooney (2012), changeset é um conceito muito importante em OSM, é um grupo de edições feitas por um usuário em um período de tempo. Edições podem ser adicionadas em um changeset desde que ele esteja aberto. Um changeset pode ser fechado explicitamente, ou fechado se não existir nenhuma edição por um período de inatividade. O OpenStreetMap Watch List (OWL) por Amos (2011) mostra mudanças comparando changesets em uma determinada área.

Changeset são desenhados para ajudar pessoas a manter um relatório de quem realizou edições, onde e quando (Topf e Chilton, 2011).

**Changeset: 20000000** ✕

Made Horace Road one-way.

Closed

- Created: Tue, 14 Jan 2014 20:39:05 +0000
- Closed: Tue, 14 Jan 2014 20:39:06 +0000

by [cosmicpop](#)

Tags

<a href="#">created_by</a>	id 1.3.5
<a href="#">imagery_used</a>	Bing

Ways (1)

- [Horace Road \(14960774, v6\)](#)

---

[cosmicpop · 20000034 >>](#)

[Changeset XML](#) · [osmChange XML](#)

Figura 2 - 23

### 2.7.11 NMEA

A Associação de Eletrônicos da Marinha Nacional Americana (NMEA), uma associação comercial para a venda de eletrônicos para a Marinha Americana (Marines) tem a publicação de um padrão (NMEA 0183) para a comunicação entre dispositivos de navegação com outros dispositivos navais. Se o dispositivo GPS for colocado em modo NMEA, ele começará a enviar um fluxo contínuo de dados, que por exemplo podem se referir à localização atual, e de correções em tempo real. Nem todos os GPS aceitam dados tipos NMEA (Lopf e Chilton, 2011).

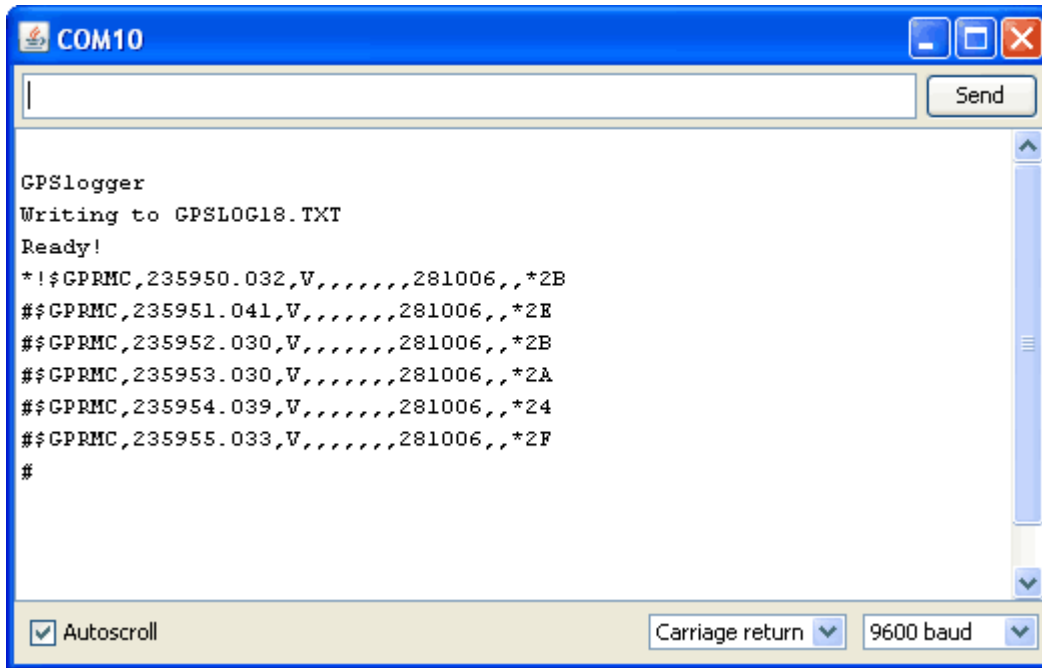


Figura 2 - 24

# Capítulo 3

## OpenStreetMap/Open Source Routing Machine

### 3.1 OpenStreetMap

Segundo OpenStreetMap (OSM), o projeto começou na Inglaterra em 2004 por Steve Coast com o objetivo de criar um mapa mundial livre, o termo livre faz referência ao movimento do Software Livre, que não necessariamente quer dizer livre de custos, mas sim livre de restrições em relação a utilização produtiva dos dados.

O modo com que o OpenStreetMap pretende criar esse mapa mundial é um pouco diferente dos métodos convencionais, pessoas que compartilham o mesmo interesse gravam os dados de GPS através de seus próprios equipamentos, dispositivos GPS e bicicletas, caminhadas e carros. Esses registros são cuidadosamente redesenhados nos computadores e então adicionadas informações adicionais, como por exemplo nomes de ruas, e então colocadas no banco de dados central.

O OpenStreetMaps já pode ser considerado o mapa mais bem sucedido. Mundialmente, inúmeras cidades já são mapeadas em um nível de detalhe que sequer pode ser alcançado por Google, Yahoo ou Microsoft (Ramm, Topf and Chilton, 2010). E ainda mais, os dados do OpenStreetMap são geralmente mais atuais do que qualquer mapa impresso ou oferecido na internet.

#### 3.1.1 Benefícios

O OSM é mantido por uma comunidade, usuários coletam dados de locais e então a informação é adicionada ao banco de dados central onde pode ser modificada, corrigida e melhorada por qualquer um que notar erros sobre a área.



Membros locais das comunidades podem melhorar funcionalidades existentes dos mapas de suas áreas adicionando informações detalhadas e realizando mudanças. Isso aumenta as chances de o mapa ser mais útil do que antigos produzidos por empresas comerciais (OpenStreetMap, 2015).

Os mapas podem ser editados para “rápida adaptação à mudanças físicas”. Por exemplo, após o terremoto no Haiti em 2010, times de voluntários de todo o mundo ajudaram na atualização do mapa devido à devastação física ocorrida. Esses mapas mais exatos das áreas afetadas ajudaram voluntários que trabalhavam na equipe local.

Os usuários mais dedicados trabalham em uma variedade de aplicações de mapeamento e melhorias são feitas diariamente. Não existe nenhuma restrição legal ou técnica. Mapas podem ser totalmente personalizados para atender as necessidades de usuários específicos.

Usuários precisam ter um bom conhecimento técnico para conseguir tirar o máximo de proveito do OSM e para editar mapas.

OSM é um trabalho que está sempre em processo, então pode ser que o mapa que o usuário procure não esteja detalhado ou então disponível.

## **3.2 Google Maps**

O Google Maps é um serviço de mapas gratuito e já bem estabelecido fornecido pela Google. O website fornece vista da rua, mapas de interiores de locais públicos (shopping centers, aeroportos, estações de trem), planejadores de rotas, localização de empresas, assim como imagens de satélites. Os seus mapas também estão disponíveis em várias plataformas móveis.

O Google Maps é uma aplicação referência no mundo todo para navegação, possui sistemas de rotas para ciclistas, pedestres, motoristas e utilizadores de sistema de transporte público local.

A utilização do Google Maps possibilitaria uma maior exatidão na coleta dos vértices e das arestas, já que é uma aplicação com grande estrutura.

O Google Maps pode ser mais detalhado em algumas áreas, é simples e fácil de usar, oferece 3 tipos diferentes de visualização de mapas (normal, satélite e terreno). Existe uma variedade de aplicativos interessantes com a utilização do Google Maps, como aplicativos para localizar pontos de internet wireless gratuita, previsão de em áreas marcadas no mapa (Google Maps, 2015).

É necessário uma conta Google para criar seu mapa ou adicionar ao seu site, é necessário a aquisição da Chave API para utilizar os serviços Google Maps. É fornecida gratuitamente e não é cobrado nenhum valor para acessar. API é uma aplicação de interface de programação que possui um conjunto de regras e especificações que o programa tem que seguir para acessar e utilizar os serviços e recursos fornecidos por um outro software que implementa este API (Google Developers, 2015).

Existem restrições de direitos autorais e usuários não podem modificar áreas do mapa que estiverem incorretas. Relatos de erros ou bugs devem ser enviados à Google.

### **3.3 Ferramentas de Comparação de Aplicativos**

Existem ferramentas na internet que possibilitam a comparação de qual o melhor mapa para o tipo de projeto que o usuário irá utilizar.

# Monitoring Tools: GeoFabrik Map Compare

25

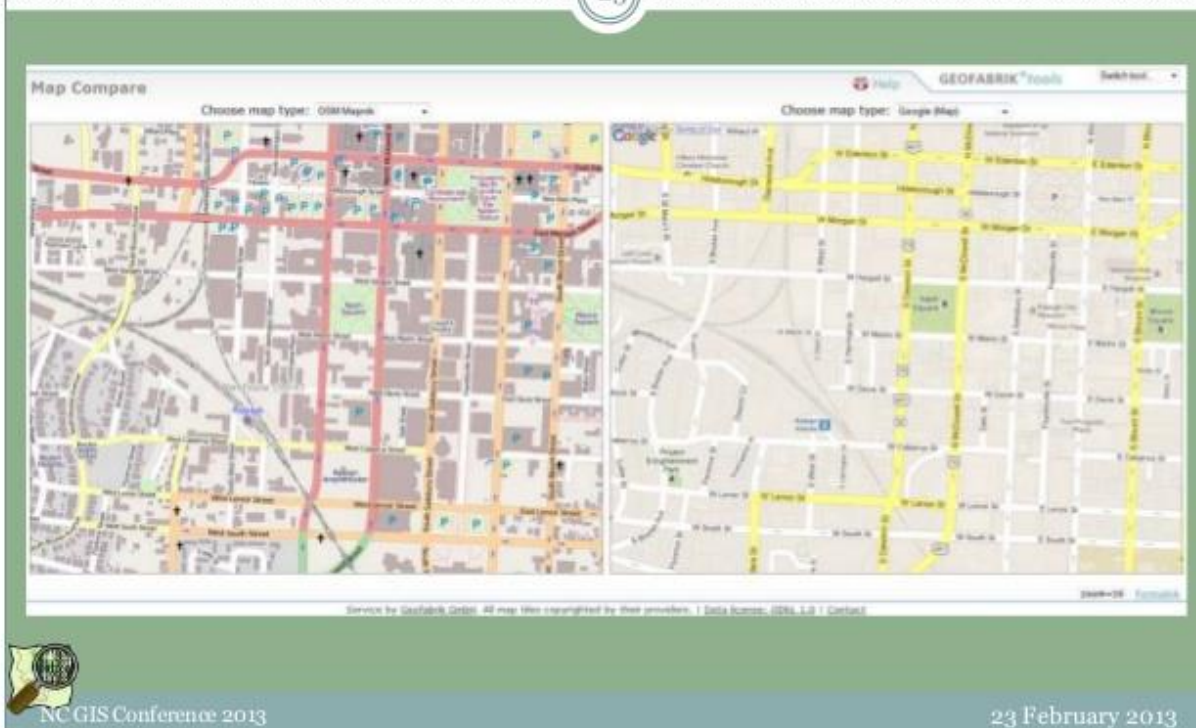


Figura 3 - 1

O Geofabrik compara diferentes camadas OSM com diferentes tipos de mapas da Google maps. Os dois mapas são sincronizados, então quando você aumentar (zoom) a tela, a mesma ação ocorrerá no outro mapa.

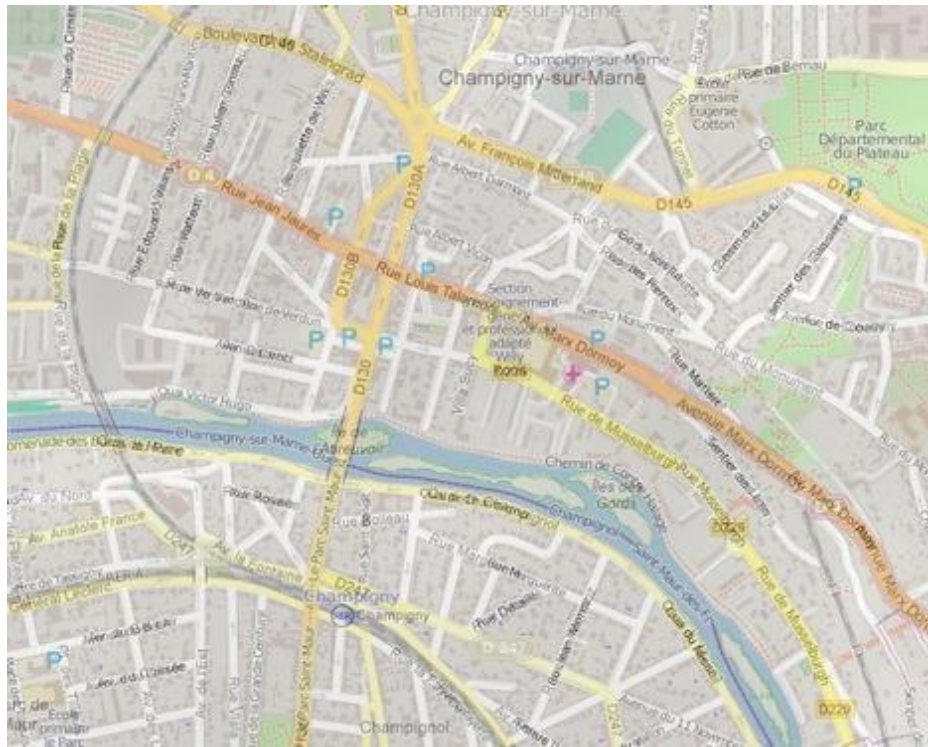


Figura 3 - 2

O Sautter coloca o OSM por cima do Google Maps para comparação. É possível a utilização da transparência para ajudar o quanto do OSM você quer ver em relação ao Google maps.

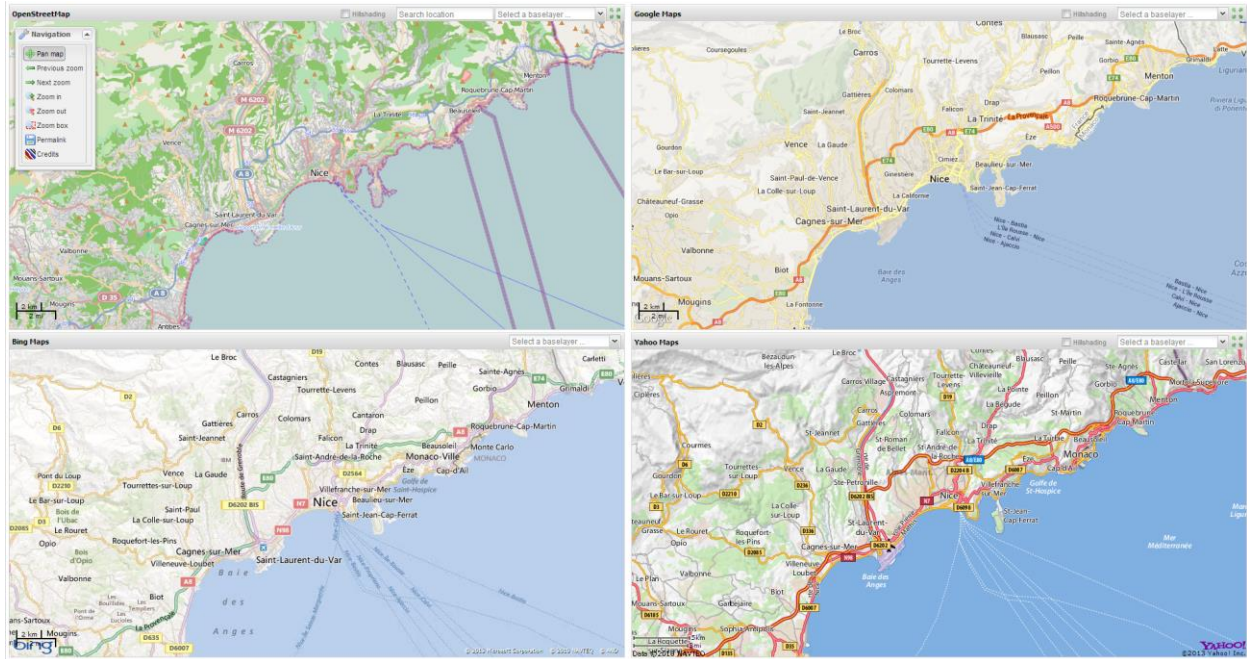


Figura 3 - 3

O Geotribu compara o OSM, Google maps, Yahoo maps e Bing Maps todos ao mesmo tempo.

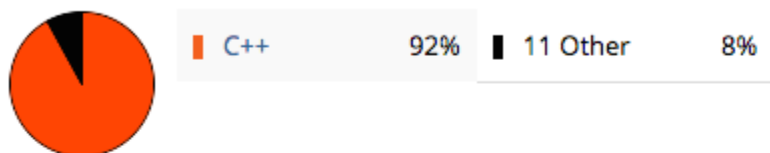
### 3.4 Open Source Routing Machine

O Open Source Routing Machine (OSRM), é uma ferramenta implementada em C++ que fornece algoritmos de menor rotas para redes de rodovias, estradas. Ele combina algoritmos sofisticados de roteamento juntamente com a rede de dados livres OpenStreepMap (Luxen e Vetter, 2011).

O projeto OSRM, como é chamado, começou a ser estruturado em Março de 2010, e, apenas 4 meses já teria sua versão v0.1 lançada. É importante ressaltar que é um trabalho em constante e crescente desenvolvimento.

Através do website OpenHub® (), é possível observar em tempo real a quantidade de linhas de código atualmente escritas no projeto, número de colaboradores, as linguagens utilizadas, colaboradores mais recentes, etc.

### Languages



### Lines of Code

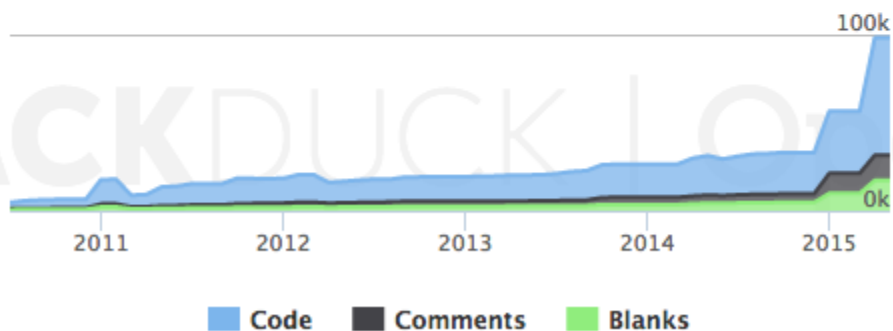


Figura 3 - 4

O projeto tem características como roteamento de alto desempenho, consegue trabalhar com redes de grandes tamanhos (estrutura de dados de continentes inteiros), sistema de importação de dados para o OpenStreetMap e foi idealizado pela comunidade acadêmica.

Existem varias opções de requisições de OSRM, ponto de inicio (latitude, longitude), ponto final (latitude, longitude), tipo de rota (carro, andando, bicicleta) e formato de saída (gpx, json).

Existem outras ferramentas na web que fornece serviços semelhantes ao OSRM, como o CloudMade, MapQuest e Google Directions.

### 3.5 CloudMade

O CloudMade é uma ferramenta construída para desenvolvedores criarem aplicações baseadas em geografia, mapas. O serviço utiliza em grande parte os dados do OpenStreetMap para fornecer seus serviços. Foi fundado em 2007 pelo criador do OSM, Steve Coast e um grande contribuinte da plataforma, Nick Black. Em janeiro de 2011, quando a companhia foi adquirida pelo grupo CloudMade Deutschland GmbH, não se sabe exatamente o que ocorreu, mas quase todos os seus serviços ficaram fora do ar e até hoje não há uma explicação para tal ocorrido.



Figura 3 - 5

### 3.6 MapQuest

O MapQuest foi criado na década de 1960 com o propósito de criar mapas de rodovias gratuitas para distribuição em postos de gasolina, em Chicago, EUA (Nielsen, 2008). Com o crescimento através dos anos e com grande utilização da internet, a empresa comanda varias aplicações web e varias plataformas de serviços geo-espaciais. Por trás de todas essa tecnologia, a MapQuest Technologies® é uma empresa que pratica desenvolvimento Ágil e Scrum, utilizando varias linguagens de programação, como JavaScript, Java, C++, Ruby, Perl, Oracle e MySQL.

A edição gratuita da plataforma fornece um conjunto de ferramentas e APIs flexíveis para os desenvolvedores, estes podem escolher a linguagem de programação, aplicação e a maneira que quiserem programar e utilizar as APIs da MapQuest.

### MapQuest Traffic Example



Select the Traffic check box to enable the traffic display.

Figura 3 - 6

### 3.7 Google Directions API

O Google Directions API fornece um serviço de roteamento com base em chamadas por Http, ele pode ser feito com vários tipos de transportes (bicicleta, carro, andando) e também pode ser requisitado através de endereços, cidades, pontos de interesse e coordenadas (Google Developers, 2015).



Para utilizar o serviço é necessário que o usuário tenha uma chave, que pode ser requerida através do website Google Developers, que tem um excelente tutorial de desenvolvimento, apresenta exemplos e varias situações de utilização.

A sua utilização é gratuita até atingir certo numero de requisições, são elas: 2500 requisições de direções em períodos de 24 horas, até 8 waypoints são permitidos para cada requisição e apenas 2 requisições por segundo. Outro termo que a Google restringe é a necessidade de exibição da rota em mapas da empresa, é proibido usar dados de direções sem que seja exibido em um mapa.

A comparação de Neis (2009), apresentada pela Figura 3-7, comprova que o serviço da Google e do OSRM são os com maior velocidade de processamento de rotas.

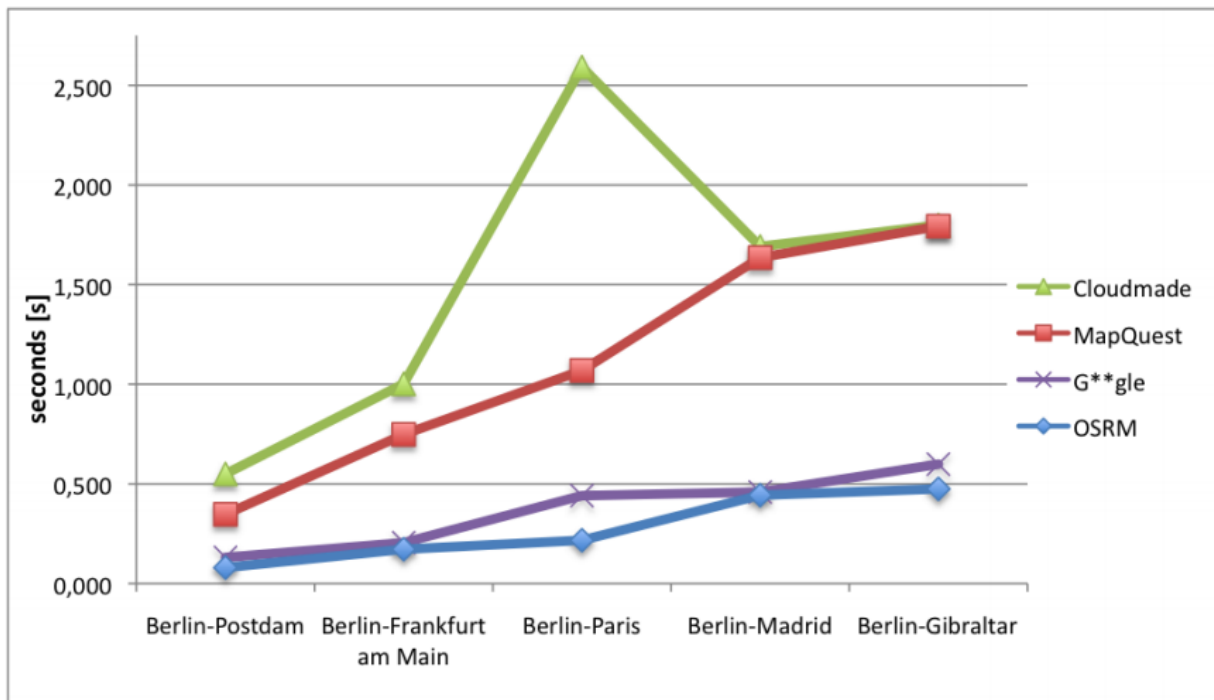


Figura 3 - 7

# Capítulo 4

## Ferramentas

Neste capítulo serão descritas todas as ferramentas utilizadas no desenvolvimento deste estudo. Serão abordadas ferramentas como Eclipse, Java, Nominatim, SwingX -WS, JxMapKit, JxMapView e MapOSMatic.

### 4.1 Eclipse

A plataforma Eclipse é um framework de código aberto e muito prática de se utilizar. Ela fornece uma fundação rica em blocos onde IDEs customizadas podem ser criadas e integradas (Eclipse Platform Technical Overview, 2008).

Apesar de um grande número de características comuns a outras IDEs, o Eclipse se diferencia das outras em várias formas importantes. Talvez a característica mais importante é a sua neutralidade com plataformas e linguagens. Além disso, por ser tão eclético em relação à mistura de linguagens suportadas pelo Eclipse Consortium (Java, C/C++, Cobol) e ainda há projetos sendo desenvolvidos para a adição de linguagens tão diversas como Python, Eiffel, PHP, Ruby e C# (Storkel, 2002).

O Eclipse também possui um grande número de características únicas, como refatoração de códigos, atualizações automáticas de códigos e instalações, lista de tarefas, testes de unidades com JUnit e integração com a ferramenta de construção Ant Jakarta.

Segundo Shelajev (2013), o projeto da plataforma se iniciou com um projeto da IBM Canada em 2001, o que se transformou muito popular entre executivos de grandes empresas. Desde 2004 é apoiado e mantido pela empresa Eclipse Foundation, uma organização sem fins lucrativos e que é patrocinado por empresas de grande expressão como Oracle, IBM, Red Hat, SAP e Google.

A Eclipse foundation mantém no radar todos os projetos baseados na plataforma ou que a utilizam, que, combinados com a disponibilidade de material educativo faz com que Eclipse seja uma escolha sólida para determinar a plataforma de projetos (Shelajev, 2013).

## 4.2 Java

Segundo Eckel, 1996, Java é uma linguagem de programação relativamente nova e bem poderosa que fornece muitas características praticas à desenvolvedores. Ela herdou os pontos fortes da linguagem C++ enquanto eliminou a maior parte de seus problemas. A sintaxe e a lógica de programação é bem parecido com sua base a C++. Java é simples e muito robusta.

Java foi concebida por James Gosling, Patrick Naughton, Chris Warth, Ed Frank e Mike Sheridan, na Sun Microsystems em 1991. A linguagem era chamada Oak mas foi renomeada em 1995 para o atual nome. A linguagem não foi criada para ser utilizada na internet, seu principal motivo era a criação de uma linguagem de plataforma independente que poderia ser utilizada para criar softwares acoplados à uma grande variedade de dispositivos eletrônicos, como microondas, televisores e controles remotos (Java Fundamentals, 2009).

É uma plataforma independente e com grande portabilidade. uma vez que compilada, pode ser utilizada em qualquer maquina que tenha a máquina virtual Java (JVM) instalada. A identificação dos erros é realizada durante a compilação do código, diferentemente de outras linguagens, que os identifica durante a execução. (Horstmann, 1998).

## 4.3 Nominatim

O Nominatim é uma ferramenta de pesquisa de dados que utiliza o dicionário geográfico do OpenStreetMap, o projeto pode ser encontrado no website <http://nominatim.openstreetmap.org>.

Ele fornece componentes como geocodificador e geocodificador reverso, ele requer osm2pgsql para abrir dados, PostgreSQL 8.4+ e PostGIS 1.5+ para funcionar (Teslya).

O Nominatim trabalha com vários tipos de parâmetros e opções, ele consegue identificar endereços incompletos e pontos de referencia e codificar para dados compatíveis com OSM. (OpenStreetMap, 2010). A tabela exhibe os parâmetros que são manipulados.

<i>Opção</i>	<i>Descrição</i>
<i>Formato</i>	<i>Output: -HTML; -JSON; -XML.</i>
<i>json-callback</i>	<i>Função Callback usada para exibir os resultados</i>
<i>q</i>	<i>String de consulta sendo pesquisada</i>
<i>addressdetails</i>	<i>Incluir uma repartição de endereços em elementos separados</i>
<i>limit</i>	<i>Limita o numero de resultados retornados para o inteiro digitado</i>
<i>contrycodes</i>	<i>Limita a pesquisa para um pais específico ou lista de países Ex: contrycodes=countrycode</i>
<i>viewbox</i>	<i>Área preferencial para encontrar resultados de pesquisa. Ex: viewbox=left,top,right,bottom</i>
<i>osm-type</i>	<i>Especifica se o recurso é um nó, caminho, ou relação em OpenStreetMap N- Node W- Way R- Relation</i>

O exemplo abaixo de uma requisição de nominatim por ponto de interesse retorna um resultado no formato JSON:

<http://nominatim.openstreetmap.org/search/Boeing%20Field?format=json>

```

{
  place_id:"127833127",
  licence:"Data © OpenStreetMap contributors, ODbL 1.0.
  http://www.openstreetmap.org/copyright",
  osm_type:"relation",
  osm_id:"537472",
  boundingbox:["47.5127926","47.5463681",-122.3183256",-122.2885146"],
  lat:"47.5295122",
  lon:"-122.300642350829",
  display_name:"Boeing Field, East Marginal Way South, Tukwila, King County,
  Washington, 98118, United States of America",
  class:"aeroway",
  type:"aerodrome",
  importance:0.54552670789228,
  icon:"http://nominatim.openstreetmap.org/images/mapicons/transport_airport2.p.20.png"
}

```

## 4.4 SwingX-WS

SwingX é um toolkit de interfaceamento para Java. É um projeto da Sublabs, criado em 2011 e atualmente existem 284 membros (referencia). Ele foi desenvolvido para trazer melhores componentes GUI do que o anterior Abstract Window Toolkit. Swing fornece um visual mais natural e emula o visual de várias outras plataformas, ele é extensível, orientado a componentes, customizável e tem uma interface bem leve (Tlachač, 2008).

O Swing Web Services (SwingX) fornece muitas classes uteis para o desenvolvimento de aplicações orientadas a web em Java. Ele introduz o um conjunto de JavaBeans e APIs para a interação de tecnologias web com aplicações Swing.

É válido destacar que a biblioteca fornece classes que foram utilizadas neste estudo, como JXMapKit e JXMapView.

## 4.6 JxMapView

O JXMapView foi a primeira classe no projeto SwingX-WS, ele implementa um mapa mundial no navegador. Ele basicamente divide todo o mapa em tiles de tamanho igual que pode ser referenciado em coordenadas x,y (Tlachač, 2008).

## 4.5 JxMapKit

É um par de JxMapViews pré configurados para facilitar a utilização de características integradas. Isto inclui botões de zoom, uma barra de zoom, um mini-mapa na parte inferior direita que exibe uma visão geral do mapa (Bernabé, 2010).

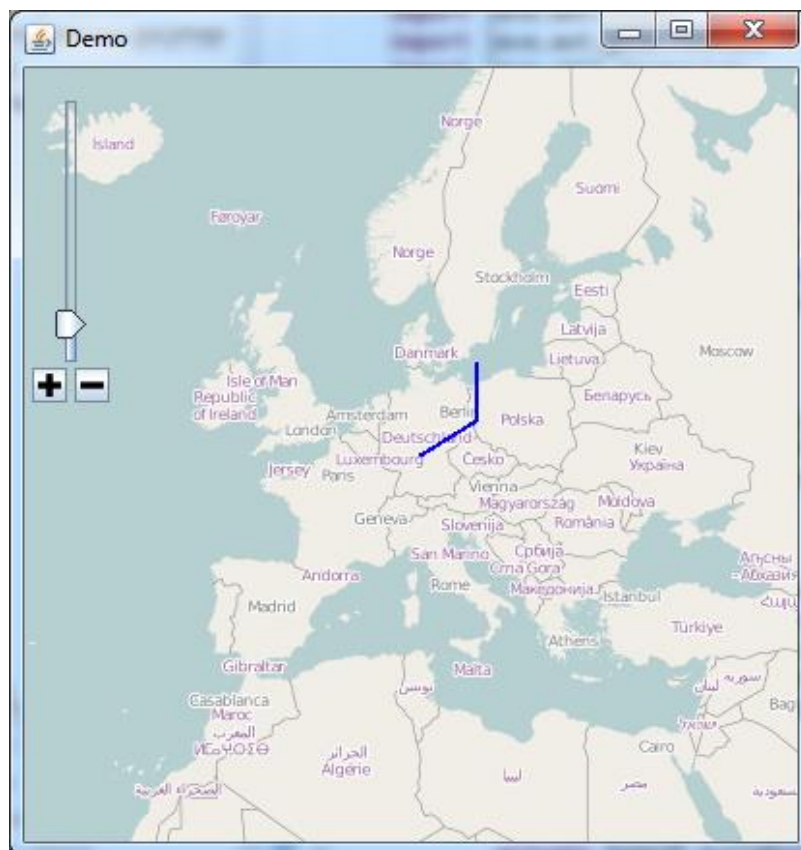


Figura 4 - 1

# Capítulo 5

## Desenvolvimento

Para o desenvolvimento do algoritmo foi utilizado como base o estudo “Sistema de Interfaceamento para Problemas de Roteamento de Veículos Utilizando o OpenStreetMap” realizado pela ex-aluna Alba Assis Campos do curso de Sistemas de Informação da UFVJM.

Em seu projeto, Alba desenvolveu uma ferramenta que calcula rotas entre cidades, e cada uma com sua demanda específica dentro de um arquivo texto e assim seria calculada a rota, conforme observado na Figura 5-1.

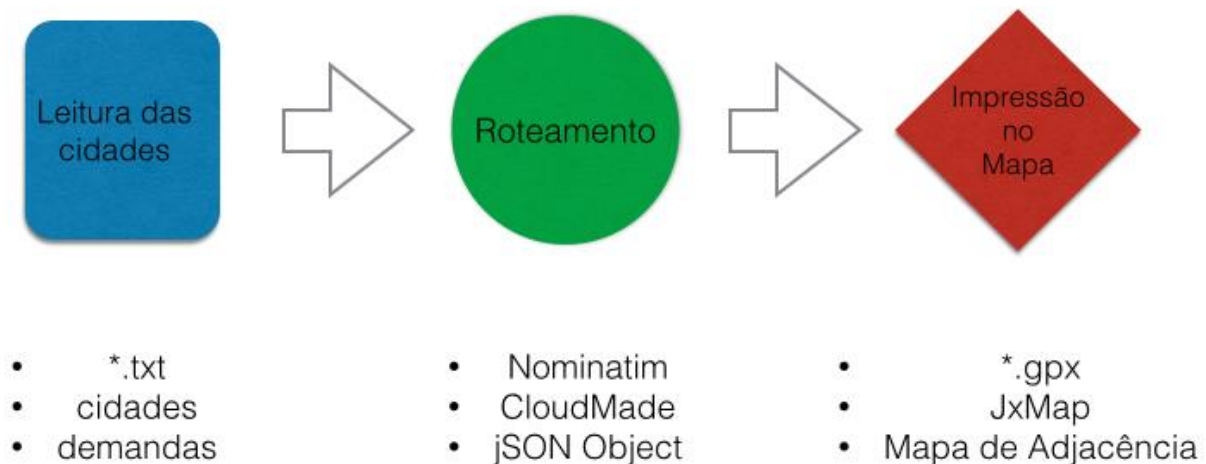


Figura 5 - 1

Em primeira instância, o arquivo de texto armazena as cidades e suas demandas - valor de prioridade para cada cidade e que não foi implementado em seu estudo por razões desconhecidas.

O roteamento é realizado com a tradução do nome da cidade para valores de coordenadas através do Nominatim, e seguindo com a consulta web da rota entre os pontos utilizando a API do CloudMade e finalmente um objeto JavaScript (*JSONObject*) é criado como retorno da rota.

Através do objeto e da especificação de cada trecho da rota, é realizado o cálculo das distâncias para que o mapa de adjacência seja criado e por final a plotagem da rota no mapa criado e customizado pelo JxMapKit e SwingX-WS.

A consulta web funciona como o pivô central do sistema, sem ela não há criação de rotas e conseqüentemente nenhum resultado final. O sistema é totalmente dependente do funcionamento do serviço de roteamento, neste caso o CloudMade, que foi extinto por razões desconhecidas e assim o estudo da Alba ficou a par de um novo sistema que realize essa função.

Assim a motivação principal deste novo estudo, a implementação de uma API para reestabelecer o funcionamento do projeto inicial, além de algumas outras melhorias.

As principais colocadas em prática neste estudo foram o desenvolvimento e implantação da ferramenta Open Source Routing Machine e no sistema de buscas de cidades. Foi desenvolvido um melhoramento para que o sistema fosse capaz de buscar e calcular rotas com endereços complexos e pontos de interesse, além de também ser capaz de captar endereços incompletos.

Outro fator de melhoria neste software foi o aprimoramento da velocidade de processamento da rota, já que como foi mostrado, a ferramenta OSRM tem uma vantagem muito superior em relação ao CloudMade.

A base do funcionamento do sistema continuou a mesma, já que o único fator que impedia o funcionamento era um sistema ativo que realizasse a função de calcular a rota entre trechos. A estrutura do sistema segue um fluxo, representado na Figura 5-2.



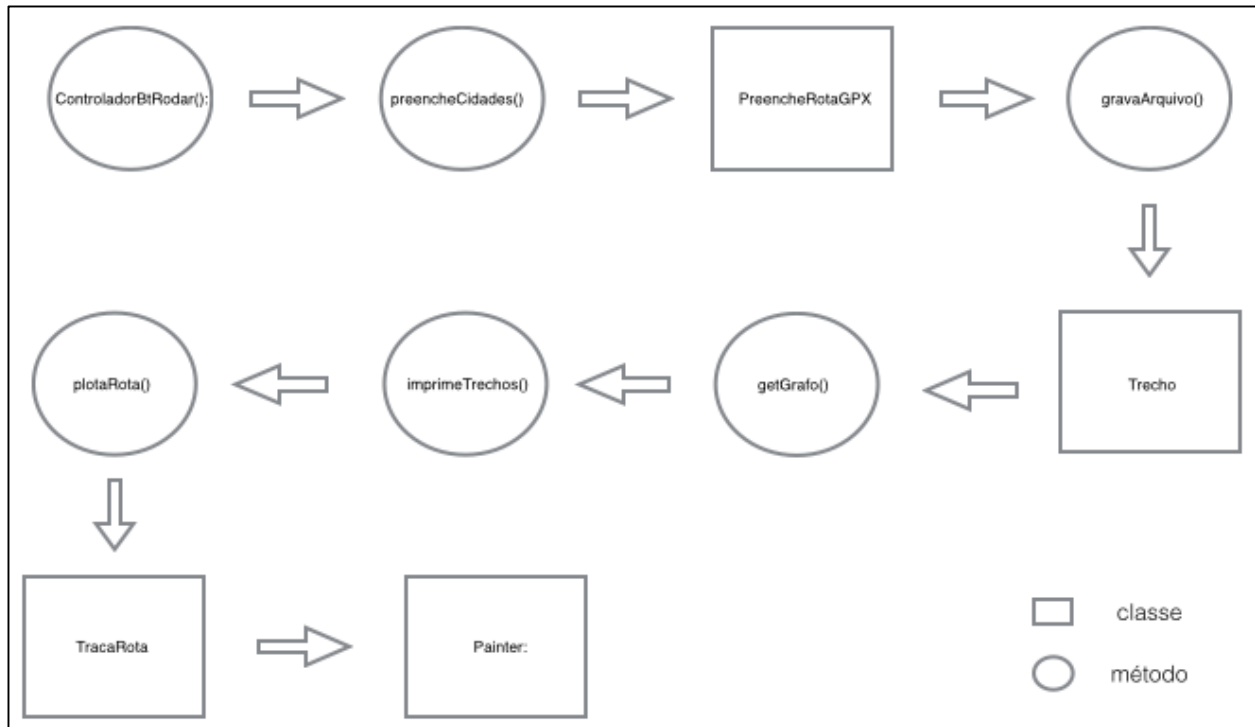


Figura 5 - 2

Ao executar a aplicação, a janela do JxMap se abrirá e será pedido que abra o arquivo.txt onde estão todos os endereços inseridos pelo usuário. Com esta ação, uma sequência de funções e classes será chamada até que as rotas sejam exibidas no mapa.

- ControladorBtRodar(): Classe chamada após clicar em rodar no painel da aplicação, uma instancia de classe JxMap sera criada com o nome do arquivo selecionado.
- preencheCidades(): Nesta função é preenchido os endereços no arquivo texto, estes são adicionados na estrutura da classe JxMap. O serviço Nominatim captura o endereço ou ponto de interesse e os transforma em coordenadas.
- PreencheRotaGPX: A classe, já com pontos de coordenadas de cada endereço lido, faz a consulta web da rota através do serviço Open Source Routing Machine. Uma requisição comum do ORSM retorna um valor JSONObject codificado, diferentemente das outras ferramentas citadas, então foi necessário utilizar a função “decode()”, da

biblioteca `polyline.encoded` criada pela equipe desenvolvedora do Leaflet (Referencia). Para o OSRM em específico, é necessário usar precisão 6 para a decodificação.

- `gravaArquivo()`: Esta função captura e grava o arquivo `.gpx` criado pela consulta acima.
- `getGrafo()`: A função gera um grafo entre os endereços inseridos e cria uma matriz de adjacência, armazenando distancias entre cada endereço. Para cada distancia é criada uma instancia da classe `Trecho`.
- `Trecho`: Esta classe possui atributos (`origem`, `destino`, `JSONArray`, `JSONObject`, `distância`) de um trecho entre endereço A para endereço B. Assim então é feita a consulta no OSRM utilizando o roteamento padrão (menor caminho de carro). O retorno é armazenado na classe e cria um elemento “`route-geometry`”, que é a geometria da interação entre trechos, também chamada de rota.
- `imprimeTrechos()`: Vai exibir na saída padrão da IDE o mapa de adjacência entre os endereços.
- `plotaRota()`: Gera o mapa com a rota entre os endereços inseridos pelo usuario.
- `TracaRota`: Recebe os trechos obtidos pela classe `JxMap` e a tela que contem o mapa. Esta tela é feita pelo `JXMapKit` com configurações de exibição, assim como os tiles de mapa, zoom, url, através da classe `TileFactory` do `SwingX`.
- `Painter`: Classe da ferramenta `SwingX` que auxilia o desenho da rota no mapa. Este desenho é feito pela função `paint`, que cria uma imagem de duas dimensões e liga um ponto ao outro, que foram passados para a classe `TracaRota`.

## 5.1 Diagrama de Caso de Uso

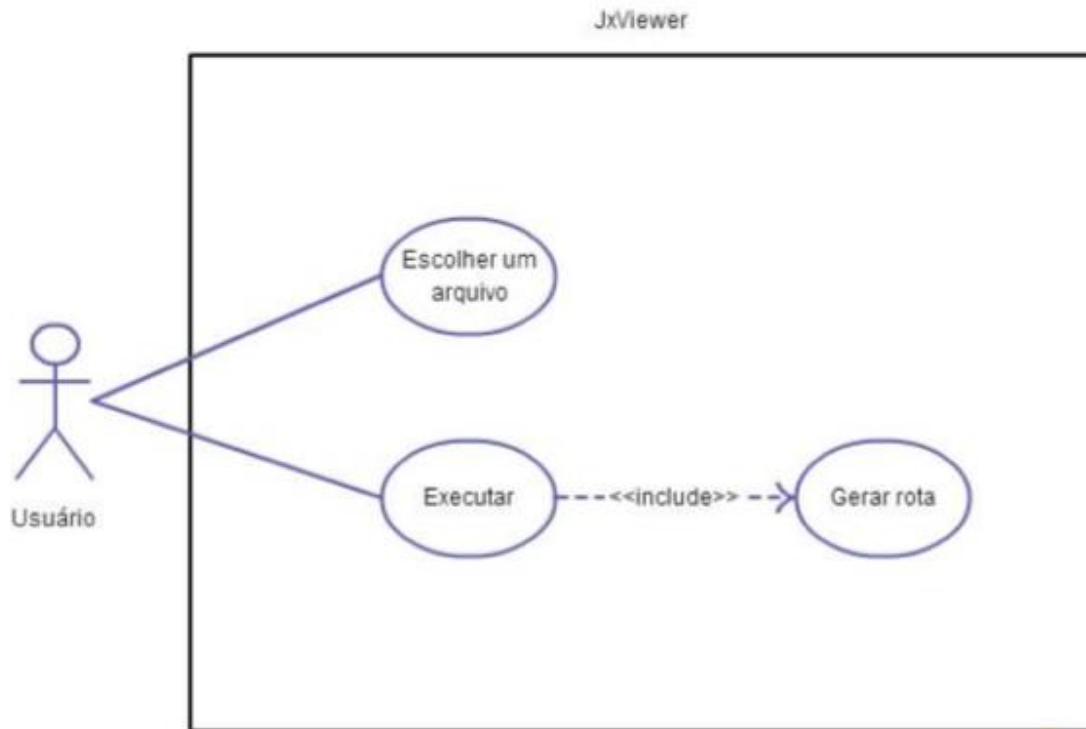


Figura 5 - 3

Descrição	<i>O caso de uso permite que o usuário crie um arquivo com uma sequencia de endereços na qual deseja encontrar a matriz de adjacência e a rota entre estes. O sistema abre o arquivo e exibe a rota em um mapa.</i>
Caso de Uso	<i>Criar rota</i>
Ator	<i>Usuario</i>
Pre-Condiciona	<i>Selecionar Arquivo</i>
Pos-Condiciona	<i>Exibir um mapa com a rota</i>
Fluxo	<ol style="list-style-type: none"> <li>1. <i>Usuário, acessa a janela principal com dois botões, escolher um arquivo e rodar.</i></li> <li>2. <i>O usuario escolhe o arquivo com os endereços e pressiona o botao escolher um arquivo.</i></li> <li>3. <i>O usuario pressiona o botao rodar</i></li> <li>4. <i>A rota é exibida</i></li> </ol>
Fluxo Alternativo	<i>4.1 O mapa não pode ser exibido por endereço inexistente ou falta de conexão à internet.</i>
Caso de Uso Complementar	<i>Exibir a Rota</i>

## 5.2 Diagrama de Classes

O diagrama de classes mostra as classes utilizadas no sistema, assim como sua estrutura, atributos e métodos.

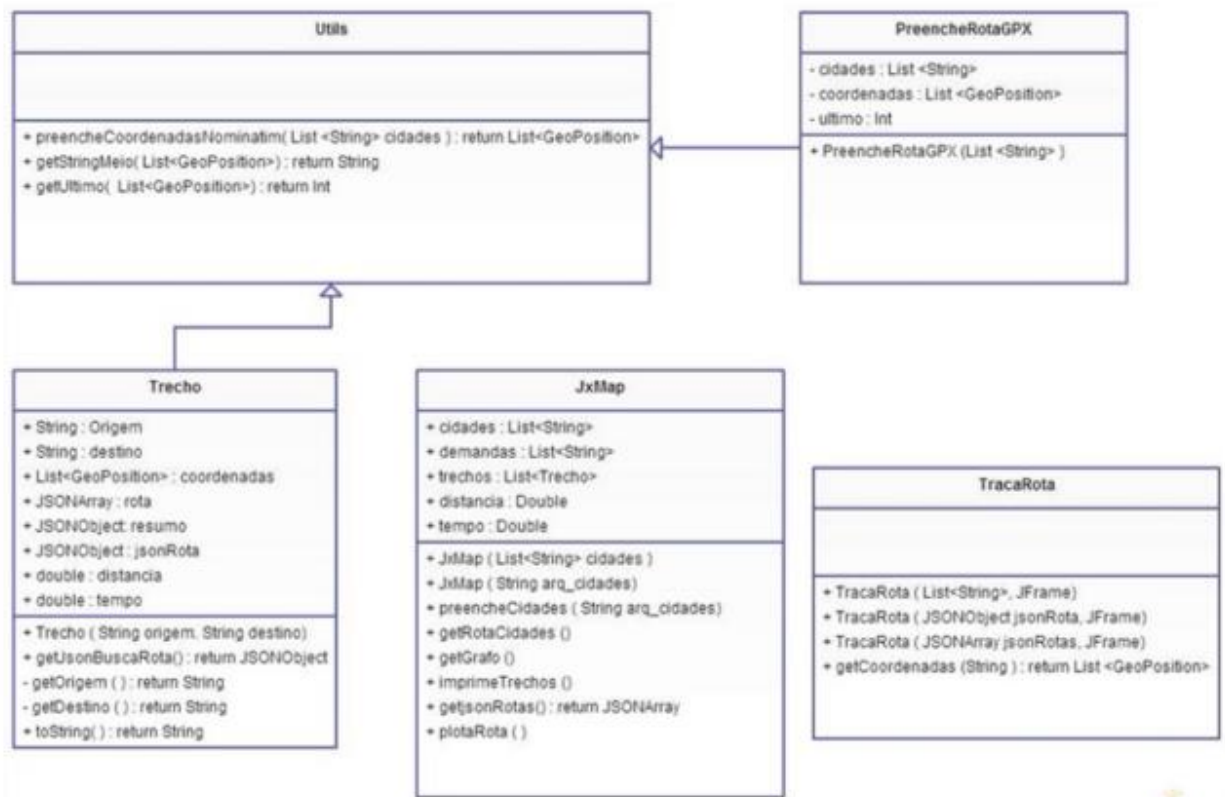


Figura 5 - 4

## 5.3 Diagrama de Frequência

O diagrama de frequência apresenta as interações entre classes assim como uma linha do tempo acompanhando estas interações.

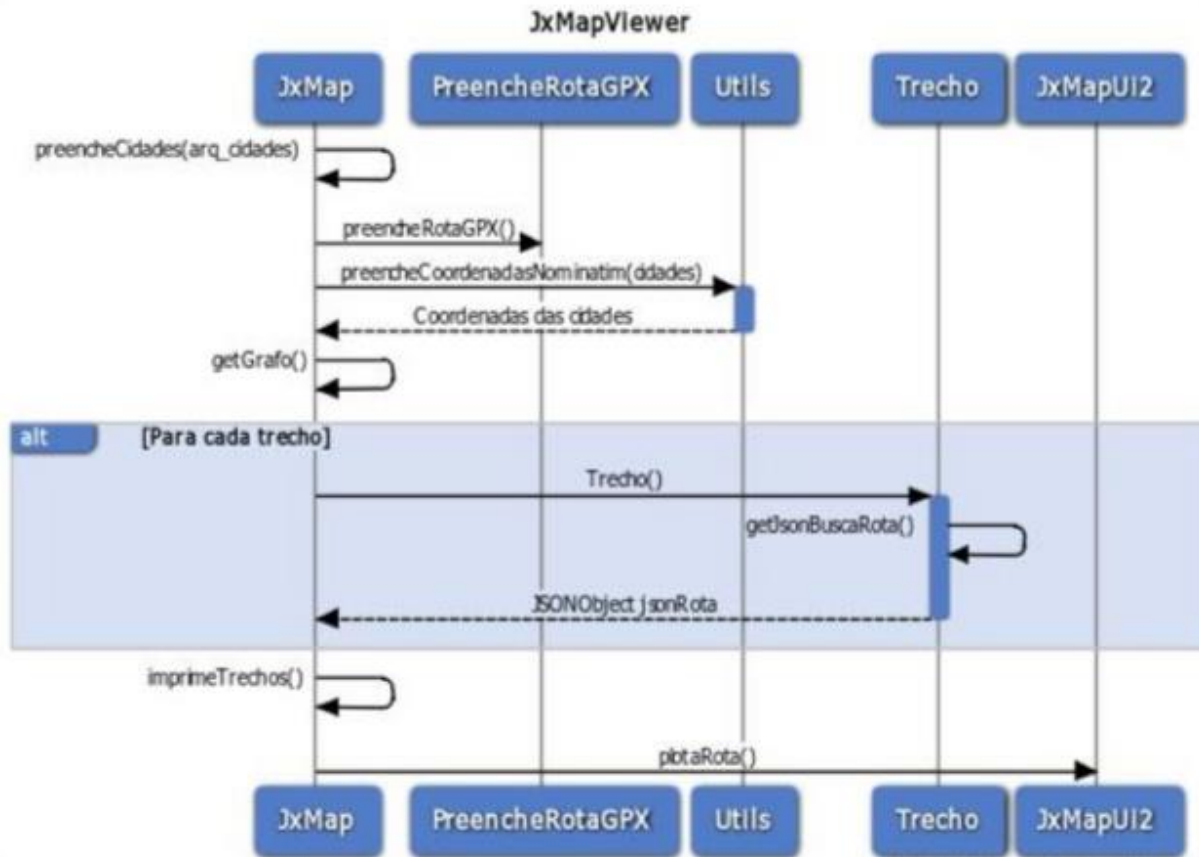


Figura 5 - 5

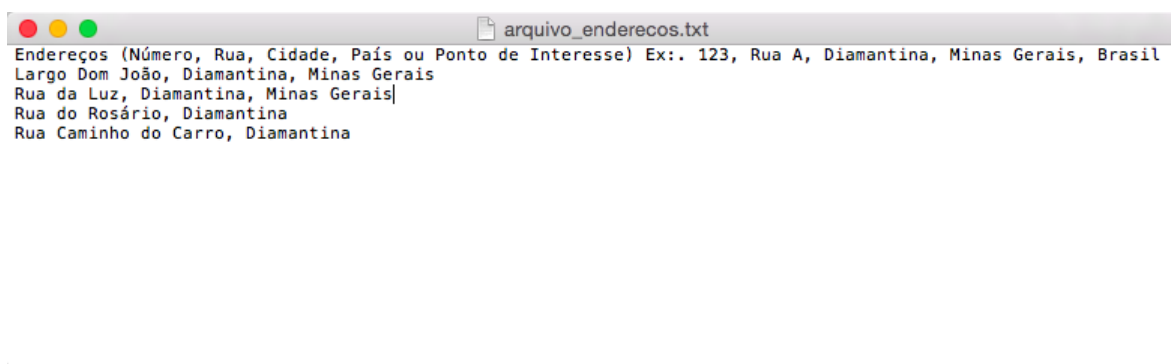
# Capítulo 6

## Testes

Este capítulo apresentará os testes realizados com o sistema.

Os principais aspectos desejados para serem testados foram a exatidão de endereços e confiabilidade do Nominatim, ou seja, realizar testes com diferentes entradas de endereços para entender o quão ampla a ferramenta consegue fazer a tradução de endereço para coordenada; a sua escalabilidade, como o sistema responde a quantidades variadas de endereços (4 a 32 endereços); a variação de tipos de endereços, como o Nominatim responde a diferentes formas de endereços, desde o formato simples contendo número, nome da rua e cidade até pontos de interesse; e por final a consulta web e impressão no mapa.

O formato necessário para a leitura dos endereços é um arquivo comum (extensão .txt), o algoritmo lê todos os strings da linha, começando pela segunda linha, a primeira é destinada ao campo de introdução do arquivo.



```
arquivo_enderecos.txt
Endereços (Número, Rua, Cidade, País ou Ponto de Interesse) Ex.: 123, Rua A, Diamantina, Minas Gerais, Brasil
Largo Dom João, Diamantina, Minas Gerais
Rua da Luz, Diamantina, Minas Gerais|
Rua do Rosário, Diamantina
Rua Caminho do Carro, Diamantina
```

Figura 6 - 1

Em seguida o usuário deve abrir este arquivo através da janela que é exibida logo após a execução do aplicativo, como mostra a Figura 6-2. Lembrando que o algoritmo lê qualquer arquivo com endereços a partir da segunda linha.

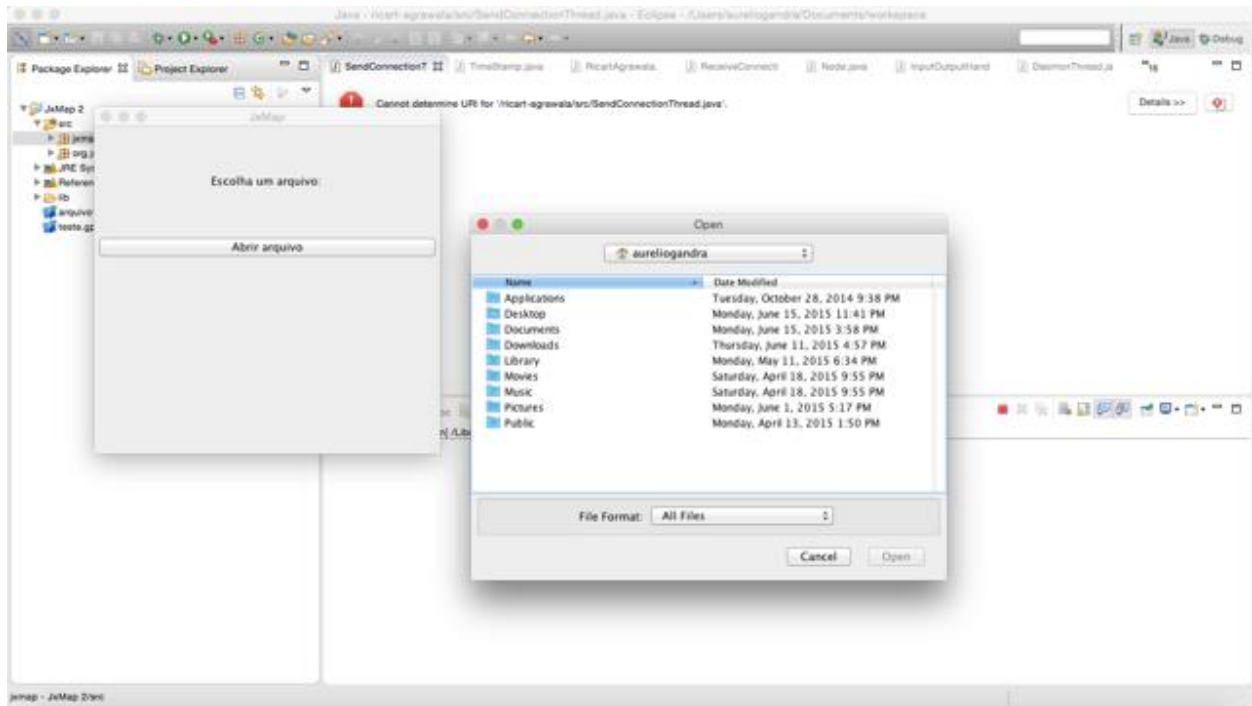


Figura 6 - 2

A utilização da ferramenta MapOSMatic é muito importante para a verificação da quantidade de dados OpenStreetMap (ruas, pontos de interesse, direção das vias, etc) existentes naquela região.

Para que o sistema pudesse ser testado com maior confiabilidade, a cidade base do estudo foi Seattle-EUA, uma cidade onde há uma grande comunidade de contribuidores do OSM, fazendo-a assim uma cidade com grande nível de detalhes, como pode ser notado através da consulta do MapOSMatic representada na Figura 6-3.

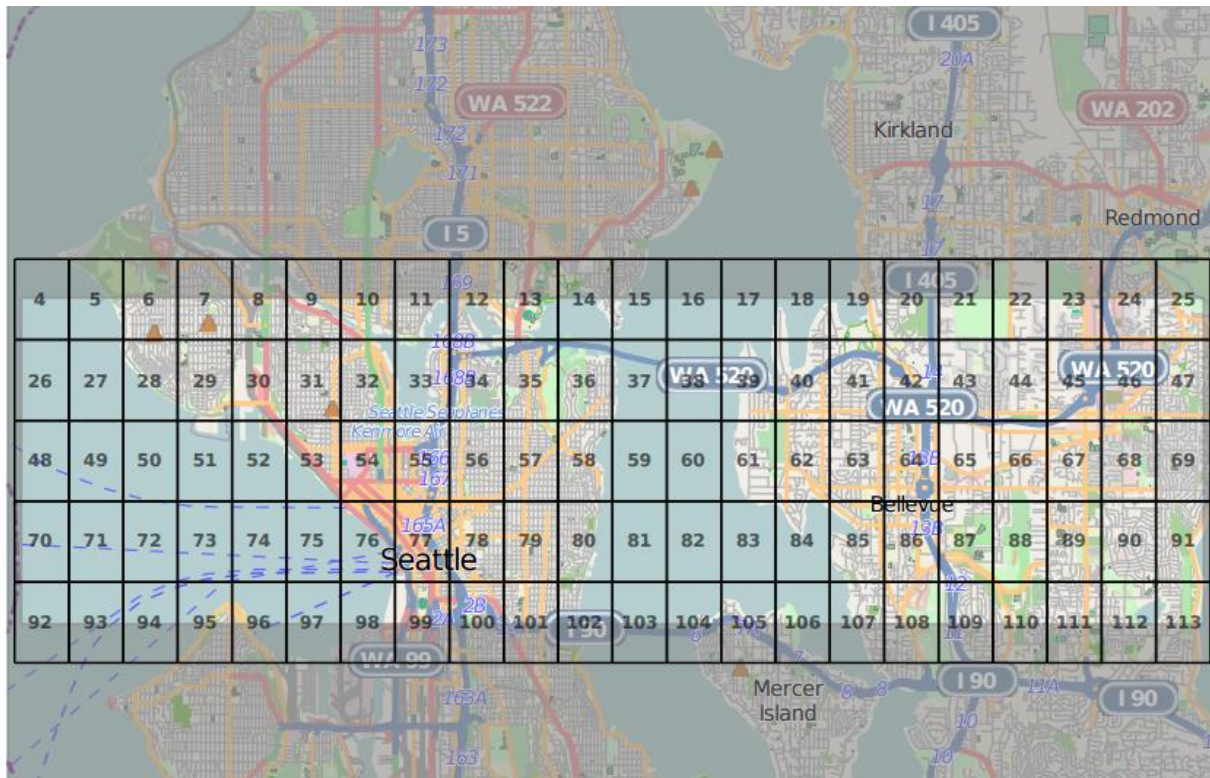


Figura 6 - 3

Para que o teste pudesse ser feito também o reconhecimento de endereços incompletos pelo Nominatim, o próximo teste foi feito com 4 endereços contendo somente Numero, Nome da Rua e Cidade, conforme mostra a Figura 6-4.



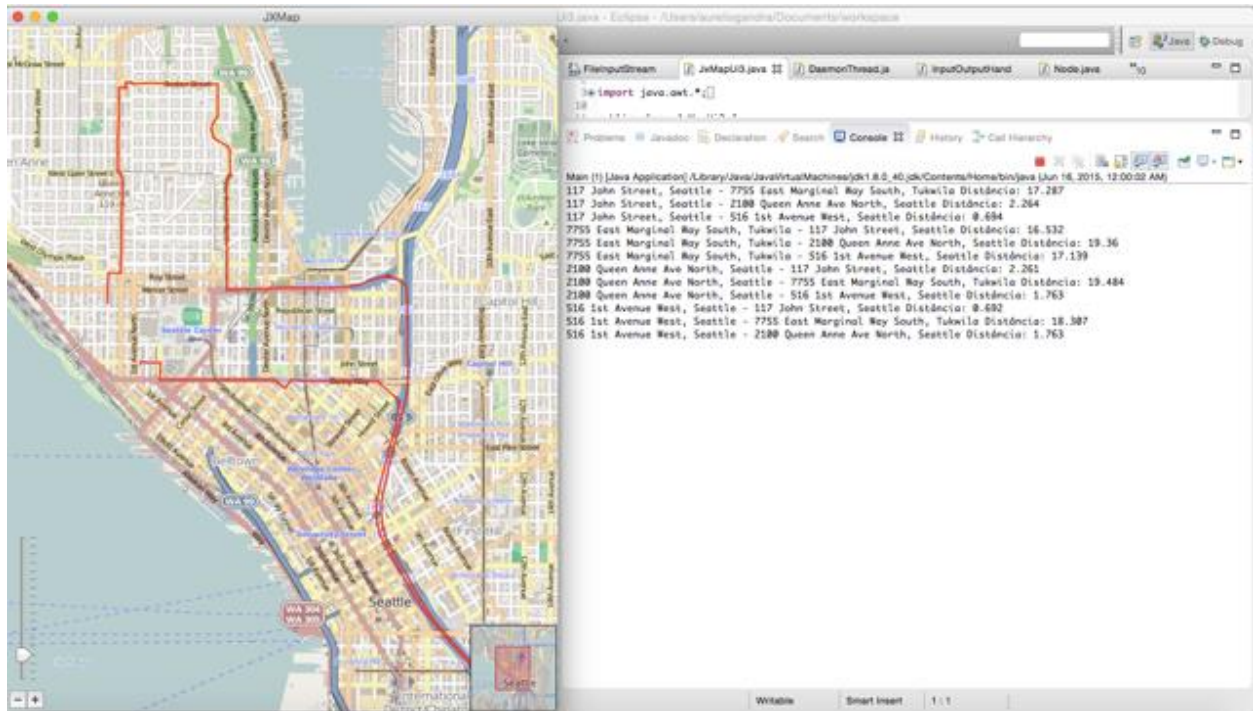


Figura 6 - 4

Pôde-se notar que as rotas entre endereços foram feitas corretamente, seguindo a direções corretas das vias da cidade e ainda exibindo a Matriz de Adjacência das distância entre endereços.

Após o teste realizado com sucesso, a próxima parte foi testar a reacao de escalabilidade do sistema, adicionando gradativamente o numero de endereços com 8 e 32 entradas:

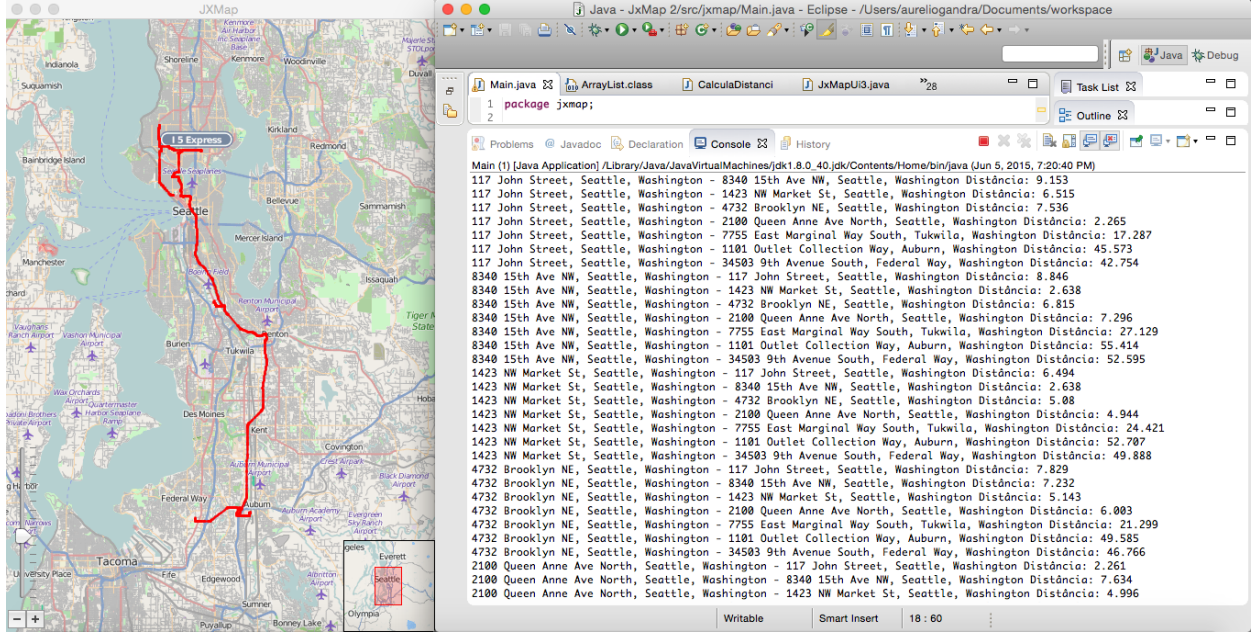


Figura 6 - 5

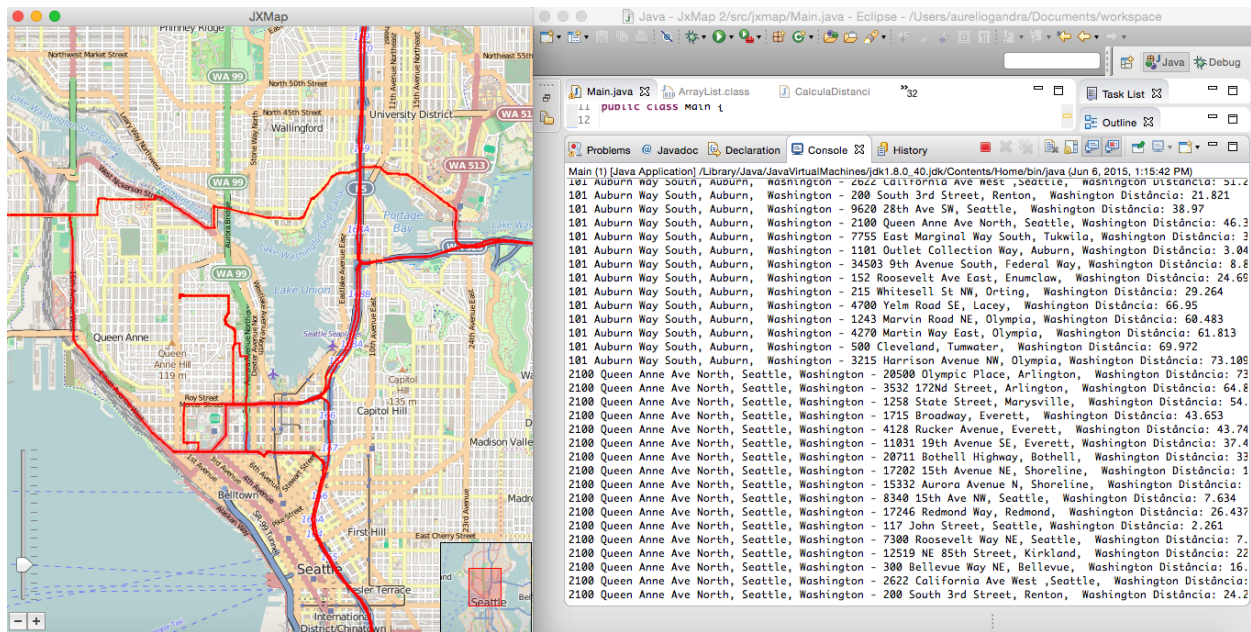


Figura 6 - 6

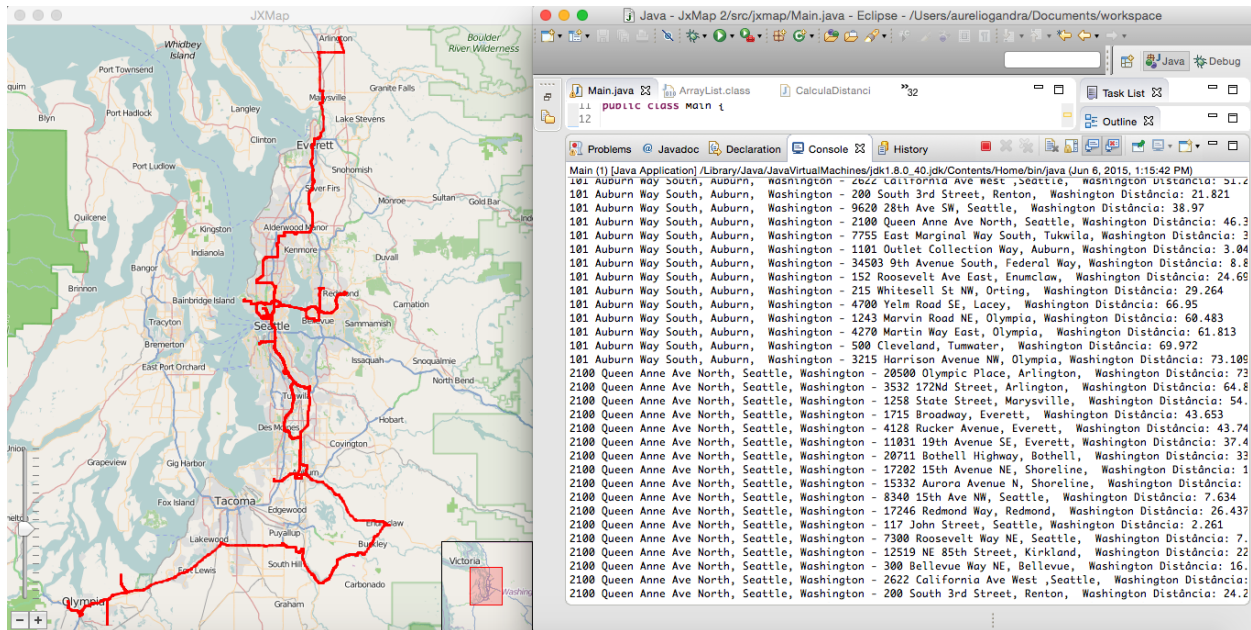


Figura 6 - 7

Para verificar a confiabilidade utilizando apenas pontos de referencia, utilizamos 4 locais na cidade de Seattle. Os pontos utilizados foram: Boeing Field (aeroporto particular da empresa), University of Washington, Aeroporto Internacional de Seattle e o Aeroporto Municipal de Renton, cidade vizinha a Seattle. O arquivo texto utilizado pode ser verificado na Figura 6-8.

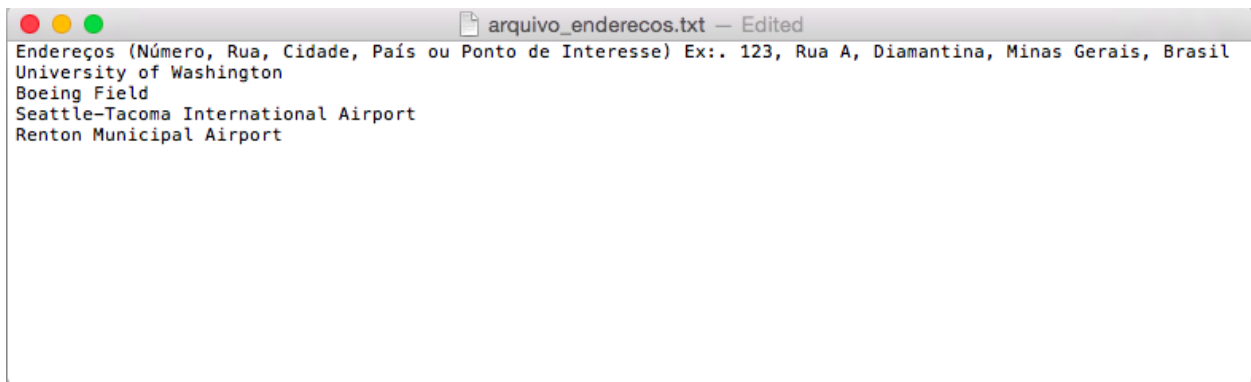


Figura 6 - 8

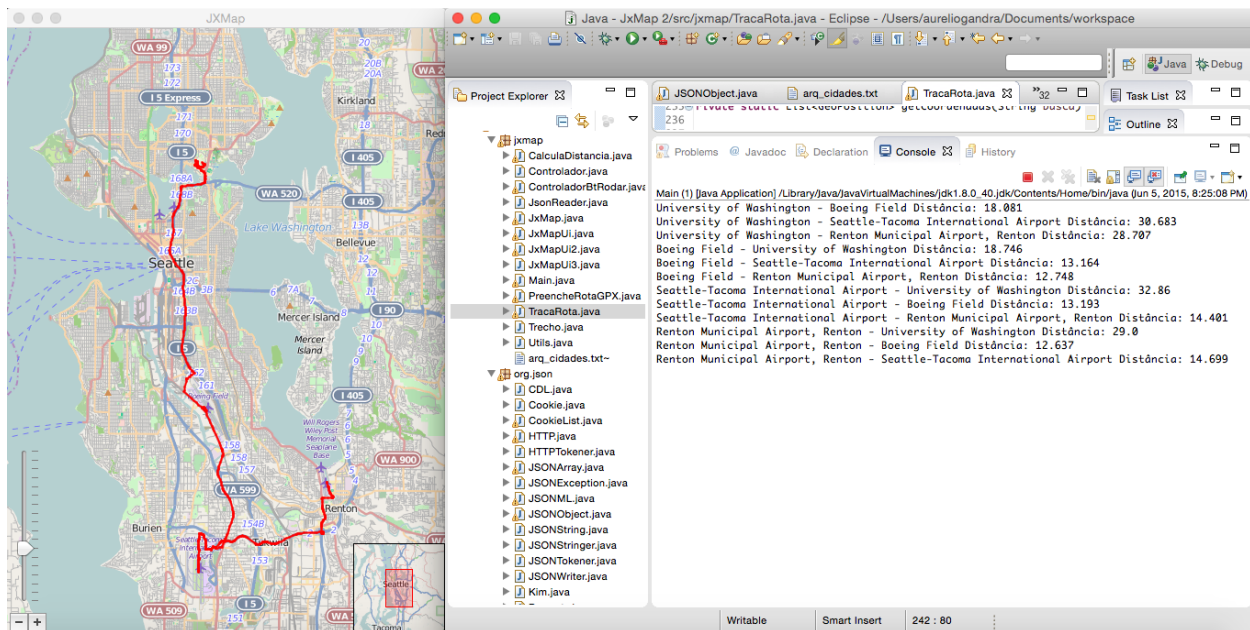


Figura 6 - 9

A ideia inicial do estudo foi da utilização de um mapa mais regional, como por exemplo a cidade de Diamantina. Foi realizado um teste na cidade, representado pela Figura 6-11.

Infelizmente, de acordo com a consulta do MapOSMatic para a região, representada pela Figura 6-10, não há dados OSM suficientes para que os testes criassem uma alta confiabilidade de resultados. Estes dados podem ser nomes de ruas, bairros, direção das vias, nomenclatura para que o Nominatim faça a tradução para coordenadas. Assim, problemas como rotas mal calculadas e dificuldade em encontrar nomes de ruas podem ser comuns na reposta do algoritmo.

Esta falta de dados significa que não existe uma comunidade ativa de contribuição para o OpenStreetMap, isto é, para que haja um maior detalhamento é necessário a inserção de informações no banco de dados do OSM.

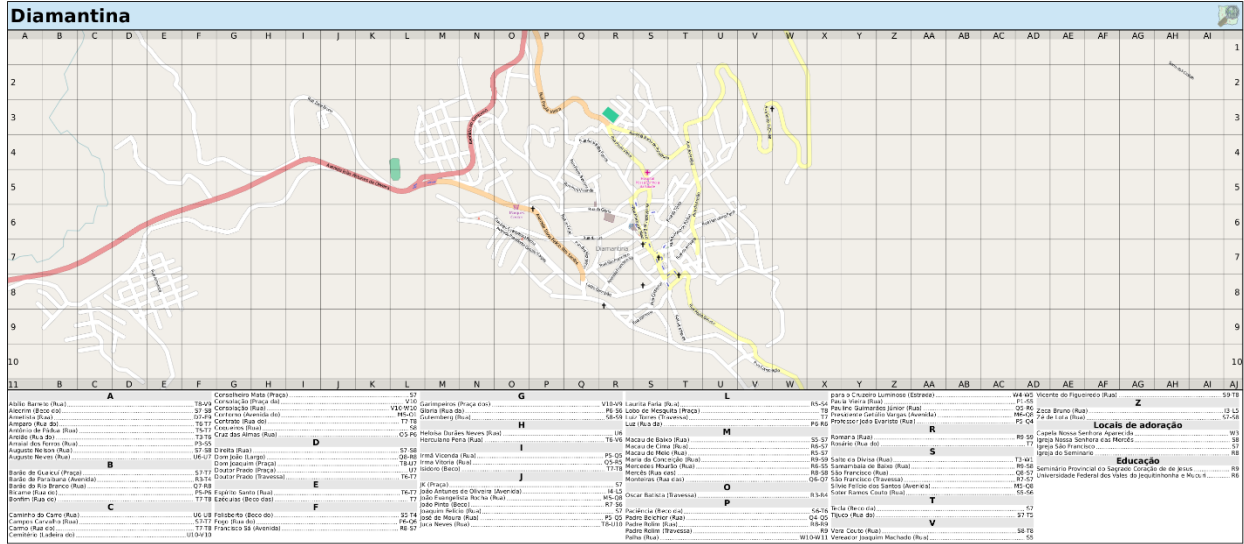


Figura 6 - 10

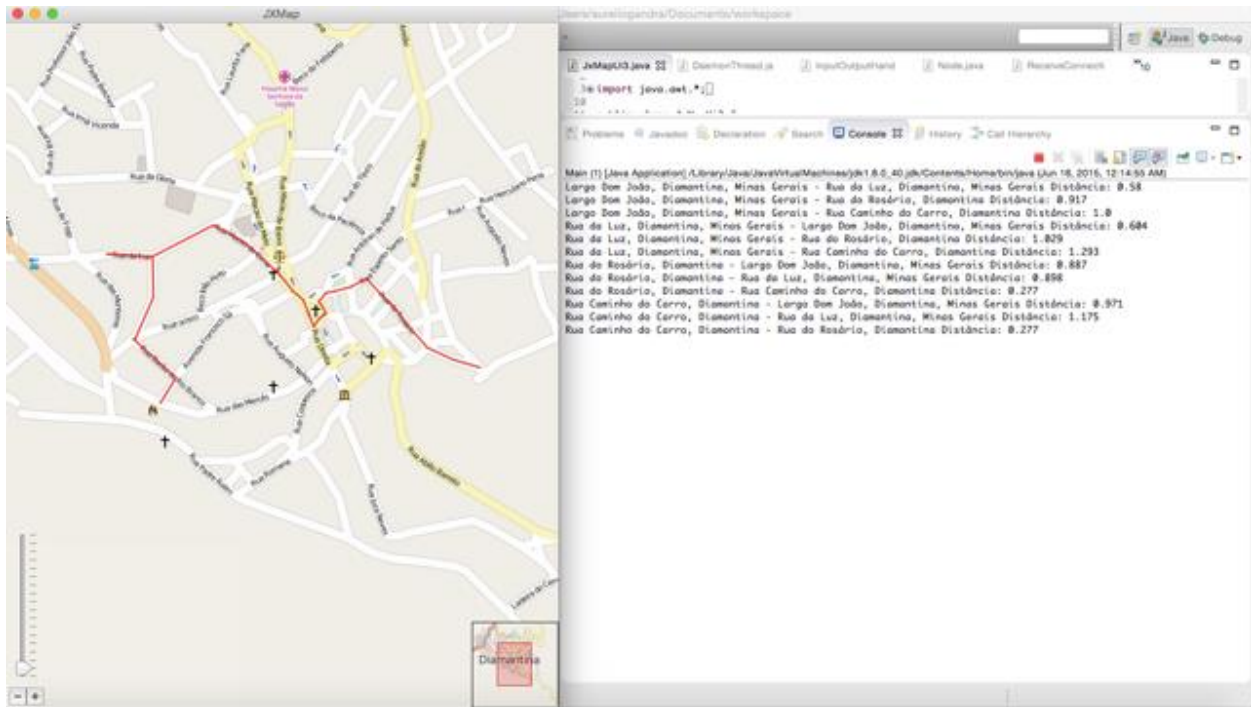


Figura 6 - 11

Pode-se notar que não há informações suficientes para criação de rotas confiáveis em relação a direção de vias dentro da cidade de Diamantina-MG.

Com a realização dos testes acima, foi possível perceber que o sistema consegue trabalhar com conjuntos pequenos de endereços até grandes números e trazendo o mapa de adjacência juntamente com a impressão do mapa. A ferramenta Nominatim consegue realizar a tradução com êxito de endereços incompletos, completos e pontos de interesse. E por final, a impressão no mapa é feita com êxito e é possível aumentar e diminuir o zoom com sucesso, assim como a exibição de nomes de ruas, bairros, cidades e pontos de interesse.

# Capítulo 7

## Conclusões Finais

Os resultados do desenvolvimento mostraram o aplicativo desenvolvido é possível criar extensivas listas de rotas para endereços, pontos de interesse, cidades, e outros. O algoritmo utilizado é bem simples e com uma boa funcionalidade, já que foram feitos diversos testes de com uma grande variação de pontos. É importante ressaltar que como resultado, o algoritmo também retorna uma matriz de distâncias, que calcula a distância entre cada ponto que o usuário colocou.

A possibilidade de se trabalhar com roteamento com dados reais traz uma grande confiabilidade em questões de aplicabilidade de algoritmos em empresas e trabalhos acadêmicos, já que a maioria dos estudos atuais trabalham com situações em que dados são simulados e que, na maioria das vezes, se tornam obsoletos ao tentarem utilizar situações reais.

Diante destes resultados, existem alguns fatores que podem ser utilizados para que programa tenha melhor desempenho e maior transparência para o usuário. O primeiro caso seria o melhoramento dos dados do OpenStreetMap, por exemplo a cidade de Diamantina não tem todos os dados completos (Nomes de Rua, Mão/Contra-mão, números e pontos de interesse) e poderia ser implementado com a captura de dados pela cidade através de um aparelho GPS e a manutenção no sistema OSM. Outro caso seria a criação de uma interface completa para a inserção dos endereços e impressão da rota, já que atualmente o usuário tem a necessidade de utilizar o arquivo texto se quiser modificar endereços para rotas e depois abrir o programa.

Este estudo também viabiliza o desenvolvimento de aplicações para empresas/projetos de logística e roteamento, por exemplo a criação de rotas dentro da cidade de Diamantina para coleta seletiva em parceria com a Prefeitura Municipal.

# Referências

- Amos, M. OWL (openstreetmap watch list), Webservice: [http://matt.dev.openstreetmap.org/owl\\_viewer/](http://matt.dev.openstreetmap.org/owl_viewer/), 2011.
- Barnhart, C., Laporte, G., Vehicle Routing, Handbook in OR & MS, vol. 14, chapter 6, pp. 1-62, Montreal, Canada, 2007.
- Bardi, E.J., Raghunathan, T.S. & Bagchi, P.K. 1994, "Logistics information systems: The strategic role of top management", Journal of Business Logistics, vol. 15, no. 1, pp. 71-85.
- Bernabé S., Plaza A., A New System to Perform Unsupervised and Supervised Classification of Satellite Images from Google Maps. Hyperspectral Computing Laboratory Department of Technology of Computers and Communications University of Extremadura, Avda. de la Universidad s/n E-10071 C´aceres, Spain, 2010.
- Brynjolfsson, E., Hitt, L., and Yang, S. "Intangible Assets: Computers and Organizational Capital," Brooking Papers on Economic Activity (1) 2002, pp 137-181.
- Ching-Sheng W., Yu-Hung S., Ching-Yang H. 2008 "A 3D Virtual Navigation System Integrating User Positioning and Pre-Download Mechanism", World Academy of Science, Engineering and Technology 40.
- Codescu, M.; Horsinka, G.; Kutz, Oliver; Mossakowski, Till; Rau, Rafaela. OSMonto - An Ontology of OpenStreetMap Tags. DFKI GmbH, University of Bremen, Germany, 2011.
- den Hengst, M; Sol G. H. "The Impact of Information and Communication Technology on Interorganizational Coordination: Guidelines from Theory." March 2001.
- Dewan, S., and Kraemer, K.L. "Information Technology and Productivity: Evidence from Country-Level Data," Management Science (46:4) 2000, pp 548-562.
- Eckel, B., Thinking in Java: A Definitive Introduction to Object-Oriented Programming in the Language of the World Wide Web. Prentice Hall, Upper Saddle River, NJ, 1998.
- Eclipse Platform Technical Overview. Eclipse Corner Whitepaper (2006). (<http://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html>)
- Fundamentos de Geodésia, <http://itc.ufes.br/geomaticsce/Modulo%20Geodesia.pdf>, 2015.
- Geobank, Serviço Geológico do Brasil – CPRM. <http://www.geobank.cprm.gov.br> , Salvador, 2013.
- Google Developers - Maps, <https://developers.google.com/maps/>, 2015.
- Google Maps, <http://www.google.com/maps/about/>, 2015.
- Google Maps Developers, <http://www.googlemapsdev.com/>, 2015.



Grover V, and William J. Kettinger. Business Process Change: Concepts, Methods, and Technologies. Hershey 1998.

Haklay, M.; Weber P. OpenStreetMap: User-Generated Street Maps. University College London. Published by the IEEE CS. 1536-1268/08. London, 2008

Horstmann S.C.,Cornell C.. Core Java 2, Volume I: Fundamentals. Prentice Hall, Upper Saddle River, NJ, 1998.

How Stuff Works, <http://electronics.howstuffworks.com/gadgets/travel/gps.htm>, 2014.

Instituto Brasileiro de Geografia e Estatística (IBGE), Nocoas Basicas de Cartografia. [http://www.ibge.gov.br/home/geociencias/cartografia/manual\\_nocoas/nocoas.pdf](http://www.ibge.gov.br/home/geociencias/cartografia/manual_nocoas/nocoas.pdf). Rio de Janeiro, 1998.

Ismo R., Simo P., Arttu H., Visualizing real-time GPS data with Internet's VRML Worlds, Proceedings of the 6th ACM international symposium on Advances in geographic information systems, p.52-56, November 02-07, 1998, Washington, D.C., USA

Karney, F. F. Charles, Geodesics on an ellipsoid of revolution. <http://geographiclib.sourceforge.net/geod.html>. SRI International, Princeton, NJ, USA, February, 2011.

Laporte, G, The Vehicle Routing Problem: “An overview of exact and approximate algorithms”, European Journal of Operational Research, vol 59, pp. 345-358. Montreal, Canada, 1991.

Laudon, Kenneth C., and Jane Price Laudon. Management Information Systems: Organizational and Technology in the Networked Enterprise. Prentice Hall, 1999.

Luxen, D.; Vetter, C. Real-Time Routing with OpenStreetMap data. 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2011). November, 2011.

Nielsen Mobile Internet Report. MapQuest History. Nielsen Top 10 Lists for 2007. comScore Media Metrix. February, 2008.

OpenHub®, <http://www.openhub.net>, 2015.

Open Source Routing Machine, <http://project-osrm.org>, 2015.

OpenStreetMap, <http://www.openstreetmap.us>, 2015.

Public release Version, “NAVSTAR GPS USER EQUIPMENT INTRODUCTION”, September 1996.

Raposo A., Neumann L., Magalhaes L., Ricarte I., 1997 "Efficient Visualization in a mobile WWW environment",. Proceedings of the WebNet 97, World Conference of the WWW, Internet, and Intranet.

- Silva, A.S., Junior J.G., Geodésia geométrica. Apostila não publicada. Universidade Federal de Viçosa, 131. 1996.
- Shelajev O., Introduction: Intro and History to the Eclipse IDE. Using Eclipse for Java development: A helpful introduction to the world's most-used IDE, <https://www.zereturnaround.com>. October, 2013.
- Smith, P.; Mooney, P. OSM-CAT: A Java tool for OpenStreetMap Contributor Analysis. Department of Computer Science, National University of Ireland Maynooth, Co. Kildare, Ireland, 2012.
- Swanson-Hysell, N. Earth and Planetary Science 101 - Field Geology and Digital Mapping, Lecture 10: Introduction to Digital Mapping Systems. University of Berkeley - California, 2014.
- Teslya, N., Web Mapping Service for Mobile Tourist Guide, Proc. of the 14th Conf. of Open Innovations Association FRUCT, State University of Aerospace Instrumentation, pp. 140-147.
- Tlachač, M. Visualization and GUI for CoUniverse. Bachelor Thesis. Masarykova Univerzita. Fakulta Informatiky, Brno, Spring, 2008.
- The GeoVRML Working Group, <http://amm.ai.sri.conffgeovrml/>.
- The Origins of Java, Java Fundamentals, Java: A beginner's Guide. Chapter 1, vol. 12, pp 1-30. Oracle, 2009.
- Topf F; Chilton, S. OpenStreetMap, Using and Enhancing the Free Map of the World. Cambridge, 2011.
- Topografix. GPX 1.1 Schema Documentation. <http://www.topografix.com/GPX/#SchemaProperties>. Kirkland, MA, 2014.
- W3C, World Wide Web Consortium, [www.w3c.org](http://www.w3c.org), 2001.
- OGC. Open Geospatial Consortium, Standards Guide, ISO/TC 211 Geographic Information/ Geomatics, ISO 19111:2007 Geographic Information — Spatial Referencing by Coordinates, Chapter 1, pp 45-49, 1998.