



ÁREA DEPARTAMENTAL DE ENGENHARIA ELECTRÓNICA E DE
TELECOMUNICAÇÕES E DE COMPUTADORES

IDENTIFICAÇÃO E REPRESENTAÇÃO AUTOMÁTICA DE PERCURSOS DE AUTOCARROS

LUÍS FILIPE DE JESUS OLIVEIRA

Trabalho Final de Mestrado para obtenção do Grau Mestre em
Engenharia de Redes de Comunicações e Multimédia

JURI:

Presidente: Doutor Arnaldo Joaquim Castro Abrantes, Prof. Coordenador

Arguente: Doutor Paulo Jorge Freitas Oliveira Novais, Prof. Associado c/Agreg.

Arguente: Doutor Paulo José de Matos Martins, Prof. Adjunto

Orientador: Doutor João Carlos Amaro Ferreira, Prof. Adjunto

DEZEMBRO DE 2015

Agradecimentos

Quero agradecer ao meu orientador, Professor Doutor João Carlos Amaro Ferreira e toda sua equipa, por se dispor na orientação, apoio e disponibilidade prestados durante este mestrado, em especial, a dedicação a este projeto.

Aos colaboradores da empresa Tecmic, António Marcelo, Joni Batista, Gabriel Saragoça, Mário Isidoro e João Costa pelo apoio prestado no enquadramento dos produtos, na oportunidade de participação no desenvolvimento de novas tecnologias da Tecmic.

A todos os colegas do curso de mestrado, que durante o mestrado contribuíram de alguma forma com apoio e motivação nas tarefas de aprendizagem.

Citação

"Making maps together means piecing together collective experiences, discovering patterns, and arriving at a collective understanding of the root causes of these shared experiences."

Eli Moore and Catalina Garzón,

Social Cartography: The Art of Using Maps to Build Community Power, in Weaving the Threads Vol. 17 , No. 2, 2010

Resumo

O transporte público de passageiros (TPP) utiliza um sistema complexo e dinâmico que exige conhecimentos de várias disciplinas. Este interdisciplinar envolve predominantemente as aplicações da geografia e topografia, planeamento de mobilidade, economia e informática na área dos transportes.

TPP é uma parte crucial de qualquer sistema de transporte que, por natureza, caracteriza espaço terrestre e é usado em ambientes geográficos e topográficos complexos. Geografia de transportes é a geografia dos sistemas de transporte. Do ponto de vista das Tecnologias de Informação, geógrafos de transporte podem ajudar no desenvolvimento da topologia de rede do transporte público, e nas avaliações subsequentes que os transportes exigem.

Planeamento da mobilidade envolve desenvolvimento da rede de rotas de transporte por parte dos engenheiros de trânsito, tais como rotas, serviços, juntamente com temporização e padrões de viagem. Bom planeamento de mobilidade resulta em rotas de transporte público mais eficientes e estimulam o uso dos meios de transporte sustentáveis.

Abstract

Public passenger transport (PPT) use a complex and dynamic system requiring expertise from multiple disciplines. This interdisciplinary predominantly involves the applications of transport geography and topography, mobility planning, transport economics, and transport informatics.

PPT is a crucial part of any transport system that, by its nature, consumes land and exists within complex geographical and topographical settings. Transport geography is the geography of transport systems. From the Information Technology point of view, transport geographers can help when modeling the elements of a public transport network's topology, and the subsequent demanding transport evaluations.

Mobility planning involves traffic engineers modeling transport routes network elements such as routes, services, together with timing and journey patterns. Good mobility plans result in efficient public transport lines and stimulate the usage of sustainable modes of transport.

Conteúdo

AGRADECIMENTOS	I
CITAÇÃO	III
RESUMO.....	V
ABSTRACT.....	VII
LISTA DE ACRÓNIMOS.....	XV
1 INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO.....	1
1.1.1 <i>Tecmic - XTraN Passenger</i>	1
1.2 CONTEXTO (ITS, PIS)	2
1.3 OBJETIVOS	2
1.4 METODOLOGIA.....	3
1.5 ORGANIZAÇÃO DO DOCUMENTO	4
2 ESTADO DA ARTE	5
2.1 INTEROPERABILIDADE	5
2.1.1 <i>Conceito de interoperabilidade</i>	6
2.1.2 <i>Normas que suportam a interoperabilidade dos ITSs</i>	7
2.1.3 <i>Requisitos para a captação de normas</i>	9
2.2 TÓPICOS RELACIONADOS COM O TEMA DO TRABALHO.....	10
2.2.1 <i>Advanced Public Transportation Systems (APTS)</i>	10
2.2.2 <i>Geographic Information System (GIS)</i>	11
2.2.3 <i>OpenStreetMap</i>	12
2.2.4 <i>OpenLayers</i>	15
2.2.5 <i>General Transit Feed Specification (GTFS)</i>	16
2.2.6 <i>Differential GPS (DGPS)</i>	16
2.2.7 <i>Algoritmo de classificação</i>	18
2.3 BASE DE DADOS DE SUPORTE.....	21
2.3.1 <i>Quantum Gis (QGIS)</i>	21
2.3.2 <i>PgRouting</i>	21
2.4 XTRAN PASSENGER.....	22
2.4.1 <i>Interpretação do Modelo de Dados</i>	23
3 MODELO CONCEPTUAL DO SISTEMA.....	27
3.1 INTERAÇÃO DO SISTEMA COM XTRAN PASSENGER.....	27
3.2 CERTIFICAÇÃO DE PARAGENS.....	29
3.2.1 <i>Differential GPS (DGPS)</i>	30
3.2.2 <i>O algoritmo KNN</i>	30

3.3	PLANEAMENTO DE ROTAS	31
3.4	PREPARAÇÃO DOS DADOS EM GTFS.....	33
4	IMPLEMENTAÇÃO DO TRABALHO	37
4.1	INTERAÇÃO DO SISTEMA COM XTRAN PASSENGER.....	37
4.2	CERTIFICAÇÃO DE PARAGENS.....	39
4.3	PLANEAMENTO DE ROTAS COM OSM.....	41
4.3.1	<i>Dados do OSM.....</i>	42
4.3.2	<i>A ferramenta PgRouting</i>	46
4.4	PREPARAÇÃO DOS DADOS EM GTFS.....	48
4.4.1	<i>Disponibilizar publicamente um Feed de transporte público</i>	52
4.4.2	<i>Enviar um feed de transporte público para o Google.....</i>	52
5	AVALIAÇÃO DA SOLUÇÃO.....	53
5.1	COMPARATIVO DA SOLUÇÃO PROPOSTA COM A ATUAL	53
5.2	COMPARATIVO COM O MERCADO	53
5.3	TESTES DA SOLUÇÃO	55
5.3.1	<i>Teste de algoritmo a utilizar</i>	55
5.3.2	<i>Comparativo de planeamento de rota manual e automático</i>	56
5.3.3	<i>Teste de ficheiros criados para o GTFS</i>	57
6	CONCLUSÕES	59
6.1	TRABALHO FUTURO	59
	ANEXOS	61
	REFERÊNCIAS.....	63

LISTA DE TABELAS

<i>Tabela 1: Comparação de propriedades entre alguns padrões [19].....</i>	<i>8</i>
<i>Tabela 2: Tabela de tipo de vias do OSM [18].....</i>	<i>12</i>
<i>Tabela 3: Exemplos de conjuntos chave/valor do OSM [18].....</i>	<i>14</i>
<i>Tabela 4: Exemplos de recursos do OSM [18].....</i>	<i>14</i>
<i>Tabela 5: Etiquetas importantes para o planejamento de rotas [18].....</i>	<i>15</i>
<i>Tabela 6: Especificação dos arquivos GTFS juntamente com seu conteúdo associado [21]:.....</i>	<i>16</i>
<i>Tabela 7: Lista de valores para a etiqueta oneway [18].....</i>	<i>32</i>
<i>Tabela 8: Exemplo de horários [21].....</i>	<i>35</i>
<i>Tabela 9: Relação entre três troços de uma carreira e devidas paragens.....</i>	<i>42</i>
<i>Tabela 10: Relação entre três troços e devidas paragens.....</i>	<i>46</i>
<i>Tabela 11: Relação das vias e as paragens.....</i>	<i>47</i>
<i>Tabela 12: Dados recolhidos em PTPARAGEM para integrar em ficheiro GTFS.....</i>	<i>48</i>
<i>Tabela 13: Características de alguns planeadores de rotas on-line [35].....</i>	<i>54</i>

LISTA DE FIGURAS

Figura 1: Diagrama do modelo proposto	3
Figura 2: FRAME – A infraestrutura da arquitetura feita para a Europa [2]	6
Figura 3: Evolução de padrões internacionais do transporte público de passageiros [19]	7
Figura 4: Modelos IFOPT e NeTeX [19]	10
Figura 5: <i>Layers</i> funções de negócios e serviços [24]	15
Figura 6: Esquema de funcionamento do DGPS [27]	18
Figura 7: Distância Euclidiana e Manhattan [22]	19
Figura 8: Modelo de dados relacional do XTraN Passenger [20]	23
Figura 9: Base de dados fsetp_fct_user	23
Figura 10: Dados Geográficos da tabela TROCO	24
Figura 11: Dados Geográficos da tabela PTPARAGEM	25
Figura 12: Dados Geográficos da tabela PTPARAGEM com camada OSM	26
Figura 13: Arquitetura do trabalho	28
Figura 14: Diagramas de Casos de Uso	29
Figura 15: Registo da localização GPS no círculo verde durante 30mn [28]	30
Figura 16: Algoritmo KNN aplicado a uma via e duas paragens	31
Figura 17: Estrutura de livrarias importadas	37
Figura 18: UI de seleção de rota a calcular	38
Figura 19: Paragem “Petrobras” no círculo azul e leitura no círculo verde	40
Figura 20: Ponto por classificar	41
Figura 21: Modelo relacional usado no planeamento das rotas	42
Figura 22: OpenLayer plugin do QGis para OSM	43
Figura 23: Dados das vias do OSM	43
Figura 24: Layer Osm2po	44
Figura 25: Via após a ferramenta “Osm2po” à direita	44
Figura 26: Ponto mais próximo entre uma <i>linestring</i> e <i>point</i>	45
Figura 27: Paragens na linha da rota	45
Figura 28: PgRouting <i>Layer</i>	47
Figura 29: Esquema de relação dos ficheiros GTFS	49
Figura 30: Criação de pontos geográficos para GTFS	50
Figura 31: <i>Linestring</i> do OSM representativa da rota	50
Figura 32: Rota GTFS	51
Figura 33: Aplicação DotNet de criação dos ficheiros GTFS	52
Figura 34: Pasta de ficheiros GTFS em zip	52
Figura 35: Cálculo de rota no GoogleMaps (Esq.) e OSRM (Dir.)	53

Figura 36: Roda de tarefas para o construtor GTFS da RTAP [31]	54
Figura 37: <i>Shortest Path A*</i> versus <i>Shortest Path Dijkstra</i>	55
Figura 38: Algoritmo " <i>Shortest Path A*</i> " e " <i>Bi-directional Shortest Path A*</i> "	55
Figura 39: Algoritmo " <i>Dijkstra Shortest Path</i> " e " <i>Bi-directional Dijkstra Shortest Path</i> "	56
Figura 40: Comparação de rota manual e automática	57
Figura 41: Teste de planeamento de rota	57
Figura 42: Teste de validação do GTFS criado.....	58
Figura 43: Teste de horário e geometrias criadas no GTFS	58

Lista de Acrónimos

API *Application Programming Interface*

APTS *Advanced Public Transportation Systems*

ATIS *Advanced Traveler Information Systems*

AVL *Automatic Vehicle Location*

B2B *Business to Business*

CEN *Comité Europeo de Normalización*

DGPS *Differential GPS*

EIF *European Interoperability Framework*

FMS *Estação Flexível de Manufatura*

GDAL *Geospatial Data Abstraction Library*

GIS *Geographic Information System*

GPL *General Public License*

GPS *Global Positioning System*

GTFS *General Transit Feed Specification*

ICT *Information and Communication Technology*

IFOPT *Identification of Fixed Objects in Public Transport*

IS *Intelligent Systems*

IT *Information Technology*

ITS *Intelligent Transportation Systems*

KNN *K-Nearest Neighbor*

NeTeX *Network Timetable and Exchanges*

ODbL *Open Database License*

ORM *Object-Relational Mapping*

OSGeo *Open Source Geospatial Foundation*

OSM *OpenStreetMap*

PIS *Passenger Information Systems*

PTSP *Public Transport Service Providers*

PPT *Public Passenger Transport*

PPTA *Public Passenger Transport Authorities*

RIA *Rich Internet Application*

RTTT *Road Transport and Traffic Telematics*

SA *Selective Availability*

SIRI *Standard Interface for Real-Time Information*

SQL *Structured Query Language*

UML *Unified Modeling Language*

URL *Uniform Resource Locator*

WWW *World Wide Web*

XML *eXtensible Markup Language*

1 Introdução

O crescimento dos transportes rodoviários, combinado com o crescimento da economia e com as exigências de mobilidade dos cidadãos, é a principal causa do congestionamento cada vez maior das vias rodoviárias e do aumento do consumo de energia, bem como dos problemas ambientais e sociais. A resposta a esses grandes desafios não se pode limitar a medidas tradicionais, incluindo, nomeadamente, a expansão da infraestrutura existente de transporte rodoviário. A inovação terá um papel importante a desempenhar na procura de soluções adequadas para a Comunidade.

Os Sistemas de Transporte Inteligentes (*Intelligent Transportation Systems* (ITSs)) são aplicações avançadas que, sem serem dotadas de inteligência enquanto tal, destinam-se a prestar serviços inovadores em matéria de modos de transporte e de gestão do tráfego, e a permitir que diversos utentes fiquem mais bem informados e utilizem as redes de transportes de uma forma mais segura, mais coordenada e mais “inteligente”.

A aplicação de tecnologias da informação (*Information Technology* (IT)) e das comunicações no sector dos transportes rodoviários e nas suas interfaces com outros modos de transporte, contribuirá significativamente para melhorar o desempenho ambiental, a eficiência, a segurança dos transportes rodoviários e a mobilidade dos passageiros.

Os progressos conseguidos nas comunicações e nas tecnologias da informação refletem uma evolução do sector do transporte rodoviário, no âmbito da integração entre o transporte rodoviário e outros modos de transporte, dando relevância à interoperabilidade das tecnologias entre os transportes.

1.1 Motivação

O presente trabalho enquadra-se numa necessidade real de uma empresa que desenvolve soluções de ITS na área dos transportes, a TECMIC.

Esta empresa disponibiliza atualmente um sistema XTraN Passenger, desenvolvida a pensar nos operadores de transportes públicos, possibilitando-lhes a consulta de tempos relacionados com as carreiras da frota, em tempo-real.

Uma operadora de transportes públicos ao iniciar um projeto novo, num determinado local que lhe é desconhecido, enfrenta algumas dificuldades, como a identificação do local das paragens que melhor assiste a população, e o traçado de rotas que garantam uma maior rentabilidade e mobilidade.

O presente trabalho pretende facilitar esta tarefa inicial de identificar as melhores rotas tendo em conta um conjunto de paragens identificadas pelo operador de transporte X, ou seja o sistema a desenvolver pretende dado um conjunto de pontos iniciais de paragens identificar percursos tendo em conta a topologia das ruas dos percursos permitidos tendo em conta as regras de trânsito.

1.1.1 Tecmic - XTraN Passenger

A Tecmic é uma empresa Portuguesa fundada em 1988 que tem como principal objetivo o desenvolvimento de soluções para a gestão inteligente de frotas. O XTraN Passenger é uma solução completa e poderosa que permite acompanhar em tempo real toda a atividade da frota de transporte públicos, comunicar por voz e dados com o tripulante, determinar e comunicar a

Capítulo 1 - Introdução

previsão de chegada dos próximos autocarros às paragens, assim como informar o utilizador dos transportes públicos sobre tempos de espera através de diversos canais.

O sistema XTraN Passenger recorre ao uso de tecnologias *Estação Flexível de Manufatura* (FMS) como o *Automatic Vehicle Location* (AVL) para obter a localização da sua frota e também a tecnologias *Advanced Traveler Information Systems* (ATIS) para providenciar informação da sua frota aos utilizadores sobre diversos canais.

O XTraN Passenger é constituído por uma plataforma modular da qual fazem parte os seguintes módulos:

Informar, uma plataforma multi-canal que fornece informação de previsões de chegada de forma precisa e em tempo real aos passageiros, através de mensagens SMS, painéis eletrónicos ou da *World Wide Web* (WWW).

Counter, um sistema inteligente de contagem de passageiros que utiliza uma inovadora tecnologia de monitorização de movimentos.

Eco-driver, que permite o controlo de eficiência energética da condução tendo em vista a redução do consumo de combustível e o aumento do conforto e segurança dos passageiros.

Infotainer, que providencia informação dinâmica a bordo do transporte relativa ao serviço, fornecendo entretenimento e publicidade baseada na localização.

1.2 Contexto (ITS, PIS)

O sistema de transporte coletivo desempenha um importante papel na mobilidade das pessoas num meio urbano. Os sistemas de informação nas áreas dos transportes públicos permitem o acesso à informação de rotas, paragens, horários, atrasos, preços entre outras. As empresas que prestam este tipo de serviço habitualmente recebem do cliente (operador de transportes públicos) uma lista com os nomes das paragens ao qual juntam as coordenadas *Global Positioning System* (GPS). Este é um caso habitual descrito pela empresa Tecmic, a qual reportou diversas melhorias necessárias na realização desta tarefa, principalmente em países que se desconhece as suas características geográficas.

1.3 Objetivos

O presente trabalho enquadra-se no desenvolvimento operacional da empresa Tecmic na área dos transportes. A sua atividade nesta área começa com uma lista de paragens de transportes públicos aos quais os técnicos fazem a correspondência com as linhas e rotas para os veículos de transporte.

Dada a globalização do mercado, esta atividade muitas vezes é feita em países estrangeiros dos quais os técnicos não têm o mínimo conhecimento físico do local. Nessa atividade inicial é consumido bastante tempo e por vezes são feitos erros que atrasam o processo de implementação do sistema de gestão dos transportes públicos.

O desafio do meu projeto é em linhas gerais tentar automatizar este processo na fase inicial com a preocupação de seguir as recomendações de normalizações para se exportar facilmente os dados de um sistema para outro, como por exemplo a representação de informação no Google Maps, requisito considerado fundamental para a maioria dos operadores de transporte.

Assim, usando uma plataforma *Open Source Software (OSS) Geographic Information System (GIS)* e tendo como ponto de partida a informação do levantamento das paragens de transportes públicos (coordenadas GPS e nomes) pretende-se desenvolver uma aplicação que permita apoiar no desenvolvimento da rede inicial de transportes, como a identificação de todas as possíveis rotas da frota, utilizando os recursos de informação existentes na Tecmic. Aproveitando o sistema existente que está dentro das recomendações e normas internacionais.

Pretende-se a criação de rotas e percursos dos autocarros, sendo atualmente uma operação demorada feita por técnicos de informática. Para além de este processo ser demorado, podem resultar em diversos erros que comprometem as etapas seguintes, de implementação do sistema de informação.

Outra questão em análise é a necessidade que a Tecmic tem em colocar a informação no Google Maps. É notória a importância das empresas atuais, poderem disponibilizar a sua informação no Google Maps, permitindo o acesso de informação importante a divulgar e assim garantindo a interoperabilidade dos dados. A Figura 1 ilustra um diagrama do modelo proposto desta aplicação (Identificação e Representação Automática de Percursos de Autocarros (IRAPA)).

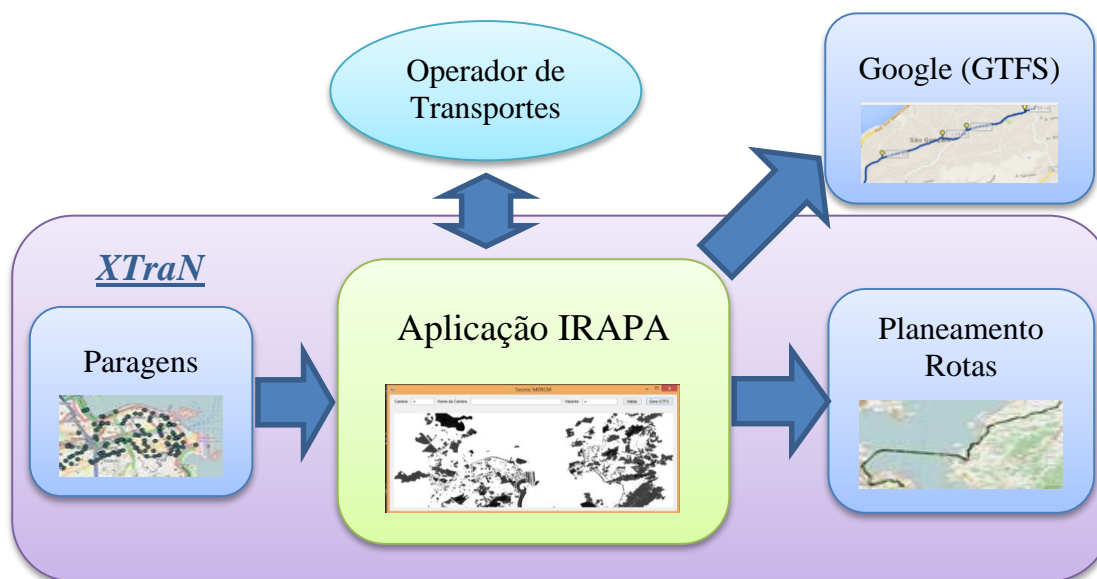


Figura 1: Diagrama do modelo proposto

1.4 Metodologia

A metodologia usada na execução deste trabalho foi a tradicional sequência "análise, desenho e desenvolvimento".

A análise detalhada do domínio do problema teve o intuito de produzir um modelo conceptual para a informação que se pretende distribuir. Nesta fase não foram consideradas as restrições de implementação, tais como concorrência, distribuição, persistência, ou a forma como o sistema deverá ser desenvolvido. Foram ainda formalmente enumerados os requisitos funcionais que constituem as premissas base da motivação envolvendo este trabalho. No fim desta fase, obteve-se uma descrição completa do que o sistema é capaz de fazer, na forma de um modelo conceptual.

1.5 Organização do Documento

Esta dissertação de mestrado está estruturada em seis capítulos, cujos conteúdos são descritos nos seguintes parágrafos.

Capítulo 1 – Introdução. Uma introdução e análise de requisitos ao tema proposto.

Capítulo 2 - Estado da Arte. Neste capítulo é feito um levantamento das soluções atualmente existentes que contribuem de alguma forma para a solução do problema em causa.

Capítulo 3 – Modelo conceptual do sistema proposto (Análise do Problema). Este capítulo começa por delinear uma estratégia de abordagem ao problema, seguindo de uma enumeração dos principais desafios que foram identificados para este trabalho.

Capítulo 4 - Proposta de Solução. Neste capítulo é feita uma análise preliminar e desenho da abordagem sugerida no capítulo anterior. De forma metódica, e de acordo com os requisitos enumerados, são apresentados os diagramas de Casos de Uso, e os mecanismos base que descrevem o funcionamento do sistema.

Capítulo 5 – Avaliação da solução. Caso de estudo onde se apresenta a forma como foi resolvido o problema em causa, através da aplicação dos conceitos propostos no capítulo anterior no desenvolvimento de um demonstrador.

Capítulo 6 - Conclusões e Trabalho Futuro. É feita uma análise conclusiva e crítica sobre o trabalho feito na ótica dos resultados obtidos, sendo propostas melhorias possíveis para trabalho futuro.

2 Estado da Arte

Este capítulo pretende dar uma visão dos tópicos relacionados com o trabalho, dando ênfase aos principais protocolos de transferência e representação da informação, arquiteturas de sistemas e aplicações de referência atualmente em uso. Assim o presente capítulo é dividido nas secções:

2.1 Interoperabilidade, onde é apresentado o desenvolvimento das ITs e como influenciaram fortemente os transportes públicos na última década. Aborda a definição de como estão capacitadas as organizações em interagirem com vista em objetivos e benefícios comuns, envolvendo a partilha de dados.

Referência a normas que suportem a interoperabilidade dos ITS's, demonstrando o esforço de países na adoção de padrões internacionais relacionados com os transportes públicos.

2.2 Tópicos Relacionados com o Tema do Trabalho. Aqui é apresentado o estudo sobre as arquiteturas de sistemas e aplicações de referência existentes, que foram utilizadas no desenvolvimento deste trabalho.

2.3 Base de dados de suporte ao sistema, apresentando as principais características de utilização, bem como a ferramenta de *routing* utilizada e a forma de interação de ambas.

2.4 Como o trabalho pretende criar um módulo para o sistema XTraN Passenger da Tecmic, nesta secção descrevo o sistema da Tecmic XTraN Passenger, orientado para a integração do trabalho que pretendo desenvolver. Assim apresento os principais componentes tendo em vista a integração do meu trabalho.

2.1 Interoperabilidade

Uma importante tarefa da informática, ligada aos transportes, dos nossos dias implica um trabalho global e o uso de padrões na integração da geografia, mobilidade e economia dos transportes, disponibiliza uma forte interoperabilidade entre os Sistemas de Informação de Passageiros (*Passenger Information Systems* (PISs)). A informática no ramo dos transportes é munida de Sistemas de Transporte Inteligentes (*Intelligent Transportation Systems* (ITSs)), Telemetria de Transporte e Trafego de Rotas (*Road Transport and Traffic Telematics* (RTTT)).

A interoperabilidade dos ITSs requer um melhor entendimento do conceito de interoperabilidade [4], em diferentes contextos (como AEC [5]) e arquitetura de sistemas [6].

Peritos dos ramos da geografia de transportes, planeamento de mobilidade, economia de transportes e informática de transportes, partilham de uma tarefa comum na maximização da utilização dos transportes públicos, no entanto, este objetivo só poderá ser alcançado se existir colaboração entre os protagonistas e as Autoridades dos Transportes Públicos de Passageiros (*Public Passenger Transport Authorities* (PTAs)), com a colaboração dos Fornecedores dos Serviços de Transportes (*Public Transport Service Providers* (PTSPs; operador)) e os seus passageiros. PISs têm sido desenvolvidos em muitos países na última década, impulsionando as arquiteturas dos ITSs ao nível básico do sector [1]. Consequentemente, o uso típico tem sido do ponto de vista do sector da autoridade, normalmente a nível nacional.

Arquiteturas ITSs descrevem os requisitos conceituais e funcionais. Estes requisitos garantem a interoperabilidade se forem implementados na construção do *software*.

Na Europa as infraestruturas das arquiteturas de ITSs (FRAME [\[2\]](#)) são mantidas pela União Europeia (UE), e por muitos dos seus estados membros, que atualmente seguem a estrutura e notação do “FRAME” (Figura 2). Contudo, é difícil implementar uma arquitetura ITS universal que responda às necessidades de todos os estados membros na Europa [\[3\]](#).

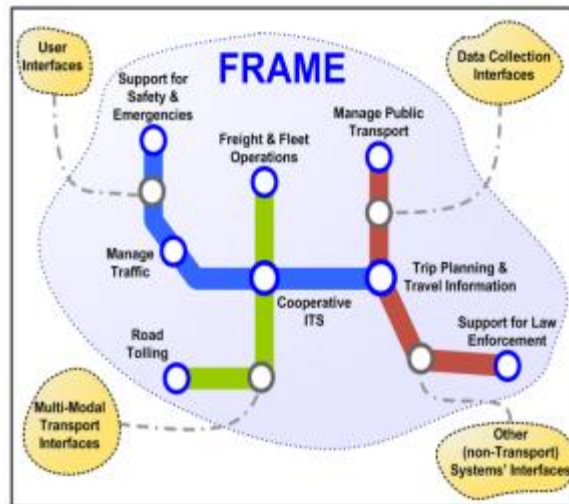


Figura 2: FRAME – A infraestrutura da arquitetura feita para a Europa [\[2\]](#)

2.1.1 Conceito de interoperabilidade

Embora existam muitas interpretações de interoperabilidade, e como o transporte público está sob supervisão governamental, uma definição adequada (como sugerido no estudo da interoperabilidade dos sistemas de informação [\[7\]](#)) pode ser encontrado no documento "*European Interoperability Framework (EIF) for pan-European e-Government Services*" [\[8\]](#). Interoperabilidade é aqui definida como "a capacidade dos sistemas de tecnologia da informação e comunicação (*Information and Communication Technology (ICT)*) e dos processos de negócio que suportam, para a troca de dados e permitir a partilha de informação e conhecimento ". Esta definição foi revista no documento final do Quadro Europeu de Interoperabilidade, publicada em 2011 [\[9\]](#). Neste documento, a interoperabilidade é definida como " a capacidade de organizações díspares e diversas interagirem com vista em objetivos e benefícios comuns, envolvendo a partilha de informações e conhecimentos entre as organizações, através dos processos de negócio que suportam, por meio da troca de dados entre os respetivos sistemas ICT ". Esta nova definição refina a anterior, salientando que as organizações durante interações são heterogêneas, que a sua interação tem um objetivo comum, e que todos eles deverão beneficiar com isso. A definição acima introduz um conceito necessário e suficiente, ainda de uma forma abstrata, pois há implementações que ainda hoje não são seguidas por completo.

PISs existentes não são interoperáveis em geral, por usarem modelos de dados próprios e mensagens definidas para a comunicação *Business to Business (B2B)*.

A interoperabilidade pode ser regulamentada ou baseada em padrões. Consórcios e organizações de desenvolvimento de padrões da indústria, como CEN/ TC 278 (*ITS standardisation* [\[10\]](#)), responderam a este problema através da publicação de normas para a troca de dados B2B interoperáveis. A seção seguinte explora os padrões da indústria mais influentes, *Comité Europeo de Normalización (CEN* [\[11\]](#)) candidatos padrão e normas publicadas em relação aos Transportes Públicos de Passageiros (*Public Passenger Transport (PPT)*).

2.1.2 Normas que suportam a interoperabilidade dos ITSs

Nos países europeus, a adoção de padrões internacionais para o transporte público é um processo em curso e em melhoria constante.

Um subconjunto de padrões ITS é adequado para utilização em PPT. A Figura 3 ilustra os modelos de dados internacionais (TransXChange, NPTG, NaPTAN), e iniciativas da indústria (Google Transit), que influenciou as especificações técnicas europeias publicadas (SIRI, IFOPT), e desenvolvimentos recentes (NeTEX). TransModel, uma norma europeia, tem sido fundamental para a evolução acima mencionada, desde a sua primeira versão em 2004.

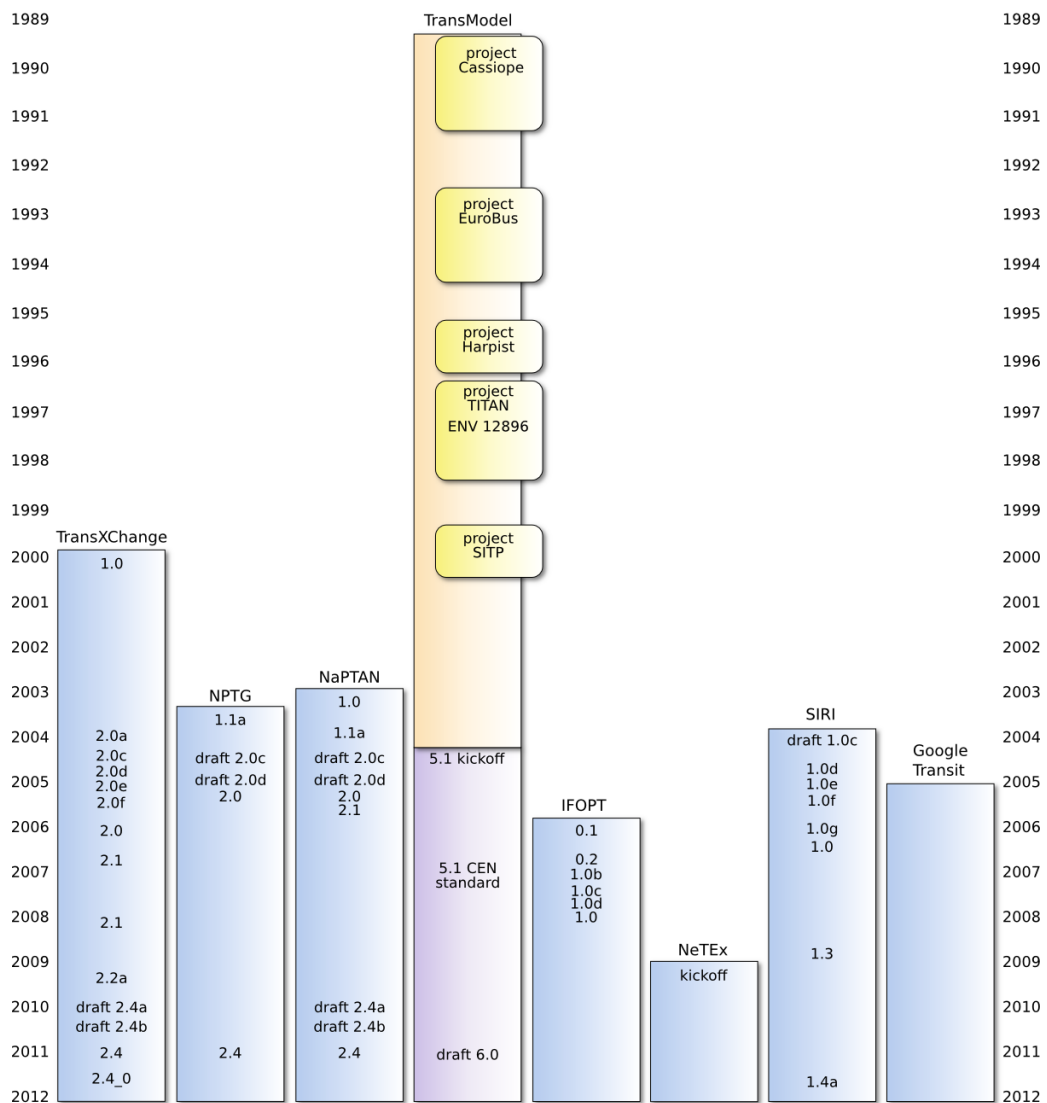


Figura 3: Evolução de padrões internacionais do transporte público de passageiros [19]

Capítulo 2 – Estado da Arte

TransModel versão 5.1 [\[12\]](#), o modelo de dados de referência europeu para informações de transporte público, foi formalmente adotada pelo CEN como norma Europeia (EN12896). Este descreve os conceitos de transportes públicos e estabelece vocabulário técnico preciso para PISs, infraestruturas de rede de transporte, horários de transportes públicos, preçários, informações sobre os passageiros em tempo real, e operações de controlo. Tecnicamente, permite a criação de um banco de dados padrão para facilitar a conectividade entre sistemas de *software* vocacionado para os transportes públicos.

IFOPT (*Identification of Fixed Objects in Public Transport* [\[13\]](#)) é uma especificação técnica (CEN/TS28701) que define um modelo contendo os principais conceitos para objetos fixos associados ao acesso do transporte público (espaços de paragem, pontos de paragem, pontos de acesso, os caminhos de ligação, etc.). Este modelo aborda principalmente a infraestrutura física e equipamentos necessários para os serviços de informação, incluindo descrições explícitas de objetos pelas suas principais características. Descreve como localizar esses objetos no espaço através de coordenadas, e através de ligações com objetos topográficos, com uma clara separação entre a camada de transporte público e a camada topográfica. Permite a designação de responsabilidades da manutenção de dados para cada objeto fixo. A localização geoespacial referencia técnicas de objetos dos transportes públicos (i.e. uso de satélites, equipamentos na estrada para posicionamento) ou técnicas de representação em mapas (projeções) que ficaram fora desta norma. IFOPT baseia-se em modelos de dados nacionais do Reino Unido, Suécia, Alemanha e França. A Tabela 1 apresenta uma comparação de propriedades entre alguns padrões.

Tabela 1: *Comparação de propriedades entre alguns padrões* [\[19\]](#)

Características	TransXChange	NaPTAN	GTFS	NeTex	IFOPT
Operadores de transportes	Sim	Sim	Sim	Sim	Sim
Espaços/Pontos de paragem	Pontos	Pontos	Pontos	Intercâmbio completo	Intercâmbio completo
Rotas	Sim	Sim	Sim	Sim	Sim
Faixas de rodagem	Sim	Sim	Sim	Sim	Sim
Linhas	Sim	Sim	Sim	Sim	Sim
Modelos de viagem	Sim	Sim	Não	Sim	Sim
Horário em tempo real	Sim	Sim	Não	Sim	Sim
Horários de viagens	Sim	Sim	Sim	Sim	Sim
Intercâmbio de viagens	Sim	Sim	Não	Sim	Sim
Condições de disponibilidade	Sim	Sim	Limitado	Sim	Sim
Dias de serviço	Proposto	Proposto	Não	Sim	Sim
Trilho de suporte completo	Não	Não	Não	Sim	Sim
Tarifas	Só etapas	Só etapas	Zonas básicas	Modelo básico	Modelo básico
Administração distribuída	Sim	Sim	Não	Sim	Sim
Trocas peer-to-peer	Sim	Sim	Não	Sim	Sim
Baseado em modelos	Sim	Sim	Não	Sim	Sim
Elementos reutilizáveis	Sim	Sim	Não	Sim	Sim
Fraco acoplamento	Sim	Sim	Não	Sim	Sim

Standard Interface for Real-Time Information (SIRI [\[14\]](#)) é uma especificação de interface para a recolha e intercâmbio de informações operacionais em tempo real sobre transportes públicos. Informações operacionais em tempo real contribuem para uma melhor compreensão do funcionamento geral do transporte público e fornece mais informações para os passageiros (i.e. painéis de informação). As informações podem ser trocadas entre veículos, objetos de infraestruturas da rede fixa, centros de controlo de transportes públicos e centros de informação para os passageiros para garantir a integridade dos *Network Timetable and Exchanges* (NeTex [\[15\]](#)), fornece uma ação complementar para os serviços de rede da SIRI e atualizações das alterações nos transportes públicos. Contém um conjunto abrangente de modelos *Unified Modeling Language* (UML) e esquemas *eXtensible Markup Language* (XML) para a execução dos serviços de transporte público multimodais. É composto de três partes: um sistema de troca de topologia da rede de transportes públicos, um sistema de troca de horários, e um esquema de troca de informações sobre os preçários. Existem exemplos de protótipos que estão focados em

volta dos modos de transporte (i.e. Autocarro), estando geograficamente limitados a PPTs nacionais. No entanto, a principal premissa das normas é alcançar a interoperabilidade técnica e multimodal em relação ao transporte internacional de passageiros. Idealmente, PISs nacional podem ser integrados para fornecer um sistema global que é capaz de fornecer dados e informações melhor do que o PISs individual operando em áreas geograficamente designadas. Até ao momento, exemplos de trabalho só foram disponibilizados a partir desses campos motivados por grupos de interesse fechados, com fortes vertentes econômicas, tais como cadeias logísticas internacionais [16] ou alianças ferroviários internacionais usando único PISs (como as DB, <http://reiseauskunft.bahn.de>, baseado em HAFAS, <http://www.hacon.de/hafas-en>). Estes sistemas utilizam soluções técnicas (globais) centralizados. Apenas existem alguns protótipos de PISs, com base na interoperabilidade entre a fraca acoplagem dos Sistemas inteligentes (*Intelligent Systems* (ISs) [17]).

2.1.3 Requisitos para a captação de normas

As bases de dados de transportes públicos, providencia dados aos ITSs nacionais, numa distribuição pela Europa num futuro próximo, terão de armazenar todos os dados referentes aos transportes públicos dentro dos seus países. Teoricamente, isso significa que cada país Europeu teria o seu próprio ITS nacional. É expectável que a maioria dos ITSs na Europa terá de adaptar os seus modelos de dados com vista nas alterações das topologias, da rede de transportes públicos, horários e informação sobre as tarifas. Normas técnicas apropriadas para atingir estes objetivos são as de IFOPT e NeTEx.

Os PISs que contribuem para PIS Europeia (SIPE) podem preservar o seu legado dos esquemas de base de dados, mas deverão estar preparados para lidar com a interoperabilidade com outros PISs com a finalidade de contribuir para os planos de viagens incluindo vários meios de transporte. A troca de documentação terá de ir ao encontro dos esquemas de IFOPT (Figura 4, lado esquerdo) e NeTEx (Figura 4, lado direito), que são compostos por:

1. Um modelo administrativo (IFOPT):
 - a. Proporcionaria um modelo organizacional para a atribuição de responsabilidade pela criação e manutenção de dados como um processo colaborativo envolvendo PTAs distribuídos e operadores de transportes públicos. Incluiria gestão “*namespace*” para gerir a emissão descentralizada de identificadores únicos.
2. Modelo para local de paragem (IFOPT):
 - a. Fornecem uma descrição detalhada de lugar de paragem dentro de transportes públicos: espaços de paragem (ou estações ou áreas) com pontos de paragem (ou paragens ou posições de paragens ou cais), os seus nomes e localizações geográficas para todos os tipos de paragens: Autocarro, elétrico, comboio, metro, aviões, barcos etc. Incluiria pontos físicos de acesso em relação a veículos e os caminhos entre os pontos (incluindo a acessibilidade). Identificadores de lugar de paragem seriam únicos dentro de cada país. Um prefixo país poderia ser usado para garantir a exclusividade global.
3. Um modelo topográfico (IFOPT):
 - a. Proporcionaria uma representação topográfica dos assentamentos, entre os quais os passageiros viajam. Seria usado para associar elementos de paragem e estação com os nomes e conceitos topográficas necessárias para apoiar as funções de planeamento da viagem, etc.
4. Um modelo de pontos de interesse (IFOPT):

- a. Descreveria as estruturas de pontos de interesse, incluindo pontos físicos de acesso, ou seja, entradas. Seria fornecer um modelo para hierarquia de classificação por pontos de interesse padronizado - um meio de fornecer taxonomia sobre diferentes tipos de ponto de interesse relevante para o planeamento da viagem.
5. Um modelo de serviços programados (NeTeX):
- a. Indicam pontos de paragem programada, ligações de serviços entre pontos de paragem programada.
 - b. Padrões de serviço: atribuições de paragem, rotas, padrões temporais, padrões de viagem, tipos de horários.
 - c. Linhas.
 - d. Os horários, calendários de serviço.

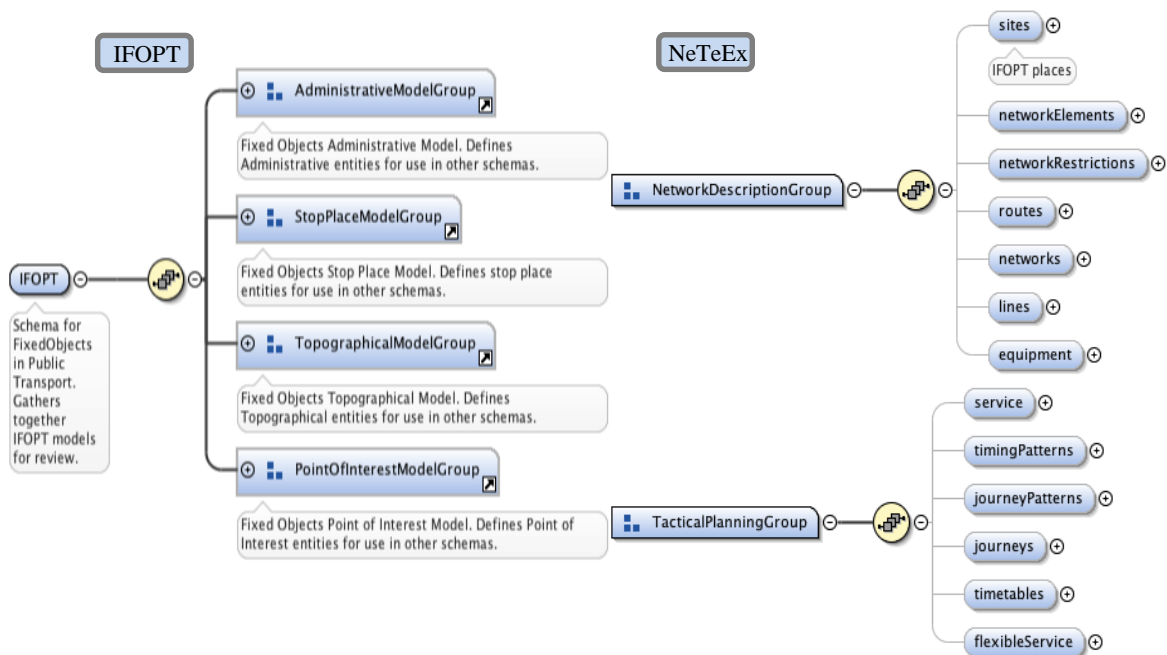


Figura 4: Modelos IFOPT e NeTeX [19]

2.2 Tópicos Relacionados com o Tema do Trabalho

2.2.1 Advanced Public Transportation Systems (APTS)

Para desenvolver um sistema de informação geográfica *Geographic Information System* (GIS) que permita modelar todo o negócio de um sistema de transportes públicos, é necessário recorrer ao uso de tecnologias dos sistemas *Advanced Public Transportation Systems* (APTS). O principal propósito destas tecnologias é providenciar aos utilizadores uma quantidade superior de informação, com maior facilidade de acesso, relativa à rede de transportes públicos. Entre outras categorias dos sistemas APTS, destacam-se duas:

- *Advanced Traveler Information Systems (ATIS)*: Utilização de tecnologias de informação para fornecer informações dos transportes públicos aos utilizadores em casa, no trabalho, na beira da estrada, no próprio transporte, nas paragens, etc. Os utilizadores podem aceder em tempo real a horários e informações de congestionamento através de telefones, painéis de mensagem variável, quiosques ou através da *World Wide Web (WWW)*. O principal objetivo é providenciar informação mais completa e de diversas formas para que os utilizadores, habituais ou ocasionais, possam escolher de forma apoiada o seu transporte.
- *Fleet Management Systems (FMS)*: Gestão e controlo da frota em tempo real recorrendo ao uso de tecnologias de informação. A tecnologia mais usada para os transportes públicos é o *Automatic Vehicle Location (AVL)*. Este sistema é constituído por vários terminais a bordo de cada veículo, que permitem a medição em tempo real da posição utilizado o sistema *Global Positioning System (GPS)*. Posteriormente, essa informação é transmitida para um sistema central.

Devido à grande evolução tecnológica que possibilitou o uso de dispositivos móveis como ferramentas para o acesso à Internet e com capacidades de realizar tarefas com alguma carga computacional, é agora possível aceder a estes sistemas de informação num contexto móvel. A grande vantagem da utilização de dispositivos móveis, para além da sua mobilidade, advém da utilização do GPS para identificar a posição do utilizador. Desta forma, o contexto do utilizador é transmitido para o sistema central facilitando a eliminação de informação irrelevante.

2.2.2 Geographic Information System (GIS)

Um Sistema de Informação Geográfica (SIG ou GIS - *Geographic Information System*, do acrónimo/acrônimo inglês) é um sistema de hardware, software, informação espacial, procedimentos computacionais e recursos humanos que permite e facilita a análise, gestão ou representação do espaço e dos fenómenos que nele ocorrem.

Fitz (2008) conceitua SIG como um sistema constituído por um conjunto de programas computacionais, o qual integra dados, equipamentos e pessoas com objetivo de recolher, armazenar, recuperar, manipular, visualizar e analisar dados espacialmente referenciados a um sistema conhecido de coordenadas.

Existem vários modelos de dados aplicáveis em SIGs (Sistemas de Informação Geográfica). Por exemplo, o SIG pode funcionar como uma base de dados com informação geográfica (dados alfanuméricos) que se encontra associada por um identificador comum aos objetos gráficos de um mapa digital. Desta forma, assinalando um objeto pode-se saber o valor dos seus atributos, e inversamente, selecionando um registro da base de dados é possível saber a sua localização e apontá-la num mapa.

O SIG separa a informação em diferentes camadas temáticas e armazena-as independentemente, permitindo trabalhar com elas de modo rápido e simples, permitindo ao operador ou utilizador a possibilidade de relacionar a informação existente através da posição e topologia dos objetos, com o fim de gerar nova informação.

Os modelos mais comuns em SIG são o modelo *Raster Maps* ou matricial e o modelo vetorial. O modelo de SIG matricial centra-se nas propriedades do espaço, compartimentando-o em células regulares (habitualmente quadradas, mas podendo ser retangulares, triangulares ou hexagonais). Cada célula representa um único valor. Quanto maior for a dimensão de cada célula (resolução) menor é a precisão ou detalhe na representação do espaço geográfico.

No caso do modelo de SIG vetorial, o foco das representações centra-se na precisão da localização dos elementos no espaço. Para modelar digitalmente as entidades do mundo real utiliza-se essencialmente três formas espaciais: o ponto, a linha e o polígono.

2.2.3 OpenStreetMap


Fazendo parte integrante do sistema XTraN Passenger, como referido na secção 1.4, há a necessidade de apresentar o OpenStreetMap (OSM [\[18\]](#)), sendo este um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo. Este sistema de mapeamento faz parte das escolhas de utilização da Tecmic, implicando que também o sistema deste trabalho se baseie na utilização dos recursos do OSM.

Os mapas são criados usando dados de recetores GPS portáteis, fotografias aéreas e outras fontes livres. Tanto as imagens obtidas por processamento de dados e os dados estão disponíveis sob uma licença *Open Database License* (ODbL). Utilizadores registados podem carregar os históricos do GPS e editar os dados usando as ferramentas disponíveis.

Este sistema colaborativo pode criar e distribuir dados geográficos livremente para todo o mundo. Este projeto foi iniciado porque a maioria dos mapas disponibilizados na internet possui restrições legais ou técnicas relativas à utilização, limitando seu uso de forma realmente criativa e produtiva.




De seguida é apresentada uma descrição dos dados OSM.

2.2.3.1 Nó (Node)

Um Nó (Node): 

- Representa um ponto específico sobre a superfície da terra definida por sua latitude e longitude. Cada Nó compreende pelo menos um número de identificação e um par de coordenadas.
- Os Nós podem ser utilizados para definir as características de pontos independentes. Por exemplo, um Nó pode representar um banco de parque ou um poço de água.

2.2.3.2 Via (Way)

Uma Via (Way):   

- É uma lista ordenada de 2 a 2.000 Nós, que define um polígono. As Vias são usadas para representar características lineares tais como rios e estradas.
- Vias também podem representar os limites das zonas (polígonos sólidos), tais como edifícios ou florestas. Neste caso, o primeiro e último Nó do caminho vai ser o mesmo. Isso é chamado de " *closed way*".


Na Tabela 2 seguinte estão representados os vários tipos de vias a ter em conta quando se constrói uma representação gráfica dos dados OSM. Nesta tabela os tipos de vias estão listadas por ordem de importância.

Tabela 2: Tabela de tipo de vias do OSM [\[18\]](#)

OSM tag	Car Routing	Pedestrian Routing	Comment
highway=motorway	yes	no	
highway=motorway link	yes	no	

highway=trunk	yes	yes	
highway=trunk link	yes	yes	
highway=primary	yes	yes	
highway=primary link	yes	yes	
highway=secondary	yes	yes	
highway=secondary link	yes	yes	
highway=residential	yes	yes	
highway=residential link	yes	yes	
highway=service	yes	yes	No through traffic
highway=tertiary	yes	yes	
highway=tertiary link	yes	yes	
highway=road	yes	yes	
highway=track	yes	yes	
highway=unclassified	yes	yes	
highway=undefined	yes	yes	
highway=unknown	yes	yes	
highway=living street	yes	yes	No through traffic
highway=private	yes	yes	No through traffic
highway=footway	no	yes	
highway=pedestrian	no	yes	
highway=steps	no	yes	
highway=bridleway	no	no	
highway=construction	no	no	
highway=cycleway	no	no	
highway=path	no	no	
highway=bus guideway	no	no	
route=ferry	yes	yes	

2.2.3.3 Relação (Relation)

A Relação (*Relation*): 

- É uma estrutura de dados “*multi-purpose*” que documenta uma relação entre dois ou mais elementos de dados (nós, caminhos, e/ou outras relações). Os exemplos incluem:
 - Uma relação rota, que lista os caminhos que formam uma grande (numerada) estrada, uma ciclovia, ou uma rota de veículos de passageiros.
 - A restrição de inversão de marcha diz não ser possível a mudança de direção numa determinada Via de apenas um sentido.
 - Um multipolígono que descreve uma área (cujo limite é a "forma externa") com espaços interiores (os “caminhos interiores”).

2.2.3.4 Etiqueta (Tag)

Uma Etiqueta (*Tag*): 

- Todos os tipos de elementos de dados (Nós, Vias e Relações) podem ter Etiquetas. Estas descrevem o significado do elemento particular à qual estão ligados.
- Uma Etiqueta é constituída por dois campos de texto de formato livre; uma “chave” e um “valor”. Cada um destes são seqüências de caracteres Unicode até 255 caracteres. Por exemplo, “*estrada = residencial*” define a maneira como uma estrada cuja principal função é dar acesso às casas das pessoas.

Capítulo 2 – Estado da Arte

A chave descreve uma ampla classe de recursos, o valor especifica os detalhes dos recursos classificados pela chave. Se forem necessários vários valores para apenas uma chave, podem ser utilizadas as funções “*semi-colon*” e “*value separator*” para garantir o registo do valor.

Na Tabela 3 são apresentados alguns exemplos de como na prática são utilizados os conjuntos chave/valor:

Tabela 3: Exemplos de conjuntos chave/valor do OSM [18]

OSM tag	Comment
highway=residential	Uma etiqueta com uma chave de "estrada", e um valor de “residencial” deve ser utilizado de forma a indicar um caminho ao longo do qual as pessoas vivem.
name=Park Avenue	Uma etiqueta para o qual o campo de valor é utilizado para transmitir o nome da rua em particular.
maxspeed=50	Uma etiqueta cujo valor é uma velocidade numérica em km/h (ou em milhas por hora, se a unidade é fornecida com um sufixo «mph»). As unidades métricas são o padrão.
maxspeed:forward=*	Uma chave que inclui um espaço para nome para 'maxspeed' para distinguir ainda mais o seu significado.
name:de:1953-1990=Ernst-Thälmann-Straße	Uma etiqueta com "Nome" chave sufixo “namespaces” para especificar o nome do local na Alemanha, que era válido entre os anos 1953 a 1990.

Cada etiqueta descreve um atributo geográfico do recurso que está a ser mostrado por esse nó, forma ou relação.

O sistema livre de etiquetas do OSM permite que o mapa inclua um número ilimitado de atributos que descrevem individualmente cada recurso. A comunidade de utilizadores concorda com certas combinações de chave e valor para as marcas mais comuns, que atuam como normas informais. No entanto, os utilizadores podem criar novas etiquetas para melhorar o estilo do mapa ou para apoiar as análises que se baseiam em atributos dos recursos em falta, no mapeamento prévio. A maioria dos recursos pode ser descrita usando apenas um pequeno número de etiquetas, como um caminho, e talvez também um nome. Mas, uma vez que este é um mapa construído por toda a comunidade mundial, pode haver muitos tipos diferentes de recursos no OSM, quase todos eles descritos por etiquetas. Para uma melhor compreensão das etiquetas existentes, são disponibilizados os detalhes das etiquetas em forma de documentação de recursos.

Se não encontrar uma etiqueta apropriada na lista da documentação, o utilizador pode criar uma nova etiqueta adequada ao local em causa. Com o tempo, pode-se achar que o nome da etiqueta deve ser alterado para se encaixar com algum consenso mais amplo. No entanto, muitas boas etiquetas foram utilizadas pela primeira vez e documentadas mais tarde.

Na Tabela 4 são apresentados alguns recursos que são muitas vezes utilizados para encontrar uma etiqueta apropriada ou explorar a sua utilização.

Tabela 4: Exemplos de recursos do OSM [18]

OSM tag	Comment
Map Features	Lista de etiquetas aceites agrupados por significado chave
How to map a	Lista alfabética de objetos do mundo real
Taginfo	Um local para explorar o uso da etiqueta atual no banco de dados do OSM, incluindo os valores das etiquetas que não são necessariamente documentados (mas inclui links para o “wiki” se houver uma documentação para uma etiqueta)
Wiki	Pesquisa por “wiki” e navegar pelas categorias das etiquetas (“tags”)
TagFinder	Motor de busca de texto para marcas OSM (Também os serviços web disponíveis)
Look how	Visualizar como um objeto semelhante de outro lugar é mapeado

Para fazer uso destas funcionalidades disponíveis pelo OSM, relacionado com o planeamento de rotas, é apresentado na Tabela 5, algumas etiquetas de atributos importantes.

Tabela 5: Etiquetas importantes para o planeamento de rotas [18]

OSM tag	Car Routing	Pedestrian Routing	Comment
oneway=yes	yes	no	Also true/1/no/false/-1 values
oneway=reversible	no	no	
junction=roundabout	yes	no	Infer Clockwise Direction Of Flow
maxspeed=*	no	no	Speed Limit (assume kph except where mph given)
bridge=yes	yes	yes	Also true/1/viaduct
tunnel=yes	yes	yes	Also true/1
surface=paved	yes	yes	No Speed Penalty. Also cobblestone/asphalt/concrete
surface=unpaved	yes	yes	Speed Penalty. Also dirt/grass/mud/earth/sand
surface=*	yes	yes	Speed Penalty (all other values)
access=private	yes	yes	Entry/exit only access
access=no	no	no	No public access

2.2.4 OpenLayers

Como referido na secção 1.4, é utilizada esta biblioteca para a integração de aplicações no sistema XTraN Passenger, adotando a OpenLayers como a ferramenta de seleção na integração deste trabalho, pois este sistema faz parte das escolhas de utilização da Tecmic, implicando que também o sistema deste trabalho se baseie na utilização dos recursos da OpenLayers.

OpenLayers é uma biblioteca JavaScript para exibir dados de mapas na maioria dos navegadores modernos, sem dependências do lado do servidor.

OpenLayers implementa uma *Application Programming Interface* (API) JavaScript para aplicações geográficas baseadas na construção *Rich Internet Application* (RIA), semelhantes aos do Google Maps e o MSN APIs do Virtual Earth, com uma diferença importante - OpenLayers é um *software* livre, desenvolvido por e para a comunidade de OSS (Figura 5).

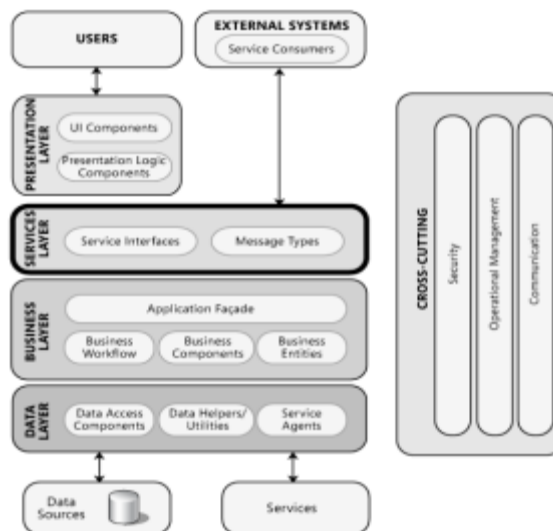


Figura 5: Layers funções de negócios e serviços [24]

Quando uma aplicação deve prestar serviços a outras aplicações, bem como características de execução para apoiar os clientes diretamente, a abordagem comum é usar uma camada de serviços que expõe a funções de negócios da aplicação, como ilustrado na Figura 4. A camada de serviços fornece efetivamente uma visão alternativa que permite aos clientes usar um canal diferente para aceder à aplicação.

2.2.5 General Transit Feed Specification (GTFS)

A pesquisa de como dispor os dados no Google Maps, respondendo a um dos requisitos da Tecmic, resulta no tratamento que os dados necessitam para serem integrados no Google Maps. General Transit Feed Specification (GTFS) [21] define um formato comum para horários de transportes públicos e informações geográficas associadas. Os feeds GTFS permitem que as empresas de transportes públicos publiquem as suas informações e que proporcionem a criação de aplicativos que utilizem esses dados com interoperabilidade.

Um feed GTFS é composto por uma série de arquivos de texto guardados num arquivo ZIP. Cada arquivo modela um aspeto específico das informações sobre o transporte público: paragens, trajetos, viagens e outros dados relativos a horário. Os detalhes de cada arquivo são definidos na referência GTFS (Tabela 6).

A empresa de transportes públicos pode produzir um feed GTFS para partilhar as suas informações sobre os transportes públicos com gestores, que criam ferramentas que usam os feeds GTFS para incorporar informações de transportes públicos nas suas aplicações. A GTFS pode ser usada para oferecer recursos mais poderosos a gestores de viagens, autores de tabelas e uma ampla variedade de aplicações que, de alguma forma, usam informações sobre transportes públicos.

Tabela 6: Especificação dos arquivos GTFS juntamente com seu conteúdo associado [21]:

Nome do arquivo	Obrigatório	Define
agency.txt	Obrigatório	Uma ou mais agências de transporte público que fornecem os dados nesse feed.
stops.txt	Obrigatório	Locais individuais em que os veículos pegam ou deixam passageiros.
routes.txt	Obrigatório	Trajetos do Google Transit. Um trajeto é um grupo de viagens exibidas aos passageiros como um único serviço.
trips.txt	Obrigatório	As viagens de cada trajeto. Uma viagem é uma sequência de duas ou mais paradas que ocorrem em um horário específico.
stop_times.txt	Obrigatório	Horários de partida e chegada dos veículos em paradas específicas em cada viagem.
calendar.txt	Obrigatório	Datas para IDs de serviço que usam uma programação semanal. Especificam quando o serviço começa e termina, bem como os dias da semana em que o serviço está disponível.
calendar_dates.txt	Opcional	Exceções para IDs de serviço definidos no arquivo calendar.txt . Se o arquivo calendar_dates.txt inclui TODAS as datas de serviço, ele pode ser especificado no lugar do calendar.txt.
fare_attributes.txt	Opcional	Informações sobre tarifas dos trajetos de uma empresa de transporte público.
fare_rules.txt	Opcional	Regras para implementação das informações de tarifa dos trajetos de uma empresa de transporte público.
shapes.txt	Opcional	Regras para desenhar linhas em um mapa para representar os trajetos de uma empresa de transporte público.
frequencies.txt	Opcional	Intervalo entre as viagens nos trajetos com frequência variável de serviços.
transfers.txt	Opcional	Regras para conexões em pontos de baldeação entre os trajetos.
feed_info.txt	Opcional	Informações adicionais sobre o feed, incluindo editor, versão e informações sobre validade.

2.2.6 Differential GPS (DGPS)

Uma abordagem para a identificação das paragens é fazer uma medição de campo mais precisa, garantindo assim uma correta identificação do local geográfico da paragem.

Para uma melhor identificação geográfica de um local, existe um sistema de leitura *Differential* GPS (DGPS [26]), que é uma forma de leitura de pontos geográficos pensada para aumentar a

Capítulo 2 - Estado da Arte

precisão na leitura da posição GPS, conseguindo melhorias de 15m para 10cm, nas melhores implementações.

O sistema DGPS envolve a cooperação entre dois receptores, um que é imóvel e outro que é móvel a fazer medições de posições geográficas. O receptor que está estacionário é a chave da solução. Este fixa todas as medições efetuadas pelo satélite, num determinado local de referência, comparando o erro das leituras de GPS, com a posição exata conhecida.

Os receptores GPS utilizam a medição do tempo dos relógios atômicos, de pelo menos quatro satélites, para definir uma posição. Cada uma dessas medições de tempo vai conter um erro (atraso), dependendo do tipo de oposição à transmissão encontrada na viagem até à posição terrestre.

Uma vez que cada sinal de sincronismo, utilizado no cálculo da posição, contém um erro, esse cálculo vai ser uma composição desses erros.

Os satélites encontram-se a uma distância significativa da terra, fazendo com que as distâncias numa viagem terrestre sejam insignificantes, minimizando os erros de tempo. Se dois receptores se encontram perto um do outro (relativo à distância do satélite à terra), a poucas centenas de quilômetros, o sinal chega a ambos praticamente ao mesmo tempo, pois viajaram através a mesma fatia atmosférica, e assim vão ter praticamente os mesmos erros.

O DGPS pode eliminar todos os erros comuns entre dois receptores de referência (estáticos) e o receptor móvel. Todos os erros, exceto, erros de caminhos múltiplos (porque ocorrem em torno do mesmo receptor) e quaisquer erros do receptor (porque são exclusivos do receptor).

A ideia por trás do DGPS é a de ter um receptor a medir os erros de temporização, e em seguida, fornecer informações da correção, aos outros receptores que estão itinerantes ao redor. Dessa forma praticamente todos os erros podem ser eliminados do sistema, mesmo o erro *Selective Availability* (SA) que adiciona um erro intencional provocado pelo *Department of Defense* (DoD).

Colocando um receptor de referência num ponto que está georreferenciado com precisão e mantê-lo lá. Este receptor de referência recebe o mesmo sinal de GPS como o receptor móvel, mas em vez de trabalhar como um receptor GPS normal, utiliza a sua posição conhecida, para calcular a medição do tempo da viagem do sinal de GPS, e compara-o com o que está a ser recebido. A diferença é o fator de correção de erro. O receptor transmite essa informação de erro, ao receptor móvel para que possa ser usada na correção das suas medições.

Uma vez que o receptor de referência não tem forma de saber qual dos satélites disponíveis ao receptor móvel, este executa o procedimento a todos os satélites visíveis e calcula os erros para cada um deles. De seguida, codifica essas informações num formato padrão e transmite para o receptor móvel, como ilustrado na Figura 6, com um exemplo utilizado em navegação marítima.

Na verdade, os receptores GPS não transmitem as correções por si mesmos. Estão ligados a radiotransmissores separados, para fazer esse trabalho. O receptor móvel obtém a lista completa de erros e aplica as correções dos satélites que estão a usar.

Transmissões de erro não incluem apenas o erro de temporização para cada satélite, incluem também a taxa de variação de erro. Dessa forma, o receptor móvel pode interpolar a sua posição entre as atualizações.



Figura 6: Esquema de funcionamento do DGPS [27]

2.2.7 Algoritmo de classificação

Para a melhor classificação do local de identificação de uma paragem, teremos de considerar os algoritmos de classificação que melhor se adaptam a esta situação.

No caso do sistema utilizado pela Tecmic, apenas é feita uma análise em função da distância. Ou seja, o local da paragem é considerado com um raio envolvente com uma determinada distância, fazendo com que a decisão de o ponto geográfico registado pelo veículo pertencer à paragem, depende apenas de se encontrar dentro da área geográfica desse raio.

Esta abordagem implica um problema quando temos duas paragens muito próximas, fazendo com que o raio envolvente coincida e provocando assim uma indecisão, quando o valor de registo do veículo pertencer à área comum das duas paragens.

Mantendo a metodologia de classificação baseada na distância, foi feita uma análise com o objetivo de melhorar a classificação na identificação da paragem.

2.2.7.1 Classificação em função de distância

Os aspetos de suporte nas instâncias a considerar é o fato de não ser necessário a construção de um modelo à-priori do conjunto de treino, pois isso implicaria uma sobrecarga de processamento. Será considerada a forma “lasy learning”, que foca a classificação apenas quando surge uma nova instância.

Recorre-se à noção de “função de distância” para determinar a distância entre duas quaisquer instâncias.

Para a classificação de uma nova instância do valor, escolhe-se a classe do(s) exemplo(s) de treino mais próximo(s) desse valor.

São apresentadas de seguida, algumas funções de distância [22].

Sejam duas instâncias A e B com atributos, respetivamente a_1, a_2, \dots, a_n e b_1, b_2, \dots, b_n :

- Distância Euclidiana (só domínios numéricos):

$$(\sum_{i=1..n} (a_i - b_i)^2)^{1/2} \quad [1]$$

- Distância “em linha reta entre dois pontos”;
- Para comparação de distâncias não é necessário o cálculo da raiz quadrada;
- Variações incluem potências maiores que 2; aumentar potência aumenta influência das maiores diferenças (à custa das menores).

- Quadrada (mesmo que Euclidiana mas sem cálculo da raiz quadrada).
- Distância de Manhattan (só domínios numéricos):

$$\sum_{i=1..n} |a_i - b_i| \quad [2]$$

- Distância “em blocos entre dois pontos”.

- Distância de Hamming (domínios numéricos e nominais):

$$\sum_{i=1..n} h(a_i, b_i), \text{ onde } h(x, y) \begin{cases} 0, & \text{se } x = y \\ 1, & \text{se } x \neq y \end{cases} \quad [3]$$

- Número de posições em que A e B diferem entre si.

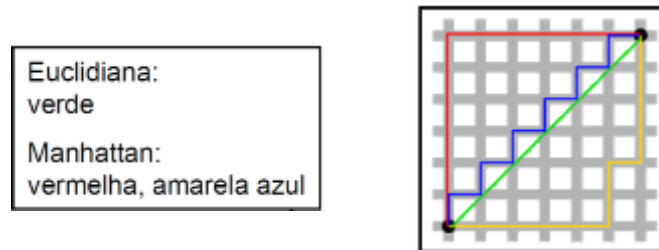


Figura 7: Distância Euclidiana e Manhattan [22]

Para procurar os vizinhos mais próximos de uma qualquer instância podemos calcular a distância a cada um dos exemplos e escolher a menor dessas distâncias. Duas das funções de distância mais utilizadas estão ilustradas na Figura 7.

Calcular distância a todos os exemplos tem complexidade temporal linear no número de exemplos, ou seja, o tempo aumenta proporcionalmente com o número de exemplos.

2.2.7.2 Estrutura em árvore (KD-Tree)

Para reduzir o tempo de procura pode organizar-se o conjunto de treino numa estrutura do tipo árvore. A KD-Tree [22] é uma árvore binária adequada para representar pontos num espaço de dimensão K (daí o KD – “*K Dimensional*” *space*), sendo K o número de atributos de cada ponto. Uma KD-Tree (árvore KD) é uma árvore binária, onde cada nó tem no máximo dois descendentes, subárvore da esquerda e subárvore da direita.

Cada nó representa um exemplo e uma dimensão (ou eixo, direção), idêntica à da árvore de pesquisa binária:

- Esquerda: exemplos com valores menores ou iguais nessa dimensão.
- Direita: exemplos com valores maiores nessa dimensão.

Para construir a árvore é preciso escolher, para cada nó, o exemplo que esse nó irá representar, e a dimensão pela qual se separam as subárvores (esquerda e direita)

- Para escolher a dimensão (ou eixo), pode-se calcular a variância dos dados em cada eixo (dimensão) individualmente e escolher o eixo (dimensão) com maior variância (uma estratégia “pesada”). Ou “rodar” nos atributos de acordo com a profundidade

(*depth*) do nó, ou seja, resto da divisão inteira da profundidade pelo número de atributos (uma estratégia “mais leve”).

- Para selecionar o valor (e exemplo) considerar a mediana, ordenar os exemplos pela dimensão escolhida e escolher o ponto (mediana) que separa a metade inferior da superior. Esta estratégia pode originar partições do tipo “retângulo fino e comprido”. Ou calcular a média e selecionar o exemplo mais próximo, implicando mais cálculo (é “mais pesado”) do que na mediana. Mas pode originar partições mais “quadradas” do que as da mediana.

Para usar a KD-Tree na procura de vizinhos mais próximos, percorre-se a KD-Tree até uma folha (dado um novo exemplo), a folha não é necessariamente o vizinho mais próximo, mas é uma boa primeira aproximação. Um melhor vizinho tem que estar a menor distância do que essa folha. Para ver se existe um melhor vizinho teremos de verificar se o nó irmão tem um melhor vizinho, retroceder ao nó pai e verificar se é um melhor vizinho. Verificar se o irmão do nó pai é melhor vizinho. Se num novo caminho um nó é melhor vizinho ver as suas subárvores neste percurso ir sempre guardando o melhor nó até ao momento.

2.2.7.3 Algoritmo kNN

Após uma análise de vários algoritmos de classificação, optou-se por utilizar o KNN pela sua simplicidade de implementação e principalmente pelo resultado positivo aplicado à questão de classificação dos locais das paragens apresentado por este trabalho [34].

Até agora neste Capítulo foram mencionados mecanismos que o algoritmo KNN tem como suporte, para uma melhor compreensão do algoritmo de classificação.

De seguida é apresentado a essência do algoritmo KNN [22], e do potencial deste tipo de classificação:

Assumindo um conjunto de treino com instâncias de forma $\langle X, f(x) \rangle$, onde X são valores de atributos e $f(x)$ é o valor da classe de X .

- Se a função f é discreta, então $f: \mathbb{R}^n \rightarrow V = \{v_1, \dots, v_n\}$
- Se a função f é contínua, então $f: \mathbb{R}^n \rightarrow \mathbb{R}$

Dada a instância, x_c , para classificar, pesquisar, no conjunto treino, as K instâncias, x_1, x_2, \dots, x_k , mais próximas de x_c (usando o KD-Tree vai melhorar desempenho na pesquisa).

Para funções f discretas, após encontrar as K instâncias x_1, x_2, \dots, x_k , devolver (“regra de maioria”):

$$f(x) = \arg \max_{v \in V} \#\{X_i: f(x_i) = v\} \quad [4]$$

Para funções f contínuas, após encontrar as K instâncias x_1, x_2, \dots, x_k , ou seja, classifica como a média dos K exemplos mais próximos de x_c , devolver:

$$f(x_c) = \frac{\sum_{i=1 \dots K} f(x_i)}{K} \quad [5]$$

Refinamento para pesar a contribuição de cada um dos K vizinhos, com base na sua distância ao ponto a classificar, considera-se para cada vizinho, x_i , o peso, w_i , como sendo uma função monótona decrescente com o aumento da distância usando, por exemplo, a inversa da função de distância adotada (i.e., $w_i = \frac{1}{\text{distância}(x_c, x_i)}$).

Para funções f discretas, após encontrar as K instâncias x_1, x_2, \dots, x_k , devolver:

$$f(x_c) = \arg \max_{v \in V} \sum_{i=1 \dots K} w_i \times \delta(f(x_i), V) \quad [6]$$

Para funções f contínuas, após encontrar as K instâncias x_1, x_2, \dots, x_k , devolver (“média ponderada”):

$$f(x_c) = \frac{\sum_{i=1 \dots K} w_i \times f(x_i)}{\sum_{i=1 \dots K} w_i} \quad [7]$$

Se o vizinho coincidente com x_c ($\text{distância}(x_c, x_i) = 0$) deve-se fazer w_i máximo.

2.3 Base de Dados de Suporte

2.3.1 Quantum Gis (QGIS)

O Quantum Gis (QGIS) é um Sistema de Informação Geográfica (SIG) de código aberto licenciado segundo a *General Public License* (GPL). O QGIS é um projeto oficial da *Open Source Geospatial Foundation* (OSGeo). É apresentado para plataformas em Linux, Unix, Mac OSX, Windows e Android, disponibilizando um vasto conjunto de funcionalidades e suportando inúmeros formatos de vetores, bases de dados e serviços geográficos.

Pode ver ou sobrepor dados vetoriais e matriciais em diferentes formatos e projeções sem conversão para um formato interno ou comum. Os formatos suportados incluídos são:

- Tabelas ativadas espacialmente e visualização usando PostGIS, SpatiaLite e MS SQL Spatial, Oracle Spatial, formatos vetoriais suportados pela biblioteca OGR instalada, incluindo arquivos de forma Esri, MapInfo, SDTS, GML (*Shapefiles*).
- Formatos de imagens e *Raster Maps* suportados pela biblioteca *Geospatial Data Abstraction Library* (GDAL) instalada, como GeoTIFF, ERDAS IMG, ArcInfo ASCII GRID, JPEG, PNG.
- Dados vetoriais e *Raster Maps* para base de dados GRASS (*location.mapset*).
- Servidores de dados espaciais *online* como Serviços Web OGC, incluem WMS, WMTS, WCS, WFS, e WFS-T.

2.3.2 PgRouting

No seguimento das anteriores opções de sistemas a utilizar, a opção para o planeamento de rotas não é diferente. O sistema foi escolhido baseado nas escolhas de OSS anteriores, para apresentação de dados geográficos, sugeridas por parte da Tecmic. Optou-se por um sistema nativo do PostGis e preparado para trabalhar mapas do OSM, de nome PgRouting.

PgRouting é uma extensão a base de dados geospacial PostGis e PostgreSQL. Adiciona funções de planeamento de rota e análise da rede.

Navegação por redes de estradas requer algoritmos complexos que suportam limitações de vias de um sentido e atributos dependentes do tempo. PgRouting é uma biblioteca de código aberto extensível que fornece uma variedade de ferramentas para a pesquisa do caminho mais curto, trabalhando como extensão do PostgreSQL e PostGIS.

PgRouting providencia funções para os seguintes algoritmos de procura:

- All Pairs Shortest Path, Johnson’s Algorithm.
- All Pairs Shortest Path, Floyd-Warshall Algorithm.
- Shortest Path A*.
- Bi-directional Dijkstra Shortest Path.

- Bi-directional A* Shortest Path.
- Shortest Path Dijkstra.
- Driving Distance.
- K-Shortest Path, Multiple Alternative Paths.
- K-Dijkstra, One to Many Shortest Path.
- Traveling Sales Person.
- Turn Restriction Shortest Path (TRSP).

2.4 XTraN Passenger

O *software* de gestão de frotas disponibilizado pela Tecmic, XTraN Passenger, contém toda a informação necessária para o desenvolvimento da aplicação deste trabalho, no apoio do operador de gestão de rotas, na escolha da melhor rota pretendida e da preparação dos dados para serem apresentados no Google. Este *software* faz uso de vários sistemas OSS como o OpenLayers e OpenStreetMap, na apresentação dos mapas ao utilizador.

XTraN Passenger contempla uma camada de acesso aos dados, referente aos transportes públicos de passageiros, denominada de PassengerDAL e ainda o servidor de estimativa de tempos. Fornece plataformas multicanal para informação em tempo real e ferramentas para uma análise fácil dos dados recolhidos nos módulos eletrónicos, existentes em todos os veículos de passageiros.

O XTraN Passenger oferece sistemas de suporte operacional integrados, para o planeamento e gestão de empresas de transportes em todas as áreas de negócio:

- Otimização dos recursos disponíveis.
- Recolha de dados críticos para a gestão dos custos de operação, em tempo real.
- Suporte a decisões corretivas e preventivas.

É neste último tópico que este trabalho foca o seu desenvolvimento, assumindo que os dados recebidos, nesta altura, já sejam o resultado dos dois tópicos anteriores. Tendo como objetivo a melhoria na análise e apresentação da informação, nos processos seguintes.

- Criação de interfaces e ferramentas assistidas de criação de percursos com parâmetros de geometria e de tempos de percurso.
- Análise e parametrização nas várias dimensões temporais do sistema, como: época do ano / calendário, tipo de dia e faixa horária.
- Mecanismo de exportação e compatibilização deste tipo de informação em formato GTFS (<https://developers.google.com/transit/gtfs/?hl=pt-br>) para compatibilização com plataformas exteriores.

Através do seu modelo de dados é possível representar toda a informação necessária para o planeamento de rotas e a preparação dos dados a serem publicados no formato de entrada do Google (GTFS).

O modelo de dados relacional UML do XTraN Passenger disponibilizado é ilustrado na Figura 8.

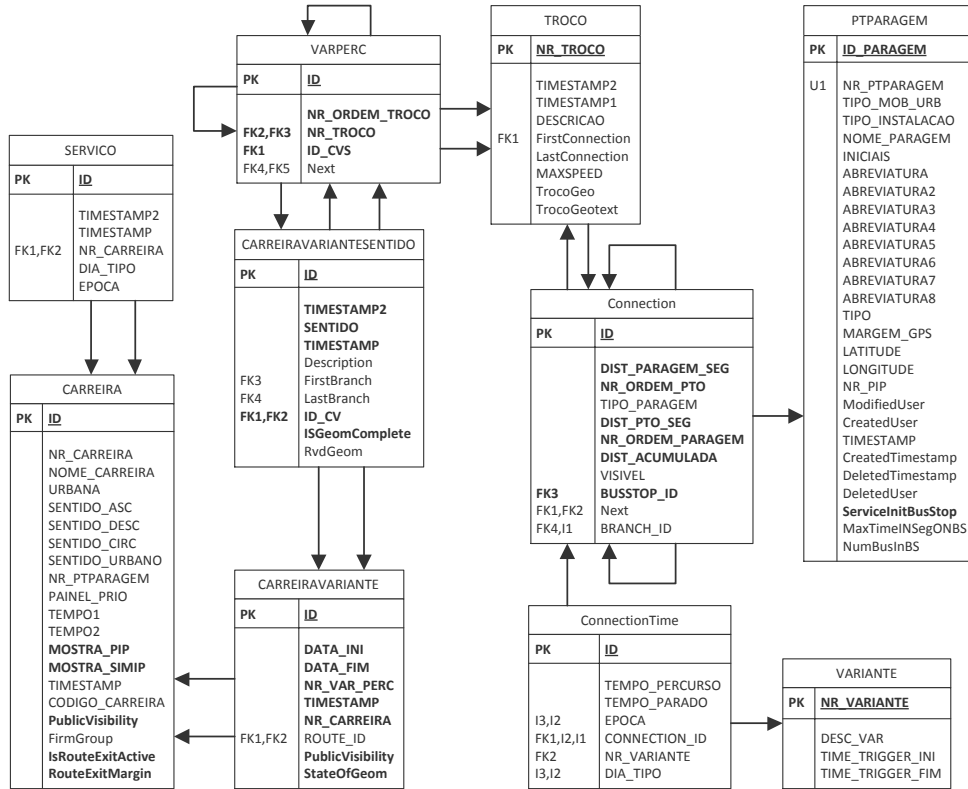


Figura 8: Modelo de dados relacional do XTraN Passenger [20]

2.4.1 Interpretação do Modelo de Dados

Os dados de pesquisa para este trabalho foram cordialmente cedidos pela Tecmic, sob a forma de acesso ao seu servidor, disponibilizando o acesso a uma base de dados para teste chamada “FSETP_Demo_Passenger”, que recria na íntegra, uma imagem da base de dados usada pelo software XTraN Passenger.

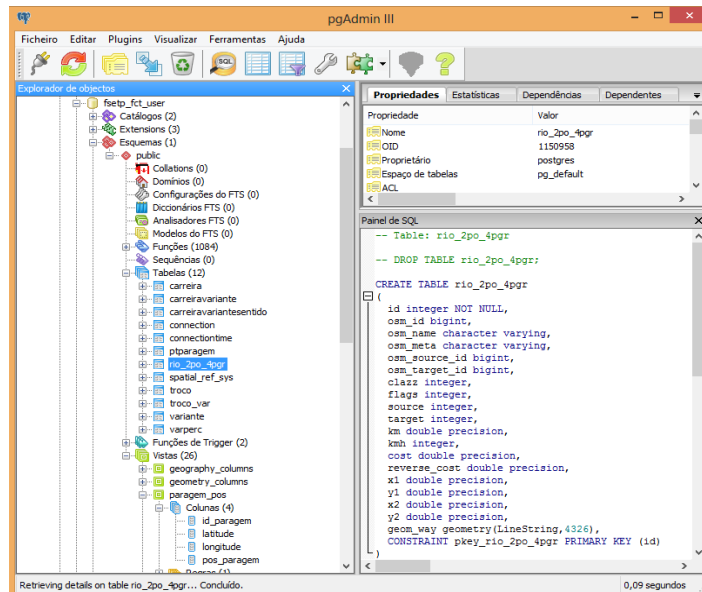


Figura 9: Base de dados fsetp_fct_user

Para minimizar o impacto no servidor de acesso, efetuou-se a extração dos dados para uma base de dados local, criada para o efeito de testes e alterações, sem que implicasse a integridade do servidor e mantendo a utilização de sistemas OSS, como ilustra a Figura 9 representando a nomenclatura da base de dados criada no PostgreSQL, a qual foi chamada de “fsetp_fct_user”.

As linhas dos transportes são representadas pela tabela CARREIRA, que poderão ter várias rotas de cada linha de transporte, sendo representadas na tabela CARREIRAVARIANTE. Cada rota é definida com duas variantes de sentido, caracterizando a ida e volta do percurso, retratadas na tabela CARREIRAVARIANTESENTIDO.

2.4.1.1 Tabela TROCO

A tabela TROCO representa a relação das rotas das carreiras, onde cada rota é caracterizada por múltiplos troços identificados por um “nr_troco” onde seguem uma ordem registada na relação com a tabela CONNECTION.

As linhas de transportes são caracterizadas por uma geometria (*MultiLineString* coluna da tabela TROCO). Todas as geometrias pertencentes aos troços do sistema XTraN Passenger, estão representados na ilustração da Figura 10.

Os dados geométricos desta tabela representam o registo final dos troços, que pertencem a uma determinada rota. Estes dados são considerados como sendo a opção final do operador, para a escolha de rota de uma determinada carreira. Logo, é nesta tabela que são recolhidos os dados das geometrias, que vão representar as rotas das carreiras, nos dados de saída para a informação do Google.

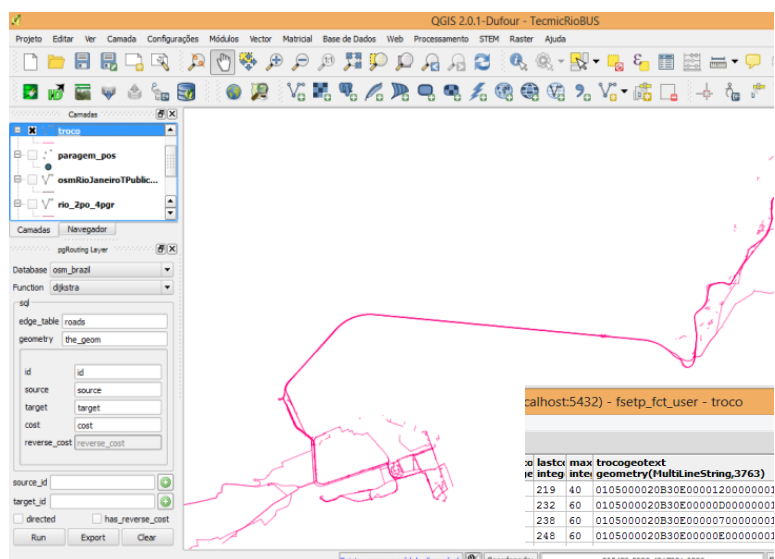


Figura 10: Dados Geográficos da tabela TROCO

2.4.1.2 Tabela PTPARAGEM

As paragens estão caracterizadas na tabela PTPARAGEM, onde se encontram os dados de referência, nome e localização, entre outros. Cada troço contém um conjunto de ligações entre paragens (tabela CONNECTION), através de cada item desta tabela, é possível obter dados importantes como a distância entre paragens (Figura 11).

Capítulo 2 - Estado da Arte

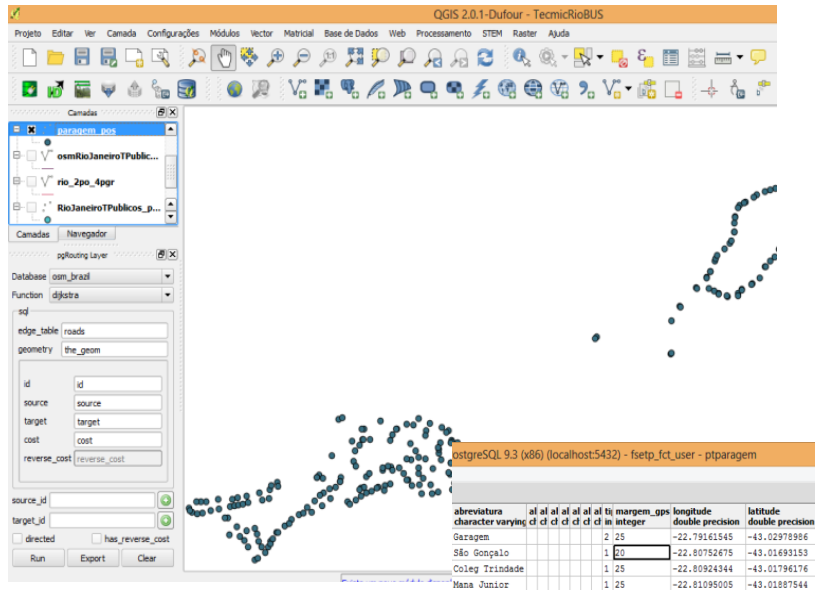


Figura 11: Dados Geográficos da tabela PTPARAGEM

Em cada percurso diário, existe a possibilidade de demorar mais tempo a realizar-se a uma determinada hora do dia ou sendo dia feriado. Tal lógica está representada na tabela CONNECTIONTIME, em que cada item desta tabela contém informação sobre o tempo estimado, para efetuar uma determinada ligação (rota).

A relação temporal ocorre na tabela VARIANTE. Como existe uma relação entre as tabelas de um para muitos entre TROCO e CARREIRAVARIANTESENTIDO, há a necessidade de criar a tabela de relação VARPERC.

Uma relação de camadas, entre os dados das paragens e o mapa do OSM, está ilustrada na Figura 12. Retrata uma camada com a informação das posições geográficas das paragens retirada da base de dados e outra camada com o mapa da zona do onde se encontram as paragens.

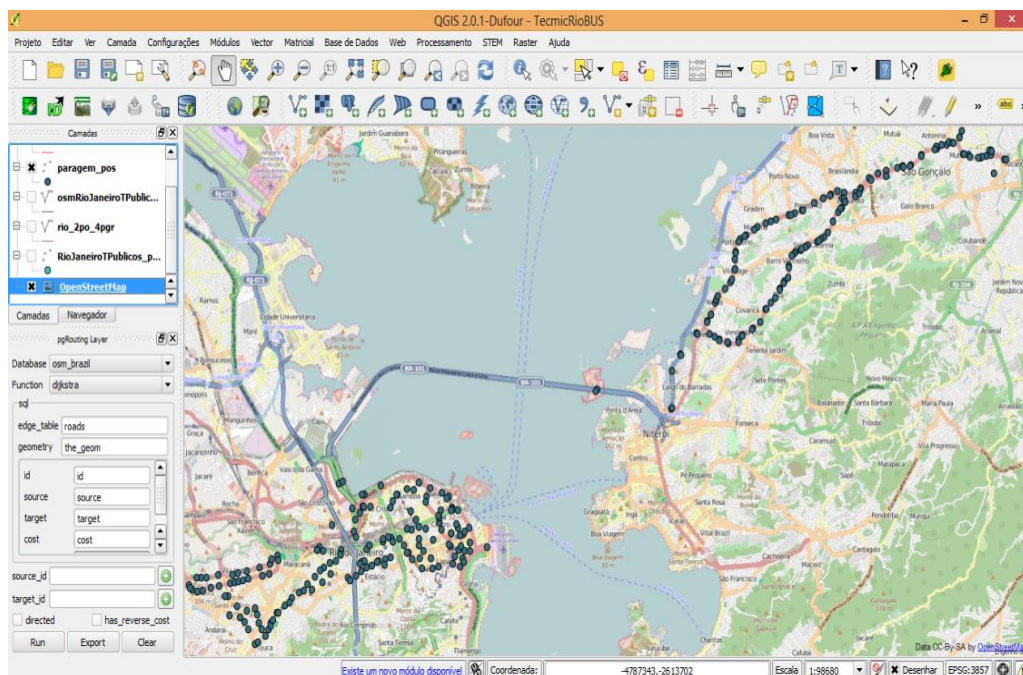


Figura 12: Dados Geográficos da tabela PTPARAGEM com camada OSM

Para completar os dados necessários a este trabalho, apenas foram criadas duas novas tabelas com o intuito de apoiar o planeamento de rotas. Estas tabelas são utilizadas para cálculo de rotas do PgRouting, fazendo proveito dos dados referentes à informação das vias do local pretendido para o planeamento de rota.

Os dados disponibilizados na base de dados “*FSETP_Demo_Passenger*”, são bastante completos para a construção da solução deste trabalho, perseguindo os objetivos inicialmente propostos:

- Calcular o percurso da carreira de uma forma automática, para dar apoio ao operador na fase de implementação dos percursos da frota.
- Melhoramento da identificação real do local da paragem do veículo, durante as várias passagens no percurso, que por vezes se torna difícil, pois são obtidas várias leituras geográficas distintas do mesmo local de paragem ao longo dos percursos, provocado por várias razões, estando entre elas o erro de leitura do GPS, paragem do veículo deslocada, etc.
- Preparação dos dados de forma a serem divulgados, seguindo as recomendações e normalizações para melhoramento na interoperabilidade do sistema. Com o objetivo de preparar os dados em formato que seja reconhecido pelo Google.

3 Modelo conceptual do sistema

Em conjunto com a Tecmic, tentou-se perceber quais os sistemas que facilmente pudessem ser integrados no *software* já existente. Onde se percebeu que o sistema XTraN Passenger faz uso do *software* do OpenStreetMap (OSM) como recurso a dados geográficos. Para a integração de mapas e/ou camadas geométricas no *software* da Tecmic, é utilizado o OpenLayers. Para além destas ferramentas usadas pelo sistema XTraN Passenger, também foi tornado um requisito a utilização de OSS com o objetivo de evitar problemas com licenças de utilização no planeamento das rotas. Também foi realçado a importância que se deveria dar, ao problema da existência de paragens muito próximas em vários locais, provocando por vezes problemas na sua identificação.

A solução apresentada contempla o desenvolvimento de uma extensão ao sistema da XTraN Passenger, criada a pensar em melhorar o processo complexo de identificação das rotas (percursos) das carreiras, por parte dos operadores de frotas de transportes. Um operador depara-se com uma grande complexidade, quando tem de alterar uma rota para um veículo de passageiros.

A escolha do melhor caminho entre duas paragens, já é um desafio respeitável, imaginemos este multiplicado por todas as paragens de um percurso completo. A escolha tem de ser ponderada entre ruas de apenas um sentido, locais onde o veículo não tem espaço de manobra, zonas demarcadas para transportes públicos, entre outras mais, acrescentado complexidade e conhecimento inerente ao local, para a tomada de decisão.

É neste contexto que se propõe uma solução de apoio, ao operador da rede de transportes, apresentando formas automatizadas de adquirir toda a informação importante para a tomada de decisão e apresentar opções de rota, baseado na informação adquirida.

O sistema proposto é pensado numa futura e completa integração no *software* da XTraN Passenger, de forma a enriquecer o sistema existente com módulos de apoio ao gestor de frota.

É apresentada detalhadamente, nesta secção, a solução proposta para cada uma das questões mencionadas no Capítulo 3. Consequentemente o presente capítulo é representado nas seguintes secções:

3.1 Interação do sistema com XTraN Passenger, onde é apresentado numa forma genérica a integração do desenvolvimento no âmbito deste trabalho.

3.2 Certificação de paragens, onde é apresentada a solução para certificar o local da paragem, numa forma mais prática e teórica.

3.3 Planeamento de rota para o apoio ao utilizador, sendo aqui apresentado de uma forma sintetizada de como é feito o planeamento das rotas.

3.4 Preparação dos dados em GTFS onde é disponibilizada toda a informação referente ao XTraN Passenger no formato Google.

3.1 Interação do sistema com XTraN Passenger

Capítulo 3 – Modelo Conceptual do Sistema

O sistema proposto tem a sua arquitetura ilustrada na Figura 13, é constituído por um módulo de certificar os locais das paragens, respeitando as orientações das normas descritas no Capítulo 2. Este módulo funciona de uma forma periódica ou quando se verificar alterações nas paragens das rotas de transportes. Sendo a paragem de seguida validada, de forma a poder ser integrada no planeamento de rotas e na informação enviada no formato GTFS.

Uma das questões que torna a decisão de uma rota complexa é saber com exatidão, o local da paragem, pois sem ela torna-se impraticável a escolha de uma rua para a rota da carreira, correndo o risco de esta sair a escassos metros adiantados da paragem real, numa rua de apenas um sentido, impossibilitando o veículo de lhe chegar sem quebrar as regras de trânsito.

Como existem variações das leituras de GPS do local onde os veículos param, na tomada e largada de passageiros das paragens (Referido na secção 3.2), a certificação das paragens vai ajudar a identificar, com precisão os locais ao longo do tempo. Quantas mais leituras existirem (passagens no local da paragem), mais preciso vai ser o local da paragem. Esta informação, para além da vantagem referida no parágrafo anterior, vai permitir um estudo de mobilidade mais preciso e aplicar melhorias nas rotas mais realistas, permitindo saber com exatidão se uma paragem pode ser mais útil à população se for mudada para outro local.

Calcular Rotas vai ser uma ferramenta essencial no apoio ao operador quando este tem de tomar as decisões de seleção das rotas. Para que estas rotas sejam apresentadas de uma forma mais realista, conta com a validação do local das paragens.

A validação das rotas encontradas é revista pelo do operador, que manualmente seleciona a que melhor se enquadra ao local, terminando o processo de validação.

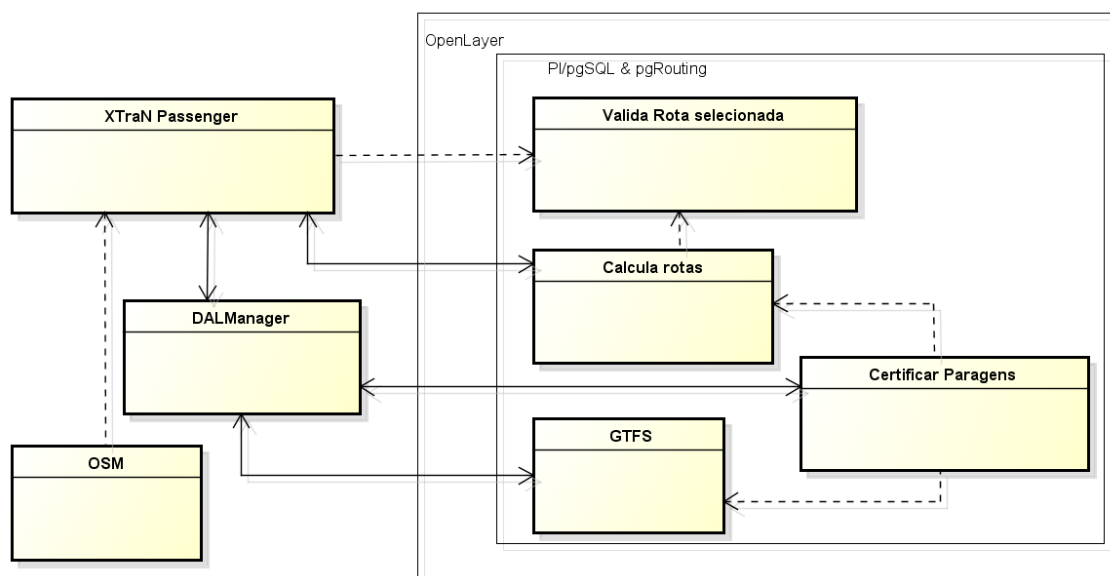


Figura 13: Arquitetura do trabalho

Estes módulos são desenvolvidos em PostgreSQL recorrendo a extensões PI/pgSQL e PgRouting, sempre com o objetivo da possível integração no *software* XTraN Passenger através da adição de camadas, que o *software* permite utilizando a ferramenta OpenLayer.

Uma análise e desenho da abordagem sugerida, de forma metódica, e de acordo com os requisitos enumerados, são apresentados nos diagramas de Casos de Uso na Figura 14, onde descrevo as funcionalidades do sistema proposto IRAPA, tendo em conta o ator identificado, operador de transportes:

- Calcula Rota, consiste no uso das coordenadas GPS das potenciais paragens levantadas pelo operador de transporte para planeamento de uma determinada rota.
- Opções de Rotas, consiste em apoiar a seleção das paragens referentes a uma determinada rota, com um menu que apresenta as rotas existentes identificando as coordenadas das paragens. Facilitando assim o planeamento das rotas.
- Guarda Rota, consiste em guardar a rota na base de dados depois de reconhecida pelo operador.
- Gera GTFS, consiste no processo em que uso o resultado do sistema (paragens e rotas) e os transforma no padrão GTFS para se exportar para o Google Maps.
- Certifica Paragens, consiste em certificar um ponto de paragem que esteja indefinido. Esta função serve de apoio a Gera GTFS e Calcula rotas.

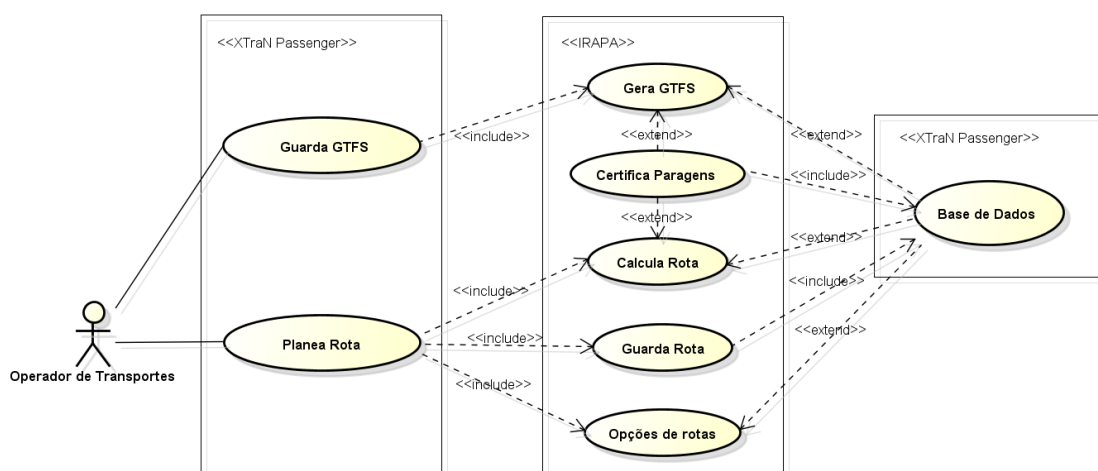


Figura 14: Diagramas de Casos de Uso

3.2 Certificação de paragens

O registo de dados GPS num ambiente urbano pode ser um grande desafio, pois baseado num estudo do *Department of Computer Science, University College London* [28], em que tinham como objetivo a medição do nível de poluição em vários pontos da cidade de Londres, depararam-se com um problema de localização dos seus sensores de Monóxido de Carbono (CO). Não conseguiram ter as posições exatas, tendo por vezes leituras de dados GPS que referenciava o lado oposto da estrada onde se encontrava (Figura 15).

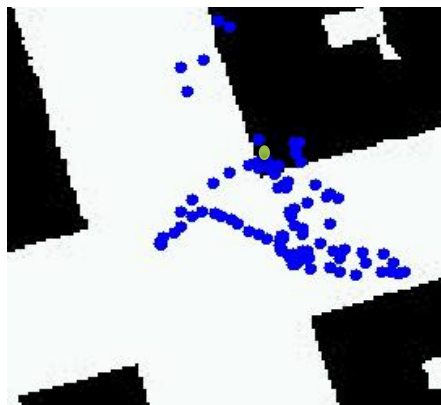


Figura 15: Registo da localização GPS no círculo verde durante 30mn [28]

No planeamento das rotas é importante garantir que exista um trabalho de campo prévio, o mais realista possível, por isso é sugerido a utilização de uma metodologia de leitura geográfica, que minimize os problemas de precisão nas leituras de GPS. Para este efeito existem duas abordagens com a finalidade de solucionar a questão:

- O sistema *Differential GPS* sendo uma solução mais técnica e dispendiosa, pelos equipamentos e recursos humanos envolvidos.
- A outra forma é a interpolação de leituras do GPS dos veículos, quando passam nas paragens. Esta consegue ter uma maior precisão ao longo do tempo, quantas mais leituras forem feitas, tornando o sistema menos dispendioso mas mais moroso.

3.2.1 *Differential GPS* (DGPS)

A identificação geoespacial com precisão é um fator importante, para a certificação dos dados das paragens. Um sistema de precisão na leitura das coordenadas geográficas, na fase inicial de implementação da frota, é essencial para poder calcular as rotas, ou seja, é uma tarefa prévia essencial para melhorar a eficácia da aplicação deste trabalho.

Existem empresas que disponibilizam este sistema que utiliza equipamentos específicos, como referido na secção 2.2.6, para cartografar os pontos necessários na identificação geoespacial das paragens, obrigando a um trabalho de campo extra. Esse sistema é chamado de DGPS [26], que é uma forma de melhorar a precisão na leitura da posição GPS, conseguindo melhorias de 15m para 10cm, nas melhores implementações.

3.2.2 O algoritmo KNN

O XTraN Passenger disponibilizado pela Tecmic, faz uso da informação em tempo real que é recolhida de cada veículo da frota, entre outros, a posição de GPS. Estes dados são utilizados para manutenção do local de paragem e confirmação da paragem efetiva do veículo, sempre que este faz uma paragem com abertura de portas. Esta abordagem traz alguns inconvenientes, devido a margens de erro GPS, paragem perto do local devido a situações pontuais, entre outros. Esta questão tem matéria para um estudo mais aprofundado, mas optou-se por simplificar o problema, recorrendo a um algoritmo muito utilizado em classificação de dados, chamado *K-Nearest Neighbor* (KNN), que é ideal para resolver problemas desta natureza.

O algoritmo KNN é utilizado para classificar com base de um conjunto de treino, onde classifica no espaço, através dos pontos mais próximos. Este algoritmo tira partido de uma estrutura que represente a noção de “nó e de árvores”, rentabilizando o processamento no cálculo do mais próximo. Como já referido na secção 2.2.7, no cálculo da distância, é boa prática o cálculo de um peso baseado na distância, ou seja não efetuar o cálculo da raiz, com o objetivo de redução de processamento e obter os mesmos resultados.

Este algoritmo tem a particularidade de conseguir classificar um ponto mais distante do grupo a que pertence, tendo outro grupo mais próximo, por não contabilizar apenas a distância entre cada grupo e dar mais peso à classe dos pontos mais próximos.

Nesta situação em particular, para a identificação do local da paragem, obtêm-se resultados bastante satisfatórios. Se pensarmos numa das causas do problema das leituras geográficas, como o facto de existirem paragens muito próximas umas das outras, obrigando por vezes à paragem do veículo um pouco fora da sua posição. É criado um padrão de leituras ao longo de várias passagens por esta paragem, como ilustrado na Figura 16. Neste exemplo estão duas paragens muito próximas (representadas por um círculo de cor azul e vermelho mais forte), e a indicação dos vários locais onde efetivamente os veículos pararam (representados por círculos coloridos mais claros). O círculo verde representa a paragem efetiva do veículo na sua última passagem pelo local. O veículo deveria parar na paragem do círculo vermelho, mas está mais perto da paragem azul, levantando o problema de classificação se apenas fosse feita uma análise de distância efetiva.

Com uma classificação utilizando o algoritmo KNN, é feita uma análise considerando e dando mais relevância, aos padrões existentes. Ou seja, mesmo que a leitura de distância atual esteja mais próxima de outra paragem, é contabilizado todos os pontos de paragem do veículo no local ao longo do tempo. Estas já estando classificadas, vão ser um forte contributo para a tomada de decisão mais acertada na identificação da classe (paragem) a que pertence.

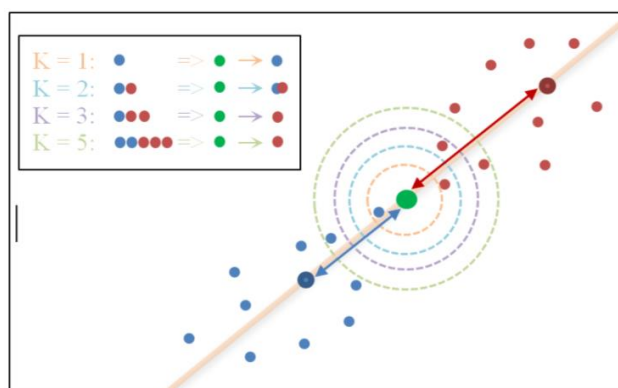


Figura 16: Algoritmo KNN aplicado a uma via e duas paragens

3.3 Planeamento de rotas

Na fase inicial de um projeto de dimensionamento de uma frota de transporte de passageiros, há a necessidade de gerar as rotas dos veículos, após o estudo prévio da posição das paragens que melhor serve a população local. É nesta fase, que a solução apresentada neste trabalho vai dar um forte contributo de automatização, proporcionando rapidez e eficiência na tarefa.

O operador poderá utilizar um serviço, que identificando um ponto de origem e destino, ser-lhe-á apresentada a rota do veículo de passageiros. Adicionalmente, o operador pode especificar

Capítulo 3 – Modelo Conceptual do Sistema

algumas preferências para a viagem, com as alterações que ache necessárias, baseado no conhecimento local.

Para concretizar esta tarefa de planeamento de rotas, é utilizado uma ferramenta chamada PgRouting, que adiciona funcionalidade de criação de rotas ao PostGIS.

No planeamento de rotas há questões importantes a ter em conta, como a identificação de vias (*highway*) que estão marcadas como sentido único (*oneway*), retirada da *Layer* do OSM quando se preparam os dados para o PgRouting, bastando para isso verificar o estado da etiqueta de sentido único:

- `Highway=*`
 - `Oneway=yes`

De notar que a etiqueta de sentido único é definida em relação à direção do elemento linear (“*Wey*”) que está dimensionado no OSM. A identificação da direção de uma via, descreve a forma como é determinada a direção da via pelos editores do OSM.

Se a restrição de um sentido for na direção oposta ao do existente da via, a correção na maioria dos casos passa por contornar essa via por outra no sentido pretendido (ferramenta usual do editores “*reverse way*”) aplicando ao valor da etiqueta “*oneway=yes*”. Em alguns casos raros, não é possível alterar a direção da via, recorrendo-se à alteração da etiqueta para “*oneway=-1*”.

Se a via apenas for de um sentido em alguns troços, tem de se dividir a via em várias partes para que todas tenham identificação de sentido única.

Em alguns casos as etiquetas (como as “*junction=roundabout*”, “*highway=motorway*” e outras) implicam a indicação de “*oneway=yes*” sendo por isso opcional a sua utilização.

Quando a etiqueta “*oneway=no*” é utilizada para confirmar que a via (ou parte dela) não é uma via de sentido único.

A etiqueta de via com sentido único pode ser utilizada em conjunto com o tipo de veículo no sentido de etiquetar exceções. Como exemplo “*oneway:moped=no*” para vias de sentido único que permitem os ciclomotores a circular no sentido oposto da via. Para as bicicletas existe um tratamento especial (“*oneway:bicycle=no*”) onde é necessário adicionar “*cycleway=opposite / opposite lane / opposite track*” para haver compatibilidade.

Para vias que são condicionadas na direção apenas em certos períodos do dia, e até alteração de sentido noutros períodos, pode ser etiquetada com “*oneway=reversible*”. Para definição dos tempos com maior precisão, ou mapear com a restrição de via de sentido único, são utilizadas as restrições condicionais.

Uma sintetização da etiqueta “*oneway*” é representada na Tabela 7.

Tabela 7: Lista de valores para a etiqueta *oneway* [18]

<i>oneway tag</i>	<i>Comment</i>	<i>Translation for routing</i>
<i>yes</i>	<i>discouraged alternative: "true", "1"</i>	<i>vehicle:backward=no</i>
<i>no</i>	<i>discouraged alternative: "false", "0"</i>	
<i>-1</i>	<i>discouraged alternative: "reverse"</i>	<i>vehicle:forward=no</i>
<i>reversible</i>	<i>If a road is one-way at certain times of day, and one-way in the other direction at other times</i>	

Assume-se que as restrições de via de sentido único, presumivelmente não são aplicadas nas vias pedestres (*path*, *footway*, *track*), mas é utilizada em países onde é necessário o controlo de movimento da população, por razões de segurança, numa altura de grande afluência da população.

3.4 Preparação dos dados em GTFS

Para a apresentação dos dados no formato que seja reconhecido no Google Maps, foi desenvolvida uma aplicação, que acede aos dados da empresa de transportes, recolhendo a informação necessária para a construção dos arquivos GTFS (*feeds*), com o objetivo de ser integrado no sistema da XTraN Passenger e automatizar o processo de carregamento dos dados XTraN para o Google Maps.

Para a construção dos *feeds*, teve de se tomar em conta os requisitos mínimos, onde é incluída a informação essencial para a construção da apresentação do Google Maps. Para isso é gerado um grupo de arquivos de texto com um formato comum dos dados de transportes públicos, a ser publicados com as informações, no seguimento das normas e recomendações mencionadas na secção 2.2.5, permitindo implementar de forma automática a passagem dos dados da BD Xtran para o Google. Para gerar esta aplicação recorreu-se a *software* .NET de forma a poder facilmente ser integrado no sistema da XTraN.

Este grupo de arquivos será composto por arquivos de texto com a informação mínima obrigatória, referente a dados sobre calendários, horários, rotas de carreiras, entre outros, que a empresa de transportes considere útil divulgar aos seus clientes e programadores.

Os arquivos construídos obedecem a um relacionamento e ordem, para que haja um encontro de dados entre estes, de forma que a informação apresentada esteja interligada.

Os dados necessários são disponibilizados na base de dados da empresa de transportes e colocada nos ficheiros seguindo os requisitos referidos na secção 2.2.5:

- Dados da empresa (agency.txt);

Campos obrigatórios:

- “*agency_name*”

Este campo contém o nome completo, da companhia de transportes. Este nome é o apresentado no Google Maps.

- “*agency_url*”

É neste campo que se apresenta o *Uniform Resource Locator* (URL) da empresa de transportes. O campo tem de estar normalizado incluindo a indicação `http://` ou `https://`.

- “*agency_timezone*”

Neste campo está contido o *timezone* do local da empresa de transportes. Neste campo o nome não pode conter espaços, tendo de se ter especial cuidado para substituir o espaço por *underscore*.

- Dados das paragens (stops.txt);

Campos obrigatórios:

- “*stop_id*”

Este campo contém, a identificação única da paragem. Cada paragem pode ser utilizada por várias rotas.

Capítulo 3 – Modelo Conceptual do Sistema

- “*stop_name*”

O nome da paragem é apresentado neste campo como sendo único e de forma perceptível para que a população local consiga entender de forma clara.

- “*stop_lat*”

Este campo contém a coordenada geográfica da latitude, do local da paragem, no formato WGS 84.

- “*stop_lon*”

Este campo contém a coordenada geográfica da longitude, do local da paragem, no formato WGS 84.

- Dados das rotas da empresa (routes.txt);

Campos obrigatórios:

- “*route_id*”

Este campo contém, a identificação única pertencente a cada rota.

- “*route_short_name*”

É neste campo que se apresenta um nome curto e abstrato de identificação da rota, de forma a ser identificativo, não sendo obrigatório a relação dos locais de passagem da rota. Este campo é opcional se o “*route_long_name*” estiver definido no ficheiro de rotas.

- “*route_long_name*”

Neste campo está contido o nome completo da rota, onde normalmente é apresentado um nome mais descritivo dos locais de passagem da rota. Da mesma forma que o “*route_short_name*”, apenas é obrigatório se não estiver definido o “*route_short_name*”.

- “*route_type*”

Neste campo é descrito o tipo de transporte utilizado na rota. Valores válidos para este campo são:

- “0” – Veículo exterior, Elétrico. Qualquer sistema exterior ou de carril elétrico numa área metropolitana.
- “1” – Metro. Qualquer veículo subterrâneo de carril numa área metropolitana.
- “2” – Comboio. Usado para ligação entre cidades ou viagens de longa distância.
- “3” – Autocarro. Usado em curtas e longas viagens.
- “4” – Cacilheiro. Usado em curtas e longas viagens de barco.
- “5” – Veículo de cabo, Usado para veículos de cabo ao nível da rua onde o cabo se encontra sob o veículo.
- “6” – Gondola, Veículo de cabo suspenso. Normalmente usado para veículos de cabo aéreo onde o veículo se encontra em suspensão.
- “7” – Teleférico. Qualquer sistema ferroviário projetado para declives íngremes.

- Dados de uma determinada rota (trips.txt);

Campos obrigatórios:

Capítulo 3 – Modelo Conceptual do Sistema

- “*route_id*”

Este campo contém uma identificação única de rota. Este valor é referenciado a partir do arquivo de rotas (*routes.txt*).

- “*service_id*”

Contém um ID que identifica um conjunto de datas em que o serviço está disponível para uma ou mais rotas. Este valor é referenciado a partir do arquivo “*calendar.txt* ou “*calendar_dates.txt*”.

- “*trip_id*”

Neste campo está contido O campo *trip_id* contém um ID que identifica uma viagem. O conjunto de dados único é *trip_id*.

- Horários de partida e chegada dos veículos nas paragens (*stop_times.txt*);

Campos obrigatórios:

- “*trip_id*”

Este campo contém, o ID que identifica as viagens. Este valor referencia o ficheiro “*trips.txt*”.

- “*arrival_time*”

Aqui é especificado o tempo de chegada a uma específica paragem para uma específica viagem de uma determinada rota. O tempo é medido a partir do "meio-dia menos 12h" (eficazmente a meia-noite, exceto para os dias em que as alterações do tempo de verão ocorrem) no início da data de serviço. Para os tempos de serviço que ocorrem após a meia-noite, introduz-se o tempo com um valor maior que 24:00:00 em HH: MM: SS na hora local para o dia em que o horário da viagem começa.

Se não existir tempos separados para a chegada e a partida das paragens, pode se inserir o mesmo valor de horário para a chegada e a partida. Paragem sem horário de chegada será agendada baseada no tempo da paragem anterior. Para garantir um traçado de rota mais precisa, deve-se fornecer os tempos de chegada e partida para todas as paragens.

É obrigatório o fornecimento do horário de chegadas e partidas, para as paragens iniciais e finais de cada viagem.

Os tempos do horário devem ter oito dígitos no HH:MM:SS (ou H:MM:SS) e não podem conter espaços (Tabela 8).

Tabela 8: Exemplo de horários [21]

Tempo	“ <i>arrival_time</i> ”
08:10:00 A.M.	08:10:00 ou 8:10:00
01:05:00 P.M.	13:05:00
07:40:00 P.M.	19:40:00
01:55:00 A.M.	25:55:00

- “*departure_time*”

Capítulo 3 – Modelo Conceptual do Sistema

Este campo especifica o tempo de partida de uma específica paragem para uma específica viagem de uma determinada rota. Segue as mesmas regras do campo “*arrival_time*” referidas no ponto anterior.

- “*stop_id*”

Este campo contém, a identificação única da paragem. Este valor referencia o ficheiro “*stops.txt*”.

- “*stop_sequence*”

Neste campo está contida a identificação da sequência de paragens para uma determinada viagem. Os valores não podem ser negativos e incrementam ao longo da viagem.

- Horários das rotas semanais (*calendar.txt*).

Campos obrigatórios:

- “*service_id*”

Este campo contém um ID que identifica um conjunto de datas em que o serviço está disponível para uma ou mais rotas. Cada valor pode aparecer no máximo uma vez no ficheiro “*calendar.txt*” sendo um conjunto de dados único. Este valor referencia o ficheiro “*trips.txt*”.

- “*monday*”, “*tuesday*”, “*wednesday*”, “*thursday*”, “*friday*”, “*saturday*”, “*Sunday*”

Nestes campos é apresentado um valor binário que indica se o serviço é válido para o dia do campo. O valor “1” indica a disponibilidade do serviço no intervalo das datas, da mesma forma, o valor “0” indica a sua indisponibilidade.

- “*start_date*”, “*end_date*”

Nestes campos são indicadas as datas de início e fim do serviço. Esta data está incluída no intervalo da data do serviço.

4 Implementação do trabalho

Nesta secção é aprofundada a solução para cada uma das propostas mencionadas nos capítulos anteriores. Consequentemente o presente capítulo é representado nas seguintes secções:

4.1 Implementação da interação do sistema com XTraN Passenger. Nesta secção é implementada a componente prática de ligação e apresentação do sistema de planeamento de rotas, com o sistema DotNet da XTraN Passenger.

4.2 Certificação de paragens, implementação da solução para certificar o local da paragem.

4.3 Planeamento de rotas para o apoio ao utilizador, apresentado a implementação do planeamento de rota automática.

4.4 Implementação da preparação dos dados em GTFS disponibilizando toda a informação referente ao XTraN Passenger no formato GTFS.

4.1 Interação do sistema com XTraN Passenger

Para a apresentação dos dados recolhidos nos pontos de implementação, utilizou-se uma ferramenta crucial chamada OpenLayers (como descrito na secção 2.2.4), que prepara todos os dados de forma a serem recebidos no *software* XTraN Passenger adicionando *layer's* para novas funções da aplicação.

Para colocação de janelas de mapa na aplicação usou-se um *LayerManager*, que é a responsável por gerir as *layer's* a serem apresentadas na janela e um *MapBox*, onde é carregado o mapa de apresentação junto com a *layer* da rota calculada. Recorrendo a livrarias personalizadas para uso específico do *software* XTraN Passenger, como o *LayerManager* pode ser obtido na livraria *Passenger.UI.LayerManager.dll*. O *Mapbox* está incluída na *SharpMap.UI.dll*. Para a utilização das livrarias disponibilizadas é necessário adicionar *SharpMap.dll* ao projeto (Figura 17).

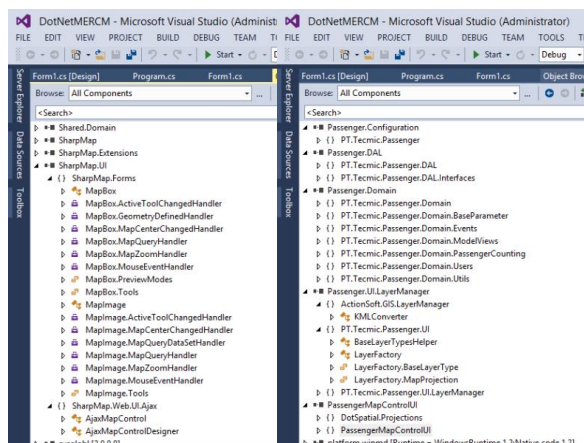


Figura 17: Estrutura de livrarias importadas

Capítulo 4 – Implementação

A aplicação deste trabalho recebe do operador a informação da carreira e variante que pretende para o planeamento da rota, como ilustra a Figura 18, recorrendo a uma *Combobox* com a informação.

Ligação da aplicação à base de dados para carregar *combobox*:

```
...
void fillCombo() {
// Create a string for DBase connection
string sql_connection =
"Host=localhost;Username=postgres;Password=1234;Database=fsetp_fct_user";
// Create a string for DBase Query
string query = "SELECT nr_carreira FROM carreira";
...
using (var conn = new NpgsqlConnection(sql_connection)) {
    conn.Open();
    using (var cmd = new NpgsqlCommand()) {
        cmd.Connection = conn;
        cmd.CommandText = (query);
        using (var reader = cmd.ExecuteReader()) {
            while (reader.Read()) {
                string sNumber = reader.GetString(0);
                comboBox1.Items.Add(sNumber);
            }
        }
    }
}
...

```

Com a ligação à base de dados são obtidos os dados para preenchimento da *combobox* com a informação das carreiras existentes, para ajudar o operador na escolha da rota pretendida.

A partir deste momento a aplicação pesquisa as paragens que fazem parte da carreira e variante selecionada pelo operador.

A aplicação toma como ponto de partida a identificação das paragens, existentes na base de dados, que pertençam à rota escolhida pelo operador. Para isso é criada uma ligação à base de dados para a pesquisa da informação necessária da mesma forma que foi feito para a *combobox*.

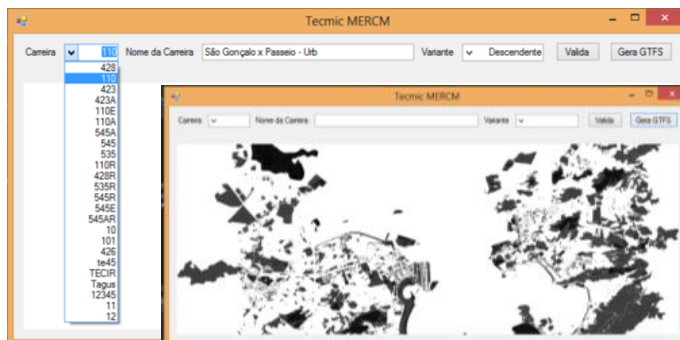


Figura 18: UI de seleção de rota a calcular

Após a identificação da rota e variante pretendida, a aplicação calcula o caminho baseado no algoritmo escolhido, e apresenta o seu resultado ao operador. Este faz uma verificação de controlo sobre a rota calculada e valida a sua introdução.

Para possibilitar ao operador a visualização da rota encontrada, fez-se uso da livreria *SharpMap.UI.dll* gerida por uma *LayerManager* de forma a possibilitar o controlo da mesma, antes de validar.

Introdução de uma Layer do PostgreSQL:

```
//Create layer
SharpMap.Layers.VectorLayer layRoads= new sharpMap.Layers.VectorLayer("Roads");

```

```
//Set the datasource to the PostgreSQL table  
layRoads.DataSource = new SharpMap.Providers.PostGis(connStr, tablename, idColumn);
```

A *LayerManager* possibilita a gestão de várias *Layer's* melhorando a aplicação com a sobreposição da rota sobre o mapa da região. A *Layer* das rotas calculadas é carregada em conjunto com a do mapa, disponibilizando ao operador um melhor entendimento das opções apresentadas.

Introdução de uma Layer do OSM:

```
mapBox1.Map.BackgroundLayer.Add(  
new SharpMap.Layers.TileAsyncLayer(new BruTile.Web.OsmTileSource(), "OSM")  
);  
mapBox1.Map.ZoomToExtents();  
mapBox1.Refresh();  
mapBox1.ActiveTool = SharpMap.Forms.MapBox.Tools.Pan;
```

Depois da rota validada pelo operador, esta é registada na base de dados sob a forma de geometria ("*MultilineString*"), podendo ser acedida em qualquer altura, pelo XTraN Passenger, utilizando a ferramenta "*PostGis*" incluída no "*SharpMap*". Esta validação é feita com um botão "Valida", que vai guardar na base de dados o registo da escolha de rota para a carreira.

A aplicação também disponibiliza um botão para a criação dos *Feeds* do Google, em que pressionado desencadeia todo o processo automático de criação, desde a pesquisa na base de dados da informação necessária para os ficheiros de texto, até ao registo do ficheiro .zip no local do servidor pretendido.

4.2 Certificação de paragens

A localização das paragens é crucial para uma boa análise na pesquisa da melhor rota. Para um melhor entendimento geral, do processo de certificação das paragens, seguiram-se as recomendações de normas existentes como referido na secção 2.3.1, nomeadamente a preocupação da existência dos nomes e localizações geográficas para todos os tipos de paragens, servindo para identificação de lugar de paragem e seriam únicos dentro de cada país.

Em conjunto com a Tecmic, foi realçado a importância que se deveria dar ao problema da existência de paragens muito próximas em vários locais, provocando por vezes problemas na fixação das rotas.

Como também faz parte integrante deste trabalho a preparação dos dados em GTFS, seguir as recomendações das normas é imperativo, sendo a solução de ter um nome único para além de um identificar, a melhor abordagem ao problema de identificação do lugar das paragens. Pois além de minimizar o processamento de dados, entre comparação de *strings* e cálculos de distâncias de vários pontos, pode ser apenas conferida a classificação da paragem pelo nome. Também é mais fácil para um operador, reconhecer a paragem, se a informação for apresentada pelo nome.

A certificação dos locais das paragens é essencial para que os dados sejam apresentados no Google, respeitando as recomendações das normas para o efeito. Para a localização geográfica do lugar da paragem, a ser registado nos ficheiros do Google, recorreu-se a duas abordagens:

- *Differential GPS (DGPS)* - Com a abordagem de cartografar os pontos geoespaciais necessários na identificação do lugar das paragens, consegue-se uma localização exata da paragem. A identificação do lugar da paragem é complementada com o nome da mesma, que em conjunto com a localização geoespacial, são uma identificação única. Neste caso, basta garantir que as paragens que servem de entrada à aplicação sejam resultado de um trabalho prévio no local, com a utilização deste processo recorrendo a empresas especializadas que fornecem esse serviço.
- Algoritmo KNN - A utilização de um algoritmo de classificação, é uma abordagem mais contínua, em que o registo de dados ao longo do tempo vai servir para uma atualização relativamente à posição das paragens. Este algoritmo é usado de forma a dar mais relevância ao padrão das leituras referentes às paragens, e menos à distância direta com a paragem, sendo este processo implementado neste trabalho.

Um exemplo real está ilustrado na Figura 19, onde a leitura está a ser feita no local do círculo verde e a paragem está localizada no círculo azul. Facilmente se percebe que existem locais de paragem mais próximos. Numa análise que contabiliza-se apenas a distância a que a paragem se encontra, este registo de paragem seria classificado como pertencente a outra paragem.

Com várias medições do local da paragem registadas ao longo do tempo (pontos amarelos), já classificadas como pertencerem à paragem “Petrobras”, torna-se mais fácil a sua classificação com a utilização do algoritmo. Estes pontos encontram-se mais próximos da leitura do círculo verde, ajudando a classificar como pertencer à paragem “Petrobras”.

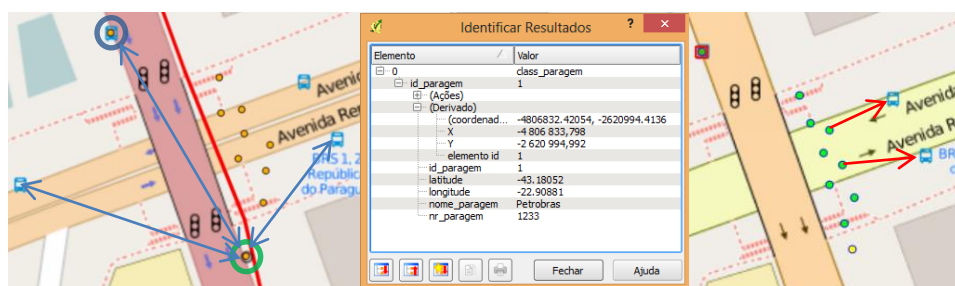


Figura 19: Paragem “Petrobras” no círculo azul e leitura no círculo verde

Para a aplicação do algoritmo, foi criada uma função “Pl/pgSQL” que testa a distância entre dois pontos e aplica o algoritmo KNN com K=5 para a classificação. É feita uma pesquisa pelos pontos mais próximos do ponto de leitura e utilizar a informação de classificação para classificar o novo ponto lido, baseado na informação registada na base de dados, sobre a classificação dos pontos de leitura mais próximos.

Função de classificação e atualizar:

```
CREATE OR REPLACE FUNCTION class_update(_tbl CHARACTER)
...
BEGIN
...
EXECUTE 'SELECT array (SELECT st1.nr_paragem
FROM ' || _tbl || ' AS st1, ' || _tbl || ' AS st2
WHERE st2.id_paragem = (SELECT id_paragem FROM ' || _tbl || ' WHERE nr_paragem IS NULL )
AND ST_Distance (st1.point_geom, st2.point_geom) != 0
ORDER BY ST_Distance (st1.point_geom, st2.point_geom) LIMIT 5)' -- Representa K=5
INTO _row_nr_par;
...
END
$func$ LANGUAGE plpgsql;
```

Com a identificação dos pontos mais próximos, analisou-se a classificação destes quanto à paragem a que pertencem fazendo um registo da classificação de todos os pontos mais próximos do ponto a classificar. A Figura 20 ilustra um novo ponto que ainda não foi classificado, realçado pelo sublinhado verde.

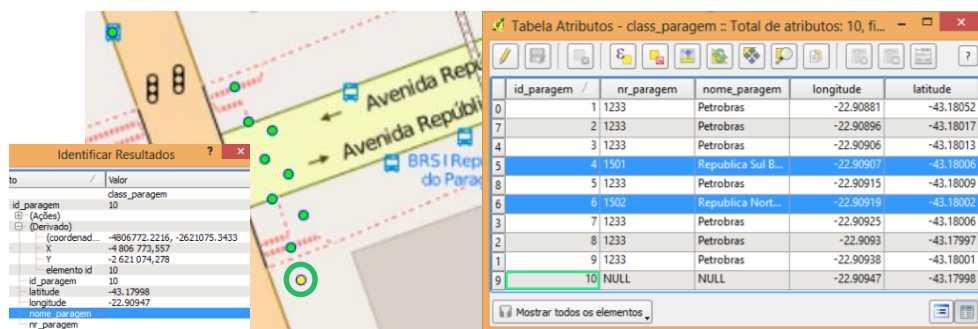


Figura 20: Ponto por classificar

O programa vai fazer uma contagem da classificação dos cinco pontos obtidos, registrando a classe mais predominante, para classificar o novo ponto.

Nesta contagem vai ser incluído um ponto que pertence a outra paragem, mas com a utilização deste algoritmo, não vai haver grande relevância para o ponto de outra paragem próxima, minimizando o problema que poderia ser causado por uma paragem próxima da pretendida.

4.3 Planeamento de rotas com OSM

O planeamento de rotas foi implementado com recurso ao *PgRouting*, ferramenta de utilização livre pertencente à plataforma da PostgreSQL. A escolha destas tecnologias deve-se ao facto de todos os componentes desenvolvidos pela Tecmic poderem usar esta tecnologia de uma forma livre de encargos.

Todos os dados estão armazenados numa base de dados relacional *Structured Query Language* (SQL) com recurso à ferramenta Microsoft SQL Server 2008. A camada desenvolvida pela Tecmic que permite o acesso aos dados, utiliza uma solução *Object-Relational Mapping* (ORM), o NHibernate. Esta camada de abstração, denominada por PassengerDAL, será utilizada para aceder aos dados presentes na base de dados do XTrAn Passenger. O PassengerDAL disponibiliza interfaces de acesso para todos os dados necessários para o planeamento de rotas.

Os dados utilizados para o planeamento da rota, são acedidos de forma a não alterar da base de dados existente, com a utilização da ferramenta para o planeamento das rotas. É preciso manter as ligações de chaves primárias e estrangeiras existente, na criação de novas tabelas e vistas na criação de novas rotas. É ilustrado na Figura 21, parte do modelo relacional de dados que incluem a informação para o planeamento das rotas.

Capítulo 4 – Implementação

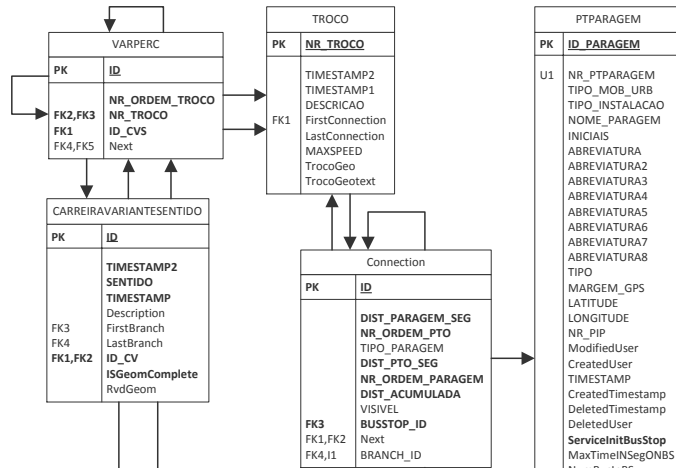


Figura 21: Modelo relacional usado no planeamento das rotas

Identificada a carreira que se pretende calcular a rota de uma forma automática, utilizando a aplicação deste trabalho (secção 4.1), estando esta informação na tabela “variantepercurso” através da ligação com a tabela “carreiravariantesentido”, que identifica a carreira. Com esta relação consegue-se obter a informação de todos os troços da via pertencentes ao percurso completo da carreira no sentido pretendido. Ficando a faltar apenas a identificação das paragens do percurso que tem a relação garantida com a tabela “connection” fazendo a ligação do troço com a paragem. Na Tabela 9 estão representados alguns dados pertencentes à base de dados, caracterizando os valores de relação identificando três troços (27,28,29) pertencentes à carreira 110 no sentido ascendente, com as devidas paragens.

Tabela 9: Relação entre três troços de uma carreira e devidas paragens

C. V.	C. V. S.	VARPERC			TROCO				CONNECTION		PTPARAGEM	
		ID	NR_TROCO	NEXT_ID	NR_TROCO	DESCRICAO	First C.	Last C.	ID	BUSSTOP_ID	ID_PARAGEM	NOME_PARAGEM
8	16	208	27	209	27	Trindade - Pç de nova Cidade	211	219	211	1054	1054	Praça Trindade
		209	28	210	28	Pç Nova Cidade – Boassú	220	232	220	1059	1059	Pç Nova Cidade
		210	29	211	29	Boassú - Clube Tamoio	233	238	233	1067	1067	Boassú

Esta informação é usada na identificação geográfica dos pontos da paragem pertencentes à rota. Juntamente com a informação da ordem das paragens, retirada da tabela “connection”, temos o ambiente criado para o planeamento da rota.

Estes dados geográficos podem ser apresentados numa nova *layer* em conjunto com o mapa carregado do OSM, para uma apresentação para que o operador tenha uma fácil compreensão da rota encontrada.

4.3.1 Dados do OSM

Para efeito de apresentação e teste, recorreu-se a ferramenta QGIS (Figura 22), que está nativamente preparada para lidar com mapas do OSM, bastando para isso instalar uma livreria OpenLayers. Com esta ferramenta foi possível analisar e retificar erros de semântica, de uma forma rápida e precisa.

Capítulo 4 – Implementação

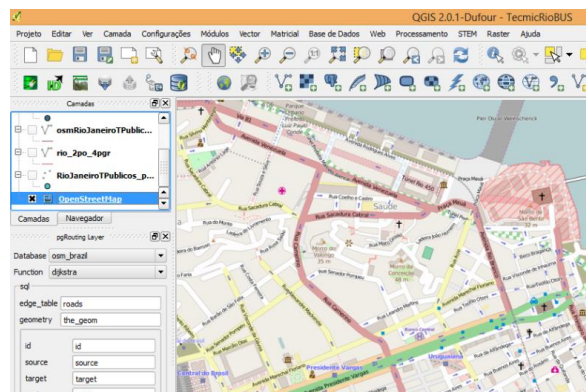


Figura 22: OpenLayer plugin do QGIS para OSM

Os dados geográficos também podem ser adquiridos em formato de ficheiro, na página do OpenStreetMap, disponibilizando uma ferramenta para descarregar esses dados OSM, num formato .osm (XML comprimido), podendo ser utilizados por vários *softwares* de GIS (Figura 23).

Com o QGIS consegue-se fazer ligação com várias bases de dados (PostGis, EspatiaLite, MSSQL, Oracle, WMS/WMTS, etc.), que suportem informação geoespacial.

Para o planeamento da rota de uma forma automática, é obtida do OSM a informação constituída por linhas geométricas (“*Multilinestring*”) representando as estradas. Uma rua desde o seu ponto inicial ao final é representada por uma linha, não estando identificados os pontos de intersecção com outras estradas, que dificulta na tarefa de calcular uma rota automática. É necessário um tratamento da informação recolhida do OSM, para que esta seja apresentada num formato preparado para planeamento de rotas.



Figura 23: Dados das vias do OSM

Após a aquisição dos dados essenciais para o planeamento de rotas automático, foi realizada a preparação dos mesmos de forma a poder ser feito o processo de planeamento de rotas com o PgRouting. Começando com a preparação dos dados, com o objetivo de introdução na base de dados SQL (sistema utilizado na empresa de transportes), disponibilizando assim o acesso à informação das vias que podem ser usadas para o planeamento da rota pretendida.

Para preparação de dados recolhidos do OSM, existem ferramentas como a *Osm2pgrouting*, sendo uma ferramenta de linha de comando para facilitar a importação de dados, incluindo-os numa base de dados apta para o bom funcionamento da PgRouting. Cria a topologia de seleção de rota automaticamente, e as tabelas para tipos de recursos e classes de estrada.

Da mesma forma o *Osm2po* é ao mesmo tempo, um conversor e um mecanismo de planeamento de rotas. O *osm2po* é capaz de converter grandes conjuntos de dados, em

Capítulo 4 – Implementação

comparação com o Osm2pgrouting, como europe.osm. Gera a implementação SQL, criando scripts de criação e inserção de dados PostGIS, compatíveis com PgRouting e QGIS (Figura 24).

Características do osm2po:

- O osm2po permite leitura de ficheiros osm.bz2, pbf, o5m and o5m.gz.
- Permite criar uma tabela PostGIS de Vertices.
- Permite criar uma tabela PostGIS de fronteiras complexas.
- Permite a criação de uma tabela PostGIS de áreas, construções, etc.

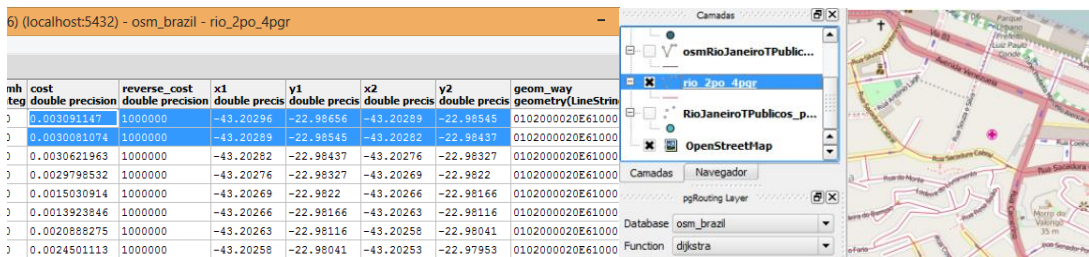


Figura 24: Layer Osm2po

Após a preparação dos dados retirados do OpenStreetMap, utilizando a ferramenta em Java chamada “Osm2po”, são gerados scripts para a inserção de dados no PostGIS (PostgreSQL), compatíveis com PgRouting e QGIS. Esta ferramenta faz a preparação dos dados geométricos, necessários para o planeamento de rotas, fazendo uma leitura exaustiva das vias, identificando pontos de interceção, vias sem ligação, para serem devidamente processadas. O que antes estava como a rua completa numa geometria “*Multilinestring*”, após o tratamento do “Osm2po” a rua é subdividida em troços referentes aos cruzamentos e entroncamentos existentes, ilustrado na Figura 25.



Figura 25: Via após a ferramenta “Osm2po” à direita

Com os dados preparados para o planeamento de rotas, pode-se agora fazer a relação da informação da empresa de transportes, sobre as suas rotas e locais de paragem ordenados. Para isso começou-se por uma parte crucial, que é a representação dos locais das paragens onde as rotas calculadas irão passar.

Uma questão que tem de ser resolvida, é a relação das paragens com as geometrias das vias que foram preparadas para o planeamento das rotas. As paragens por norma estão posicionadas fora das vias, obrigando a que seja feita a identificação da via mais próxima da paragem.

Capítulo 4 – Implementação

Para que o planeamento da rota seja feito sem problemas, é essencial que o ponto de início e fim de rota pertençam às linhas das geometrias representativas das vias.

A função de destaque na realização desta tarefa é o método “ST_ClosestPoint”, que devolve o ponto mais próximo entre duas geometrias (Figura 26).

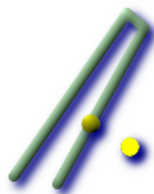


Figura 26: Ponto mais próximo entre uma *linestring* e *point*

Esta função tem como entrada um ponto e uma linha, devolvendo o ponto contido na linha mais próximo do ponto de entrada. O ponto de entrada pode ser considerado como sendo o ponto da paragem e a linha como sendo a via mais próxima.

Com o ponto da paragem não existe qualquer problema pois este é conhecido e está bem documentado, mas quanto à via surge um problema, pois não se sabe qual a via a usar. Ou se utiliza todas as vias da base de dados, que não é aconselhado porque em dimensões muito grandes torna-se impraticável, ou procura-se uma ferramenta para limitar as geometrias de cálculo.

Então procurou-se quais as geometrias a utilizar para o cálculo das distâncias. Começando classificar vias como candidatas a mais próximas que se encontram num raio envolvente do ponto da paragem. Foi criada uma geometria circular envolvendo o ponto da paragem, de forma a coincidir apenas com as vias candidatas a mais próximas, em vez de utilizar todas as vias da base de dados.

Com este processo resolveu-se o problema de identificação da via que representava a paragem (“estrada” ilustrado na Figura 27).

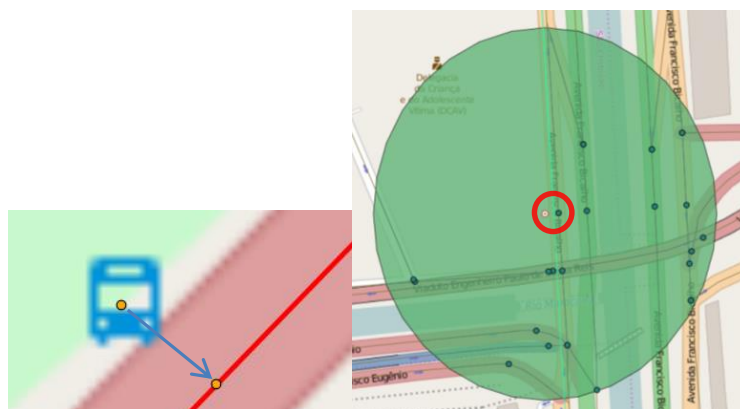


Figura 27: Paragens na linha da rota

Para implementar esta parte, recorreu-se às ferramentas de manipulação de geometrias do PostgreSQL.

Para uma maior eficiência de processamento, procedeu-se com a criação de um *buffer* em torno do ponto geográfico da paragem de raio igual a 250m, de forma a reduzir a pesquisa das vias candidatas à paragem mais próxima (Figura 27 à direita). Após encontrar a geometria da via mais

próxima, esta é registada como sendo a via para o cálculo das rotas que passem na paragem em causa.

Vista ways_buf_4pgr:

```
CREATE VIEW ways_buf_4pgr
AS
SELECT
  CAST(CAST(a.id AS TEXT) || CAST(b.id_paragem AS TEXT) AS numeric(24,0)) AS id_ways,
  a.id, b.id_paragem AS id_prg, a.source, a.target,
  ST_Intersection(a.geom_way, b.buf) AS intersect_ab
FROM rio_2po_4pgr a
JOIN point_buf b ON ST_Intersects(a.geom_way, b.buf);
```

Na Tabela 10 é apresentado um exemplo de dados gerados para o planeamento de rotas, incluindo já a identificação da via para a devida paragem. Os troços estão identificados com as paragens e vias a que pertencem para o planeamento. Estas vias têm a informação de *source* e *target* criada na fase de preparação dos dados para planeamento de rotas. Esta informação é utilizada essencialmente para identificar a via no planeamento da rota, podendo ser a origem, destino ou passagem de uma dada rota.

Tabela 10: Relação entre três troços e devidas paragens

ID	CONNECTION			TROCO_AUTO		
	BUSSTOP_ID	Next	BRANCH_ID(TROCO)	ID_VIA	Source	Target
211	1054	212	27	118906	87480	87575
220	1059	222	28	129642	87284	87241
233	1067	234	29	111373	4234	3727

Com esta tarefa concluída está criado o ambiente para o planeamento de rotas, permitindo a criação de tabelas e vistas usando a ferramenta “Pl/pgSQL” e PgRouting.

4.3.2 A ferramenta PgRouting

Uma vez feita a preparação dos dados, estes encontram-se prontos a ser utilizados no planeamento de um percurso. Como vimos na secção anterior, é importante para esta fase saber a identificação de “*source*” e “*target*” de cada via.

PgRouting é uma extensão da base de dados geoespacial PostGIS/PostgreSQL para fornecer a funcionalidade de planeamento de rotas (Figura 28). Estão disponíveis vários algoritmos de pesquisa, como referido anteriormente na secção 2.2.1.

Depois da preparação dos dados geográficos, como questões de pontas sem ligação, vias de sentido único e cruzamentos entre vias a ser utilizados nas rotas, está o ambiente preparado para se utilizar a ferramenta de planeamento de rotas PgRouting.

O PgRouting é nativo para lidar com dados da rede rodoviária do OSM, após a recolha dos dados, são guardados na base de dados toda a informação necessária para o planeamento das rotas, utilizando funções “Pl/pgSQL”, e faz uso da ferramenta OpenLayers para apresentação numa aplicação “*web-mapping*”.

Cada via representativa do mapa OSM, tem uma identificação única, que depois da ferramenta de preparação para o planeamento de rota, mantém essa identificação e é criada uma indicação de *source* e *target* que vai alimentar a ferramenta de pesquisa da melhor rota de um ponto de origem para um destino.

A forma geral de uma consulta de rota é:

```
“select pgr_<algorithm>(<SQL for edges>, start, end, <additonal options>)”
```



Figura 28: PgRouting Layer

Em vias de manter o sistema deste trabalho coerente e compatível com o existente na Tecmic, todo este processo de pesquisa das rotas é efetuado para cada trecho, determinado por ponto de paragem atual até ao ponto de paragem seguinte. Para cada trecho entre paragens foi calculada a variante possível, ficando registada a geometria do trecho na base de dados (Tabela 11, AUTO_TROCO), com o objetivo de poder ser apresentada como rota possível para uma determinada carreira.

Tabela 11: Relação das vias e as paragens

CONNECTION				TROCO_AUTO		
ID	BUSSTOP_ID	Next	BRANCH_ID(TROCO)	ID_VIA	Source	Target
211	1054	212	27	118906	87480	87575
220	1059	222	28	129642	87284	87241
233	1067	234	29	111373	4234	3727

O processo de pesquisa de rota começa com a introdução da rota e da sua variante pretendida (ex. carreira 110 sentido ascendente), onde são identificados os treços e as devidas paragens. Perante a informação das paragens e a sua sequência, obtida pelos treços, consegue-se obter as vias das paragens que se pretendem para o planeamento da rota.

Para o planeamento das rotas foi realizado um método, que com a identificação da via de origem e destino (paragem de origem e destino), este procura o melhor caminho de forma automática.

Método de cálculo de rota:

```
CREATE OR REPLACE FUNCTION geometry_tr( source integer, target integer )
RETURNS geometry
AS $$
    SELECT geom_way
    FROM pgr_astar(
        'SELECT id, source, target, st_length(geom_way) as cost, x1, y1, x2, y2 FROM
        rio_2po_4pgr', source, target, false, false
    ) as di
    JOIN rio_2po_4pgr pt
    ON di.id2 = pt.id
$$ LANGUAGE 'sql';
```

Este método tem como entrada as identificações “source” da via de origem e “target” da via de destino, que levou à criação de dois métodos que com a entrada da identificação do trecho, é devolvido o “source” e o “target” desse trecho, recorrendo à relação de tabelas “TROCO” e “CONNECTION” que identifica o “firstconnection” e o “lastconnection” por cada trecho.

Capítulo 4 – Implementação

```
CREATE OR REPLACE FUNCTION source(conne integer)
RETURNS integer
AS $$
    SELECT lmp.source
...
CREATE OR REPLACE FUNCTION target( conne integer )
RETURNS integer
AS $$
    SELECT lmp.target
    FROM troco t, connection c, linha_mais_proxima lmp
    WHERE lmp.id_paragem=c.busstop_id and t.lastconnection=c.id and
t.firstconnection=conne;
$$ LANGUAGE 'sql';
```

Este cálculo de rota é feito de forma progressiva e sequencial entre paragens troço a troço, com o resultado de vários troços que juntos vão formar a rota da carreira pretendida.

4.4 Preparação dos dados em GTFS

Na preparação dos dados para GTFS, são efetuadas *Query's* à base de dados da empresa de transportes públicos, recolhendo informação referente aos requisitos dos vários ficheiros GTFS. Cada ficheiro modela um aspeto específico das informações sobre o transporte público: paragens, trajetos, viagens e outros dados relativos a rotas de carreiras, horários e mais dados que as empresas de transportes queiram divulgar. Para isso é criada uma aplicação, que vai correr no programa do operador, de criação dos ficheiros em texto seguindo as normas GTFS.

Código C# de escrita do ficheiro agency.txt:

```
...
string[] agency =
{
    "\"agency_id\", \"agency_name\", \"agency_url\", \"agency_timezone\", \"agency_lang\", \"agen
cy_phone\", \"agency_fare_url\"",

    "\"TC\", \"Tecmic\", \"http://www.tecmic.pt/\", \"Europe/Lisbon\", \"PT\", \"+
351.244845533\", \"\"\"
};

// WriteAllLines creates a file, writes a collection of strings to the file,
// and then closes the file. You do NOT need to call Flush() or Close().

System.IO.File.WriteAllLines(@"C:\Users\Public\PastaTeste\agency.txt", agency);
...
```

Para que seja possível a utilização do serviço do Google, todos os ficheiros de texto têm de seguir as normas de escrita, para um perfeito reconhecimento, como referido na secção 2.2.5.

Na Tabela 12 é apresentado um exemplo de dados necessários, da base de dados da Tecmic, para implementar os ficheiros no formato do Google. Neste exemplo estão referenciadas duas paragens da rede de transportes.

Tabela 12: Dados recolhidos em PTPARAGEM para integrar em ficheiro GTFS

PTPARAGEM				
ID_PARAGEM	NR_PARAGEM	NOME_PARAGEM	LATITUDE	LONGITUDE
1233	26297	Petrobras	-22.90954076	-43.18004077
1235	8070	Passeio	-22.91244619	-43.17615229

Depois de feita a *query* que devolve os dados pretendidos, a implementação guarda essa informação num ficheiro de texto “*stops.txt*” para ser integrado num ficheiro .zip onde se encontram os restantes ficheiros de dados.

Código GTFS de escrita do ficheiro stops.txt

```
"stop_id","stop_code","stop_name","stop_lat","stop_lon","location_type","stop_desc","parent_station"
"1053","SG01118","Garagem",-22.79161545",-43.02978986",0,"",""
"1054","SG01381","Praça Trindade",-22.80752675",-43.01693153",0,"",""
"1055","3","Colegio Trindade",-22.80924344",-43.01796176",0,"",""
"1056","4","Av Mana Junior",-22.81095005",-43.01887544",0,"",""
"1057","5","Nova Viseu",-22.81243045",-43.01967116",0,"",""
"1058","6","Ponte Seca",-22.81385904",-43.02041522",0,"",""
"1059","SG04773","Pç Nova Cidade",-22.81552854",-43.02443714",0,"",""
```

Existe uma relação entre a criação dos ficheiros de uma elevada importância para que não haja discrepâncias na apresentação dos horários e formas geométricas das viagens.

Uma viagem (*Trip*) está individualmente relacionada com uma rota (*Route*). Por sua vez uma rota contém um grupo de paragens (*Stops*) que a constitui. Essas paragens, para além de estarem individualmente identificadas e registarem o seu ponto geométrico, estão contidas numa viagem respeitando um tempo e uma ordem (*Stop Times*), que ligadas criam a forma geométrica (*Shapes*) da via que constitui a viagem. A forma geométrica da viagem é composta por vários pontos intermédios às paragens, também estes, seguindo uma ordem no sentido do início ao fim da viagem. Um esquema de relação dos ficheiros está ilustrado na Figura 29.

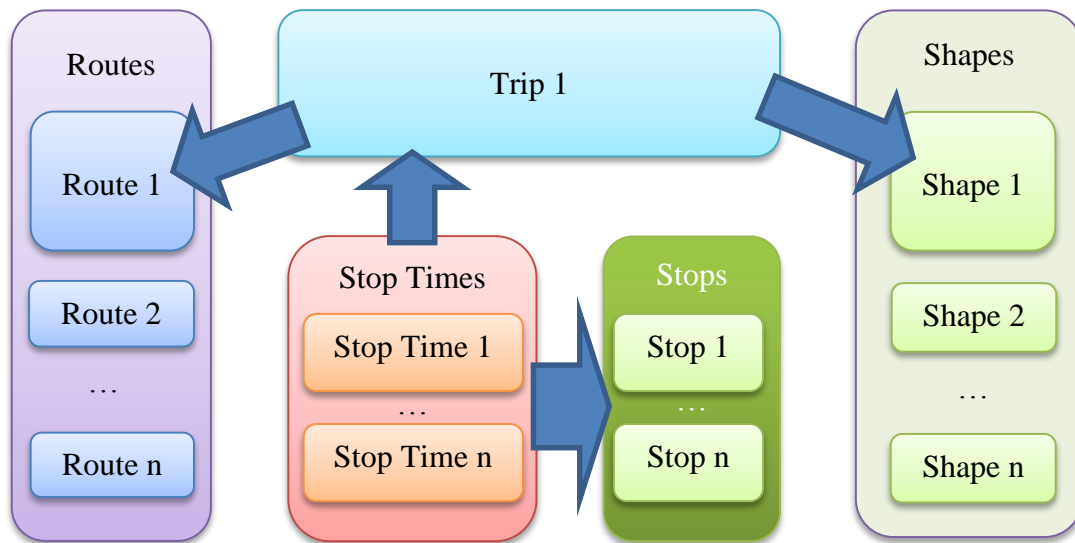


Figura 29: Esquema de relação dos ficheiros GTFS

Um dos desafios encontrados, na criação destes dados, foi a preparação destes para formar a rota no GoogleMaps, pois esta entrada é feita apenas com dados de pontos geométricos referentes à linha da rota. Têm de se retirar pontos ao longo da linha da rota, criando uma sequência de leitura, para que o GoogleMaps consiga reproduzir a linha da rota. A Figura 30 ilustra a rota da carreira 10 descrita por pontos.

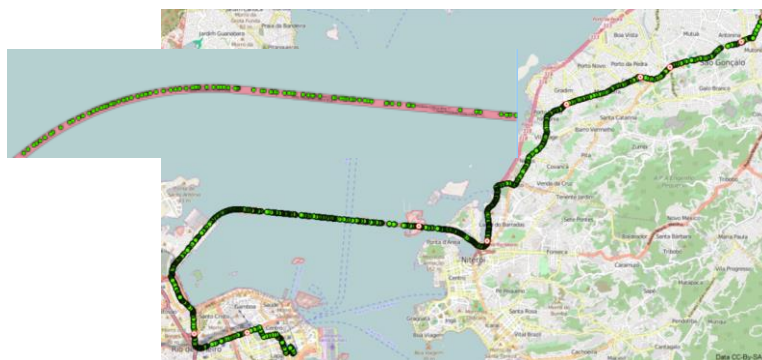


Figura 30: Criação de pontos geográficos para GTFS

Como o OSM é um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo, em que qualquer pessoa pode participar na atualização do sistema, a ordem das formas não segue qualquer regra, tornando a tarefa de ordenar uma linha de rota um desafio. A criação de uma *Linestring*, através da *Multilinestring* representativa da rota, com a ordem original do OSM está ilustrada na Figura 31.

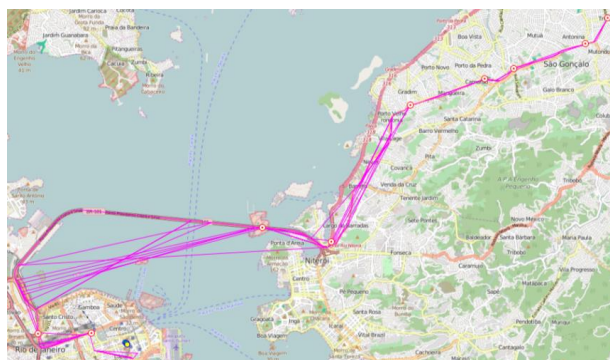


Figura 31: *Linestring* do OSM representativa da rota

Para colmatar esta questão, recorreu-se à criação de uma vista (“ordena_pontos”), com o registo das distâncias entre pontos. Todos os pontos dentro de uma determinada distância, equivalente à maior distância encontrada são registados de forma a incluir todos os pontos seguintes mais próximos.

Como esta seleção se baseia na distância, para além dos pontos seguintes também vai incluir pontos anteriores, obrigando a registar todos os pontos incluídos na ordem, como sendo anteriores, para serem descartados.

Código C# ordena as shapes:

```

...
// Create a string array that consists of lines for file shapes.txt
List<Object> shapes = new List<Object>();
List<Int32> plido = new List<Int32>();
...
cmd.CommandText = (
"SELECT id_order, gtfs_ini, gtfs_fin, mproximo, id_cvs, primeira_par FROM ordena_pontos
"
);
var pontos_ord = new Dictionary<Int64, Object[]>();
Int32 primeira_par = 0;
using (var reader0 = cmd.ExecuteReader())
{
    // Check is the reader has any rows at all before starting to read.

```

Capítulo 4 – Implementação

```
if (reader0.HasRows)
{
while (reader0.Read())
{
...

pontos_ord.Add(Int64.Parse(id_order), new Object[] {
    Convert.ToDouble(mproximo), Int32.Parse(gtfs_ini), Int32.Parse(gtfs_fin) });
primeira_par = Int32.Parse(first_stop);
}

Int32 atual = primeira_par;
//Order by the next closest point
for (int i = 1; i < pontos_ord.Count+1; i++)
{
    var dict = pontos_ord[i];
    if (!plido.Any() ||
        (plido.IndexOf((Int32)dict.GetValue(2)) == -1 && atual == (Int32)dict.GetValue(1))) {
        if (plido.Any()) {
            atual = (Int32)dict.GetValue(2);
            i = 0;
            plido.Add(atual);
        }
        else { // first item to enter
            plido.Add(atual);
            i = 0;
        }
    }
}
}
...
}
```

Outro desafio foi a preocupação da uniformização do sistema de coordenadas utilizadas, quando se trabalha com várias plataformas, o Sistema de Referência de Coordenadas (SRC) tem que ser o mesmo, correndo o risco de na interação entre sistemas estes não reconhecerem o tipo de dados, podendo assim trazer muitos dissabores no desenvolvimento de projetos.

Uma outra questão pertinente a ter em conta, é o rigor da informação dos horários, pois pode-se correr o risco de uma viagem terminar abruptamente por ter uma hora anterior na paragem seguinte (ilustrado na Figura 32).



Figura 32: Rota GTFS

Para o apoio da realização das tarefas deste Capítulo, foi implementada uma aplicação em .NET, com o intuito futuro de integração no sistema da Tecmic. Nesta aplicação é colocado um botão (“Gera GTFS”) que efetua todo o processo necessário para o registo dos ficheiros GTFS (Ilustrado na Figura 33), incluindo a ordenação essencial dos pontos da rota par apresentação no Google.

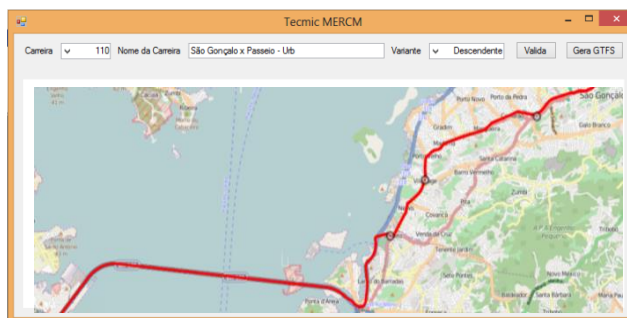


Figura 33: Aplicação DotNet de criação dos ficheiros GTFS

Após a escrita de todos ficheiros necessários para apresentação do serviço nos mapas do Google, apenas é necessária a colocação destes numa pasta com o formato recomendado “Nome_google_transit.zip” (*Feeds*) como se ilustra na Figura 34.

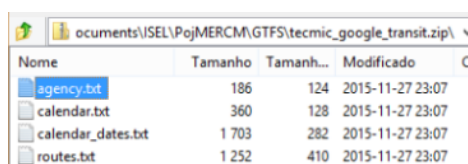


Figura 34: Pasta de ficheiros GTFS em zip

Após a criação da pasta comprimida com os ficheiros de dados da empresa de transportes, esta pode ser testada com várias ferramentas disponibilizadas no Google, como é demonstrado na secção 5.3.3.

4.4.1 Disponibilizar publicamente um Feed de transporte público

A forma mais acessível de disponibilizar um feed é ser introduzido num servidor web e publicar um anúncio que o disponibilize. Há locais específicos na web para os programadores poderem divulgar os seus feed's.

Uma lista de agências de transportes públicos que fornecem feed's públicos está disponível na página do projeto GoogleTransitDataFeed [32].

4.4.2 Enviar um feed de transporte público para o Google

Para as empresas de transportes públicos, com interesse em publicar a informação no GoogleMaps e a outras aplicações que publiquem informações sobre os transportes públicos, o Google disponibiliza ferramentas para testar e verificar a coerência dos dados.

Após os testes concluídos, tem de se colocar o *feed* num local do servidor e fazer o pedido ao Google, com a informação do local do ficheiro para a sua validação:

- http://myserver.agency.com/current/google_transit.zip [33]

Depois de validado é enviado para a empresa que pretende a publicação, e só é publicado depois de a empresa confirmar os dados a publicar.

5 Avaliação da solução

5.1 Comparativo da solução proposta com a atual

Este trabalho vai trazer uma forte melhoria de funcionamento, relativamente a facilidade e tempo despendido, na fase de uma operadora de transportes públicos pretender criar as suas rotas. O que outrora era feito de uma forma manual por um técnico, que na grande maioria desconhece o local, agora está à distância de um clique no botão da função.

Neste momento, quando a operadora se encontra na fase de calcular rotas para a frota de veículos e já na posse da localização das paragens, a criação das rotas é feita pelo operador de uma forma manual, olhando para o mapa e identificando a melhor rota entre cada paragem. Este processo para além de ser moroso acarreta algumas falhas por desconhecimento do local.

Com a aplicação deste trabalho, integrada no sistema da operadora, a tarefa de calcular as rotas da frota, torna-se uma tarefa rápida e precisa. O operador tem apenas de identificar as paragens pretendidas que serve de entrada para a rota e a aplicação faz o planeamento automático da rota, deixando apenas a tarefa de validação final da rota ao operador, se verificar a inexistência de erros.

Nesta aplicação também é acrescentado uma funcionalidade de grande importância, no ponto de vista comercial, pois são disponibilizados os dados para o Google de uma forma automática. À distância de um “click” tem a saída dos ficheiros no formato normalizado e no local pretendido.

5.2 Comparativo com o mercado

No mercado existem algumas ferramentas que disponibilizam o serviço de cálculo de rotas, como o GoogleMaps [30] e OSRM [36], mas estes serviços são pagos se forem utilizados a nível comercial. Um exemplo de cálculo de rota no GoogleMaps está ilustrado na Figura 35.

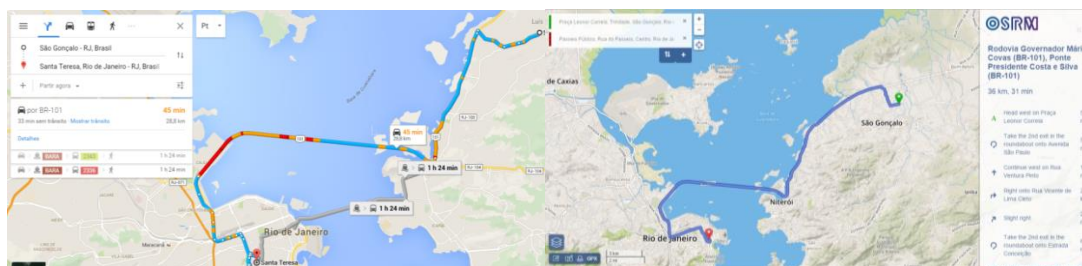


Figura 35: Cálculo de rota no GoogleMaps (Esq.) e OSRM (Dir.)

A aplicação deste trabalho (IRAPA) é baseada no OSM, como o planeador OSRM, logo herda grande parte das funcionalidades. A grande diferença está no desenvolvimento específico para os transportes públicos de passageiros, com uma função acrescida de preparação dos “inputs” para o Google.

Na Tabela 13, tendo em conta as definições propostas por [35] é feito um comparativo do protótipo desenvolvido (IRAPA) face a outros produtos comerciais existentes. Saliente-se que o

Capítulo 5 – Avaliação da Solução

trabalho foi desenvolvido para a área dos transportes públicos onde se procurou desenvolver uma solução sem custos que satisfizesse os requisitos da empresa Tecmic.

Tabela 13: Características de alguns planejadores de rotas on-line [35]

Funcionalidade (Descrição)	IRAPA (App)	MapQuest [6]	OSRM [7]	TripGo [8]	OpenTrip-Planner[9]	Finnder [11]
Cobertura	Global	Global	Global	Cidades Seleccionadas	Cidades Seleccionadas	Cidades Seleccionadas
Modos de transporte						
Carro (mais rápido)	Não	Sim	Sim	Sim	Não	Não
Carro (mais curto)	Não	Sim	Não	Sim	Não	Não
Pontos da Via	Sim	Sim	Sim	Apenas em Mode-change	Limitada	Não
Tráfego em tempo real	Não	Não	Não	Não	Não	Não
Planeamento em rotas férreas	Não	Sim	Sim	Sim	Sim	Não
Transito (transportes publicos)						
Rotas	Sim	Indeterminado	Não	Sim	Sim	Não
Horários	Sim	Indeterminado	Não	Sim	Sim	Não
Entrada de GTFS	Sim	Indeterminado	Não	Sim	Sim	Não

Algumas páginas *web* que disponibilizam as ferramentas de planeamento de rotas, também contemplam ferramentas para a criação dos dados no formato GTFS, como é o caso da Nationalrtap [31]. Esta ferramenta apenas carece dos dados em Excel para a exportação dos dados em GTFS. Esta empresa apenas pede dois ficheiros com os dados formatados de forma a serem reconhecidos pela aplicação (Figura 36).

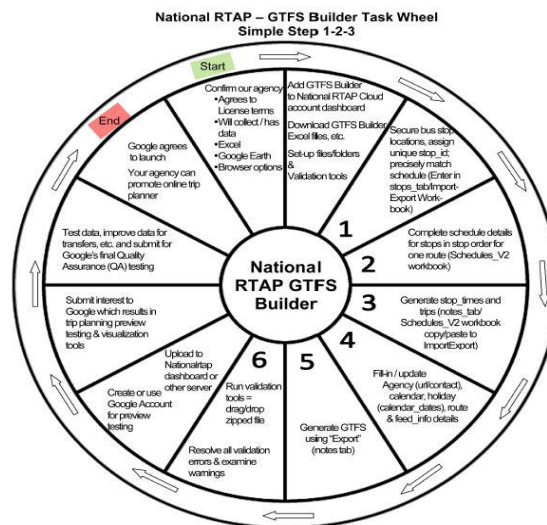


Figura 36: Roda de tarefas para o construtor GTFS da RTAP [31]

5.3 Testes da solução

5.3.1 Teste de algoritmo a utilizar

Para a escolha do algoritmo utilizado, efetuaram-se testes e comparação entre os algoritmos suportados pelo PgRouting. Em questões da área abrangida no processamento da procura da rota, conclui-se que a melhor opção de uso é o algoritmo de “*Shortest Path A**”.

Ilustrado à esquerda da Figura 37, este algoritmo apresenta uma maior eficiência em questão de tempo e área na pesquisa, para o planeamento da mesma rota comparando com o algoritmo “*Dijkstra Shortest Path*”.



Figura 37: *Shortest Path A** versus *Shortest Path Dijkstra*

Fazendo uma análise apenas em questão de tempo, conclui-se que os algoritmos “*Bi-directional Shortest Path A**” e “*Bi-directional Dijkstra Shortest Path*” são mais rápidos, mas em questão de eficiência do resultado, não apresenta resultados tão satisfatórios como os seu homónimos sem a bidirecionalidade.

A Figura 38 ilustra um teste do algoritmo “*Shortest Path A**” (à esquerda). Este é baseado no algoritmo de Dijkstra com uma heurística que lhe permitem resolver a maioria dos problemas de caminho mais curto por avaliação apenas de um subconjunto do gráfico geral.

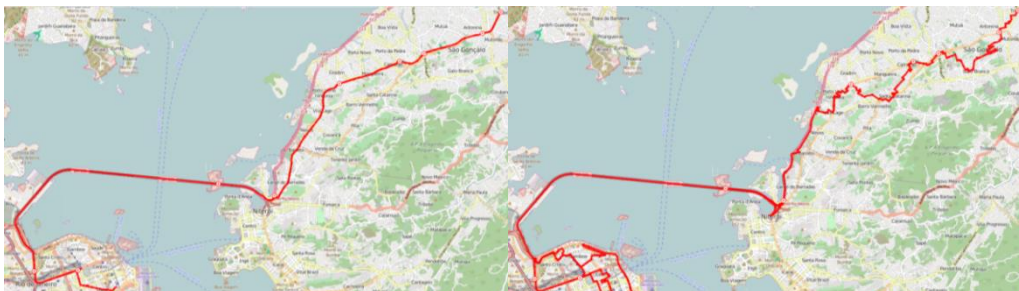


Figura 38: Algoritmo “*Shortest Path A**” e “*Bi-directional Shortest Path A**”

À direita da Figura 39 está ilustrado o algoritmo “*Bi-directional Shortest Path A**”. Este procura a partir da origem em direção ao destino e ao mesmo tempo a partir do destino para a origem e termina quando ambas as pesquisas se encontram a meio. Como se pode verificar o comportamento da pesquisa bidirecional, apesar de mais rápido, não apresenta o caminho mais curto.

O algoritmo de “*Dijkstra Shortest Path*”, concebido pelo cientista da computação holandês Edsger Dijkstra em 1956. É um algoritmo de pesquisa a solução num gráfico para o caminho mais curto desde a origem ao destino, com os custos não-negativos, produzindo uma árvore de pesquisa do caminho mais curto. O teste realizado com este algoritmo está ilustrado à esquerda da Figura 39.

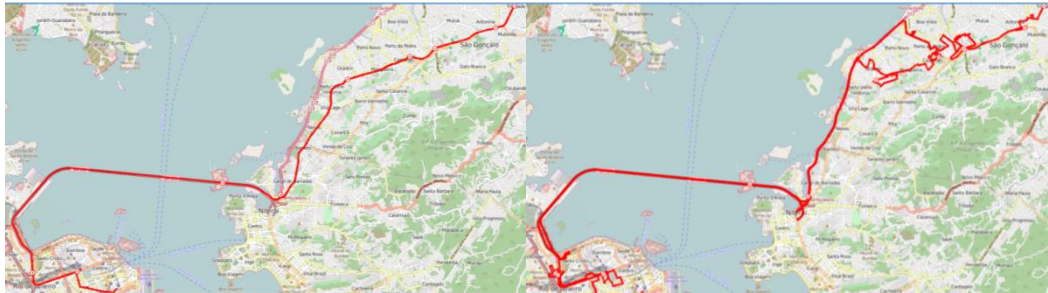


Figura 39: Algoritmo “*Dijkstra Shortest Path*” e “*Bi-directional Dijkstra Shortest Path*”

O “*Bi-directional Dijkstra Shortest Path*” é um algoritmo de busca bidirecional Dijkstra. Este procura a partir da fonte em direção destino e ao mesmo tempo a partir do destino para a origem e termina a pesquisa quando se encontram no meio. Retorna um conjunto de `pgr_costResult` (seq, id1, id2, custo) linhas, que compõem um caminho.

À semelhança do algoritmo “*Bi-directional Shortest Path A**”, o algoritmo “*Bi-directional Dijkstra Shortest Path*” também procura o caminho mais curto em dois sentidos encontrando-se a meio. Da mesma forma verificou-se que o comportamento da pesquisa bidirecional, apesar de mais rápido, não apresenta o caminho mais curto.

Conclui-se com estes testes o que foi referido no primeiro parágrafo deste Capítulo, que a melhor opção de uso é o algoritmo de “*Shortest Path A**”.

5.3.2 Comparativo de planeamento de rota manual e automático

Para avaliar a usabilidade da aplicação deste trabalho, foi realizado um planeamento de rota que normalmente fosse desempenhado por um técnico da operadora, em que os dados das rotas são criados com a leitura manual da passagem dos veículos (a amarelo na Figura 40), retirado da base de dados disponibilizada pela Tecmic.

O processo de calcular as rotas começa com a leitura geográfica dos locais das paragens pretendidas para cada rota. Estas leituras são recolhidas por um veículo que se desloca no local e regista a posição GPS de cada paragem, transmitindo de seguida para a base de dados da empresa, de modo a ficar registado.

Após a obtenção da informação GPS das paragens, é iniciado o processo de planeamento da rota para a carreira em causa. Neste momento entra o operador que tem agora a possibilidade de calcular a rota de modo automático, possibilitando uma rápida visualização da rota (a vermelho na Figura 40).

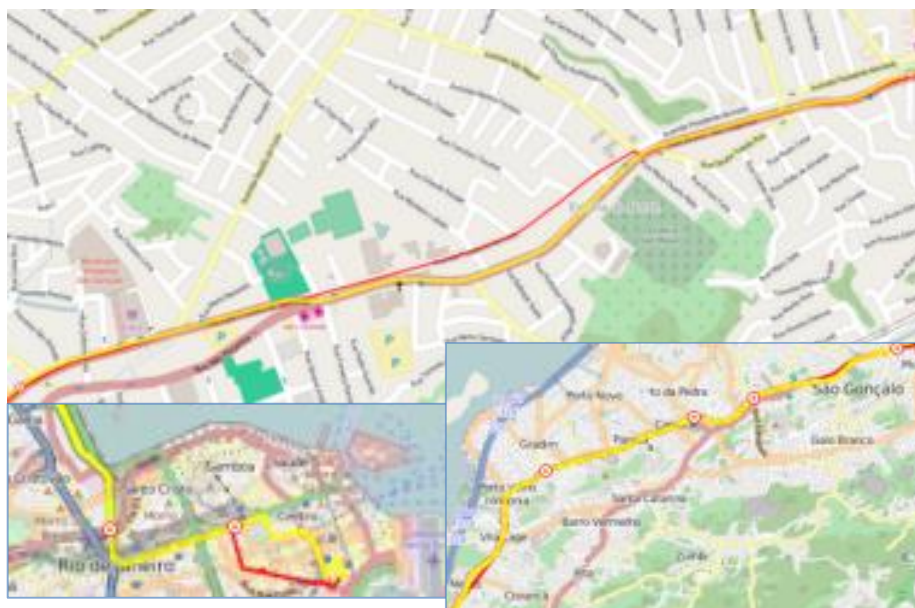


Figura 40: Comparação de rota manual e automática

Com este teste conclui-se que não há grandes alterações entre a rota criada de uma forma manual e a gerada pela aplicação deste trabalho. Verificou-se algumas mudanças de vias em determinados trechos que implicariam uma melhoria no trajeto. Nesta fase da análise, o operador de transportes públicos, faz o seu juízo das diferenças optando, ou não, pela rota automática.

Foram selecionadas as leituras das paragens pertencentes à rota nº110 sentido ascendente e fez-se correr a aplicação. O resultado está ilustrado na Figura 41.

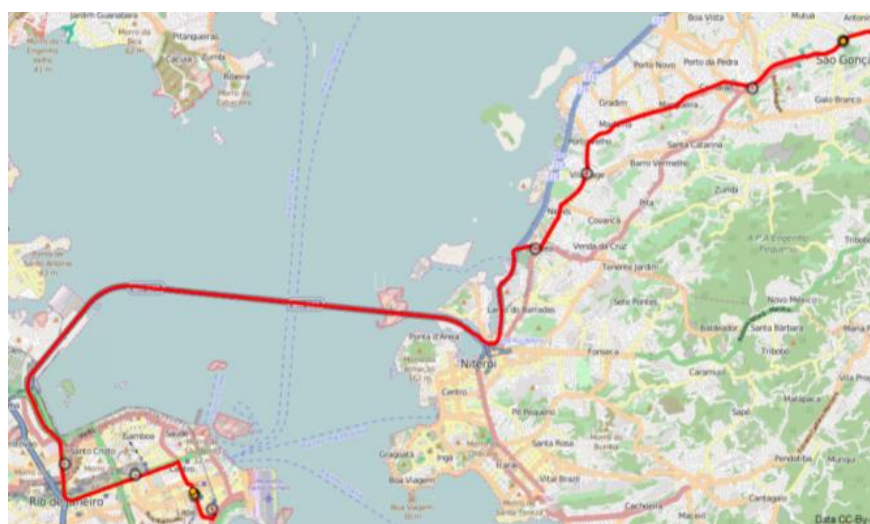


Figura 41: Teste de planeamento de rota

5.3.3 Teste de ficheiros criados para o GTFS

No caso de testar os resultados obtidos na criação dos dados a importar para o Google, usei os dados fornecidos pela Tecmic, com as informações necessárias à criação dos *Feeds*.

Capítulo 5 – Avaliação da Solução

Durante o processo de criação utilizei uma ferramenta para teste disponibilizada pelo Google, que possibilita a compilação de erros existentes nos *Feeds* GTFS criados (“*FeedValidator*” [29]). Esta ferramenta vai procurar problemas e reporta-os gerando um HTML (ilustrado na Figura 42) permitindo uma rápida correção. Esta ferramenta é muito útil, pois não só identifica o problema como também o local onde se encontra, poupando muito tempo na procura do problema.

GTFS validation results for feed:
C:\Users\Filipe\Documents\ISEL\PojMERC\GTFS\transifed-wir
FeedValidator extension used: extensions.googletransit

Agencies: [Tecmic](#)
Routes: 24
Stops: 310
Trips: 2342
Shapes: 33
Effective: January 02, 2015 to January 02, 2016

During the upcoming service dates Sun Nov 15 to Fri Jan 01:
Average trips per date: 1316
Most trips on a 2342, on 24 service dates (Mon Nov 16, Tue Nov 17, date: Wed Nov 18, ...)
Least trips on a 0, on 16 service dates (Sun Nov 15, Sat Nov 21, Sun date: Nov 22, ...)

Found these problems:
2650 warnings

- 1 [Expiration Date](#)
- 4 [Invalid Values](#)
- 2342 [Other Problems](#)
- 2 [Stops Too Close](#)
- 301 [Unused Stops](#)

Warnings:

Expiration Date

- This feed will soon expire, on January 02, 2016 (last service date in calendar.txt)

Invalid Value

- Invalid value 101 in field route_long_name
route_long_name shouldn't be the same the route_short_name value, as both fields are often displayed side-by-side. It's OK to omit either the short or long name (but not both).
in line 18 of routes.txt

route_id	agency_id	route_short_name	route_long_name	route_type
101	TC	101	101	3

- Invalid value 426 in field route_long_name
route_long_name shouldn't be the same the route_short_name value, as both fields are often displayed side-by-side. It's OK to omit either the short or long name (but not both).
in line 19 of routes.txt

route_id	agency_id	route_short_name	route_long_name	route_type
426	TC	426	426	3

Figura 42: Teste de validação do GTFS criado

Após a retificação de todos os erros e avisos, podemos visualizar o conteúdo da forma como irá aparecer no Google com outra ferramenta disponibilizada para os *Feeds* GTFS (“*schedule_viewer*” [29]).

Tecmic

Time: 8:00
Date: select
Find Station: Search
Find Trip ID: Search

10 São Gonçalo x Estácio 101 101 11:11
110 São Gonçalo x Passeio - Urb 110 110 11:53
Praça Trindade to Central Brasil, 9 stops, 1 trips 11:53
110A São Gonçalo x Passeio - V 110A 110A 12:04
110E São Gonçalo x Passeio - V 110E 110E 12:09
110R São Gonçalo x Passeio via 12 12
12345 TESTE 12345 12345

trips.txt direction_id=0 route_id=110 trip_headsign= shape_id=16 service_id=Mon-Fri trip_id=110ASC160 trip_short_name=São Gonçalo x Passeio - Urb
routes.txt route_type=1 route_id=110 route_short_name=110 route_long_name=São Gonçalo x Passeio - Urb agency_id=TC

Figura 43: Teste de horário e geometrias criadas no GTFS

Esta ferramenta apresenta uma página HTML com os dados das rotas e horários no mapa (ilustrado na Figura 43), da mesma forma como irá ser apresentado no GoogleMps depois de validado pela equipa do Google.

6 Conclusões

O presente trabalho resultou de uma necessidade real de uma empresa e constitui um trabalho aplicado com sistema ao desenvolvimento de uma solução feita com ferramentas *Open Source Software* (OSS) preparada para ser usada pela empresa. Uma das grandes dificuldades sentidas foi o curto espaço de tempo e a necessidade de interação da Tecmic, e das limitações de escolha de *software* para poder ser integrado no sistema existente. O trabalho consistiu em criar uma ferramenta para ajudar a Tecmic na implementação do seu sistema XTraN Passenger, que fornece ferramentas de apoio a operadoras de transportes. O presente trabalho foi apresentado na Tecmic no dia 24.09.2015 onde se mostrou as funcionalidades desenvolvidas a integração com o XTraN e a exportação de resultados para o Google. O feedback recebido foi positivo e a ferramenta permite reduzir o trabalho feito pelos técnicos da Tecmic. Foi identificado o técnico da Tecmic (Daniel Crespo) que ira seguir o processo de implementação (da integração da aplicação que cria os *feed's* GTFS) e coordenar o posterior uso do produto desenvolvido na Tecmic. De igual forma foram identificadas mais requisitos os quais vou identificar como trabalho futuro.

A solução desenvolvida apresenta uma grande modularidade e encontra-se otimizada para o modelo de dados do XTraN Passenger embora consista numa primeira aproximação para o problema e irá necessitar de melhorias, função dos resultados obtidos com os testes reais a realizar. Foi acordo separar o módulo GTFS na sua aplicação XTraN Passenger visando desta forma complementar a sua oferta ao mercado com novas funcionalidades.

O presente trabalho pretende ser implementado em situações reais em empresas de transporte onde a solução proposta vai servir para disponibilizar uma ferramenta rápida para os operadores dos transportes públicos. Dado a crescente expansão do negócio da Tecmic no Brasil esta ferramenta será usada na próxima implementação comercial do XtraN no Brasil. Este tipo de serviço também providencia uma extração da informação no formato do Google, útil para o negócio pois vai levar a informação a um grupo maior de clientes. Este serviço vai ser aplicado em situações reais onde a Tecmic tinha pendente o processo de exportação para o Google, como é o exemplo do operador Auto Lotação Inga no Rio De Janeiro.

O trabalho consistiu numa aplicação que permitisse recolher coordenadas GPS de paragens e partir destas estabelecer rotas tendo em conta as estradas existente e corrigindo erros de GPS. Usei como ferramenta base de dados geográfica, o *OpenLayers* e algoritmos para determinar percursos.

A parte de exportação para o Google permitiu o estudo de padrões para a exportação de dados e verificar o conceito de interoperabilidade dos ITSs de uma forma prática, aplicando regras normativas nos registos dos dados para exportação.

Para finalizar apenas referir que no presente trabalho foi feito uma abordagem pratica na área do GIS a um caso concreto de uma necessidade de uma empresa, na qual desenvolvi em código aberto uma solução que permite o planeamento automático de rotas, como uma ferramenta de apoio ao operador de transportes, e criar os ficheiros (Feeds) de *input* para o Google (GTFS) de uma forma automática.

6.1 Trabalho Futuro

A Tecmic definiu novas necessidades, as quais são essencialmente relacionadas com otimização tendo em conta o conjunto das rotas definido na globalidade para o operador e as existentes na zona de outros operadores. Este processo pode ser atingido pela corelação das rotas

definidas com rotas de outras operadoras de transportes instaladas no local, identificando percursos comuns.

Adicionalmente será possível estender o trabalho à determinação de percursos baseado em todas as rotas existentes, de forma a serem otimizadas, economizando troços desnecessários. Essa vertente é claramente uma mais-valia para a Tecmic e todas as operadoras de transportes de passageiros.

Uma variante possível será a inclusão de dados de mobilidade dos utentes, inferidas a partir das necessidades levantadas dos utentes e qual permitia uma aproximação maior entre a oferta e as necessidades reais dos utentes de transporte.

ANEXOS

Anexo 1: Código SQL/PlpgSQL para manipulação de Base de Dados

Por favor consultar CD do Projeto na pasta de Anexos 1

Anexo 2: Código C# para aplicação de *User Interface (UI)*

Por favor consultar CD do Projeto na pasta de Anexos 1

Anexo 3: CENTC 278, CEN TC278WG3 - 2008

Por favor consultar CD do Projeto na pasta de Anexos 3

Anexo 4: Correcting GPS Readings from a Tracked

Por favor consultar CD do Projeto na pasta de Anexos 4

Anexo 5: QGIS-2.2-QGIS Training Manual-en

Por favor consultar CD do Projeto na pasta de Anexos 5

Anexo 6: PgRouting Documentation

Por favor consultar CD do Projeto na pasta de Anexos 6

Anexo 7: GTFS (Especificação Geral sobre Feeds de Transporte Público)

Por favor consultar CD do Projeto na pasta de Anexos 7

Anexo 8: Identification of Fixed Objects in Public Transport (IFOPT)

Por favor consultar CD do Projeto na pasta de Anexos 8

REFERÊNCIAS

- [1] M.K. Natvig, A. Vennesland, Flexible organisation of multimodal travel information services, IET Intelligent Transport Systems 4 (4) (2010) 401–412.
- [2] E-Frame, European ITS Framework Architecture, 2011 Available at: <http://www.frame-online.net/> (Acedido em Julho de 2015).
- [3] Z. Belinova, P. Bures, P. Jesty, Intelligent transport system architecture different approaches and future trends, Advances in Intelligent and Soft Computing 81 (2010) 115–125.
- [4] R. Jardim-Goncalves, A. Grilo, C. Agostinho, F. Lampathaki, Y. Charalabidis, Systematisation of interoperability body of knowledge: the foundation for EI as a science, Enterprise Information Systems (2012), <http://www.tandfonline.com/doi/pdf/10.1080/17517575.2012.684401> (Acedido em Julho de 2015).
- [5] A. Grilo, R. Jardim-Goncalves, Value proposition on interoperability of BIM and collaborative working environments, Automation in Construction 19 (5) (2010) 522–530, <http://www.sciencedirect.com/science/article/pii/S0926580509001733> (Acedido em Julho de 2015).
- [6] R. Jardim Goncalves, A. Grilo, A. Steiger-Garcao, Challenging the interoperability between computers in industry with MDA and SOA, Computers in Industry 57 (2006) 679–689, <http://www.sciencedirect.com/science/article/pii/S0166361506000753> (Acedido em Julho de 2015).
- [7] D. Soares, L. Amaral, Information systems interoperability in public administration: identifying the major acting forces through a Delphi study, Journal of Theoretical and Applied Electronic Commerce Research 6 (2011) 61–94.
- [8] European Commission, European Interoperability Framework for Pan-European eGovernment Services, IDABC Program, European Commission, 2004 Available at: <http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529> (Acedido em Julho de 2015).
- [9] European Commission, European Interoperability Framework (EIF), European Union, 2011 Available at: http://ec.europa.eu/isa/documents/eif_brochure_2011.pdf (Acedido em Julho de 2015).
- [10] <http://www.itsstandards.eu> (Acedido em Julho de 2015).
- [11] <http://www.cen.eu> (Acedido em Julho de 2015).
- [12] European Committee for Standardization, Transmodel v5.1, Reference Data Model For Public Transport, CEN EN12896.
- [13] European Committee for Standardization, IFOPT, Identification of Fixed Objects in Public Transport, CEN/TS 28701.
- [14] European Committee for Standardization, SIRI, Standard Interface for Real-Time Information, CEN/TS 15531, parts 1–5.
- [15] European Committee for Standardization, NeTEx, Network and Timetable Exchange, CEN TC 278 WG3 SG9, work in progress.
- [16] P. Levjakangas, J. Haajanen, A.M. Alaruikka, Information service architecture for international multimodal logistic corridor, IEEE Transactions on Intelligent Transportation Systems 8 (4) (2007) 565–574.
- [17] EU-Spirit, 2005. Available at: <http://eu-spirit.eu/> (Acedido em Setembro de 2015).
- [18] http://wiki.openstreetmap.org/wiki/About_OpenStreetMap (Acedido em Setembro de 2015).
- [19] Dr. Andrej Tibaut, Standard driven approach for sustainable interoperability of public transport passenger information systems, Tibaut-CI_final.pdf, 2013.
- [20] <http://www.tecmic.pt/> (Acedido em Setembro de 2015).
- [21] <https://developers.google.com/transit/gtfs/> (Acedido em Setembro de 2015).
- [22] Algoritmos Baseados em Instâncias. (Disciplina AMD do Curso MERCM – Prof. Dr. Paulo Trigo Silva).
- [23] <http://geops.de/blog/mapping-public-transit-networks?language=en> (Acedido em Setembro de 2015).
- [24] <http://anitagraser.com/2013/07/06/pgrouting-2-0-for-windows-quick-guide/> (Acedido em Setembro de 2015).
- [25] <https://msdn.microsoft.com/en-us/library/ee658109.aspx> (Acedido em Setembro de 2015).
- [26] http://www.trimble.com/gps_tutorial/dgps-how.aspx (Acedido em Setembro de 2015).
- [27] http://www.anmpn.pt/informacoes/2002/rede_dgps.htm (Acedido em Setembro de 2015).
- [28] Richard Milton and Anthony Steed, Correcting GPS Readings from a Tracked Mobile Sensor, Correcting GPS Readings from a Tracked Mobile Sensor.pdf, 2004.
- [29] <https://github.com/google/transitfeed/wiki/FeedValidator> (Acedido em Setembro de 2015).
- [30] <https://www.google.pt/maps/> (Acedido em Setembro de 2015).
- [31] <http://nationalrtap.org/> (Acedido em Setembro de 2015).
- [32] <https://code.google.com/p/googletransitdatafeed/wiki/PublicFeeds> (Acedido em Setembro de 2015).
- [33] <https://maps.google.com/help/maps/mapcontent/transit/participate.html#host> (Acedido em Setembro de 2015).
- [34] https://en.wikipedia.org/wiki/Category:Classification_algorithms (Acedido em Setembro de 2015).
- [35] http://wiki.openstreetmap.org/wiki/Routing/online_routers (Acedido em Setembro de 2015).
- [36] <http://project-osrm.org/> (Acedido em Setembro de 2015).