

Universidade Federal dos Vales do Jequitinhonha e Mucuri
Faculdade de Ciências Exatas e Tecnológicas
Departamento de Computação
Bacharelado em Sistemas de Informação

Sistema de Interfaceamento para Problemas de Roteamento de Veículos Utilizando o OpenStreetMap

Alba Assis Campos

Diamantina, 21 de Fevereiro de 2014

Universidade Federal dos Vales do Jequitinhonha e Mucuri
Faculdade de Ciências Exatas e Tecnológicas
Departamento de Computação
Bacharelado em Sistemas de Informação

Sistema de Interfaceamento para Problemas de Roteamento de Veículos Utilizando o OpenStreetMap

Alba Assis Campos

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 Andrade

Diamantina, 21 de Fevereiro de 2014

Alba Assis Campos

**Sistema de Interfaceamento para Problemas de Roteamento de
Veículos Utilizando o OpenStreetMap**

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.

COMISSÃO EXAMINADORA

Alessandro Vivas Andrade (Orientador)

Universidade Federal dos Vales do Jequitinhonha e Mucuri

Rafael Santin

Universidade Federal dos Vales do Jequitinhonha e Mucuri

Luciana Pereira de Assis

Universidade Federal dos Vales do Jequitinhonha e Mucuri

Diamantina, 21 de Fevereiro de 2014

Agradecimentos

Agradeço a Deus por mais essa vitória, pelo conhecimento que adquiri e pelo crescimento que obtive ao fim desta jornada. Agradeço também aos meus pais, pelo amor e apoio nos momentos difíceis. Agradeço ao Professor Alessandro Vivas, pela atenção e empenho nas orientações deste trabalho. Agradeço aos professores e colegas do curso de Sistemas de Informação da UFVJM que contribuíram para minha formação.

Resumo

A globalização tem desencadeado uma transformação no mundo e na economia. O número de pessoas que compram pelo comércio eletrônico cresceu de maneira extraordinária, gerando um impacto na organização das empresas. Devido a isso, as empresas tem se preocupado com os gastos em relação ao transporte, pois se o processo decisório em relação ao transporte for bem planejado as empresas conseguem minimizar seus custos e maximizar seu tempo, além dos consumidores ficarem satisfeitos com a entrega no prazo. Uma das maneiras que tem contribuído para isso é com aplicação do Problema de Roteamento de Veículos.

O trabalho apresentado tem como objetivo desenvolver um sistema utilizando a linguagem de programação Java e a API OpenStreetMap para a exibição de dados. Neste caso, será desenvolvido um sistema no qual poderá aplicar-se a problemas de roteamento de veículos. O usuário fornecerá um arquivo com um conjunto de cidades e o sistema calculará a distância entre os trechos das cidades e exibirá no mapa a rota entre elas.

Palavras-Chave: Roteamento de Veículos, OpenStreetMap, Sistema de Posicionamento Global (GPS).

Abstract

The globalization initiated a transformation in the world and economy. The number of people who buy through e-commerce increased in an extraordinary way, impact on organizational structure of firms. Because of this, firms has been concerned with transport costs, whereas if decision making process in relation to transport planning has success, they could minimize their costs and maximize their time. Additionally, customers would be happy with on-time delivery. One of the ways that has contributed to this is with the implementation of the Vehicle Routing Problem.

The work presented aims system development using java programming language and OpenStreetMap API to data display. In this case, would be a system development that you can apply to the Vehicle Routing Problem. User provides a file with a group of cities and system will calculate distance between sections of cities. After that displays map route between them.

Keywords: Vehicle Routing, OpenStreetMap, Global Positioning System (GPS).

Sumário

Sumário	vii
Lista de Figuras	ix
Lista de Tabelas	x
1 Introdução	1
2 Conceitos Iniciais	4
2.1 Software Livre	4
2.2 Conceitos básicos de dados geográficos	5
2.2.1 Sistema de Coordenadas Geográficas	5
2.2.2 Dados Vetoriais	6
2.2.3 Dados Matriciais	7
2.2.4 Banco de Dados	7
2.2.5 Banco de Dados Geográfico	7
2.3 Dispositivos e Softwares	8
2.3.1 GPS	8
2.3.2 GPSBabel	8
2.3.3 Google Earth	8
2.4 Formato dos Dados	9
2.4.1 XML	9
2.4.2 KML	10
2.4.3 GPX	11
2.4.4 TCX	12
3 OpenStreetMap	14
3.1 Motivações	15
3.2 História	15

3.3	Produção dos Mapas	16
3.4	Fontes de dados públicos	17
3.5	Fontes de dados comerciais	17
3.6	Elementos básico	17
3.7	Aplicações	18
3.8	Cálculo da rota e navegação	19
3.9	Software de Mapeamento e Estrutura do Projeto	20
3.10	Formato dos Dados	21
3.11	Ferramentas de Edição	21
3.12	Futuro do Projeto OpenStreetMap	25
4	Ferramentas Utilizadas	26
4.1	Eclipse	26
4.2	Java	27
4.3	CloudMade	28
4.4	Nominatim	29
4.5	SwingX-WS	31
4.6	JXMapKit	32
4.7	JXMapView	33
5	Sistema Desenvolvido	35
5.1	Modelagem	37
5.1.1	Diagrama de Caso de Uso	37
5.1.2	Diagrama de Classes	38
5.1.3	Diagrama de Sequência	39
6	Testes Realizados	41
6.1	Execução do Sistema	41
7	Conclusão	46
	Referências Bibliográficas	47

Lista de Figuras

2.1	Representação do Ponto	6
2.2	Representação da Linha	6
2.3	Representação da Área	7
3.1	Representação Nó	18
3.2	Representação Caminho	18
3.3	Representação Relação	18
3.4	Site de Roteamento OSRM	19
3.5	Site de Roteamento Routino	20
3.6	Site do Projeto OpenStreetMap	22
3.7	Editor Potlatch	23
3.8	Editor JOSM	24
3.9	Editor Merkaartor	25
4.1	O JXMapKit: O mapa principal e outro mapa de menor dimensão na parte inferior direita	32
5.1	Diagrama de Caso de Uso	38
5.2	Diagrama de Classe	39
5.3	Diagrama de Sequência	40
6.1	Tela Principal	41
6.2	Selecionar Arquivo	42
6.3	Arquivo texto	42
6.4	Tela Principal com arquivo selecionado	43
6.5	Rota plotada	43
6.6	Rota com menos zoom	44
6.7	Rota Completa	44
6.8	Distância entre trechos de cidade	45

Lista de Tabelas

4.1	Opções de solicitações com CloudMade	29
4.2	Retorno JSON	29
4.3	Parâmetros Nominatim	30
5.1	Descrição Caso de Uso	38

Capítulo 1

Introdução

A globalização, as megafusões, as parcerias, as concentrações de varejo e o crescimento do mercado eletrônico vêm desencadeando um processo de transformação na economia, cujo impacto maior é na organização da empresa (Malaquias, 2006).

De acordo com de Oliveira (2011), os gastos relacionados com o transporte de pessoas e mercadorias recebem a cada dia maior importância nos processos decisórios das empresas, em que a minimização dos custos é um importante foco para aquelas que visam sobreviver e crescer no mercado atual. Pesquisas vêm contribuindo de maneira significativa nas últimas décadas para a solução de problemas relacionados à logística, em que o Problema de Roteamento de Veículos PRV, proposto por Dantzig and Ramser (1959), possui considerável importância.

De forma geral, o problema de Roteamento de Veículos consiste em alocar uma frota de veículos para visitar um conjunto de clientes com demandas específicas. Em outras palavras, dado um conjunto N de clientes, cada qual com uma demanda e uma frota de veículos homogênea com capacidade Q , tem-se como objetivo, estabelecer os trajetos de custo mínimo a serem percorridos pelos veículos, de forma a atender completamente a demanda dos clientes numa única visita (Mine, 2010).

Na última década, as empresas de transporte se multiplicaram com o considerável aumento das vendas no comércio eletrônico. Estima-se que, em 2012, as vendas deste setor alcançaram um montante de 335 bilhões de dólares em todo o mundo. No primeiro semestre de 2010, o comércio eletrônico brasileiro registrou faturamento de 6,7 bilhões de reais, representando 40% de crescimento em comparação com os seis primeiros meses do ano de 2009 (Braun, 2010). O custo anual de distribuição de mercadorias nos Estados Unidos foi esti-

mado em 400 bilhões de dólares. Deste valor, estima-se que 45 bilhões de dólares representam desperdícios, podendo chegar a até 30% do valor de um produto. Em um contexto regional, pesquisas indicam que serviços com transporte representam 2,53% do produto interno bruto (PIB) do Brasil (Fraga, 2007). Por motivos como estes, o PRV é um dos problemas mais estudados na área de Pesquisa Operacional (Steiner, 2000).

Como o problema de distribuição de mercadorias no mundo real se apresenta de variadas formas, há, naturalmente, uma subdivisão nos problemas de roteamento de veículos, de acordo com as características e restrições presentes nas situações reais. As mais encontradas são adição de janela de tempo em que clientes, armazém e motoristas podem trabalhar; frota de veículos mista, cuja determinação do tamanho e a composição é parte do problema, sendo atribuídos custos fixos e variantes de acordo com o veículo; entregas parciais permitidas, ou seja, mais um veículo pode atender cada cliente; múltiplos armazéns, de onde podem iniciar as rotas dos caminhões; entre outros (de Oliveira, 2011).

Além do avanço nos estudos sobre problemas baseados no PRV, houve também uma rápida evolução tecnológica, como, por exemplo, a popularização dos equipamentos de GPS (do acrônimo do inglês Global Positioning System), telefones celulares, sistemas de mapeamento geográficos. Tais avanços oferecem suporte nas decisões de um sistema de roteamento de veículos ao longo do dia de serviço (Ichoua, 2000).

No mercado existem duas interfaces de Programação de Aplicativos (API) mais utilizadas, que disponibilizam aos programadores um conjunto de funções e métodos para a manipulação de suas funcionalidades sem a necessidade de distribuição de seu código fonte. Elas são: Google Maps e OpenStreetMap.

O Google Maps disponibiliza atualmente aos seus utilizadores uma API denominada Google Maps API V3, esta biblioteca contém todas as funções necessárias para incorporar um mapa a uma página da web. Como o código da API não é carregado na aplicação do usuário se faz necessário o uso de referências, para acessar este serviço o Google disponibiliza a geração de uma chave de utilização. A chave gerada é única para o domínio onde está hospedado a aplicação e deve ser obtido por inscrição online. Depois de gerada a chave deve ser inserida no código fonte da aplicação, este é o único modo de utilizar a API do Google Maps atualmente. O utilizador da API do Google necessita principalmente de conhecimentos em Javascript e programação orientada a objetos para utilizar o máximo das funcionalidades ofe-

recidas pela ferramenta. O Google disponibiliza uma documentação completa e detalhada para os desenvolvedores mostrando e ilustrando passo a passo as funcionalidades desta ferramenta.

O projeto OpenStreetMap é uma poderosa e atualizada fonte de dados de geoprocessamento, por outro lado, empresas como CloudMade adotam estas informações para criação de API's a fim de disponibilizar rotinas de fácil utilização para os seus usuários. As API's possibilitam aos seus usuários a personalização de mapas em um nível de detalhamento que nem mesmo o Google Maps API atinge na em sua atual versão V3.

Diante do que foi exposto, o objetivo do trabalho é desenvolver um sistema utilizando a linguagem de programação Java e a API OpenStreetMap para a exibição de dados. Neste caso, será desenvolvido um sistema que poderá ser aplicado em problemas de roteamento de veículos. O usuário fornecerá um arquivo com um conjunto de cidades e o sistema calculará a distância entre os trechos das cidades e exibirá no mapa a rota entre elas.

Este trabalho está dividido em seis capítulos. O capítulo 1 especifica alguns conceitos básicos sobre as tecnologias envolvidas no desenvolvimento deste trabalho. O capítulo 2 apresentará sobre a ferramenta OpenStreetMap, a qual é a base para o desenvolvimento do sistema. O capítulo 3 apresenta o conceito das ferramentas utilizadas no desenvolvimento do sistema. O capítulo 4 mostra o funcionamento do sistema desenvolvido e diagramas de caso de uso, classe e sequência. O capítulo 5 apresenta os testes realizados. Finalmente, no capítulo 6 são discutidas as conclusões, limitações e propostas para trabalhos futuros.

Capítulo 2

Conceitos Iniciais

Este capítulo apresenta alguns dos conceitos principais que serão abordados neste trabalho e que são importantes para compreender o objetivo do mesmo.

O capítulo inicia-se com a definição de software livre, pois o sistema desenvolvido faz uso de informações geográficas gratuitas disponibilizadas pelo projeto OpenStreetMap. Em um segundo tópico, serão abordados os conceitos de básico de dados geográficos, como: latitude, longitude, dados vetoriais e matriciais, banco de dados e banco de dados geográficos. Em seguida serão abordados sobre os dispositivos de captura de dados geográficos e aplicativos para a sua manipulação. E finalmente serão definidos os formatos dos dados geográficos.

2.1 Software Livre

Segundo a GNU (2014), "o projeto GNU foi lançado em 1983 por Richard Stallman, com o objetivo de criar um sistema operacional baseado em software livre". O sistema Operacional seria compatível com Unix, porque seu design geral já era testado e portátil, além de facilitar a migração dos usuários do Unix para o GNU por ser compatível os sistemas operacionais.

Em 1985, foi fundada a Free Software Foundation, uma empresa que inicialmente iria levantar fundos para ajudar a desenvolver o GNU.

Por volta de 1990, o projeto GNU já havia desenvolvido todos os componentes principais, exceto o kernel. Então o Linux, um kernel do tipo Unix, foi desenvolvido por Linus

Torvalds em 1991 e transformado em software livre em 1992. Assim, foi desenvolvido um sistema operacional completo chamado GNU/Linux.

De acordo com GNU (2014), um software para ser considerado livre deve garantir a qualquer usuário a premissa de: (i) executar o programa para qualquer propósito, (ii) a qualquer momento modificar o software para atender as novas necessidade, (iii) redistribuir livremente cópias originais, (iv) distribuir livremente cópias de suas versões modificadas.

O software livre é bastante distinto dos softwares proprietários, pois, segundo Saleh (2004), " No software proprietário em geral a única liberdade garantida ao usuário é a de usar o programa, mesmo assim somente após o seu licenciamento e normalmente com o compromisso de não redistribuí-lo e nem de modificá-lo ". Portanto, um software livre não é sinônimo de software grátis. O software apenas é considerado livre se atender os quatros requisitos citados acima.

2.2 Conceitos básicos de dados geográficos

2.2.1 Sistema de Coordenadas Geográficas

Existem vários sistemas de coordenadas que permitem a localização precisa de um ponto qualquer na superfície terrestres, porém a mais usual é a Coordenadas Geográficas (latitude e longitude).

A latitude pode ser definida como a distância em graus de qualquer ponto da Terra em relação à linha do equador. É também chamada de paralelo por se tratar de linhas imaginárias traçadas paralelamente ao equador. Os principais paralelos são: o círculo polar ártico, o círculo polar antártico, o trópico de câncer e o trópico de capricórnio. As latitudes variam em 180° , sendo contadas a partir da Linha do Equador, que é a latitude 0° , responsável por dividir o planeta nos hemisférios Norte (Boreal ou Setentrional) e Sul (Austral ou Meridional).

A longitude pode ser definida como a distância em graus de qualquer ponto da Terra em relação ao Meridiano de Greenwich. As longitudes também são chamadas de meridianos. As longitudes variam entre 0° e 360° , sendo contadas a partir do Meridiano de Greenwich, que é responsável por dividir o planeta nos hemisférios Leste (Oriental) e Oeste (Occidental).

Apesar de as longitudes e latitudes serem traçadas cartograficamente a cada 15°, qualquer ponto da superfície terrestre possui latitude e longitude específicas. No âmbito das representações gráficas, o único ponto de referência utilizado são as coordenadas geográficas traçadas a partir dessas linhas.

2.2.2 Dados Vetoriais

Segundo a DPI (2014) "as estruturas vetoriais são utilizadas para representar as coordenadas das fronteiras de cada entidade geográfica, através de três formas básicas: pontos, linhas, e áreas (ou polígonos)".

Ponto: Pode ser definido por um par ordenado xy . Ele representa uma localização ou uma ocorrência no espaço. A Figura 2.1 representa o ponto.

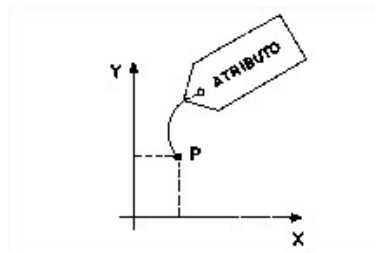


Figura 2.1: Representação do Ponto

Linha: Definida por no mínimo dois pares de par ordenados xy (dois pontos). Pode representar uma rodovia, um rio. A Figura 2.2 representa a linha.

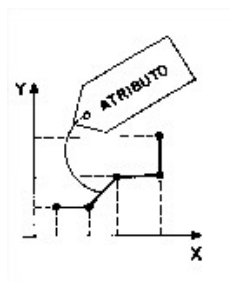


Figura 2.2: Representação da Linha

Área: Definida como um conjunto de coordenadas (x,y) formando segmentos de retas que fecham uma área. Pode representar um lago, uma determinada área. A Figura 2.3 representa a área.

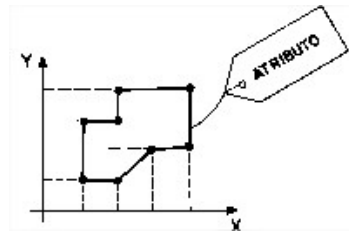


Figura 2.3: Representação da Área

2.2.3 Dados Matriciais

É uma estrutura de dados geográficos que representa uma partição do espaço em células (frequentemente designadas também por pixels) de igual dimensão. Cada célula armazena um valor que indica o tipo de objeto ou condição que é encontrada naquela localização, o que é válido para toda a célula, independente de sua dimensão física.

O espaço é ocupado por um grande número de células distribuídas regularmente, cada uma das quais pode ter um valor diferente. A área que cada célula representa define a resolução espacial, que é inversamente proporcional ao tamanho da célula. Quanto maior a célula, menor a resolução espacial.

Ao contrário do modelo vetorial, as entidades no modelo matricial não correspondem às entidades espaciais que representam no mundo real. Isto porque as entidades espaciais no modelo matricial, são as células individuais.

2.2.4 Banco de Dados

O sistema de Banco de Dados pode ser definido como um sistema computadorizado de manutenção de registros. O banco de dados, por si só, pode ser considerado como um repositório que armazena uma coleção de arquivos computadorizados e permite aos usuários do sistema realizar diversas ações como: inserir, alterar, excluir e buscar dados em arquivos existentes (Date, 2004).

2.2.5 Banco de Dados Geográfico

O Banco de Dados Geográfico (BDG) são coleções de dados georeferenciados, manipulados por Sistemas de Informação Geográficas (SIG). O Sistema de Informação Geográficas são sistemas computacionais capazes de capturar, armazenar, recuperar, manipular, analisar e apresentar dados geográficos (Gazola and Furtado, 2007).

2.3 Dispositivos e Softwares

2.3.1 GPS

Global Positioning System (GPS), primeiro sistema de navegação por satélite do mundo. Foi desenvolvido pelo Departamento de Defesa do governo dos EUA em 1973.

O GPS é um sistema de navegação baseado em um conjunto de 24 satélites operacionais (há outros, sobressalentes), que têm órbitas com raio aproximado de 20 mil quilômetros e que completam aproximadamente duas voltas ao redor da Terra por dia. As órbitas são arranjadas de modo a possibilitar que, em um ponto qualquer da superfície terrestre, pelo menos quatro satélites estejam "visíveis", ou seja, acima da linha do horizonte. Cada satélite transmite continuamente um sinal de rádio que inclui sua identificação, informações de sua órbita e o instante de tempo (horário) em que a transmissão foi feita, com precisão de um bilionésimo de segundo. Quando o sinal chega a um receptor GPS, este com base nas informações do sinal, calcula a que distância do satélite ele se encontra. É necessário receber o sinal de no mínimo três satélites para que o receptor determine a longitude e latitude (Canto, 2010).

2.3.2 GPSTabel

O GPSTabel é um software livre e está disponível para download no site <https://www.gpsbabel.org/>. Este sistema é multiplataforma, ou seja, é possível instalá-lo no Windows, Mac OS X e Linux. Ele é responsável em converter dados de GPS em outros formatos de dados (Bennett, 2010).

O exemplo a seguir mostra a conversão de um arquivo NMEA para GPX.

```
1 gpsbabel -w -t -i nmea -f<input
2 filename> -o gpx -F <output
3 filename>
```

2.3.3 Google Earth

O software Google Earth é um sistema multiplataforma e possui a função de apresentar um modelo tridimensional do globo terrestre. Trata-se de um software gratuito que faz uso de imagens obtidas por satélite e fotografias aéreas em um modelo tridimensional da Terra. Dados geográficos do usuário podem ser representados facilmente no Google Earth através do uso da documentação da linguagem KML (Keyhole Markup Language). Estes documentos podem ser

utilizados para mostrar os pontos, os caminhos, polígonos e superposições de solo. O software permite rotacionar a superfície de uma região, marcar locais que se consegue identificar para visitá-los posteriormente, medir a distância entre dois pontos e até mesmo ter uma visão tridimensional de uma determinada localidade (Lorbiesk, 2010).

2.4 Formato dos Dados

2.4.1 XML

Segundo Medina (2014) "Extensible Markup Language (XML), é um padrão para a formatação de dados, ou seja, uma maneira de organizar informações. Os documentos XML podem ser facilmente compreendidos por programadores facilitando o desenvolvimento de aplicativos compatíveis. Todas as informações contidas no XML estão dentro de tags".

Ele foi criado pelo World Web Consortium para trazer para a Web uma forma simples de superar as limitações do HTML e permitir novos tipos de aplicações para a Internet. O XML é um padrão de armazenamento de dados em um formato de texto simples, isto significa que ele também pode ser aberto em qualquer computador (Medina, 2014).

O código abaixo representa um exemplo de uma receita de pão em XML.

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <receita nome = "pão" tempo-de-preparo = "5 minutos" tempo-de-
   cozimento = "1 hora" >
3 <titulo>Pão simples</titulo>
4 <ingredientes>
5 <ingrediente quantidade = "3" unidade = "xícaras" > Farinha</
   ingrediente>
6 <ingrediente quantidade = "7" unidade = "gramas" > Fermento</
   ingrediente>
7 <ingrediente quantidade = "1.5" unidade = "xícaras" estado = "
   morna" > Água</ingrediente>
8 <ingrediente quantidade = "1" unidade = "colheres de chá" >
   Sal</ingrediente>
9 </ingredientes>
10 <instrucoes>
11 <passo>Misture todos os ingredientes, e dissolva bem.</passo>
```

```
12     <passo>Cubra com um pano e deixe por uma hora em um local
13         morno.</passo>
14     <passo>Misture novamente, coloque numa bandeja e asse num
15         forno.</passo>
16 </instrucoes>
17 </receita>
```

2.4.2 KML

KML é uma gramática especializada da linguagem de marcação XML (Extensible Markup Language) 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 geográficos que podem ser representados utilizando a KML estão pontos, linhas e polígonos (de Jesus Júnior and Marozzi, 2014).

Exemplo de código KML, onde se pode observar que é composto neste caso por um marcador (placemark) que tem um nome (name), uma descrição (description) e neste caso um ponto (point) com as suas coordenadas.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://www.opengis.net/kml/2.2">
3   <Placemark>
4     <name>
5       casa1
6     </name>
7     <description>
8       Casa de campo
9     </description>
10    <Point>
11      <coordinates>
12        -8.7724220811056,41.7689908978,0
13      </coordinates>
14    </Point>
15  </Placemark>
16 </kml>
```

2.4.3 GPX

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

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.

O código abaixo representa um exemplo de um arquivo GPX, no código fonte é possível visualizar a latitude, a longitude, a elevação, a data e a hora que o ponto foi registrado.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <gpx xmlns="http://www.topografix.com/GPX/1/1" creator="GPS
   TrackMaker "
3 version="1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
   "
4 xsi:schemaLocation="http://www.topografix.com/GPX/1/1
5 http://www.topografix.com/GPX/1/1/gpx.xsd">
6   <metadata>(?)</metadata>
7   <trk>
8     <name>
9       ACTIVE LOG
10    </name>
11   <trkseg>
12     <trkpt lat="41.399574289" lon="-8.667590563">
13       <ele>
14         54.451780
15       </ele>
16     <time>
```

```

17 2006-02-08T14:21:43Z
18 </time>
19 </trkpt>

```

2.4.4 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 série 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).

O código abaixo representa um exemplo de um arquivo TCX.

```

1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <TrainingCenterDatabase xmlns="http://www.garmin.com/xmlschemas/
   TrainingCenterDatabase/v2" xmlns:xsi="http://www.w3.org/2001/
   XMLSchema-instance" xsi:schemaLocation="http://www.garmin.com/
   xmlschemas/TrainingCenterDatabase/v2 http://www.garmin.com/
   xmlschemas/TrainingCenterDatabasev2.xsd">
3   <Activities>
4     <Activity Sport="Running">
5       <Id>2007-08-18T21:45:45Z</Id>
6       <Lap StartTime="2007-08-18T21:45:45Z">
7         <TotalTimeSeconds>6343.400000</TotalTimeSeconds>
8         <DistanceMeters>22297.0312500</DistanceMeters>
9         <Calories>1098</Calories>
10        <Intensity>Active</Intensity>
11        <TriggerMethod>Manual</TriggerMethod>
12      </Lap>
13      <Training VirtualPartner="true">
14        <Plan IntervalWorkout="false" Type="Course">
15          <Name>00 Paradise Loo</Name>
16        </Plan>
17      </Training>
18      <Creator xsi:type="Device_t">
19        <Name>Forerunner305</Name>

```

```
20     <UnitId>3332601296</UnitId>
21     <ProductID>484</ProductID>
22     <Version>
23         <VersionMajor>2</VersionMajor>
24         <VersionMinor>60</VersionMinor>
25         <BuildMajor>0</BuildMajor>
26         <BuildMinor>0</BuildMinor>
27     </Version>
28 </Creator>
29 </Activity>
```

Capítulo 3

OpenStreetMap

Segundo [Bennett \(2010\)](#), "o OpenStreetMap, também conhecido por OSM, é um projeto colaborativo para construção de banco de dados geográfico livre". Apesar do projeto ter iniciado apenas com mapeamento de estradas, hoje em dia ele já está bem além, mapeando praias, cachoeiras, construções, entre outras características geográficas. Seu objetivo principal é ter um registro de cada característica geográfica do planeta. Membros desse projeto também produzem vários softwares que podem criar, editar, manipular ou usar, os dados que foram coletados, para alguma finalidade. O projeto OpenStreetMap permite criar mapas gratuitos e editáveis de forma colaborativa.

Em janeiro de 2013, o projeto atingiu cerca de 1.000.000 usuários cadastrados, em fevereiro de 2011, foi constatado aproximadamente 12 mil edições mensais no banco de dados. O número de usuários cresce cerca de 10% ao mês. Países como Alemanha e Reino Unido são as maiores comunidades locais, mas há grupos de usuários em todo o mundo ([Bennett, 2010](#)).

Os mapas são criados usando informações geográficas enviadas pelos usuários registrados, que são conhecidos internamente como "mappers", através de dispositivos GPS. Este mapeamento, tanto as imagens criadas como os dados vetoriais são armazenados em seu banco de dados, perante uma licença de código aberto, permitindo qualquer pessoa copiar, alterar e redistribuir os dados.

Toda semana são adicionados 90 mil quilômetros de novas estradas, com um total de quase 24 milhões de quilômetros de estrada, sem contar com outros tipos de dados. O tamanho do banco de dados levantado em fevereiro de 2011 foi acima de 205 Gigabytes, aumentando diariamente cerca de 10 Megabytes de dados compactados.

3.1 Motivações

A grande motivação em utilizar o OpenStreetMap é devido ser gratuito. O OpenstreetMap praticamente não tem restrição do que pode ser feito com seus dados. É possível usá-lo para qualquer propósito, incluindo para atividades comerciais, sem ter que pagar taxas de licenças. O OpenStreetMap também permite o próprio usuário corrigir bugs e adicionar novos dados.

3.2 História

Em julho de 2004, o inglês Steve Coast começou o projeto OpenStreetMap em resposta aos altos preços cobrados pela Ordnance Survey, agência de mapeamento da Grã-Bretanha.

Em agosto de 2006, o OpenStreetMap começou o processo para se tornar uma fundação visando possuir a infraestrutura necessária para executar o projeto de aceitar doações. Em dezembro, a empresa Yahoo! confirmou que OpenStreetMap poderia usar as suas fotografias aéreas como base para a produção de mapas.

Em abril de 2007, a empresa holandesa Automotive Navigation Data, doou um conjunto completo de dados dos países baixos e as principais estradas da Índia e da China para o projeto. Em julho do mesmo ano, durante a primeira conferência internacional sobre o OpenStreetMap "The State of the Map", realizada em Manchester, havia 10.000 usuários registrados no site OpenStreetMap. Em agosto, foi lançado um projeto separado chamado OpenAerialMap visando a obtenção de um banco de dados de fotografias aéreas disponíveis para a sua distribuição sob uma licença livre, e em dezembro, a Universidade de Oxford tornou-se a primeira grande instituição a utilizar os dados do OpenStreetMap em seu site.

Em janeiro de 2008, o projeto adquiriu uma nova funcionalidade. Através de dispositivos GPS era possível o mapeamento OpenStreetMap voltado para o ciclismo.

Em fevereiro, realizaram-se oficinas de capacitação para a produção cartográfica de dados OpenStreetMap na Índia.

Em março, O OpenStreetMap Foundation anunciou que receberam financiamento de 2,4 milhões de euros da CloudMade, uma empresa comercial que iria utilizar os dados do OpenStreetMap. Em julho, durante a conferência anual State of the Map 2008, em Limerick, observou-se que o projeto OpenStreetMap havia atingido 45 mil usuários cadastrados.

Em janeiro de 2009, a agência cadastral francesa permitiu o uso de OpenStreetMap no Web Mapping Service (WMS) para dados geográficos e vetorização.

Em abril de 2009, o escritório alemão da Wikimedia anunciou aos desenvolvedores na sua reunião realizada em Berlim, o financiamento de 15.000 euros para um projeto piloto de integrar o OpenStreetMap com a Wikipédia e montar um servidor para as ferramentas que atuarão sobre os dados do OpenStreetMap. Atualmente os artigos da Wikipédia são georreferenciados em diferentes idiomas e implementados com o mapeamento do OpenStreetMap com marcadores no local referido no texto, bem como imagens com geotags do Wikimedia Commons.

Em 2012, o Google mudou sua política de uso de produto do Google Maps, passaram a cobrar pelo uso da API para sites que geram tráfego. Este fato fez com que empresas de referência como o Foursquare deixassem o Google Maps para utilizar dados do OpenStreetMap. Além disso, a Apple lançou uma nova versão do software iPhoto, que poderia ser visto como geolocalização de fotos que é baseada em mapas do OpenStreetMap.

3.3 Produção dos Mapas

De acordo com Bennett (2010), "os primeiros dados do projeto foram compilados a partir do zero, por voluntários através de um trabalho de campo. Foram utilizados dispositivos GPS, notebooks e gravadores de voz. Estas informações foram incorporadas ao banco de dados do OpenStreetMap".

Mais recentemente, a disponibilidade de fotografias aéreas e outras fontes de dados públicos e comerciais aumentaram consideravelmente a velocidade do trabalho, permitindo a coleta de informações com uma maior precisão.

A coleta de informações no campo é feita por voluntários, que consideram a contribuição para o projeto um hobby viciante. Aproveitando-se das suas viagens a pé, de bicicleta ou de carro e de usar um dispositivo GPS, eles capturam traçados e pontos de interesse, aproveitando também para gravar as informações associadas a esses traçados ou pontos de interesse através de bloco de notas, gravador de voz ou uma câmera digital. Eles também tendem a interrogar os moradores para o seu conhecimento local das especificidades locais que são desconhecidas. Mais tarde, no computador esta informação é enviada para o projeto de banco de dados comum (Bennett, 2010).

Além destes levantamentos de informações, o projeto baseia-se principalmente no grande número de pequenas edições feitas pela maioria dos contribuintes, como: corrigir os erros e acrescentar novos dados para o mapa.

3.4 Fontes de dados públicos

A existência ou a liberação de dados públicos das instituições governamentais com o tipo de licença compatível com o OpenStreetMap tem permitido a importação de informações geográficas no banco de dados do projeto. Assim, a maioria das informações relativas aos Estados Unidos vem dessas fontes, onde as leis exigem que o governo federal tornem os dados públicos.

Várias autoridades locais também têm lançado suas fotografias aéreas, tornando-as acessíveis ao público através do OpenAerialMap.

Na Espanha, o Instituto Geográfico Nacional, um órgão público responsável pela criação, manutenção e comercialização de mapas oficiais do país, modificou em abril de 2008, a licença de uso dos seus dados, liberando parte desses gratuitamente para qualquer tipo de utilização.

3.5 Fontes de dados comerciais

Algumas empresas doaram dados para o projeto sob licenças apropriadas para o uso. Em particular, os dados da empresa holandesa Automotive. Os dados de navegação, que foram doados possuem a cobertura total para os países baixos e as principais estradas da China e da Índia. Além disso, Yahoo! e Bing permitiram ao OpenStreetMap fazer uso de suas fotografias aéreas. Elas estão disponíveis para serem utilizadas como base de fotografia em que, para "desenhar". Elas são utilizadas em diferentes softwares de edição.

Os colaboradores podem criar seus mapas vetoriais utilizando as fotografias aéreas como imagem de fundo, mas em uma licença livre e aberta.

3.6 Elementos básico

Os elementos básico de mapeamento dos OpenStreetMap são:

- Nós: São pontos no espaço. Eles são elementos primitivos para ter informações sobre a posição e todos os outros elementos dependem de nós para a sua localização. Um nó pode ser usado em si mesmo para mapear um ponto de interesse, tal como uma junção entre duas formas, ou apenas como uma mudança de direção em uma forma. Cada nó tem a sua latitude e longitude e é representado por um ponto.

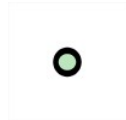


Figura 3.1: Representação Nó

- Caminhos: São uma lista ordenada de nós. Eles podem descrever características lineares tais como estradas, vias fluviais. E também podem ser fechados de modo a formar as áreas.

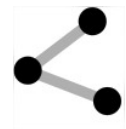


Figura 3.2: Representação Caminho

- Relações. Estes são grupos de nós, caminhos e outras relações para o qual é possível atribuir certas propriedades comuns. Por exemplo, ramificações de ruas, rotas de longas distâncias.



Figura 3.3: Representação Relação

- Tags. Podem ser atribuída a nós, caminhos ou relações, consistem em uma chave e um valor. Qualquer elemento básico pode ter um número arbitrário de tags associado a ele. Por exemplo: `maxspeed = 110`.

3.7 Aplicações

Os dados do projeto OpenStreetMap não só podem mapear estrada, mas também trilhas, mapas de ciclismo, mapas náuticos, mapas de esqui, etc. Eles também são usados em apli-

cações para calcular as melhores rotas para veículos e pedestres. Devido a licença aberta, o OpenStreetMap permite o desenvolvimento de outras aplicações gratuitamente.

Existem diferentes serviços em sites que usam dados do OpenStreetMap para exibir mapeamento on-line por meio de diferentes estilos de renderização e visualização. Estes serviços vão desde as próprias aplicações de mapeamento web que são disponibilizadas pela Fundação OpenStreetMap, como também as oferecidas por outras empresas, por exemplo, a que é oferecida pelo Yahoo para cidades como Bagdá, Pequim, Sydney e Tóquio.

3.8 Cálculo da rota e navegação

O cálculo de rotas otimizadas utilizando dados do OpenStreetMap não está totalmente desenvolvido, mas o progresso nessa direção nos últimos anos tem sido muito importante. Em muitas partes dos dados existentes atualmente ainda não está detalhado o suficiente para que eles se tornem totalmente confiáveis, pois muitas vezes não há informações sobre os nomes das ruas por exemplo. Além disso, pode haver problemas de consistência topológica com erros de digitalização involuntárias ou rotulagem como: cruzamentos sem nó comum. O projeto OpenStreetMap possui vários usuários para fornecer ferramentas revisoras para detectar e corrigir a maioria desses problemas (Bennett, 2010).

Existem diversos sites que oferecem serviços de roteamento baseados em dados do OpenStreetMap através de algoritmos conhecidos. Como por exemplo:

- Open Source Routing Machine (OSRM)

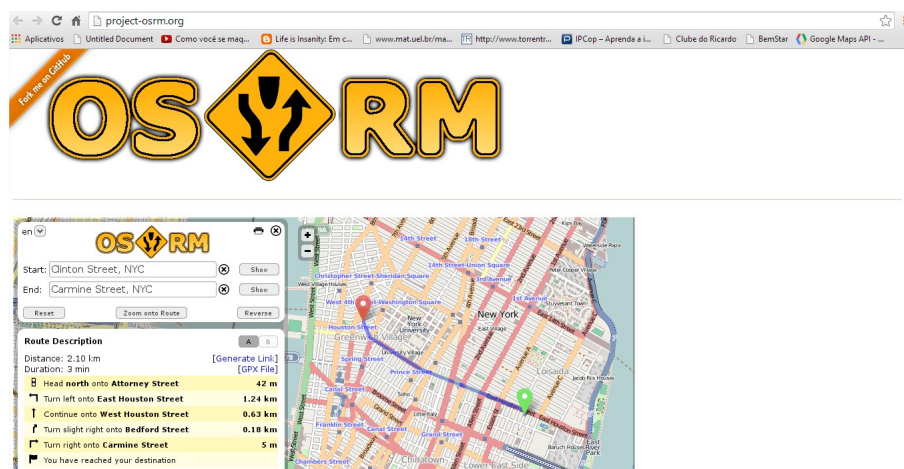


Figura 3.4: Site de Roteamento OSRM

- Routino

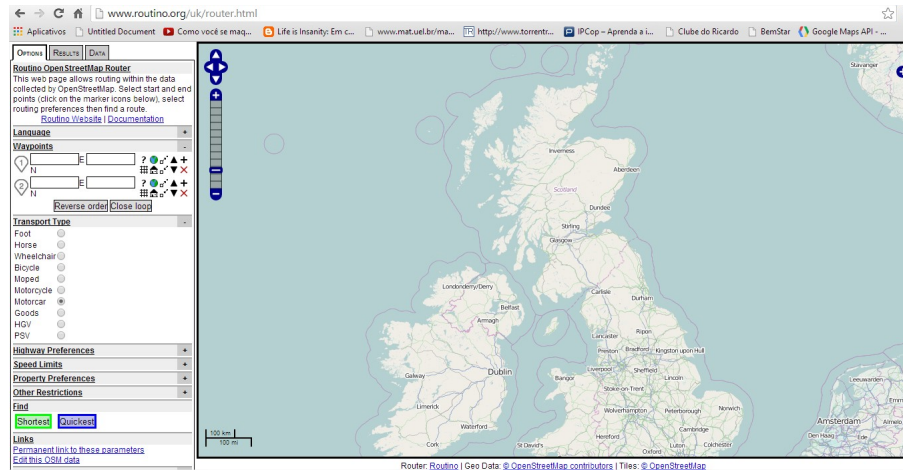


Figura 3.5: Site de Roteamento Routino

Na maioria dos casos, estas implementações não necessariamente lidam com o caminho mais curto, mas sim com a variável em questão, dependendo do problema considerado. Existem algumas experiências na análise de mapeamento otimizado com OpenStreetMap em rotas multimodal. Existem também inúmeras aplicações para dispositivos móveis projetados para calcular caminhos ideais, cujo a lista pode ser encontrada no site do projeto.

Alguns membros da comunidade também fornecem mapas gerados a partir de dados do OpenStreetMap Garmin. Os vários modelos de GPS da marca Garmin podem usar esses recursos de mapeamento para o cálculo da rota. Estes mapas podem ser criados com a ajuda do programa mkgmap, que permite a qualquer usuário compor os seus próprios mapas personalizados para Garmin a partir de dados do OpenStreetMap (Garmin, 2010).

3.9 Software de Mapeamento e Estrutura do Projeto

OpenStreetMap fornece os dados brutos para download a partir do seu próprio site. Estes podem ser modificados para cada projeto e apresentado com estilos de renderização personalizados.

A pequena estrutura formal do OpenStreetMap reflete a natureza colaborativa do projeto. Os usuários possuem o mesmo nível de acesso para adicionar conteúdo. Há apenas administradores de sistema que cuidam da infraestrutura do projeto.

A OpenStreetMap Foundation é uma fundação sem fins lucrativos que mantém o projeto OpenStreetMap. Este projeto não possui empregados remunerados e não determina o que deve ser feito, ou quando deve ser feito, a fundação apenas regula os grupos de trabalhos que planejam desenvolvimento de projetos a longo prazo. A única intervenção da fundação é quando usuários individuais põem em risco o projeto.

Os principais servidores e a infraestrutura associada ao projeto OpenStreetMap Foundation estão alojados principalmente na University College London e outros geograficamente distribuídos em todos os continentes. Todos dedicados à hospedagem de banco de dados, aplicativos para sites, aplicativos para a API e mapa de renderização. Há outros servidores de menor desempenho para os serviços auxiliares, tais como o projeto wiki, a busca repositório e as funções do mapa (Bennett, 2010).

3.10 Formato dos Dados

OpenStreetMap apenas aceita traçados em formato GPX (GPS Exchange), que é um XML que lida com dados geográficos e waypoints. Caso o aparelho de GPS não crie os dados em arquivo GPX, será necessário convertê-los.

Para converter os dados em um formato adequado para o projeto pode ser usados programas de conversão como GPSTabel.

A cada semana, está disponível uma cópia atualizada do banco de dados chamado planet.osm.

3.11 Ferramentas de Edição

Para contribuir com dados de mapeamento no OpenStreetMap é preciso registrar-se gratuitamente no site <http://www.openstreetmap.org/>, usando um endereço de e-mail válido. Agora se o usuário não estiver planejando fazer qualquer contribuição de mapeamento e está apenas interessado em baixar os dados não é necessário registrar-se. A Figura 3.6 mostra a página inicial do site OpenStreetMap.

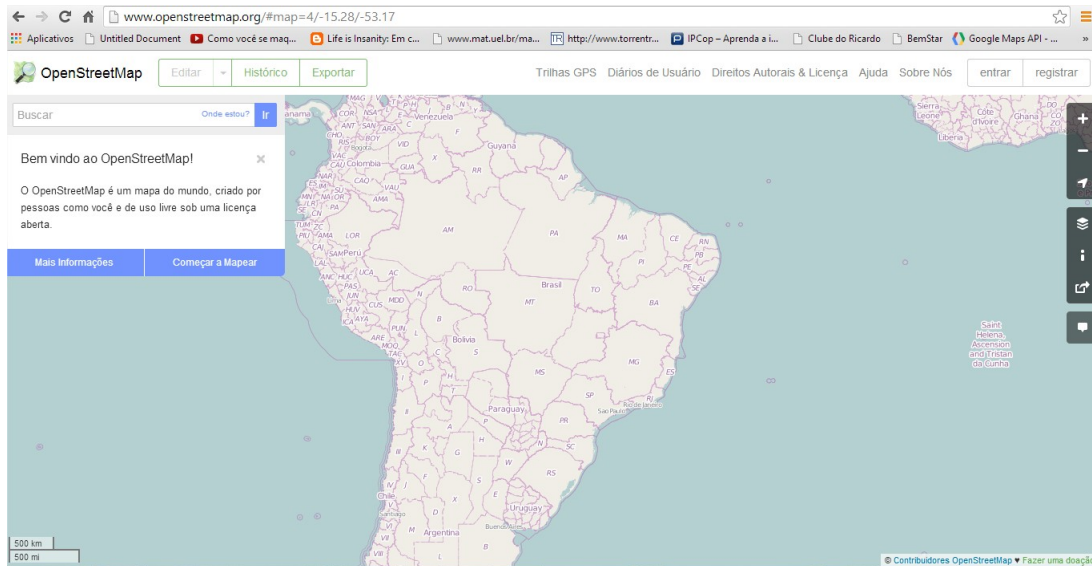


Figura 3.6: Site do Projeto OpenStreetMap

Os dados geográficos que os contribuintes capturam com os seus dispositivos de GPS servem como um guia para desenhar no mapa um caminho. Ao utilizar o GPS ele grava a latitude e longitude ao longo do tempo, gerando um ponto no mapa. Quando o colaborador se movimenta com o GPS esse ponto vira uma linha, criando um dado geográfico. Ele é utilizado para mapear um caminho, ou uma rua. Em seguida, esses dados geográficos são enviados ao servidor a partir do computador local do colaborador. Ele também podem baixar do servidor OpenStreetMap os dados geográficos na área que ele irá editar e que foram enviados por outros contribuintes do OpenStreetMap. Isso permite que erros sejam corrigidos ou tenha um aumento na precisão, já que com mais dados é possível delimitar o melhor caminho.

Paralelamente a estes dados brutos em forma de GPS, podem ser utilizadas ferramentas de edição que permitem o download de fotografias aéreas e imagens de satélite livres, que podem ser acrescentados ao mapa. Assim que temos a informação geográfica básica, podemos adicionar itens no mapa representados por nó.

Para editar os dados é preciso de um editor. Para tal, existem diferentes opções:

Potlatch: É um editor online baseado em flash para o OpenStreetMap. Sua vantagem sobre os outros editores é que ele não precisa ser instalado. Ou seja, é possível usá-lo a partir de qualquer computador que tenha navegador com plugin Flash. Para utilizar o editor Potlatch é necessário que o usuário esteja logado no openstreetmap.org e acessar a guia editar. O editor

Potlatch tem menos recursos do que os editores de desktop, mas ainda é capaz de criar e editar os dados como qualquer outro editor (Bennett, 2010).

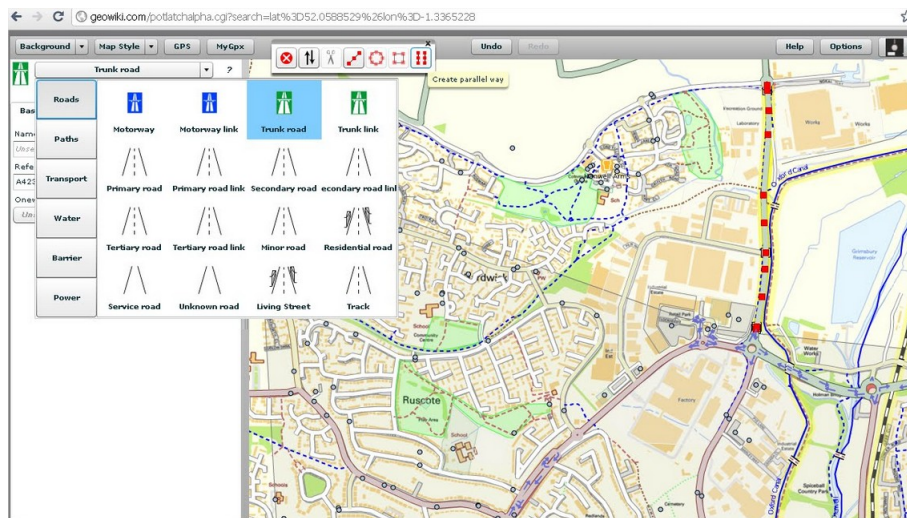


Figura 3.7: Editor Potlatch

O OpenStreetMap Editor Java (JOSM): é quase universalmente conhecido por suas iniciais, é um aplicativo de edição de desktop escrito em Java. Como tal ele roda em Mac OS, Linux e Windows. O projeto tem seu próprio site website em [HTTP://josm.openstreetmap.de/](http://josm.openstreetmap.de/) onde é possível baixar a versão mais recente do software, porém é necessário ter o Java instalado no computador. Ele tem mais recursos embutidos, como áudio e suporte de mapeamento fotográfico, que ajudam a transformar o levantamento de informações em dados do mapa. Ele também suporta plugins que adicionam informações extras, como a coleta direta de traçados do GPS em tempo real e ligações a site de terceiros. Em geral, há duas versões do JOSM disponível: a versão testada e a versão mais recente. A versão testada é mais estável e é o que o usuário deve usar caso tenha uma necessidade particular de ter algumas características de ponta. A última versão é mais recente, mas provavelmente irá conter bugs e pode até mesmo travar enquanto o colaborador está trabalhando, ocasionando a perda de todo o trabalho. Ambos são distribuídos como arquivos JAR executáveis (Bennett, 2010).

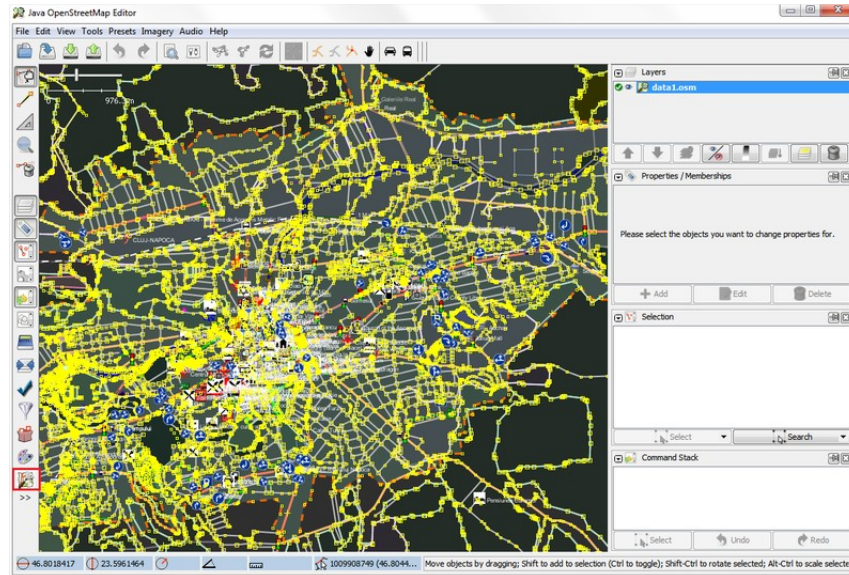


Figura 3.8: Editor JOSM

Merkaartor é outra aplicação desktop de edição. Implementada em C++, está disponível para Windows, várias versões de Linux e uma versão para o Mac OS. É um aplicativo nativo, o que implica uma necessidade de instalá-lo antes de usar. O editor é muito mais leve na aparência e na utilização de um sistema GIS tradicional em relação a outros editores OpenStreetMap. Ele tem um renderização de alta qualidade, várias camadas, gerenciamento sofisticado de alterações de mapas e ferramentas de desenho. No entanto, não é extensível da mesma maneira como o JOSM, e algumas características, tais como mapeamento de áudio não estão incluídos. Junto com a edição, é possível utilizar o Merkaartor para processar mapas, incluindo a personalização de estilos (Bennett, 2010).

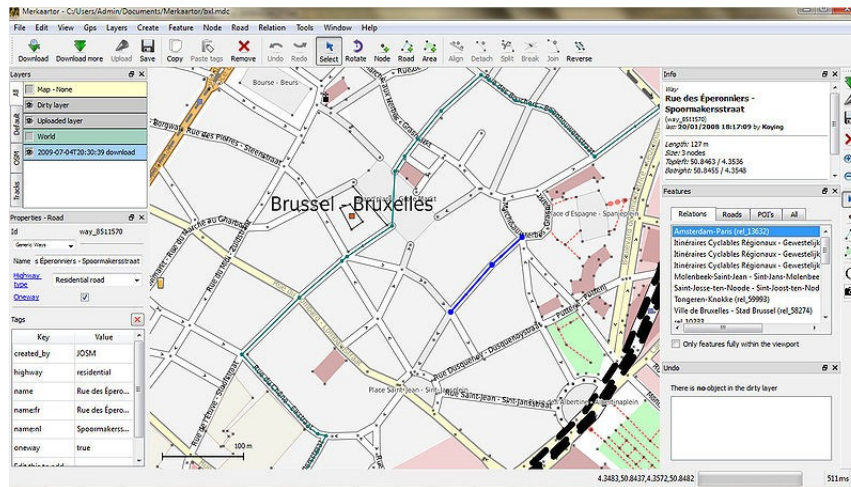


Figura 3.9: Editor Merkaartor

Estes são os principais editores. Além disso, há outros de terceiros desenvolvido como Mapzen, Iloee para iPhone, Android Vespucci, extensões para GIS e ArcGIS QGIS, etc.

3.12 Futuro do Projeto OpenStreetMap

O OpenStreetMap alcançou muito em um curto espaço de tempo, mas ainda há muito o que melhorar. A tarefa mais importante do projeto é aumentar a quantidade e precisão dos dados, e para isso, são necessárias melhorias nas ferramentas utilizadas no sentido de facilitar a colaboração.

Atualmente não há projetos em andamento para fazer grandes mudanças no modelo de dados do OpenStreetMap ou em suas APIs. As mudanças nos softwares de edição seguem em uma taxa constante.

A principal mudança esperada para o OpenStreetMap é jurídica ou legal. Essa mudança se faz necessária para que haja uma maior interação entre o OpenStreetMap e o banco de dados geográficos proprietários.

Como os dados do OpenStreetMap são simples e flexíveis, uma abordagem proposta é aplicativos de edições diversificadas. Cada tipo com suas diferentes funcionalidades voltadas para um grupo de colaboradores, como os ciclistas, caminhoneiros ou taxistas. Isso poderia possibilitar uma maior facilidade aos diferentes grupos de colaboradores ao ficarem mais simples as aplicações e com uma melhor usabilidade.

Capítulo 4

Ferramentas Utilizadas

Este capítulo descreve sobre as ferramentas de desenvolvimento utilizadas para a construção do sistema. Os tópicos abordados serão sobre as ferramentas Eclipse, Java, CloudMade, Nominatim, SwingX- WS, JXMapKit e JXMapView.

4.1 Eclipse

O Eclipse é um IDE (Integrated Development Environment), ambiente de desenvolvimento, que foi criado pela empresa americana IBM (International Business Machines), em novembro de 2001. A empresa investiu cerca de 40 milhões de dólares para o desenvolvimento de um IDE de código aberto (open source) e de grande usabilidade. Ao longo do tempo, o Eclipse tornou-se um dos IDEs mais utilizados no mundo, especialmente por utilizar SWT (The Standard Widget Toolkit) como biblioteca gráfica e também por não ser tão pesada como os outros IDE, Netbeans e Visual Basic.

Uma das grandes propriedades do Eclipse é o desenvolvimento baseado em plugins. O projeto Eclipse foi e continua sendo um projeto tão grandioso que em janeiro de 2004, foi criada uma fundação sem fins lucrativos com membros que fazem o suporte do projeto Eclipse, a The Eclipse Foundation (IBM, 2014). Ela fornece em geral quatro serviços à comunidade Eclipse:

- IT (Information Technology) Infrastructure, que fornece aos usuários e projetistas do Eclipse uma tecnologia evolutiva e segura;

- Intellectual Property (IP) Management, que coordena a propriedade intelectual do software, gerencia os serviços e a comercialização do mesmo;

- Development Community Support, que contribui de maneira direta com melhorias no desenvolvimento do software;

- Ecosystem Development, onde a Fundação promove reuniões, conferências e afins para promover toda a comunidade Eclipse.

O Eclipse é um IDE de código aberto utilizado para desenvolvimento da linguagem java, mas é também multilinguagem, ou seja, suporta outras linguagens de programação tais como C/C++ instalando-se os devidos plug-ins adicionais para cada linguagem. É um projeto livre de patentes por ser um software livre. E é portátil, ou seja, sua aplicação funciona em vários ambientes, como: Windows, Linux, Solaris (Lozano, 2002).

A plataforma de desenvolvimento Eclipse fornece vários pacotes de desenvolvimento, tais como, Eclipse JDT, que é a base para qualquer plug-in na linguagem java, o Eclipse SDK, que é o pacote de distribuição da IDE java, o Eclipse WTP (Web Tools Platform), que é usado para desenvolvimento de linguagem para web, e o compilador do JDT, que é o próprio compilador java, mais rápido e de código aberto.

4.2 Java

Java é uma linguagem de programação orientada a objetos, formada por um conjunto de elementos que envolvem mais do que sintaxe da linguagem de programação. A linguagem Java possui os formatos de arquivos (.java e .class), um conjunto de APIs (classes, componentes, frameworks), e uma Máquina Virtual Java (JVM), que permite a portabilidade dos programas. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual. (Dias and Fontes, 2003).

Segundo Gomes (2014), a Sun Microsystems desenvolveu uma linguagem orientada a objetos denominada Oak, em 1990, cujo propósito era o desenvolvimento de pequenos aplicativos e programas para controle de eletrodomésticos e dispositivos portáteis. Com essa linguagem de programação seria possível programar novas funcionalidades para os aparelhos

domésticos, tornando-os mais capazes. Por exemplo, um forno de microondas poderia ser programado para preparar diversos tipos de alimentos (Magan and Vargas, 1999).

A linguagem de programação Java foi lançada pela primeira vez, em 1995, pela Sun Microsystems. A Sun disponibilizou a maioria das distribuições Java gratuitamente. Em 13 de novembro de 2006, a Sun liberou partes do Java como software livre, sob a licença GNU General Public License. Em 2009, a Oracle comprou a Sun, fortalecendo a marca. A Oracle sempre foi, junto com a IBM, uma das empresas que mais investiram e fizeram negócios através do uso da plataforma Java.

Atualmente, a linguagem Java consiste em uma das mais utilizadas no mundo, possuindo um dos maiores repositórios de projetos de código-aberto, o java.net. Por questão de sua portabilidade, o aplicativo Java é capaz de rodar em qualquer tipo de aplicação, entre elas: aplicativos móveis, jogos, web, servidores, desktop, etc (Pamplona, 2009).

4.3 CloudMade

De acordo com CloudMade (2014), a empresa foi fundada em 2007 para permitir aos colaboradores desenvolverem aplicações e serviços com localização, utilizando a sua API. A mesma está disponibilizada no site <http://cloudmade.com/>.

A CloudMade disponibiliza várias ferramentas e APIs aos colaboradores, dentre elas a API de roteamento. Apesar de não ser tarifada até uma quantidade de requisições mensais, para fazer uso da API é necessário fazer um registro no site.

A API CloudMade utiliza mapas fornecidos pela OpenStreetMap e retorna uma rota entre um conjunto de coordenadas. A rota retornada pode ser em formato GPX ou JSON.

A tabela 4.1 apresenta uma típica solicitação. Ela possui: ponto inicial, ponto final, tipo de rota e formato de retorno.

Um exemplo de requisição é:

```
http://routes.cloudmade.com/8ee2a50541944fb9bcedded5165f09d9/api/0.3/47.25976,9.58423,47.26117,9.59882/car/shortest.js
```

Tabela 4.1: Opções de solicitações com CloudMade

Opções de solicitações	Tipo de retorno
start-point	(lat,lon)
end-point	(lat,lon)
route-type	("car", "foot", "bicycle")
output-format	("js", "gpx")

O JSON retornado tem os parâmetros mostrados na Tabela 4.2.

Tabela 4.2: Retorno JSON

Status	0 - OK 1 - Error
status-message	Mensagem que descreve o erro.
route-summary	Resumo com alguns dados da rota como tempo total estimado distância total e pontos intermediários.
route-geometry	Geometria da rota. (usada para traçar a rota no mapa). Contém os nós e vetores de direção.
route-instructions	Instruções textuais para se chegar ao destino partindo da origem.

Exemplo onde a resposta é um arquivo Json:

```

1 {"status":0,"route_instructions":[
2 ["Head south on Perckhoevelaan",111,0,13,"0.1 km","S",160.6],
3 ["Turn left at Laarstraat",112,3,13,"0.1 km","NE",58.1,"TL"
4   ,269.0],
5 ["Turn right at Goudenregenlaan",70,5,8,"70 m","SE",143.4,"TR"
6   ,89.8]],
7 "route_summary":{"total_time":34,"total_distance":293,"end_point":
8   "Goudenregenlaan","start_point":"Perckhoevelaan"},"
9   route_geometry":
10  [[51.17702,4.3963],[51.17656,4.39655],[51.17639,4.3967],
11  [51.17612,4.39696],[51.1764,4.39767],[51.17668,4.39828],
12  [51.17628,4.39874],[51.17618,4.39888]],
13  "version":"0.3"}

```

4.4 Nominatim

Nominatim (do latim, "por nome"), é uma ferramenta de pesquisa para encontrar dados do OpenStreetMap pelo seu nome e endereço. Ele pode ser encontrado no site [http:](http://)

[//nominatim.openstreetmap.org](http://nominatim.openstreetmap.org). O Nominatim também fornece uma função de geocodificação reversa capaz de obter endereços ou nomes de seus pontos geográficos. Ao solicitar um nome ou endereço, ele é capaz de manipular várias opções e parâmetros (OpenStreetMap, 2010).

A Tabela 4.3 mostra os parâmetros mais importantes:

Tabela 4.3: Parâmetros Nominatim

Opção	Descrição
Format	Especifica o formato de saída. - Html; - Json; - XML.
json-callback	Função Callback usada para exibir os resultados abaixo
Q	String de consulta que está sendo pesquisada.
addressdetails	Incluir uma repartição de endereços em elementos separados. 0- Não exibir os dados. 1- Sim, exibir os dados
Limit	Limita o número de resultados retornados para o número inteiro digitado.
Countrycodes	Limita a pesquisa para um país específico ou uma lista de países. Ex: countrycodes=countrycode.
Viewbox	Área preferencial para encontrar resultados de pesquisa. Ex: viewbox=left,top,right,bottom
osm-type	Especifica se o recurso é um nó, caminho, ou relação em OpenStreetMap. N - Node; W - Way; R - Relation.

Exemplo Nominatim:

<http://nominatim.openstreetmap.org/search?q=135+pilkington+avenue,+birmingham&format=xml&polygon=1&addressdetails=1>

```

1 <searchresults timestamp="Sat, 07 Nov 09 14:42:10 +0000"
  querystring="135 pilkington, avenue birmingham" polygon="true"
  >
2 <place
3   place_id="1620612" osm_type="node" osm_id="452010817"
4   boundingbox="
      52.548641204834,52.5488433837891,-1.81612110137939"
5   polygonpoints="[[ '-1.81592098644987', '52.5487429714954' ]...]"

```



```
6   lat="52.5487429714954" lon="-1.81602098644987"
7   display_name="135, Pilkington Avenue, Wylde Green, City of
      Birmingham, West Midlands (county), B72, United Kingdom"
8   class="place" type="house">
9   <house>135</house>
10  <road>Pilkington Avenue</road>
11  <village>Wylde Green</village>
12  <town>Sutton Coldfield</town>
13  <city>City of Birmingham</city>
14  <county>West Midlands (county)</county>
15  <postcode>B72</postcode>
16  <country>United Kingdom</country>
17  <country_code>gb</country_code>
18 </place>
19 </searchresults>
```

4.5 SwingX-WS

A biblioteca SwingX WS é um subprojeto desenvolvido pela SwingLabs, um laboratório que se dedica a realizar projetos com código aberto e explorar novas formas de fazer aplicações com Swing. A SwingX WS é desenvolvida na linguagem de programação Java e contém um conjunto de JavaBeans (componentes reutilizáveis de software que podem ser manipulados visualmente com a ajuda de uma ferramenta de desenvolvimento) para interagir com os serviços web (García, 2010).

É possível encontrar vários beans incluídos no projeto SwingX WS, porém um deles, o JXMapView será utilizado no projeto. O JXMapView é um visualizador genérico de mapas. Devido a este componente é possível obter o mapa do sistema desenvolvido.

Para o correto funcionamento da biblioteca SwingX WS é necessário utilizar outra biblioteca. Ela também foi desenvolvida pela SwingLabs, chamada de SwingX. A SwingX é um série de aprimoramentos e novos componentes Swing, que irão auxiliar na construção de aplicativos Java para Desktop (García, 2010).

É importante destacar que a biblioteca SwingX WS proporciona bastantes classes interessantes, que serão utilizadas no desenvolvimento do sistema. Sendo elas: JXMapKit, JXMapView.

4.6 JXMapKit

O JXMapKit é um par de JXMap Viewers pré-configurado para facilitar na construção e visualização dos elementos. Por um lado temos o mapa principal no qual o usuário irá utilizar e por outro lado, um mapa com dimensões menores, com um menor nível de zoom, cuja funcionalidade é marcar sobre ele mesmo a parte visível no mapa principal. Em outras palavras, o JXMapKit inclui botões de zoom, um controle deslizante de zoom e um mini-mapa no canto inferior direito, mostrando uma visão geral do mapa (García, 2010).

A Figura 4.1 representa o JXMapKit.

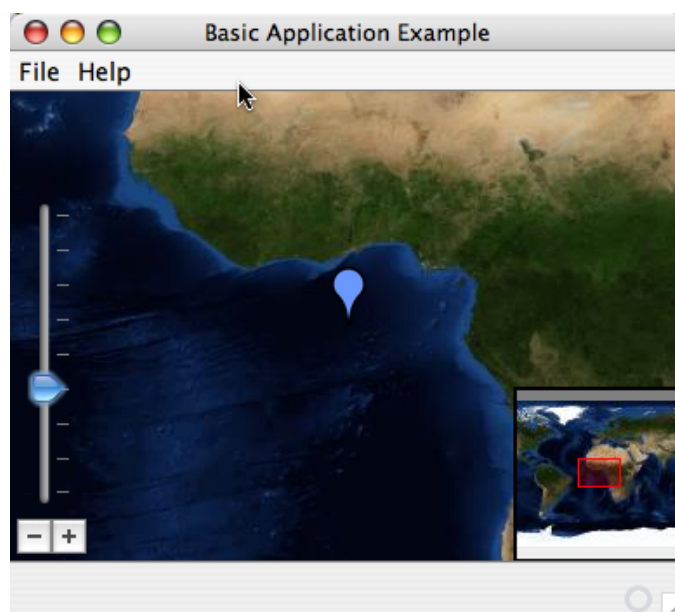


Figura 4.1: O JXMapKit: O mapa principal e outro mapa de menor dimensão na parte inferior direita

O JXMapKit herda a classe JPanel, pertencente a biblioteca SwingX. Algumas funções herdadas são:

- Obtenção da coordenada central do mapa determinada por latitude e longitude.
- Obtenção do mapa principal (componente JXMapView).

- Obtenção do mapa de menor dimensão (componente JXMapView).
- Centrar o mapa em uma coordenada determinada.
- Definição das características do mapa.
- Ajustar o nível de zoom.

4.7 JXMapView

O JXMapView é um componente Swing open-source (LGPL) criado pelos programadores no SwingLabs. No seu núcleo, o JXMapView é um JPanel especial que é capaz de carregar mapas de um servidor de imagens, como OpenStreetMap, GoogleMaps, Yahoo Maps (García, 2010).

O JXMapView possui três propriedades para o ponto central:

- A propriedade `addressLocation`. Representa no mapa um centro abstrato. (É o primeiro resultado obtido ao realizar a busca)
- A propriedade `centerPosition`. Representa o ponto central atual no mapa em latitude e longitude. Caso o usuário arrasta o mapa, o ponto altera o local, porém o `addressLocation` não.
- A propriedade `center`. Representa o mesmo ponto da propriedade `centerPosition`, porém em píxel.

Este componente também herda a classe `JXPanel`. Os métodos mais importantes da classe estão descritos abaixo:

- Calcular o nível de zoom a partir de um conjunto de pontos especificados, de modo que todos eles sejam visíveis na janela.
- Ativar ou desativar a opção de arrastar o mapa com o mouse.
- Obter as características atuais do mapa.

- Obter os limites da parte visível do mapa em um determinado momento.
- Obter as características atuais do mapa.
- Obter o nível de zoom atual.
- Ajustar o nível de zoom.
- Ajustar o ponto central do mapa em uma posição especificada. (tanto em píxel como em coordenadas), assim como obter os valores em um determinado momento.
- Converter o ponto geográfico em píxel e vice e versa.
- Estabelecer e obter o ponto addressLocation.

Capítulo 5

Sistema Desenvolvido

Ao rodar o main da aplicação é aberto a janela principal do sistema. Nessa tela deverá ser escolhido um arquivo e então o usuário deverá clicar no botão rodar.

Essa ação chama a classe `ControladorBtRodar()`. Nela é criada uma instância da classe `JxMap` passando o nome do arquivo selecionado.

O construtor de `JxMap` que recebe uma string, chama a função `preencheCidades()` passando o nome do arquivo. Nessa função que preenche as cidades, são lidas as cidades e as demandas do arquivo texto. Estes dados adicionados são preenchidos na estrutura da classe `JxMap`. Para cada cidade no arquivo é obtida as coordenadas das cidade com o serviço `Nominatim`.

Com a lista de cidades preenchidas na classe `JxMap`, é então instanciada a classe `PreencheRotaGPX`.

Essa classe ao ser instanciada faz uma consulta ao `CloudMade` tendo como pontos de origem e destino a primeira e última cidade do arquivo. Pontos intermediários são definidos utilizando as cidades intermediárias caso sejam mais de duas cidades preenchidas na classe `JxMap`. Isto é feito para se adequar a API do `CloudMade`.

O `CloudMade` retorna o arquivo com a rota no formato GPX que é então gravado conforme foi definido na função `gravaArquivo()`.

A função `getGrafo()` gera uma grafo com as cidades que foram obtidas da estrutura `JxMap`.

São preenchidos os trechos entre as cidades passadas no arquivo. Supondo três cidades A, B e C e sendo Origem->Destino, teríamos os seguintes trechos: A->B, A->C, B->A, B->C, C->A, C->B. Para cada um desses trechos será então criada uma instância da classe `Trecho`.

Essa classe contém origem, destino, coordenadas da origem e do destino, um objeto `JSONArray` com dados para traçar a rota, um objeto `JSONObject` com o resumo da rota, outro `JSONObject` com a rota em formato json além da distância e do tempo estimado pelo `CloudMade`.

O construtor dessa classe recebe uma string com o nome da cidade de origem e outra string com o nome da cidade de destino. As coordenadas destes 2 pontos são obtidas através do serviço `Nominatim`. Já sabendo as coordenadas dos pontos de origem e destino, é feito uma consulta ao `CloudMade`. Nessa consulta são passados as coordenadas da origem e do destino além de algumas outras informações como chave de inscrição para uso da API, versão utilizada, tipo de roteamento e tipo modificador de roteamento.

Em relação ao tipo do roteamento é passado como meio de transporte "carro" e como tipo modificador de roteamento é passado como sendo o "caminho mais rápido".

O que é retornado pelo `cloudMade` é armazenado na classe `Trecho` como sendo `json-Rota`. Para obter a geometria da rota, que é utilizada para traçar a rota no mapa, é extraído do `jsonRota` o elemento "route-geometry" e armazenado na classe `Trecho` como sendo `rota`. Essa rota está em formato `JSONArray`.

Do `jsonRota` também é possível extrair o elemento "route-summary" que é utilizado para obter o tempo estimado e a distância da origem ao destino. Cada instância da classe `Trecho` é então adicionada na lista de trechos da classe `JxMap`.

A função `imprimeTuchos()` exibe, na saída padrão, quais trechos foram criados.

A função `plotaRota()` tem como finalidade gerar uma mapa com a rota entre as cidades passadas no arquivo.

Para isso é criada uma nova classe JxMap passando as cidades que foram extraídas do arquivo texto. Então são preenchidos os trechos entre cada cidade e a cidade subsequente. Supondo 3 cidades A, B e C e sendo Origem->Destino, teriam se os seguintes trechos: A->B, B->C. Assim como ao preencher o grafo, para cada trecho criado é feita uma consulta ao CloudMade e cada instância da classe Trecho é então adicionada na lista de trechos da classe JxMap.

Preenchida a classe JxMap é então criada a tela que irá conter o mapa. O JXMapKit é criado e recebe algumas configurações iniciais como qual será o provedor dos fragmentos(tile) do mapa, zoom, url, entre outros através da classe TileFactory do SwingX. Essa tela criada e os trechos obtidos pela classe JxMap são então passados para a classe TracaRota.

A classe TracaRota pega as rotas, que são uma lista de pontos, presentes em cada trecho e a sequência é apenas uma lista de pontos.

O SwingX tem uma classe Painter que auxilia a desenhar sobre o mapa, e traçar a rota nada mais é do que desenhar uma reta ligando alguns pontos. Esse "desenho" é feito através da função paint, que cria uma figura 2d e traça uma reta sobre os pontos passados à classe TracaRota.

Após isso é então exibido o mapa com a rota das cidades.

5.1 Modelagem

As funcionalidades do sistema desenvolvido são demonstradas através da modelagem Unified Modeling Language (UML), utilizada para especificar, visualizar, construir e documentar os artefatos de um sistema, geralmente com orientação a objetos (Uml.org, 2014).

5.1.1 Diagrama de Caso de Uso

A Figura 5.1 representa a interação do usuário com o sistema, no caso, o usuário poderá escolher o arquivo com as cidades, demandas e o sistema calculará a distância entre seus trechos. O sistema também exibirá o mapa com a rota traçada. Na sequência a tabela 5.1 apresentará a descrição do caso de uso, onde é explicado o fluxo principal da aplicação, para melhor entendimento dos eventos.

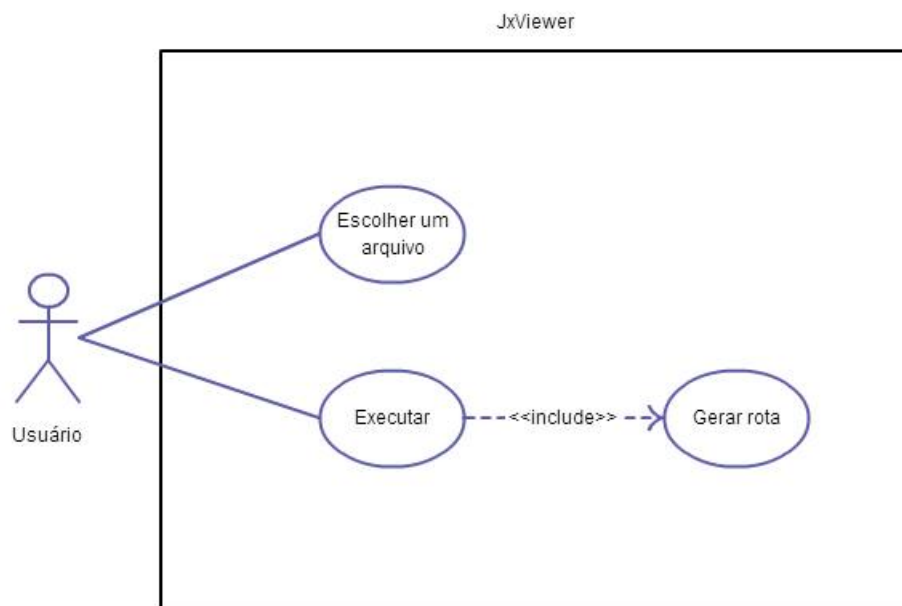


Figura 5.1: Diagrama de Caso de Uso

Tabela 5.1: Descrição Caso de Uso

Descrição	Este caso de uso permite que o usuário selecione um arquivo com as cidades na qual deseja calcular a distância entre seus trechos. Em seguida o sistema carrega o mapa com a rota traçada.
Caso de Uso	Gerar Rota
Ator	Usuário
Pré-condição	Selecionar arquivo
Pós-condição	Exibir o mapa com a rota traçada
Fluxo Principal	<ol style="list-style-type: none"> 1. Usuário acessa a tela principal, a qual exibirá dois botoões escolher um arquivo e executar 2. O usuário seleciona um arquivo com as cidades e pressiona o botão escolher um arquivo. 3. O usuário pressiona o botão rodar. 4. O mapa é exibido com a rota traçada.
Fluxo Alternativo	4.1 O mapa não consegue ser exibido pois o computador está sem internet.
Caso de Uso Complementar	Exibir a rota

5.1.2 Diagrama de Classes

No diagrama de classes são apresentadas as classes utilizadas para o desenvolvimento do sistema, a estrutura das classes, com seus atributos e métodos e a relação entre as classes, conforme a Figura 5.2.

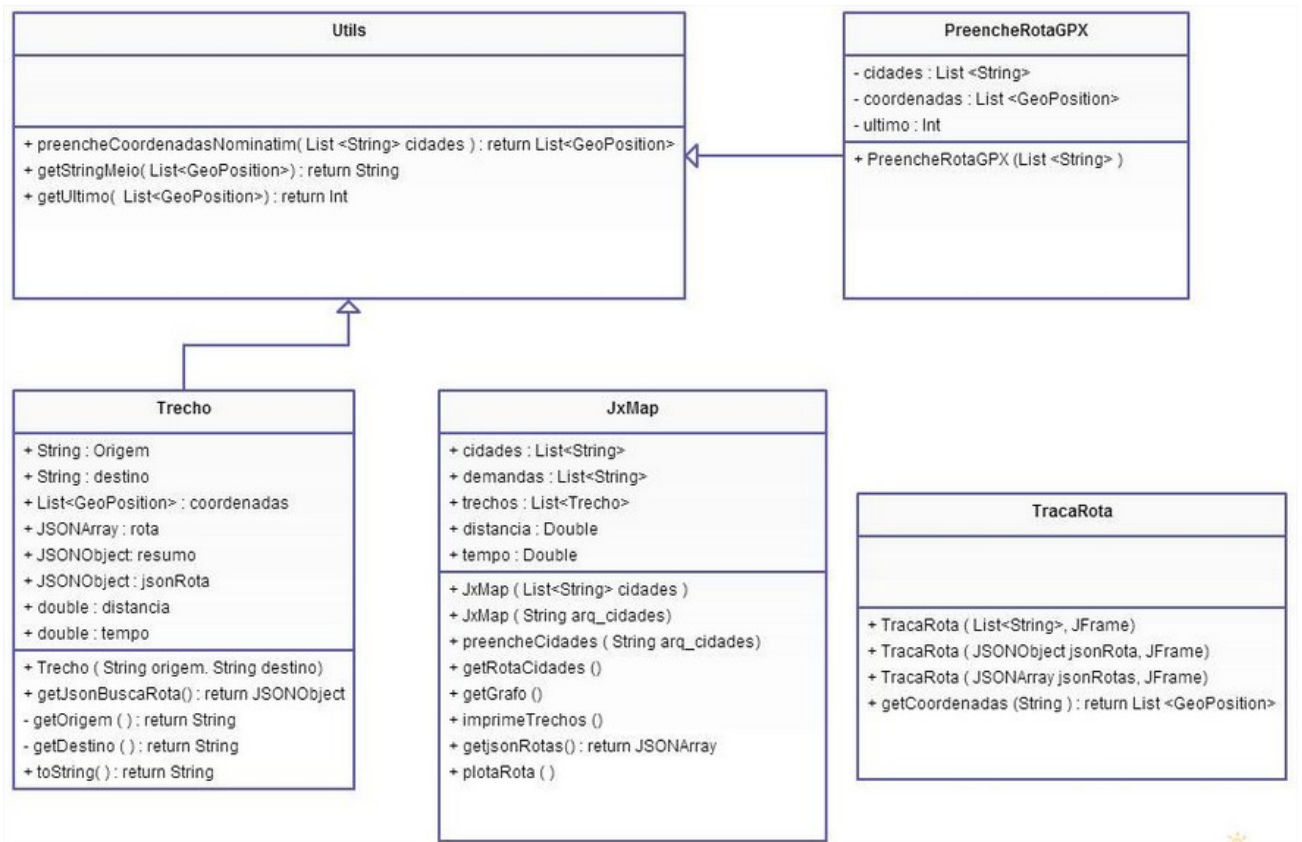


Figura 5.2: Diagrama de Classe

5.1.3 Diagrama de Sequência

A Figura 5.3 mostra as interações das classes do sistema e suas funções, com envio de mensagens de solicitação e respostas de retorno entre eles, em uma linha de tempo para acompanhar o processo.

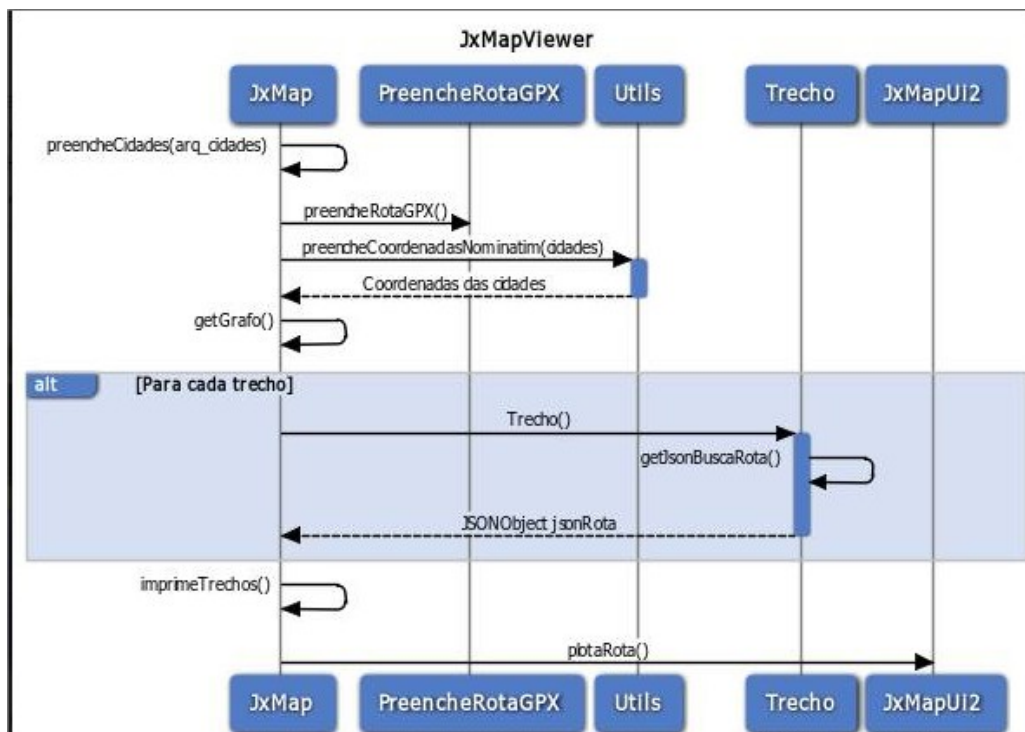


Figura 5.3: Diagrama de Sequência

Capítulo 6

Testes Realizados

Este capítulo apresenta o funcionamento do sistema desenvolvido, através da apresentação de suas telas iniciais.

6.1 Execução do Sistema

A Figura 6.1 apresenta a tela principal do sistema. O usuário deverá selecionar um arquivo ".txt" para iniciar a execução do sistema.

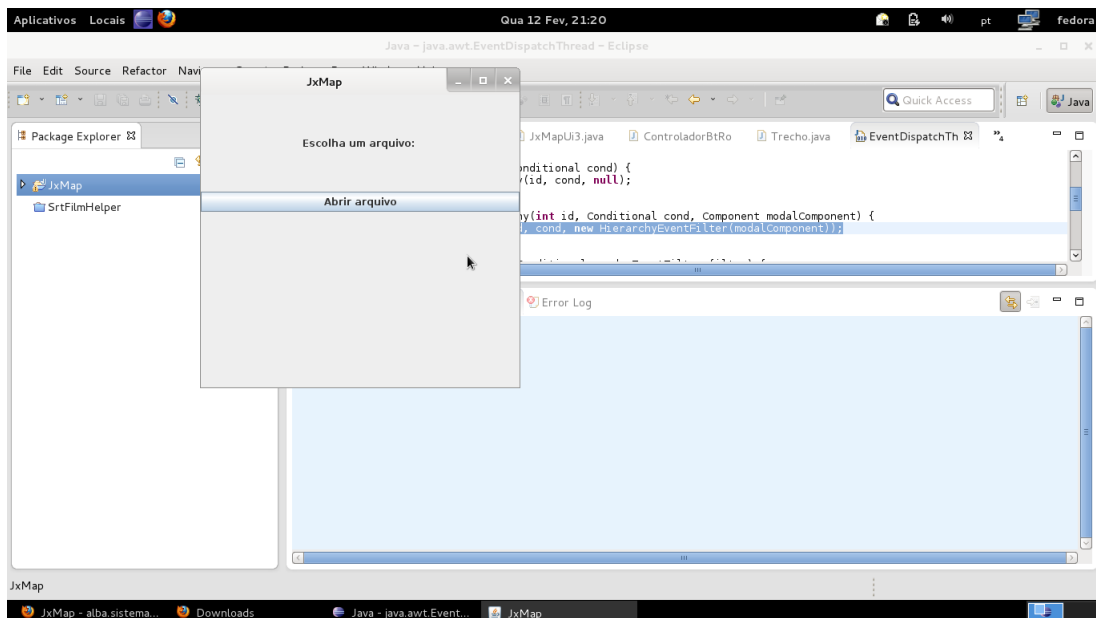


Figura 6.1: Tela Principal

6.1. Execução do Sistema

Em seguida abrirá uma janela, onde o usuário deverá escolher o arquivo e clicar no botão "Abrir".

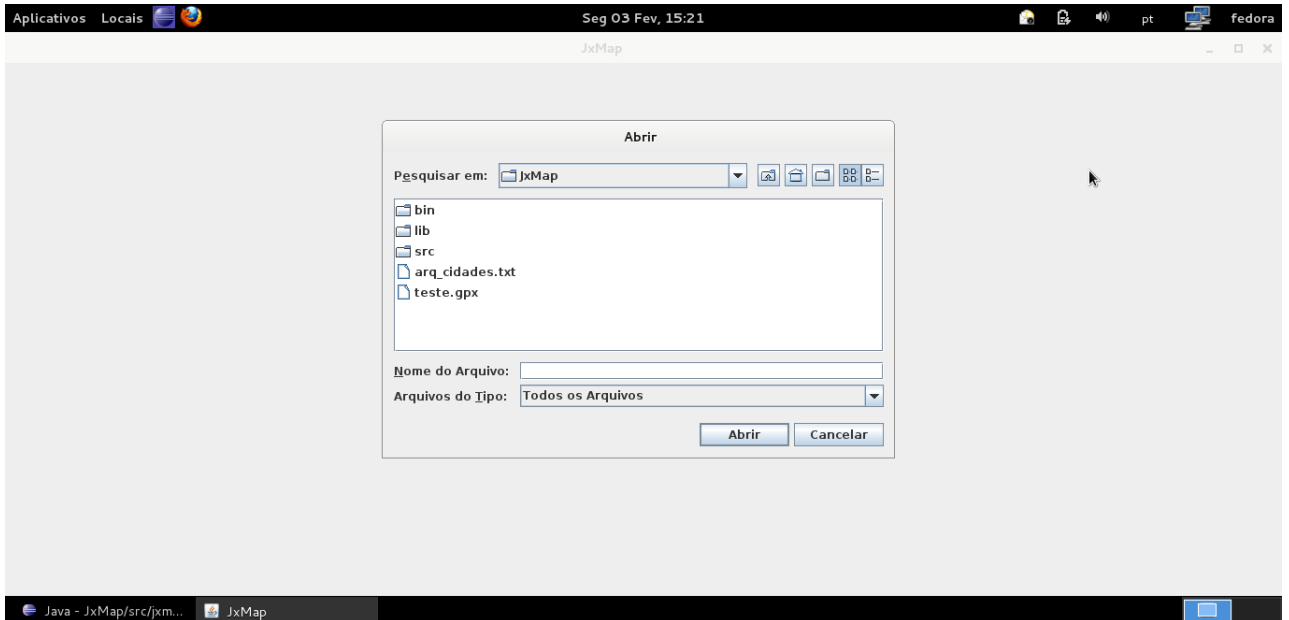


Figura 6.2: Selecionar Arquivo

Conforme a Figura 6.3 o arquivo ".txt" apresentará as cidades e as demandas. A lista das cidades deverá estar na ordem que o usuário deseja gerar a rota.

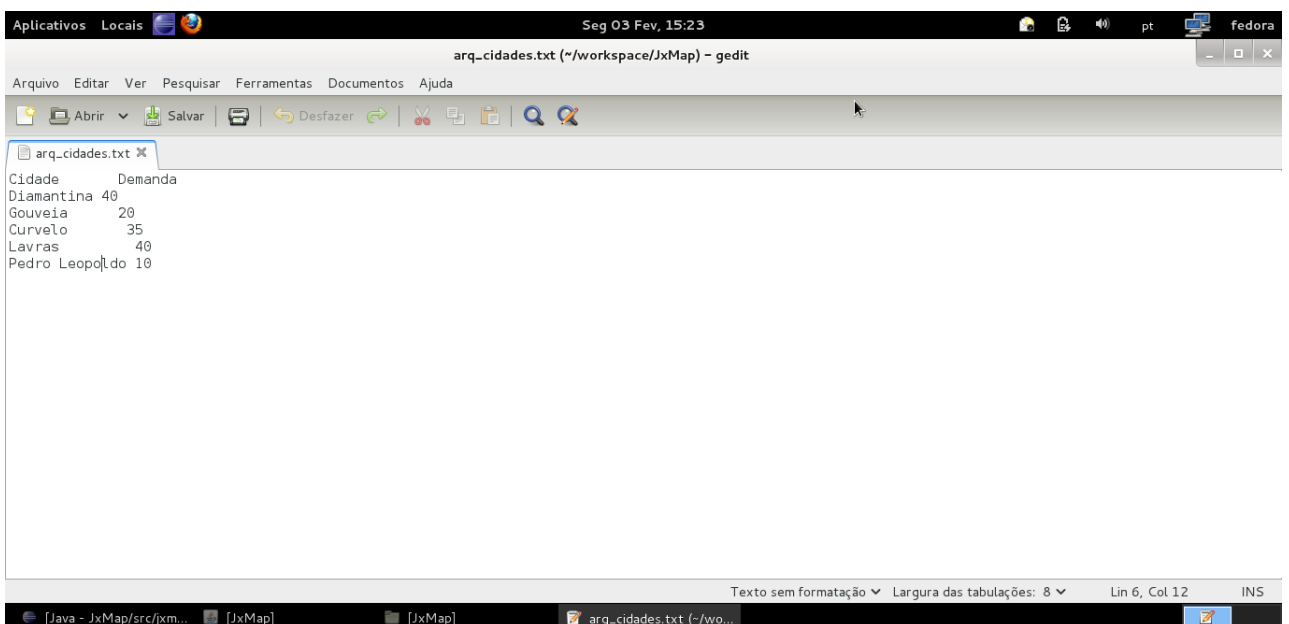


Figura 6.3: Arquivo texto

O sistema retornará a tela principal com o arquivo ".txt" já selecionado. Dessa forma, o usuário deverá clicar no botão rodar para iniciar a execução do sistema. Neste momento será gerado a rota com as cidades determinadas no arquivo e gerado um arquivo gpx com a rota traçada.

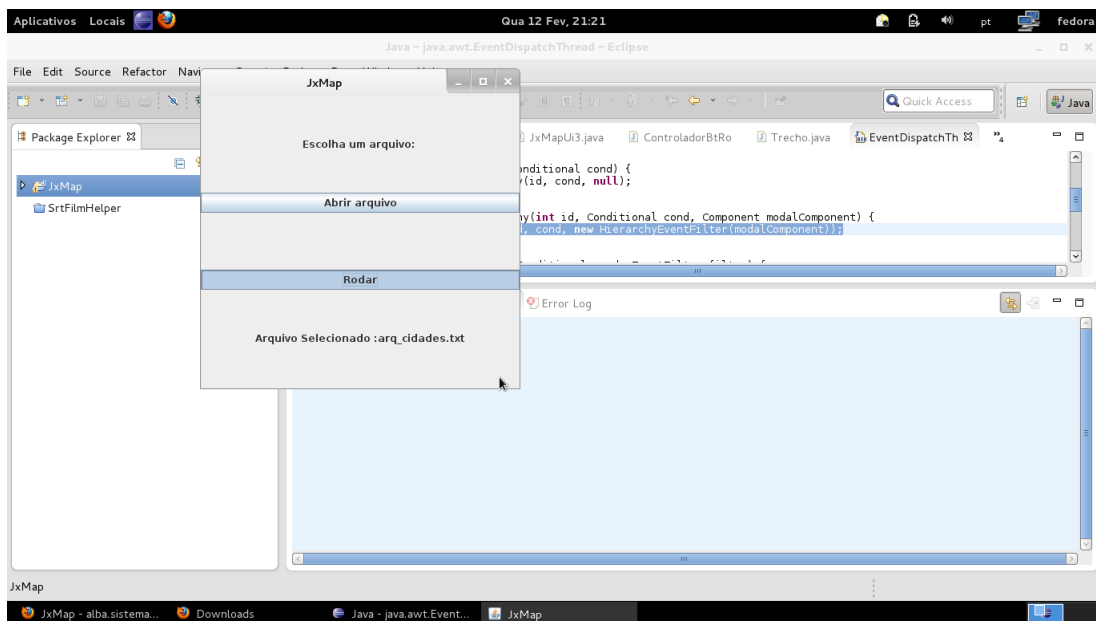


Figura 6.4: Tela Principal com arquivo selecionado

As Figuras 6.5, 6.6, 6.7 representam a rota dos trechos plotados no mapa.

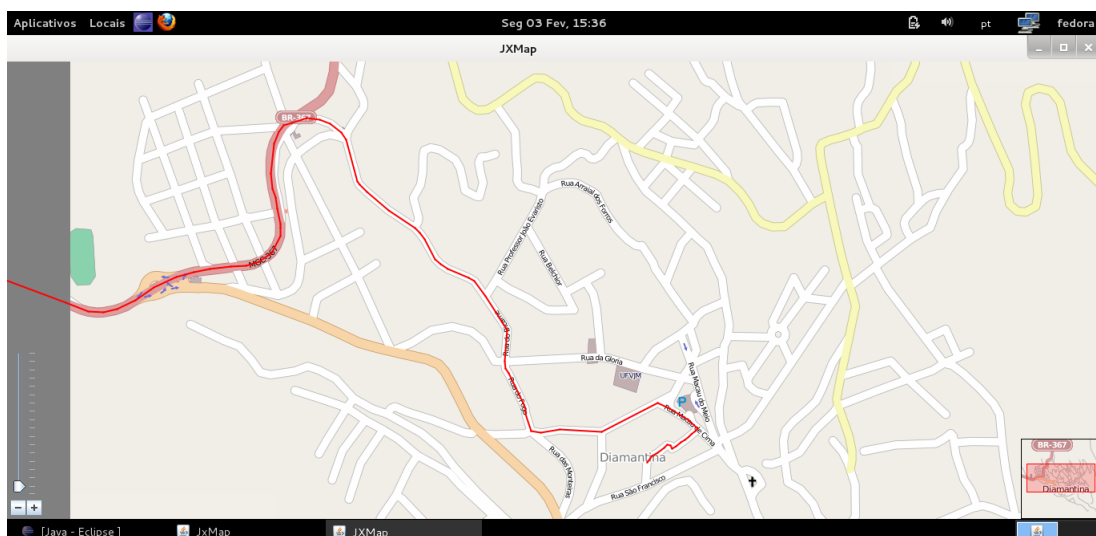


Figura 6.5: Rota plotada

6.1. Execução do Sistema

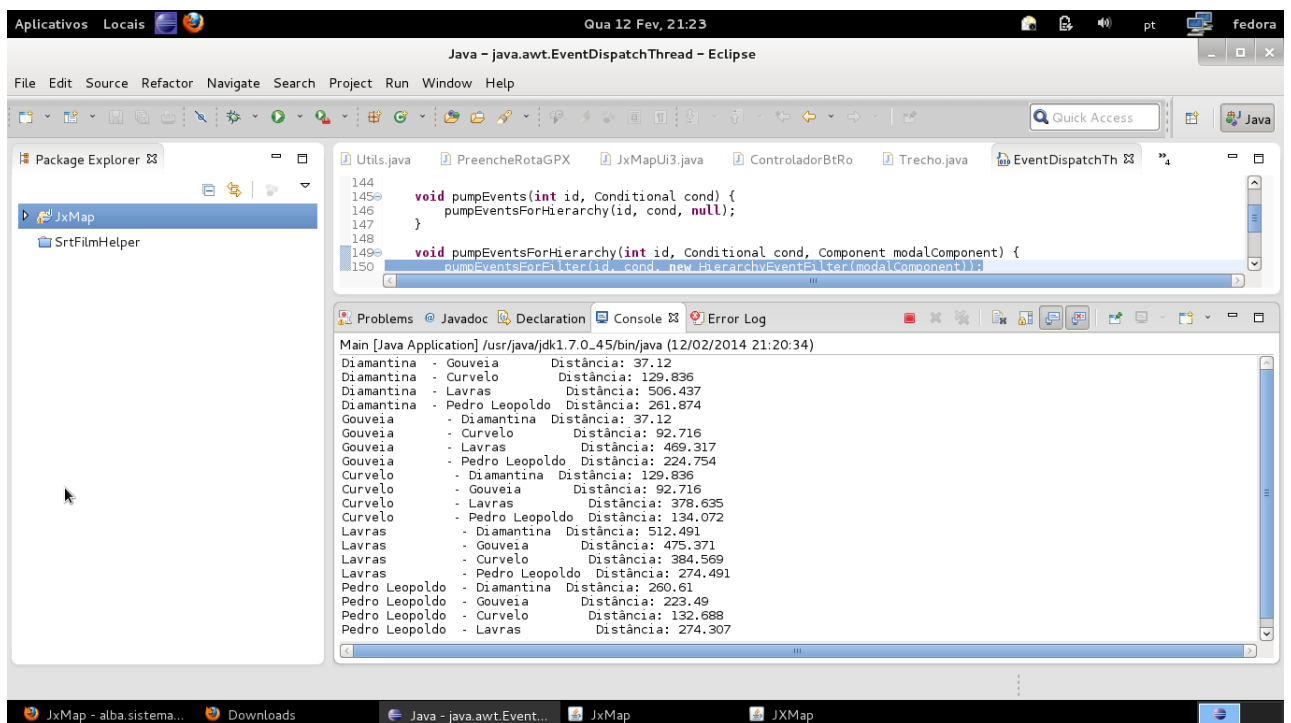


Figura 6.8: Distância entre trechos de cidade

Capítulo 7

Conclusão

Os problemas de Roteamento de Veículos estão sendo estudados e aplicados cada vez mais. Este fato acontece devido a globalização e o impacto que a mesma causa na economia e nas empresas. A aplicação de algoritmos de Roteamento de Veículos permitem a minimização dos gastos e maximização dos recursos e do tempo.

No presente trabalho foi possível conhecer mais sobre a poderosa e atualizada fonte de dados que é o OpenStreetMap e algumas tecnologias que o cercam. Visualizar através de interface gráfica as rotas entre os pontos roteados e os plotar no mapa, oferece uma nova dimensão ao problemas relacionados à logística.

Através do software proposto foi possível obter a rota traçada e a plotar sobre o mapa que foi fornecido através de um arquivo contendo um conjunto de cidades. Também foi calculado a distância entre os trechos das cidades. O software desenvolvido utilizou a linguagem de programação java, a API OpenStreetMap e CloudMade.

São necessários estudos mais aprofundados no que tange o roteamento utilizando a base do OpenStreetMap. Isso permitiria não utilizar a API fornecida pela empresa CloudMade e aplicação dos diferentes algoritmos de roteamento. Outro estudo que merece uma atenção é a definição da ordem das cidades no arquivo passado para o software aqui proposto. A ordem entre elas poderia ter um impacto na distância total a ser percorrida no trajeto.

Aplicativos que permitam a visualização de informações em um mapa, como a visualização de endereços dos clientes plotados no mapa, ou aplicativos que permitam visualizar estruturas no mapa de acordo com o perfil do usuário são um novo horizonte a ser explorados.

Referências Bibliográficas

- Bennett, J. (2010). OpenStreetMap. [citado na(s) páginas(s) 8, 14, 16, 19, 21, 23, 24]
- Braun, D. (2010). Comércio eletrônico fatura 6,7 bilhões no 1º semestre. O Globo. [citado na(s) páginas(s) 1]
- Canto, E. (2010). Como funciona o gps? [citado na(s) páginas(s) 8]
- CloudMade (2014). Cloudmade. [citado na(s) páginas(s) 28]
- da Costa, M. A. F. (2011). Sistema de apoio à mobilidade de utilizadores de bicicleta. page 7. [citado na(s) páginas(s) 11]
- Dantzig, G. B. and Ramser, J. H. (1959). The Truck Dispatching Problem, volume 6. [citado na(s) páginas(s) 1]
- Date, C. J. (2004). Introdução a Sistemas de Banco de Dados. Campus - RJ. [citado na(s) páginas(s) 7]
- de Jesus Júnior, I. D. and Marozzi, J. W. (2014). Kml: Linguagem de marcação. [citado na(s) páginas(s) 10]
- de Oliveira, H. C. B. (2011). Algoritmo online para o problema dinâmico de roteamento de veículo. [citado na(s) páginas(s) 1, 2]
- Dias, K. L. and Fontes, W. P. (2003). Desenvolvimento de aplicações para dispositivos móveis utilizando a plataforma j2me. [citado na(s) páginas(s) 27]
- DPI (2014). Teoria: Estrutura de dados. [citado na(s) páginas(s) 6]
- Fraga, M. C. P. (2007). Uma metodologia híbrida: Colônia de formigas – busca tabu – reconexão por caminhos para resolução do problema de roteamento de veículos com janela de tempo. [citado na(s) páginas(s) 2]

- García, S. B. (2010). Desarrollo de un sistema automático para búsqueda por contenido en imágenes de satélite. [citado na(s) páginas(s) 31, 32, 33]
- Garmin (2007). Training center database xml. [citado na(s) páginas(s) 12]
- Garmin (2010). Garmin. [citado na(s) páginas(s) 20]
- Gazola, A. and Furtado, A. L. (2007). Banco de dados geográficos inteligentes. [citado na(s) páginas(s) 7]
- GNU (2014). Gnu operating system. [citado na(s) páginas(s) 4, 5]
- Gomes, A. (2014). Tutorial: J2me – visão geral. [citado na(s) páginas(s) 27]
- IBM (2014). Introdução à plataforma eclipse. [citado na(s) páginas(s) 26]
- Ichoua, S. (2000). Diversion issues in real-time vehicle dispatching. [citado na(s) páginas(s) 2]
- Lorbiesk, R. (2010). Uma contribuição ao controle de dengue em cascavel/pr através de um software com informações georreferenciadas. [citado na(s) páginas(s) 9]
- Lozano, F. (2002). Eclipse visual com swt. Java Magazine. [citado na(s) páginas(s) 27]
- Magan, M. and Vargas, P. (1999). Técnicas para desenvolvimento de aplicações orientadas a objetos utilizando a linguagem java. [citado na(s) páginas(s) 28]
- Malaquias, N. G. L. (2006). Uso dos algoritmos genéticos para a otimização de rotas de distribuição. page 17. [citado na(s) páginas(s) 1]
- Medina, R. D. (2014). A linguagem xml. page 3. [citado na(s) páginas(s) 9]
- Mine, M. T. (2010). Problema de roteamento de veículos com coleta e entrega simultânea (prvces). [citado na(s) páginas(s) 1]
- OpenSteetMap (2010). Nominatim. [citado na(s) páginas(s) 30]
- Pamplona, V. F. (2009). O que é java free. [citado na(s) páginas(s) 28]
- Saleh, A. M. (2004). Adoção de tecnologia: Um estudo sobre o uso de software livre nas empresas. page 13. [citado na(s) páginas(s) 5]
- Steiner, M. T. A. (2000). Problema de roteamento no transporte escolar. [citado na(s) páginas(s) 2]
- Uml.org (2014). Introdução ao uml. [citado na(s) páginas(s) 37]