



Projeto

Mestrado em Engenharia Informática - Computação Móvel

*Desenvolvimento de solução tecnológica para
recolha e disponibilização de informação de
transportes públicos*

Tiago José Jesus Pedro

Leiria, Março de 2017



Projeto

Mestrado em Engenharia Informática - Computação Móvel

*Desenvolvimento de solução tecnológica para
recolha e disponibilização de informação de
transportes públicos*

Tiago José Jesus Pedro

Projeto de Mestrado realizado sob a orientação do Doutor Luís Filipe Fernandes Silva Marcelino e da Doutora Catarina Helena Branco Simões da Silva, professores da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, Março de 2017

Resumo

Com as crescentes dificuldades em utilizar os automóveis como meios de transporte pessoal, devido a problemas ambientais e financeiros, uma das alternativas que mais tem resistido ao longo dos anos consiste nos transportes públicos, especialmente nas grandes metrópoles. Ao mesmo tempo, com a necessidade destas expandirem e melhorarem os seus serviços, passaram a ser usados vários sistemas tecnológicos para informar os seus utentes do estado dos seus transportes e atrasos. No entanto, é possível verificar que em cidades pequenas e médias, grande parte da informação acerca dos seus transportes ainda é disponibilizada em papel nas paragens ou num panfleto *online*, onde estão discriminados os horários das paragens. Para resolver este problema, foi desenvolvida uma solução capaz de auxiliar as companhias de transportes públicos a desenvolverem e difundirem os seus dados em formato digital e a melhorarem o bem-estar dos utentes dos transportes públicos, disponibilizando-lhes ferramentas que lhes permitam usar estes serviços com mais facilidade.

Palavras-chave: Planeador de rotas, GTFS, Android, Mobile, Transportes

Abstract

With the increasing difficulties of using cars as a mean of personal transportation, due to environmental problems and gas/maintenance costs, one of the best alternatives to this mean of transportation, especially in big cities, are the public transports. At the same time, with the necessity to expand and improve their transports, this cities are employing various technological systems to inform their users of the state and delays of their vehicles. However, it is noticeable that in smaller cities, most of the information about the public transports are available at the various stops in paper format and in *online* brochures with the timetables. To solve this problem, it was developed a solution capable of not only helping the public transport companies create/update the digital information about their transports but also provide various tools for the transport users to improve their usability.

Keywords: Trip Planner, GTFS, Transports, Android, Mobilis

Lista de Figuras

2.1	Ligação entre os vários ficheiros GTFS	7
2.2	Arquitetura do Lisboa Bus System	10
2.3	Descrição de um plano de trajeto no OpenTripPlanner	11
2.4	Percurso e paragens de uma linha de autocarro no OpenStreetMap	13
2.5	Descrição de um plano de trajeto no Google Maps	14
2.6	Arquitetura de sistema do Transitr[9][13]	19
2.7	Interface da aplicação Moov.it	19
2.8	Página principal do OneBusAway	21
2.9	Interface da aplicação UK Bus Checker	22
2.10	Interface da aplicação Next Bus - Porto	23
2.11	Interface da aplicação TransitApp	24
2.12	Interface da aplicação CityMapper	26
2.13	Interface da aplicação Sapo Transportes	27
2.14	Interface da aplicação Move-Me	29
2.15	Interface da aplicação OpenTripPlaner for Android	30
2.16	Interface da aplicação StarMetro	32

3.1	Arquitetura do sistema	47
4.2	Estrutura do ficheiro calendar.txt	75
4.3	Estrutura do ficheiro frequenciasecra.csv	76
4.4	Estrutura do ficheiro periodosle.csv	76
4.5	Estrutura do ficheiro internetMobilis.csv	76
4.6	Estrutura do ficheiro calendar_dates.txt	78
4.7	Estrutura do ficheiro feriados.csv	78
4.8	Estrutura do ficheiro feriadosle.csv	79
4.9	Estrutura do ficheiro stops.txt	80
4.10	Estrutura do ficheiro MasterStageList.csv	80
4.11	Estrutura do ficheiro desc_paragens2.csv	80
4.12	Estrutura do ficheiro routes.txt	82
4.13	Estrutura do ficheiro servicos.xml	82
4.14	Estrutura do ficheiro stop_times.txt	84
4.15	Estrutura do ficheiro internet_mobilis.csv para a <i>trip</i> 101001	84
4.16	Estrutura do ficheiro stop_times.txt para a <i>trip</i> 101001	85
4.17	Horários com o mesmo ID mas frequência diferentes	87
4.18	Horários com o mesmo ID partidas iguais e chegadas diferentes	87
4.19	Estrutura do ficheiro trips.txt	88

4.20	Estrutura do ficheiro stops.txt	89
4.21	Estrutura do ficheiro stops.txt	89
4.22	Estrutura resultante de origem para o Agency.txt	89
4.23	Estrutura do ficheiro fare_attributes.txt	90
4.24	Estrutura resultante de origem para o Fare_attributes.txt	90
5.1	Estatísticas da mudança de horário de um planeamento	98
5.2	Estatística da mudança de linha na atividade "Paragens"	98
5.3	Estatísticas de utilização das textboxes, GPS e mapas no planeamento de uma rota	99
5.4	Estatísticas de utilização da Infowidget VS Botão de descrição no pla- neamento de um trajeto	100
6.1	Protótipos iniciais do Mobilis Trip Planner	109
6.2	Protótipos segunda fase do Mobilis Trip Planner	110
6.5	Divisão das pessoas nos teste principal	113
6.6	Quantidade de erros para conseguir o primeiro horário	114
6.7	Quantidade de erros para conseguir o primeiro planeamento	114
6.8	Tempo para conseguir o primeiro horário	114
6.9	Tempo para conseguir o segundo horário	115
6.10	Tempo para conseguir o primeiro planeamento	115
6.11	Tempo para conseguir o segundo planeamento	115

6.12	Tempo para ver a descrição dum planeamento	116
6.13	Tempo para ver a descrição dum planeamento	116

Lista de Tabelas

2.1	Características das plataformas de difusão de informação	16
2.2	Características mais importantes dos clientes de informação de informação	34
3.1	Tabela de atores e sistemas	39
3.2	Requisitos funcionais do planeador de rotas do ponto de vista do utilizador	40
3.3	Requisitos funcionais do planeador de rotas do ponto de vista do administrador	41
3.4	Requisitos funcionais do Planeador Android	42
3.5	Requisitos funcionais do Template	43
3.6	Requisitos funcionais do Template para os ficheiros GTFS opcionais . .	44
3.7	Requisitos funcionais do Tradutor GTFS Mobilis	44
3.8	Requisitos funcionais do Servidor de informação dinâmica	45
3.9	Requisitos não funcionais do sistema	46

Lista de Siglas

Acrónimos

API Application Programming Interface

CSV Comma-Separated Values

GPS Global Positioning System

GTFS General Transit Feed Specification

GTFS-Realtime General Transit Feed Specification - Realtime

NTFS Navitia Transit Feed Specification

RAM Random-Access memory

REST REpresentational State Transfer

URL Uniform Resource Locator

XML eXtensible Markup Language

Índice

Resumo	III
Abstract	V
Lista de Figuras	X
Lista de Tabelas	XI
Lista de Siglas	XIII
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação e objetivos	2
1.3 Estrutura do documento	3
1.4 Metodologia de desenvolvimento	4
2 Estado da Arte	5
2.1 Estruturação e validação de dados dos transportes públicos	5
2.1.1 General Transit Feed Specification	6
2.1.2 GTFS-Realtime	8
2.1.3 FeedValidator	9
2.2 Plataformas de difusão de informação de transportes públicos	9
2.2.1 Retrieving real-time information: Lisbon bus system	9
2.2.2 OpenTripPlanner	11
2.2.3 Navitia.io	12
2.2.4 OpenStreetMap	13
2.2.5 Google Transit e Google Maps	14
2.2.6 OneStopTransport	15
2.2.7 Resumo de funcionalidades	16
2.3 Clientes de informação de transportes públicos	18
2.3.1 Transitr	18
2.3.2 Moov.it	19
2.3.3 One Bus Away	21
2.3.4 Bus Checker	22
2.3.5 Next Bus - Porto	23
2.3.6 TransitApp	24
2.3.7 CityMapper	26
2.3.8 Sapo Transportes	27
2.3.9 Move-Me	29
2.3.10 OpenTripPlanner for Android	30
2.3.11 StarMetro	32
2.3.12 Resumo de funcionalidades	33

2.4	Conclusão	35
3	Proposta de Solução	37
3.1	Objetivos da solução proposta	37
3.2	Requisitos do Sistema	38
3.2.1	Atores e sistemas envolvidos neste trabalho	38
3.2.2	Lista de requisitos funcionais	39
3.2.3	Lista de requisitos não-funcionais	45
3.3	Arquitetura do sistema	47
3.4	Expansão do Mobilis	49
3.4.1	Características introduzidas com a atualização do Mobilis	49
3.4.2	Implicações a nível do GTFS	50
3.4.3	Especificação de um formato e estrutura para um template auxiliar de criação/atualização de dados GTFS	51
3.5	Conclusão	52
4	Implementação	55
4.1	GTFS	55
4.1.1	Correção e reestruturação dos ficheiros em formato GTFS	55
4.1.2	Template Excel	59
4.2	OpenTripPlanner	65
4.2.1	Desafios e requisitos do OpenTripPlanner	66
4.2.2	Mapa para a zona de Leiria	67
4.2.3	Tradução para Português do OpenTripPlanner	67
4.2.4	GTFS-Realtime	70
4.3	Componentes dedicadas aos transportes Mobilis	71
4.3.1	Mobilis Trip Planner	71
4.3.2	Mobilis GTFS Creator	73
4.4	Conclusão	90
5	Avaliação e Testes	93
5.1	Fases de teste	93
5.2	Testes preliminares	94
5.2.1	Correção do Mobilis Trip Planner	96
5.3	Testes com utilizadores	96
5.3.1	Correção do Mobilis Trip Planner depois dos testes principais	101
5.4	Conclusão	101
6	Conclusão	103
	Bibliografia	107
	Anexos	109

Capítulo 1

Introdução

Neste capítulo é apresentado o enquadramento, explicando a situação atual sobre os transportes públicos, motivação e objetivos deste trabalho, seguido da estrutura deste relatório e da metodologia de desenvolvimento.

1.1 Enquadramento

O transporte de pessoas sempre foi algo essencial para qualquer cidade, grande ou pequena. Com os crescentes custos da utilização de automóveis para deslocação pessoal, existe a necessidade de disponibilizar meios de transportes alternativos a estes [11][3][6]. Uma das soluções para este problema, especialmente nas grandes cidades, consiste na utilização de transportes públicos, apresentando vantagens a vários níveis, nomeadamente:

- **Economia:** dado o reduzido custo dos bilhetes dos transportes públicos face aos custos de manutenção, combustíveis e tarifas de parque de estacionamento para os automóveis [11][3][6];
- **Ambiente:** dado o impacto ambiental causado por transportes públicos como os autocarros ou comboios ser inferior ao de cada pessoa usar um automóvel para se deslocar [11][3][6];
- **Turismo:** dado que muitos turistas durante as suas visitas procuram uma solução de mobilidade acessível e integrada no ambiente urbano a explorar[14][7].

Apesar das vantagens apresentadas pelos transportes públicos, também existem

alguns desafios, nomeadamente:

- O facto de muitas vezes estes transportes não possuírem paragens próximas do local de origem ou destino, principalmente para pontos fora das cidades, obrigando a que o início e o fim de trajeto sejam percorridos a pé ou por outro meio [8][15][10];
- A necessidade de espera por um transporte público, principalmente quando há atrasos desconhecidos [9][4];
- O desconhecimento dos horários dos transportes públicos, por exemplo, em dias de feriado ou em ocasiões especiais;
- O desconhecimento do serviço prestado como novas linhas ou paragens disponibilizadas pela agência de transportes.

Como forma de minimizar estes problemas, algumas cidades (principalmente de grandes dimensões) decidiram optar por sistemas tecnológicos que lhes permite difundir nova informação, como por exemplo novas paragens. No entanto, para pequenas e médias cidades, é possível notar que ainda não são muito utilizados sistemas informáticos e/ou dispositivos móveis para difusão e leitura de informação acerca dos transportes públicos. Os dispositivos móveis, como *smartphones* e *tablets*, podem tornar-se excelentes plataformas com a capacidade de fornecer informação aos seus utilizadores acerca dos transportes públicos a qualquer hora e qualquer lugar graças à Internet [9][4][5]. É neste contexto de pequenas e médias cidades que pretendam melhor difundir os seus transportes públicos, que surge a solução desenvolvida neste trabalho.

1.2 Motivação e objetivos

Apesar das grandes cidades já utilizarem ferramentas que oferecem aos utilizadores dos transportes públicos a capacidade de planear rotas, ver a localização, atrasos e saber da disponibilidade destes [9][4], nas pequenas e médias cidades ainda são apenas disponibilizadas *online* brochuras com informação estática sobre os transportes públicos. Este trabalho tem desta forma o intuito de auxiliar as companhias de transportes públicos/cidades a desenvolverem e difundirem os seus dados em formato digital e de melhorar o bem-estar dos utentes destes transportes, disponibilizando-lhes ferramentas

que lhes permitam usar estes serviços com mais facilidade. Para alcançar estes objetivos, esta solução disponibiliza uma plataforma Cliente-Servidor *Open-source* para o planeamento de rotas multi-modal capaz de difundir informação estática, como a localização das paragens, um planeador de rotas em Android e um Template capaz de ajudar na criação de dados para suportar estas plataformas. Dado o interesse por parte dos responsáveis dos transportes públicos Mobilis da cidade de Leiria neste trabalho, também foi desenvolvida uma aplicação capaz de traduzir dados já existentes nos seus servidores para a plataforma online. Com isto, é possível criar uma solução capaz de ser aplicável em várias cidades sem um elevado esforço adicional e de ser acessível tanto num ambiente fixo (num computador de secretária), como num ambiente móvel (*smartphone* ou *tablet*).

1.3 Estrutura do documento

Em termos de estrutura, este relatório é dividido em 6 capítulos:

- Introdução;
- Estado da arte;
- Proposta de solução;
- Implementação;
- Avaliação e testes;
- Conclusão.

No primeiro capítulo, onde se encontra esta secção, é feito o enquadramento deste trabalho através da introdução do problema, motivação e objetivos, estrutura do relatório e metodologia de trabalho. No segundo capítulo, é feito o estudo e análise de várias soluções disponíveis em mercado, verificando que características estas oferecem assim como as suas limitações. No terceiro capítulo é apresentada a proposta de solução deste trabalho, onde são descritos os atores e sistemas envolvidos neste trabalho, os requisitos e arquitetura de sistema por fim a descrição da expansão dos transportes Mobilis da cidade Leiria, caso de teste deste trabalho. No quarto capítulo é descrito o processo de implementação das várias plataformas/aplicações utilizadas e que desafios estas apresentaram. No quinto capítulo são apresentados os vários testes reais com

utilizadores sobre a aplicação Mobilis Trip Planner. No ultimo capítulo é apresentada a conclusão e trabalho futuro desta solução.

1.4 Metodologia de desenvolvimento

Em termos de metodologia de desenvolvimento aplicada neste trabalho, apesar de se identificar mais com as metodologias ágeis, não foi usada nenhuma em específico. Inicialmente, foi feita uma discussão dos objetivos para este trabalho e que funcionalidades interessantes poderiam ser aplicadas nele. Posteriormente deu-se lugar ao desenvolvimento do trabalho, implementando as várias componentes necessárias enquanto iriam sendo feitas reuniões semanais ou a cada duas semanas (que poderiam ou não ser presenciais) onde era discutido o que tinha sido feito, o que estava a ser feito e o que iria ser feito no futuro. Em caso de ser preciso realizar um ajuste nas características/funcionalidades do trabalho, estas eram discutidas nas reuniões semanais e depois então aplicadas no trabalho. No final, foram realizados vários testes junto de possíveis utilizadores e posteriormente feitas várias correções às plataformas/aplicações de forma a responder às dificuldades e sugestões que estes tiveram durante os testes.

Capítulo 2

Estado da Arte

Neste capítulo é apresentado o estado da arte, onde é feito o estudo e análise das várias plataformas disponíveis em mercado e posteriormente feito um pequeno comparativo entre estas. Dada a necessidade de integrar diferentes plataformas neste trabalho, este capítulo foi dividido em várias secções de forma a agrupar as plataformas com as mesmas características. Na secção de estruturação e validação de dados dos transportes públicos, são apresentadas algumas estruturas de dados capazes de armazenar informação estática ou dinâmica acerca dos transportes públicos assim como ferramentas capazes de validar estes mesmos dados. Na secção de plataformas de difusão de informação acerca dos transportes públicos, são apresentadas várias plataformas capazes de interpretar dados estruturados da secção anterior e posteriormente difundi-los. Por último, na secção de clientes de informação acerca dos transportes públicos, são apresentadas várias plataformas capazes de apresentarem ao utilizador final informação acerca dos transportes públicos, podendo ou não utilizar as plataformas de difusão anteriormente apresentadas.

2.1 Estruturação e validação de dados dos transportes públicos

Nesta secção serão apresentadas várias estruturas de dados e ferramentas de validação de dados de transportes públicos em formato digital de forma a serem lidos posteriormente pelas plataformas de difusão de informação. Assim, vamos de seguida apresentar algumas soluções disponíveis no mercado e analisar as suas características.

2.1.1 General Transit Feed Specification

O GTFS¹, originalmente denominado de Google Transit Feed Specification, foi criado em 2006 por Chris Harrelson, colaborador da Google, em conjunto com Tim e Bibiana McHugh, ambos *IT managers* da companhia de transportes Trimet da cidade de Portland, surgindo da necessidade de adaptar os dados dos transportes públicos em formato CSV com o sistema de localização geoespacial do Google Maps [12][antrim2013many]. Em termos estruturais, o GTFS é constituído por 6 ficheiros essenciais:

- `agency.txt` - possui dados acerca da companhia de transportes, como nome e URL do *Website* da companhia;
- `calendar.txt` - possui dados acerca da regularidade dos serviços. Indica por exemplo, que um dado serviço de transportes funciona todos os dias da semana (segunda a sexta) entre de 1 de Janeiro de 2016 até 31 de Julho de 2016;
- `routes.txt` - possui os dados acerca das linhas, como o nome curto e nome;
- `stops.txt` - possui os dados acerca das paragens, como o nome da paragem e sua posição geoespacial (coordenadas GPS);
- `stop_times.txt` - possui os dados dos horários, como a hora de chegada e partida e nome da paragem a que se refere;
- `trips.txt` - tem como objetivo ligar os horários com as linhas e serviços. Interliga os ficheiros `stop_times.txt`, `calendar.txt` e `routes.txt`.

Também é importante referir que existem outros ficheiros opcionais úteis, como:

- `calendar_dates.txt` - possui os dias de funcionamento extra ou de não-funcionamento dos serviços. Por exemplo, todos os feriados em que um dado serviço não funciona, deverão ser aqui registados. Para isso, é necessário indicar o serviço a que se refere, data e tipo de exceção (se é uma dia de funcionamento ou não);
- `fare_attributes` - caracteriza várias classes de pagamento, possuindo dados como a quantia a pagar e qual a moeda utilizada;
- `fare_rules.txt` - associa uma classe de pagamento a uma linha de transportes.

¹<https://developers.google.com/transit/gtfs/>

Estruturalmente, podemos relacionar todos estes ficheiros da seguinte forma:

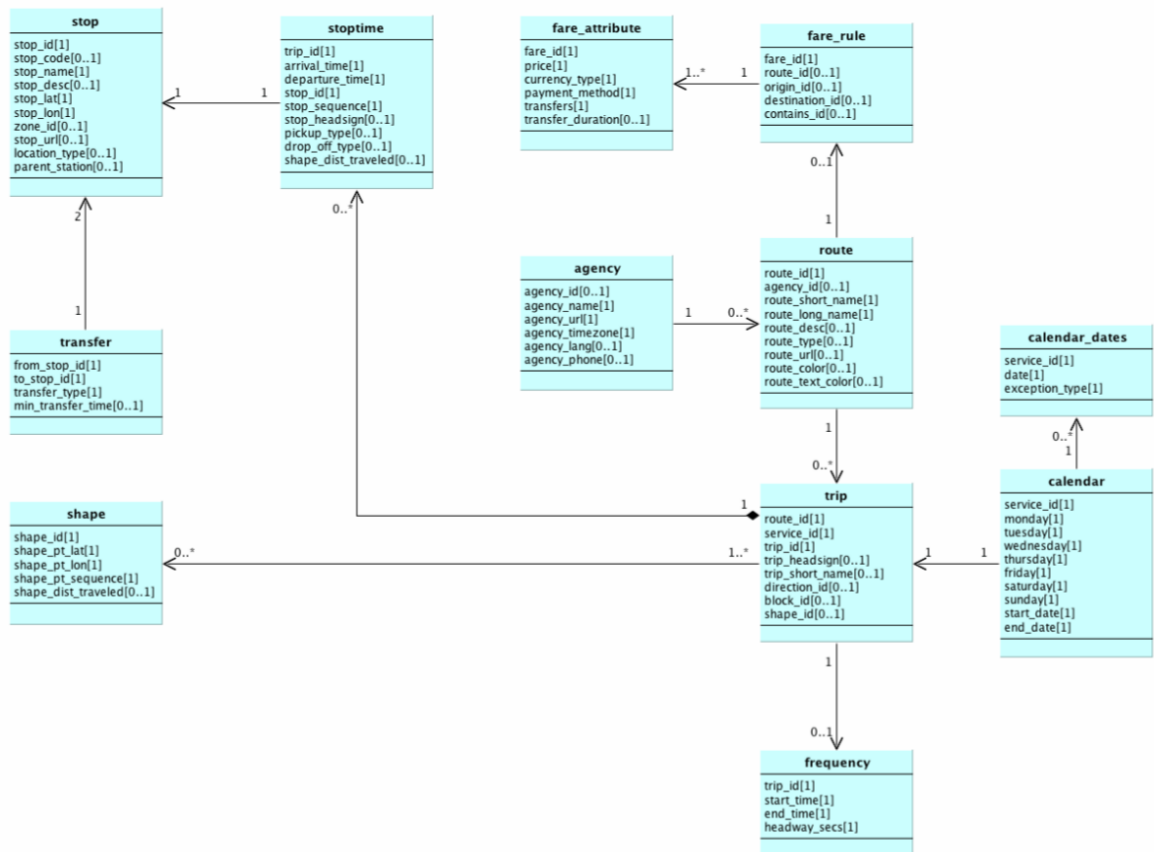


Figura 2.1: Imagem representativa da ligação entre os vários ficheiros GTFS [Adaptado de <https://developers.google.com/transit/images/gtfs-feed-diagram.png>]

Isto resulta numa estrutura de dados capaz de lidar com redes de transportes bastante complexas, existente em cidades como Portland ou Paris. No entanto, o GTFS é caracterizado por possuir uma curva de aprendizagem inicial algo acentuada visto este não possuir qualquer ferramenta de auxílio à criação/atualização dos vários ficheiros e o desenvolvimento dos horários (stop_times.txt) e das viagens (trips.txt) poderia ser simplificado, podendo por exemplo, definir a que horas um conjunto de paragens são repetidas. Ao mesmo tempo, como se pode verificar pela sua estrutura da figura 2.1, o GTFS possui apenas dados estáticos (linhas, paragens entre outros). Para criar uma estrutura com dados dinâmicos como a localização dos vários veículos, é necessário utilizar outras soluções, como a apresentada de seguida.

2.1.2 GTFS-Realtime

O GTFS-Realtime², lançado em 2011 pela Google, consiste numa estrutura de dados com o objetivo de difundir dados dinâmicos acerca dos transportes públicos e assim complementar o GTFS base. Através deste formato, é possível disponibilizar a várias plataformas informação como a localização dos transportes, alerta de eventos e estado de uma viagem (informar por exemplo, se está atrasada). No entanto, em comparação com o GTFS, o GTFS-Realtime é mais difícil e complexo em termos de criação, atualização e partilha de informação, por motivos como:

- Necessidade de constante disponibilização na Internet;
- Complexidade e dificuldade em termos de criação e atualização dos dados;
- Necessidade de dispositivos que disponibilizem as coordenadas geoespaciais ao GTFS-Realtime.

Ainda que o problema de constante disponibilização na Internet possa ser facilmente resolvido pela companhia de transportes ou cidade ao contratar serviços de disponibilização dos dados num servidor Web, as questões de dificuldade e complexidade na criação e atualização dos dados mantêm-se, devido à necessidade de preencher e serializar para um ficheiro a estrutura de dados do GTFS-Realtime no formato "Protocol Buffer Message". Ao mesmo tempo, este ficheiro precisa de ser disponibilizado e atualizado regularmente na Internet, logo, também é necessário desenvolver uma rotina para o caso [12]. Por estas razões, as companhias de transportes ou cidade têm dificuldades em lidar com estes desafios, visto não possuírem pessoas especializadas para o caso [12]. Ao mesmo tempo, para além deste trabalho de desenvolvimento, todos os veículos precisarão de estar equipados com um dispositivo que forneça ao sistema as suas coordenadas geoespaciais.

Apesar desta dificuldade e complexidade extra necessária para gerar as mensagens em Protocol Buffer Message, isto permite gerar mensagens mais leves e fáceis de transmitir através da Internet em relação a estruturas como o XML [12].

²<https://developers.google.com/transit/gtfs-realtime/>

2.1.3 FeedValidator

O `feedvalidator`³ consiste numa pequena ferramenta desenvolvida pela Google para a verificação de *feeds* GTFS por erros e warnings presentes nos vários ficheiros e que encaixa num pacote de aplicações chamado "transitfeed"⁴, com o objetivo de fornecer várias ferramentas capazes de ajudar no desenvolvimento de ficheiros GTFS.

Apesar de serem disponibilizadas várias ferramentas relacionadas com GTFS no `transitfeed`, esta é particularmente interessante para este projeto visto a capacidade de gerar um relatório com quase todos os erros e *warnings* presentes nos ficheiros GTFS e assim facilitar a tarefa de correção destes. Para fazer no entanto a verificação destes ficheiros, será necessário executar a aplicação através de uma linha de comandos, apesar de serem disponibilizadas várias formas de verificar os dados e com diferentes níveis de detalhe.

O `Feedvalidator` é compatível com sistemas operativos Windows, Linux e Mac (estes dois últimos através da execução do *script* python⁵)

2.2 Plataformas de difusão de informação de transportes públicos

Nesta secção são apresentadas as plataformas de difusão de informação, que têm como principal objetivo difundir os dados acerca dos transportes públicos para posteriormente serem interpretados, organizados e apresentados pelos clientes de informação aos seus utilizadores. Tendo em vista estes objetivos, vamos de seguida apresentar várias soluções disponíveis no mercado e analisar as suas características.

2.2.1 Retrieving real-time information: Lisbon bus system

Neste trabalho desenvolvido por David Alves, Luis M. Martinez e José M. Viegas[1][13] é apresentada uma proposta de solução para a cidade de Lisboa com o objetivo de fornecer um sistema de planeamento de trajetos em tempo real para os transportes públicos

³<https://github.com/google/transitfeed/wiki/FeedValidator>

⁴<https://github.com/google/transitfeed>

⁵<https://www.python.org/>

de Lisboa. Para isso, os autores desta solução fizeram um o estudo inicial dos logs dos transportes da companhia de transportes Carris ⁶ durante quatro meses, onde através do processo de data mining destes foram capazes de analisar e classificar os tempos de viagem e velocidades praticadas.[1][13] Ao mesmo tempo, visto a necessidade de o sistema evoluir os seus dados ao longo do tempo, os dados vindos dos vários dispositivos GPS localizados nos autocarros são utilizados no planeamento de rota feito por um utilizador e também armazenados num *data-center* para futuras previsões, como é demonstrado na figura 2.2. Através destas características, é possível ao sistema definir um tempo espera de chegada de um autocarro a uma paragem com bastante precisão. Em termos de requisitos desta solução, é necessário que os transportes possuam dispositivos capazes de obter coordenadas GPS, para posteriormente serem enviados para um *data-center* capaz de armazenar e calcular os tempos de previsão de chegada dos transportes, utilizando para isso o algoritmo Dijkstra ⁷. Depois de gerados os tempos previstos para os transportes, estes são enviados para um planeador de trajetos e então fornecidos ao utilizador dos transportes[1][13].

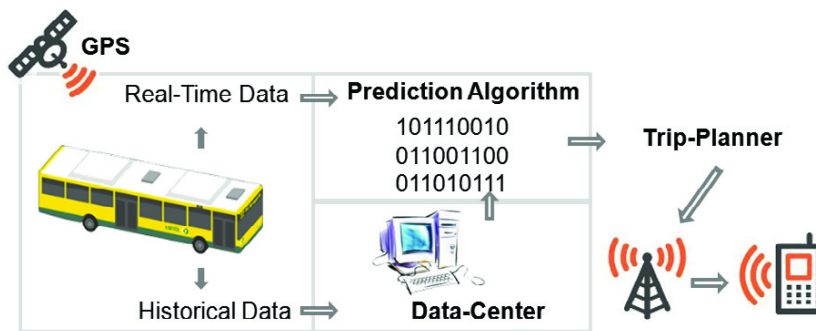


Figura 2.2: Arquitetura do Lisboa Bus System [1]

No entanto, para além de não se conseguir encontrar online qualquer implementação deste sistema *online* de forma a ser testado, a adaptação desta solução neste trabalho seria bastante difícil, porque obrigaria desde logo a que todos os autocarros de uma dada companhia de transportes possuíssem sistema de localização GPS e existissem *logs* de vários meses com a informação inicial. Ao mesmo tempo, sempre que houvesse a alteração/expansão dos serviços de uma dada companhia dos transportes, teria de ser feito novamente este estudo inicial para depois serem capaz de gerar previsões com grande precisão.

⁶<http://www.carris.pt/>

⁷https://en.wikipedia.org/wiki/Dijkstra%20s_algorithm

2.2.2 OpenTripPlanner

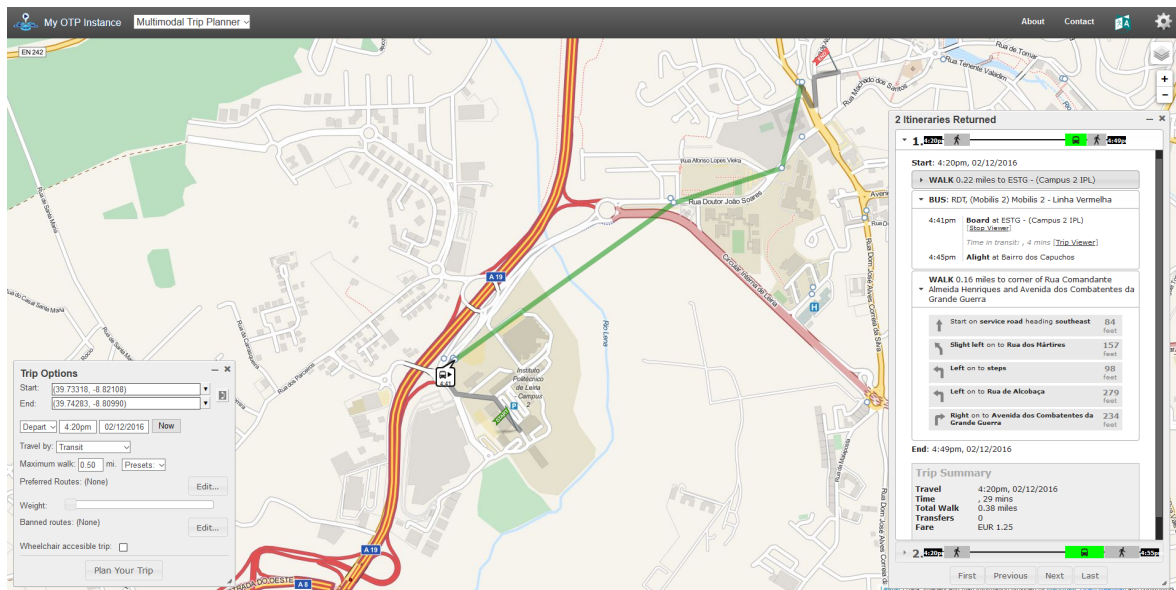


Figura 2.3: Descrição de um plano de trajeto no OpenTripPlanner

O OpenTripPlanner ⁸ (Figura 2.11) consiste numa plataforma *open-source* para planeamento de rotas multi-modais lançado em 2009, originalmente suportada pela agência Oregon's Trimet ⁹. Como plataforma de difusão de dados acerca dos transportes público, o OpenTripPlanner disponibiliza funcionalidades como:

- Planeador de trajetos, que nos permite definir um trajeto entre um local de origem e destino, bem como sugerir alternativas, utilizando transportes públicos;
- Horários das paragens e rotas dos transportes públicos;
- Informação em tempo real acerca dos transportes através de GTFS-Realtime¹⁰.

O OpenTripPlanner fornece estas funcionalidades com uma plataforma acessível através de REST API (podendo ser usado por aplicações terceiros) e num cliente Web que pode ser visitado através de um simples Web Browser. De forma a alimentar o servidor com dados acerca dos transportes, são utilizados dados em formato GTFS (para informação estática, como os horários) e GTFS-Realtime (para informação dinâmica, como alertas e localização dos transportes)[2]. De forma a representar toda esta informação num mapa, o OpenTripPlanner faz uso do OpenStreetMap¹¹, colocando pos-

⁸Disponível em: <http://www.opentripplanner.org/>

⁹<http://trimet.org/>

¹⁰<https://developers.google.com/transit/gtfs-realtime/>

¹¹disponível em: <https://www.openstreetmap.org/>

teriormente sobre este a localização das paragens e rotas resultantes do planeamento de um trajeto. Ao mesmo tempo, para além do OpenTripPlanner disponibilizar uma versão base pronta a executar, também é possível desenvolver uma versão personalizada ao obtermos o projeto através da página GitHub. No entanto, o OpenTripPlanner possui uma interface de cliente Web pouco clara em termos de utilização, trazendo dificuldades em aceder a funcionalidades como os horários de cada paragem, aos planos alternativos para uma rota, ou mesmo à mudança de linguagem. Ao mesmo tempo, a interface do cliente Web num dispositivo móvel, dado o pequeno tamanho de ecrã, torna ainda mais difícil a sua utilização. O OpenTripPlanner encontra-se disponível para as plataformas Windows, Linux e Mac.

2.2.3 Navitia.io

O Navitia .io ¹² consiste numa plataforma online com dois objetivos:

- Ser uma plataforma de difusão de dados acerca dos transportes públicos, onde os seus utilizadores podem submeter dados acerca dos transportes a nível mundial;
- Fornecer uma REST API utilizando a informação submetida pelos seus utilizadores, para posteriormente ser utilizada por aplicações terceiras;

Os conjuntos de dados fornecidos pelo Navitia.io são disponibilizados em formato GTFS ou NTFS, contendo informação como:

- Horários;
- Localização das paragens;
- Informação acerca das linhas e de construções;
- Ligações entre as paragens;

Apesar deste conjunto de funcionalidades ser bastante interessante para este projeto assim como a capacidade de disponibilizar nesta plataforma online a informação dos transportes públicos, o Navitia.io apenas permite 1 chamada por segundo à API e não possui qualquer aplicação cliente para aceder aos dados.

¹²Disponível em: <http://www.navitia.io/>

Convém no entanto referir que no decorrer deste trabalho, o Navitia.io alterou alguns dos seus serviços e deixou de ser possível a qualquer pessoa poder fornecer informação GTFS para ser usada online, visto a necessidade de autenticação e registo por parte de uma companhia de transportes. Em termos de limitações sobre a utilização da API, deixou de ser 1 chamada por segundo¹³ para um máximo de 90.000 chamadas por mês¹⁴.

2.2.4 OpenStreetMap

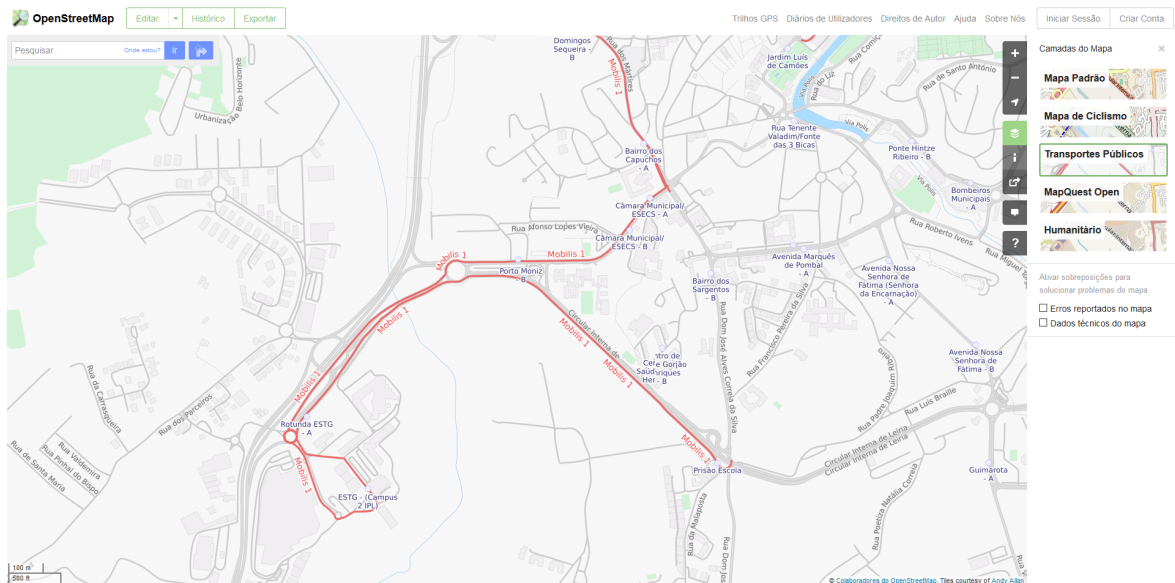


Figura 2.4: Percurso e paragens de uma linha de autocarro no OpenStreetMap

O OpenStreetMap ¹⁵ (Figura 2.3) consiste numa plataforma de criação e representação de mapas com o objetivo de criar e distribuir vária informação geográfica de forma livre, dado que grande parte dos mapas disponibilizados noutros serviços têm limitações em termos legais e técnicos, o que inviabiliza trabalhos mais criativos sobre os mapas. Para conseguir atingir estes objetivos, o OpenStreetMap é desenvolvido por uma comunidade constituída por vários mapeadores, profissionais das áreas geográficas e engenheiros que voluntariamente mapeiam vários locais (novos ou que foram alterados), inclusive áreas onde ocorreram grandes desastres recentemente¹⁶. Ao ser uma plataforma livre e desenvolvida pela comunidade, existem bastantes soluções de terceiros que tiram partido dele, como é o caso do OpenTripPlanner que faz uso dos mapas

¹³<https://github.com/SNCFdevelopers/API-trains-sncf/blob/master/source/index.rst>

¹⁴<http://www.navitia.io/pricing>

¹⁵Disponível em: <https://www.openstreetmap.org/>

¹⁶<https://www.openstreetmap.org/about>

para representação dos trajetos e paragens. Apesar de não ser uma plataforma dedicada à difusão de dados, o OpenStreetMap ainda assim é capaz de representar rotas dos transportes públicos assim como o local e nome das paragens, como se pode verificar na Figura 2.3. No entanto, visto ser uma plataforma desenvolvida principalmente pela comunidade, seria interessante fornecer mais informação acerca dos transportes públicos como horários e locais das paragens, por exemplo.

2.2.5 Google Transit e Google Maps

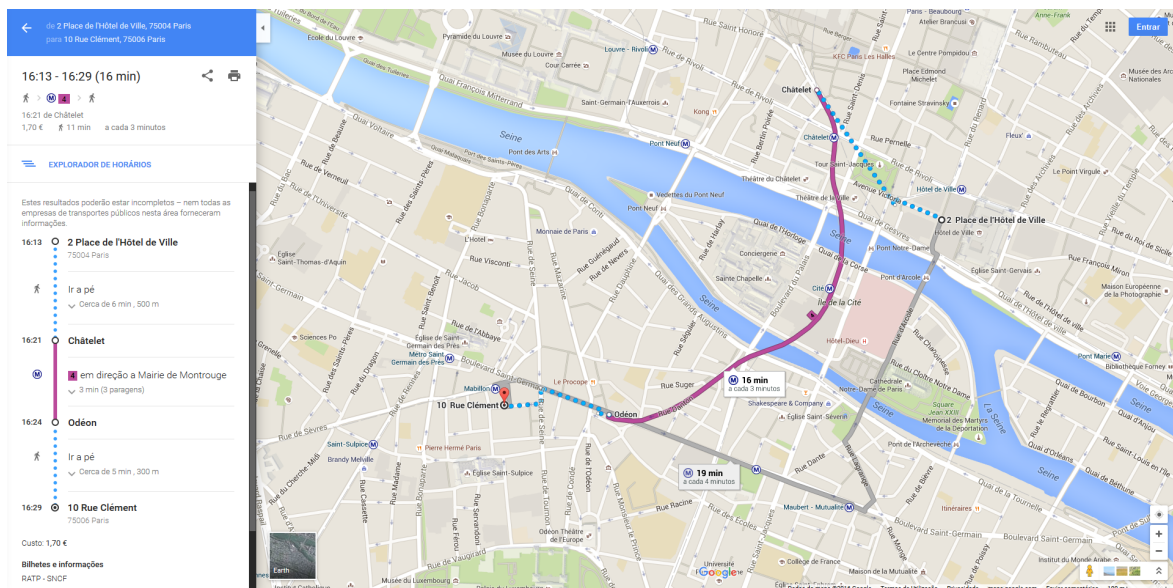


Figura 2.5: Descrição de um plano de trajeto no Google Maps

O Google Transit¹⁷ (Figura 2.13) é a plataforma que suporta o modo autocarro/comboio disponibilizado pelo Google Maps. Os dados dos transportes urbanos colocados no Google Transit, podem ser utilizados no planeamento de trajetos do Google Maps, facilitando a comparação com outros meios de transportes como o carro e assim obter a funcionalidade de planeamento de trajetos. Para disponibilizar a informação dos transportes urbanos, o Google Transit possui um programa onde as agências de transportes podem fornecer informação estática acerca dos seus transportes em formato GTFS. Após a submissão do ficheiro GTFS, este é avaliado e no caso de uma validação positiva, é implementado no Google Maps. Desde 2011 que o Google Transit também permite o fornecimento de informação dinâmica dos transportes urbanos em tempo real, sendo para isso necessário disponibilizar periodicamente os dados em formato GTFS-Realtime. Esta funcionalidade só está disponível às agências que já te-

¹⁷Disponível em: <https://maps.google.com/landing/transit/index.html>

nam informação estática disponibilizada no Google Maps. Com isto, é possível dotar o Google Maps com funcionalidades de planeamento de rotas a nível dos transportes públicos e oferecer outras capacidades como:

- Horários das paragens e rotas dos transportes públicos;
- Informação em tempo real acerca dos transportes através de GTFS-Realtime (como alertas);
- Pesquisa pelas paragens assim como pontos de interesse.
- Capacidade de ver fotos de vários locais numa cidade
- Capacidade de visitar uma cidade como se andássemos nas ruas (Google Street View)
- Não é necessário um servidor dedicado

No entanto, para submeter informação dos transportes públicos no Google Transit obriga ao registo da companhia de transportes, impedindo que pessoas terceiras possam submeter a informação. Posteriormente, a Google verifica se a informação está corretamente construída e válida para ser colocada online, logo, é necessário esperar algum tempo até que a informação seja disponibilizada online. O Google Transit/Maps está disponível em Windows, Linux e Mac.

2.2.6 OneStopTransport

O OpenStopTransport¹⁸ consiste numa plataforma desenvolvida por uma equipa Portuguesa com o objetivo de criar uma iniciativa em termos de *open data* no campo dos transportes urbanos¹⁹. O OneStopTransport disponibiliza uma REST API capaz de fornecer funcionalidades como:

- Planeador de trajetos, que nos permite definir um trajeto entre um local de origem e destino, bem como sugerir alternativas, utilizando transportes públicos
- Horários das paragens e rotas dos transportes públicos
- Informação em tempo real acerca dos transportes públicos através de GTFS-Realtime (como alertas)

¹⁸disponível em: <https://www.ost.pt/>

¹⁹<https://github.com/OneStopTransport/OneStopTransport/wiki>

Para além da API, o OneStopTransport disponibiliza uma plataforma online com várias aplicações disponíveis para dispositivos móveis, relacionadas com cidades e transportes. É através destas aplicações que o OneStopTransport também fornece informação, ainda que estática, acerca dos transportes públicos da cidade do Porto, Coimbra e dos comboios de Portugal entre outros[13]. Infelizmente, apesar das funcionalidades desta plataforma e API, ela deixou de ser desenvolvida em Janeiro de 2014, não está documentado como podemos utilizar a API com informação personalizada criada por terceiros e a plataforma online em si apenas fornece um local onde possamos partilhar uma aplicação para outros utilizadores utilizarem.

2.2.7 Resumo de funcionalidades

De forma a comparar mais facilmente as várias soluções disponíveis e assim tornar mais fácil a sua comparação, apresenta-se de seguida uma tabela com algumas das características mais importantes e diferenciadoras:

	OpenTripPlanner	Navitia.io	OpenStreetMap	Google Transit	OneStopTransport	Lisbon Bus System
Planeamento de rotas e horários	X	X	X	X	X	X
Transportes em tempo-real	X	X	X			X
Multimodal	X	X	X	X	X	
Aplicação cliente incluída	X			X		

Tabela 2.1: Características mais importantes das plataformas de difusão de informação acerca dos transportes públicos

Como é possível observar na tabela acima, existem bastantes funcionalidades em comum entre as várias plataformas apresentadas. A capacidade de planear trajetos e verificar os horários e linhas são características importantes para uma plataforma de difusão de dados, porque para além de serem algo comum entre as várias plataformas, são estas que originalmente organizam estes dados para depois enviá-los para os clientes de informação para serem apresentados. Ao mesmo tempo, ao fornecermos estes dados ao utilizador, podemos ajudá-lo a conhecer melhor os transportes públicos

disponíveis, sendo estes um dos objetivos deste trabalho. A característica de transportes em tempo-real consiste em difundir alertas e a localização dos veículos, sendo esta característica comum a várias plataformas permitindo simultaneamente reduzir os tempos de espera nas paragens. Por último, a funcionalidade de aplicação cliente também é bastante interessante visto permitir facilmente explorar na totalidade a plataforma implementada. Destas aplicações, todas exceto o OpenTripPlanner e o Google Maps²⁰ são assentes em API exclusivamente. O OneBusAway utiliza a API para enviar informação e como apresentado anteriormente, também possui uma aplicação Android mas com poucas funcionalidades em relação a outras soluções disponíveis. O Navitia.io inicialmente apresentou uma proposta bastante interessante ao ser possível enviar para a plataforma online dados acerca dos transportes em formato GTFS, que depois viria a ser acessível online. No entanto, devido à pequena dimensão da plataforma, era colocado uma limitação de no máximo 5 chamadas à API por segundo por cada utilizador. À alguns meses no entanto, o Navitia.io alterou os seus serviços, e já não é possível qualquer pessoa possa fornecer informação GTFS para ser usada online, visto a necessidade de autenticação e registo por parte de uma companhia de transportes. Em termos de limitações de sistema, passou-se de 5 chamadas por segundo para um máximo de 90.000 chamadas por mês²¹. O OneStopTransport também se apresentava como uma solução interessante, com uma plataforma online que oferecia variadas aplicações assentes na sua API. No entanto, houve a paragem no desenvolvimento das API e das aplicações, não está documentado como podemos utilizar a API com informação criada por terceiros e a plataforma online em si apenas fornece um local onde possamos partilhar uma aplicação para outros utilizadores utilizarem. Desta forma, a decisão caiu sobre duas soluções: OpenTripPlanner ou Google Transit. Em ambos os casos, as plataformas incluem aplicação cliente. No caso do OpenTripPlanner, junto com o servidor e no caso do Google Transit, é possível usar o Google Maps para aceder à informação disponibilizada. Ambas plataformas utilizam informação estática em formato GTFS e dinâmica em GTFS-Realtime. As duas plataformas possuem formas de aceder à informação através de Web Browser ou no caso de um dispositivo móvel, através de aplicações dedicadas, como a aplicação do Google Maps para o Google Transit e o OpenTripPlanner for Android ou Mobilis Trip Planner (disponível neste trabalho) para o OpenTripPlanner. No entanto, enquanto o OpenTripPlanner é implementado num servidor privado, o Google Transit é implementado na nuvem, o que necessita do upload dos dados para os servidores da Google para depois serem disponibilizados online. Apesar da vantagem do Google Transit em não necessitarmos de um servidor privado, é necessário que a informação seja disponibilizada pela companhia de transportes detentora dos dados. O Lisbon bus System apesar de ser altamente preciso nos

²⁰<https://www.google.com/maps/>

²¹<https://www.navitia.io/pricing>

seus cálculos de tempo de espera em tempo real, não se adapta à esta solução visto não ser multimodal, um dos requisitos deste projeto. De forma a não ser necessário depender das companhias de transportes para difundir os seus dados (um dos objetivos para este trabalho) e permitir que sejam outras entidades a partilhar os dados acerca dos transportes públicos desejados, foi decidido optar pelo OpenTripPlanner.

2.3 Clientes de informação de transportes públicos

Nesta secção vamos apresentar os clientes de informação de transportes públicos, tendo como principal objetivo organizar e apresentar informação acerca dos transportes públicos fazendo se possível uso dos dados disponibilizados pelas plataformas de difusão anteriormente apresentadas. Tendo em vista estes objetivos, vamos então de seguida apresentar várias soluções disponíveis no mercado e analisar as suas características.

2.3.1 Transitr

Neste trabalho desenvolvido por Jerald Jariyasunant, Daniel B. Work, Branko Kerkez, Raja Sengupta, Steven Glaser, e Alexandre Bayen tem como objetivo de descrever o desenvolvimento de um planeador de rotas para dispositivos móveis chamado Transitr[9][13]. Em termos de funcionamento, é previsto ser usado como outros clientes de informação, onde podemos adicionar um local de origem e destino e posteriormente é devolvido uma descrição do trajeto. Para conseguir esta rota, o local de origem e destino são enviados para um servidor encarregue de calcular o trajeto mais curto entre os dois pontos, utilizando para isso o algoritmo K-Shortest path baseado nos vários trajetos e dados em tempo real fornecidos pelos transportes equipados com dispositivos GPS. Depois de calculado a rota, esta é enviada à aplicação Transitr e depois apresentada. No entanto, o planeador de rotas do Transitr não permite pesquisar trajetos para outras datas ou horas nem definir quais meios de transporte utilizar para planear uma rota. O Transitr estaria disponível em dispositivos móveis iOS e em plataforma Web[9][13].

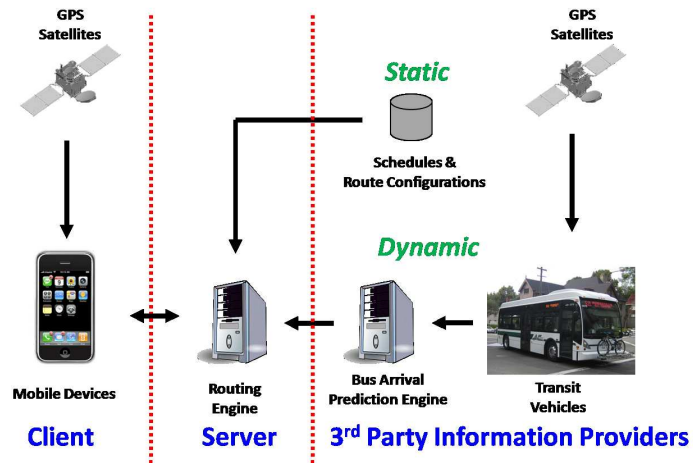
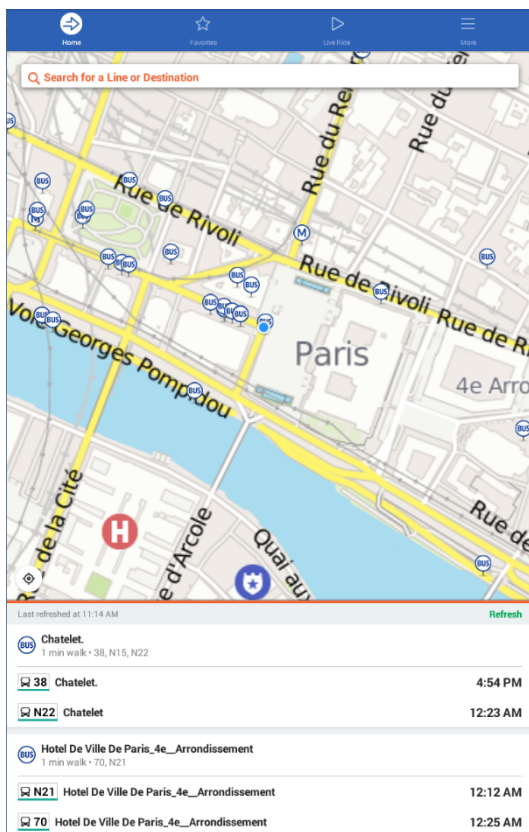
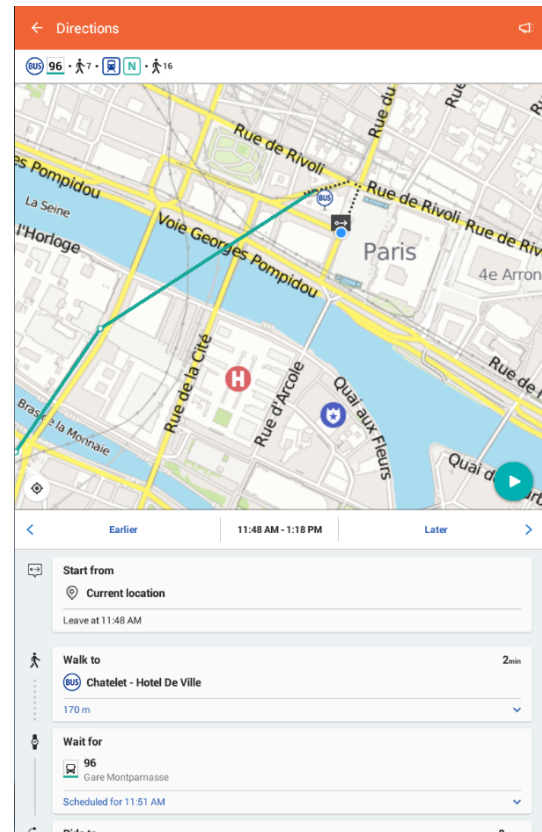


Figura 2.6: Arquitetura de sistema do Transitr[9][13]

2.3.2 Moov.it



(a) Ecrã principal do Moov.it



(b) Descrição de um plano de trajeto no Moov.it

Figura 2.7: Interface da aplicação Moov.it

O Moov.it ²² (Figura 2.5) apresenta-se desde logo como uma solução capaz de ser utilizada em várias cidades de vários países, disponibilizando uma aplicação com funcionalidades como:

- Planeador de rotas, que nos permite definir trajetos entre locais, bem como sugerir alternativas, utilizando transportes públicos
- Guia de rotas, que permite ao dispositivo móvel guiar-nos por um trajeto definido pelo planeador de rotas
- Horários de cada paragem e rotas que passam por elas
- Posição GPS dos autocarros e chegada destes às paragens
- Sistema de favoritos
- Verificação do estado dos serviços de transporte para cada linha
- Sistema de comunidade que permite partilhar informações acerca das rotas, indicando acidentes, atraso e mudanças de plataformas

O Moov.it também oferece a possibilidade das companhias de transportes disponibilizarem informação acerca dos seus transportes e serviços, sendo para isso necessário registar a companhia no website oficial do Moov.it. Apesar de ser bastante completo, o Moov.it poderia utilizar uma interface com um aspeto mais moderno. O Moov.it encontra-se disponível para as plataformas Android, iOS e Web Browser.

²²disponível em: <http://moovitapp.com/>

2.3.3 One Bus Away

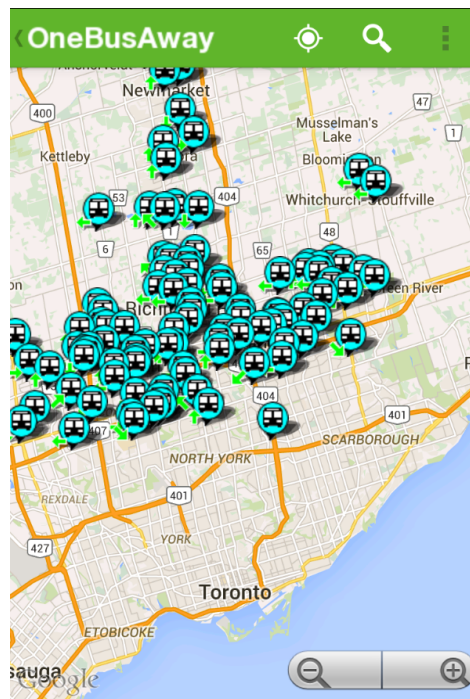


Figura 2.8: Página principal do OneBusAway

O One Bus Away ²³, cujo o interface se apresenta na Figura 2.2, é uma plataforma *open source* com informação em tempo real sobre os transportes de algumas cidades americanas. Para suportar esta plataforma, foram desenvolvidas várias API, promovendo assim o desenvolvimento de aplicações terceiras. O One Bus Away resulta de um projeto conjunto de várias agências públicas, companhias comerciais e investigadores académicos, com o objetivo de desenvolver uma plataforma robusta e open-source, com usabilidade para as agências de transportes e capaz de suportar períodos de desenvolvimento a curto e longo prazo²⁴.

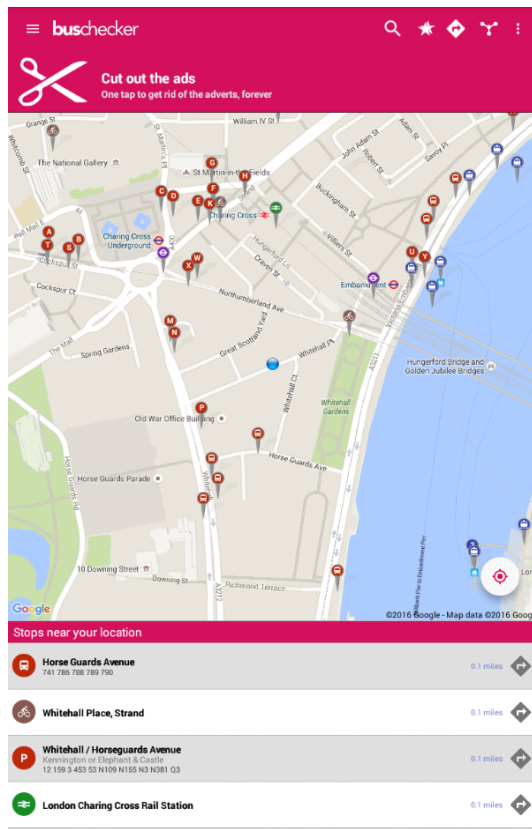
O OneBusAway para Android é uma aplicação capaz de fornecer os horários das paragens dos autocarros, das rotas percorridas, um sistema de favoritos onde podemos adicionar as nossas paragens favoritas, um sistema de notificação dos problemas com as rotas disponibilizadas e da chegada de um autocarro a uma paragem definida pelo utilizador. Recentemente, o OneBusAway foi atualizado com uma interface mais moderna e ao nível de outras aplicações aqui apresentadas e apesar de já possuir um planeador de rotas (algo que não possuía anteriormente), ainda está numa versão beta e não permite planejar um trajeto através do mapa da aplicação. O OneBusAway su-

²³disponível em: <http://onebusaway.org/>

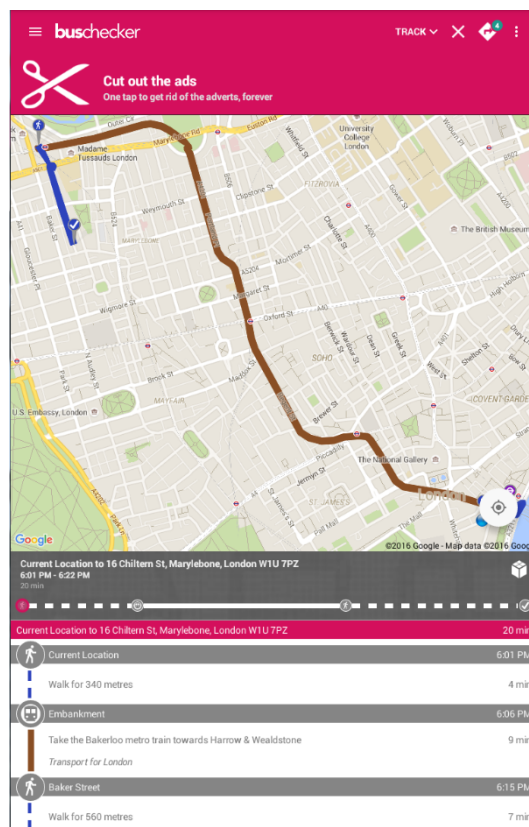
²⁴<http://onebusaway.org/the-onebusaway-project/vision/>

porta apenas algumas cidades americanas como Atlanta, Puget Sound, Tampa, York, San Diego, Washington D.C e Rogue Valley. O OneBusAway encontra-se disponível para as plataformas Android e iOS.

2.3.4 Bus Checker



(a) Écran principal do UK Bus Checker



(b) Descrição de um plano de trajeto no Uk Bus Checker

Figura 2.9: Interface da aplicação UK Bus Checker

O Bus Checker ²⁵ (Figura 2.7) é um conjunto de aplicações desenhadas para várias cidades como Londres e Chicago, mas também países, como o UK Bus Checker para o Reino Unido. Apresenta funcionalidades como:

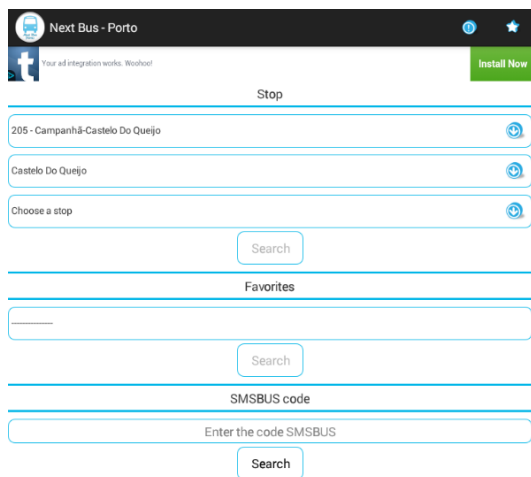
- Planeador de rotas;
- Guia de rotas;
- Horários de cada paragem e rotas;

²⁵disponível em: <http://www.buschecker.com/app/UK/>

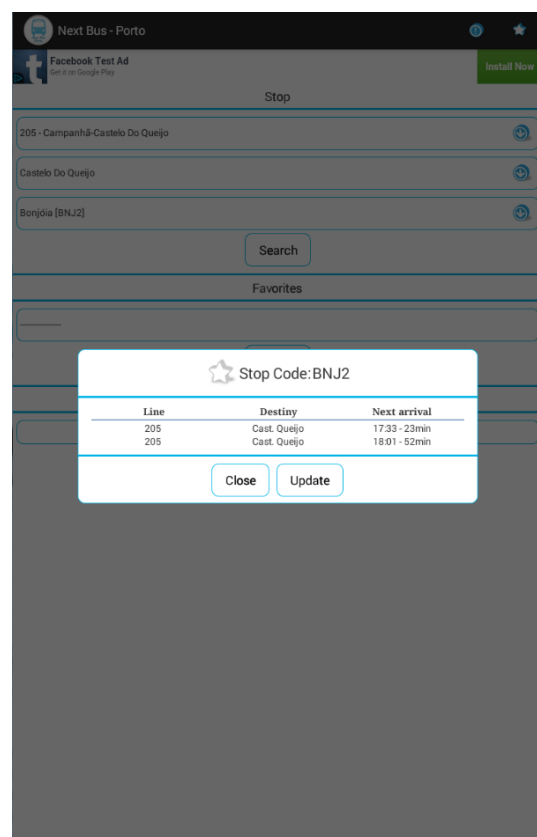
- Posição GPS dos autocarros;
- Sistema de favoritos;
- Verificação do estado dos serviços de transporte para cada linha.

Apesar de o Bus Checker possuir grande parte das funcionalidades de outros planeadores de rotas como o Moov.it e ao mesmo tempo, utilizar um interface único e intuitivo, ele possui informação de poucas cidades a nível mundial e obriga a instalar uma nova aplicação quando nos mudamos para outra cidade. O Bus Checker encontra-se disponível para as plataformas Android, iOS, Windows Phone, Kindle Fire e Windows PC/Tablet.

2.3.5 Next Bus - Porto



(a) Écran principal do NextBus - Porto

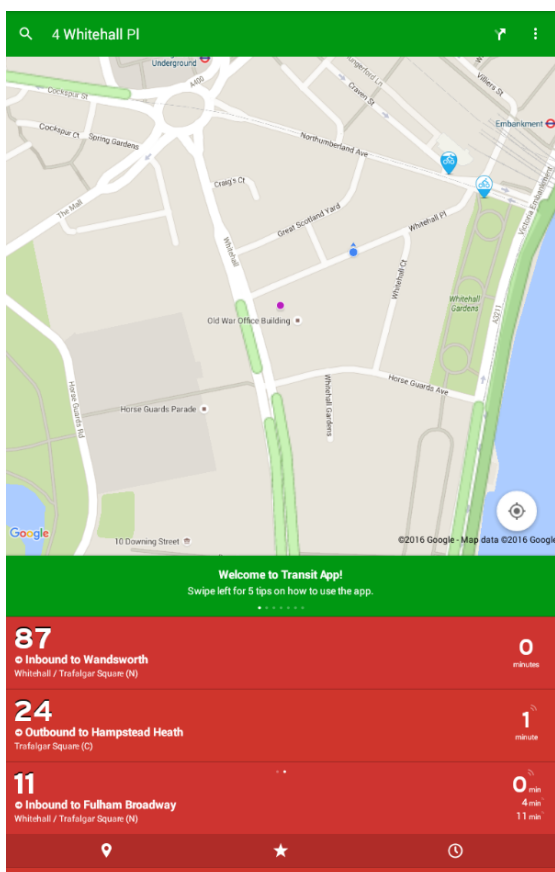


(b) Página com informação acerca de uma paragem

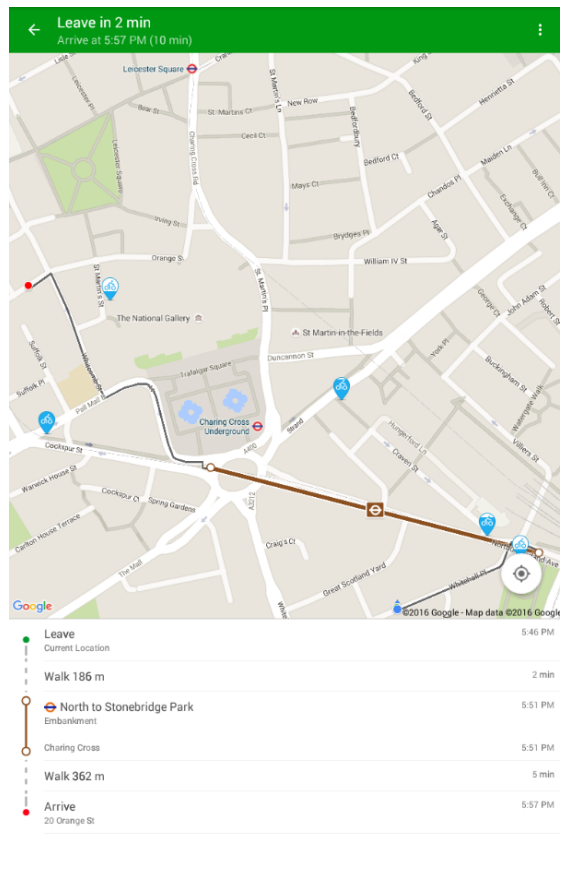
Figura 2.10: Interface da aplicação Next Bus - Porto

Na Figura 2.8 apresenta-se um exemplo de interesse do Next Bus - Porto²⁶, uma aplicação desenvolvida em Android para a cidade do Porto, permitindo saber os horários em cada uma das paragens, usando a seleção de linhas e direções para mostrar informação de determinada paragem. Esta aplicação possui também a funcionalidade única de através da inserção de um código disponibilizado no próprio local, podermos obter informação sobre determinada paragem. No entanto, o Next Bus - Porto possui bastantes funcionalidades em falta como o planeador de rotas, estado dos transportes e um guia de rotas. O Next Bus - Porto encontra-se disponível apenas em Android.

2.3.6 TransitApp



(a) Écran principal do TransitApp



(b) Descrição de um plano de trajeto no TransitApp

Figura 2.11: Interface da aplicação TransitApp

O Transit App²⁷, apresentado na Figura 2.9, é uma aplicação desenvolvida para várias cidades de vários países, capaz de funcionalidades como:

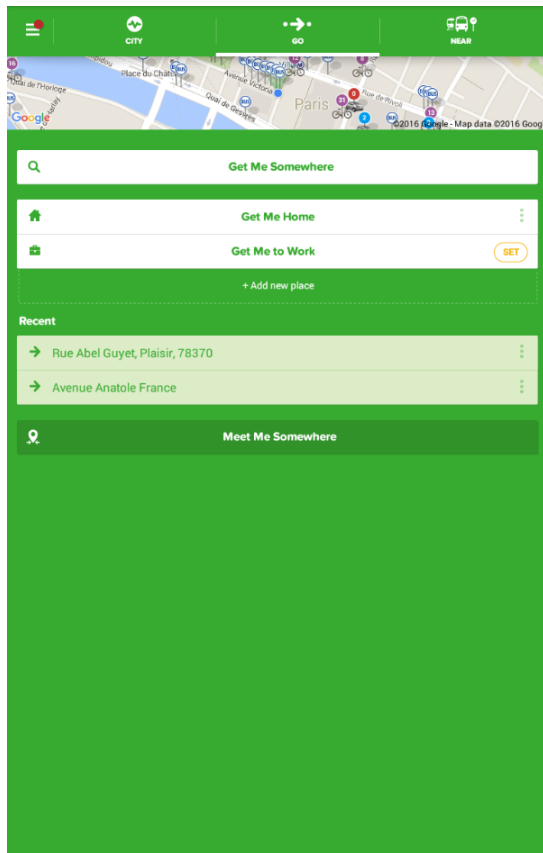
²⁶disponível em: https://play.google.com/store/apps/details?id=pt.pgsoft.nextbus_porto

²⁷disponível em: <http://transitapp.com/>

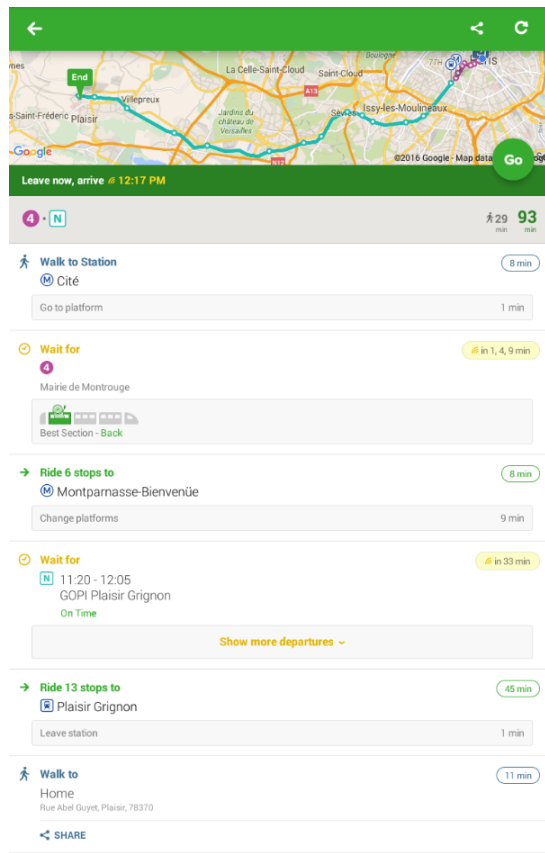
- Planeamento de rotas;
- Guia de rotas;
- Horários de cada paragem e rotas que passam por elas;
- Sistema de favoritos;
- Posição GPS dos autocarros e chegada destes às paragens;

Apesar de outras aplicações também possuírem estas funcionalidades, o TransitApp diferencia-se ao fornecer no ecrã principal as paragens mais próximas com transportes prestes a chegar e também um mini-tutorial com 5 dicas para começarmos a utilizar a aplicação. No entanto, apesar de bastante completo e comparável a aplicações como o Moov.it e Bus Checker, este não permite verificar os estados dos transportes públicos para uma dada companhia. O TransitApp encontra-se disponível para as plataformas Android e iOS.

2.3.7 CityMapper



(a) Écran principal do CityMapper



(b) Descrição de um plano de trajeto no CityMapper

Figura 2.12: Interface da aplicação CityMapper

O CityMapper ²⁸, apresentado na Figura 2.10, assim como outras aplicações anteriormente referidas, apresenta-se como uma aplicação para várias cidades e bastante completa, funcionando principalmente como um planeador de rotas e guia de rotas ligado com as redes sociais. O CityMapper possui funcionalidades como:

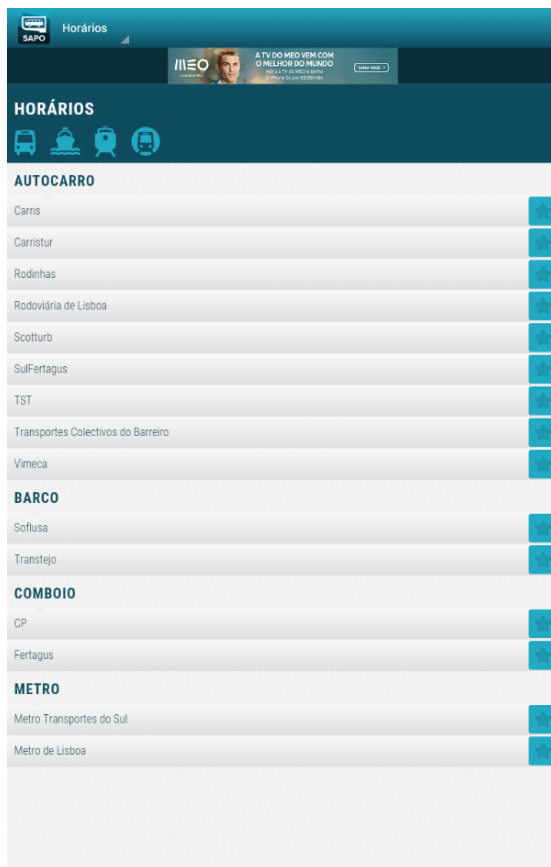
- Planeamento de rotas;
- Guia de trajetos;
- Horários de cada paragem e rotas que passam por elas;
- Sistema de favoritos;
- Posição GPS dos autocarros e chegada destes às paragens;

²⁸Disponível em: <https://citymapper.com>

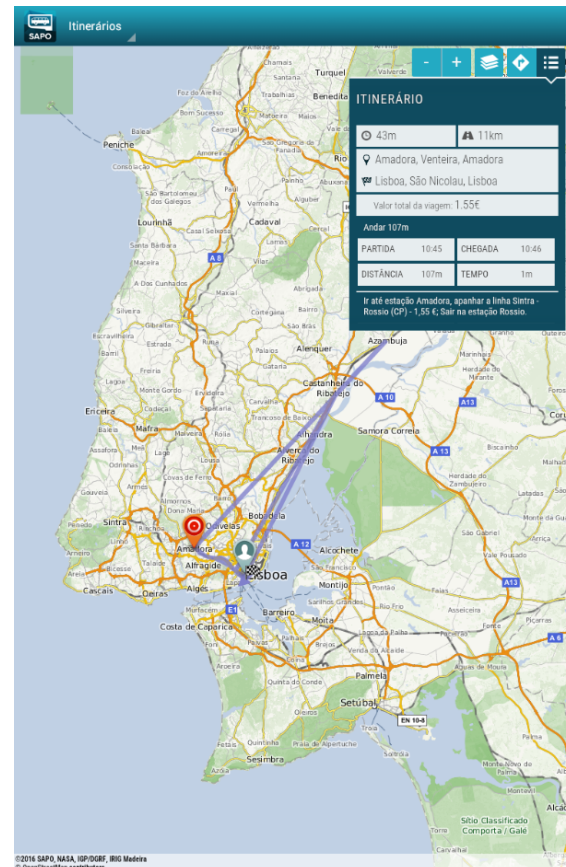
- Verificação do estado dos serviços de transporte para cada linha;

Para além destas funcionalidades, o CityMapper ainda apresenta a capacidade de apresentar uma estimativa de calorías gastas para possíveis trajetos feitos de bicicleta ou a pé e sugestão de locais interessantes para pontos de encontros com os amigos. No entanto, à alguns meses, foi alterado o aspeto da aplicação Web e sacrificou alguma da sua usabilidade e pode ser pouco intuitivo como fazer o planeamento de uma rota usando unicamente o mapa. O CityMapper está disponível para as plataformas Android, iOS e Web Browser.

2.3.8 Sapo Transportes



(a) Écran principal do Sapo Transportes



(b) Descrição de um plano de trajeto no Sapo Transportes

Figura 2.13: Interface da aplicação Sapo Transportes

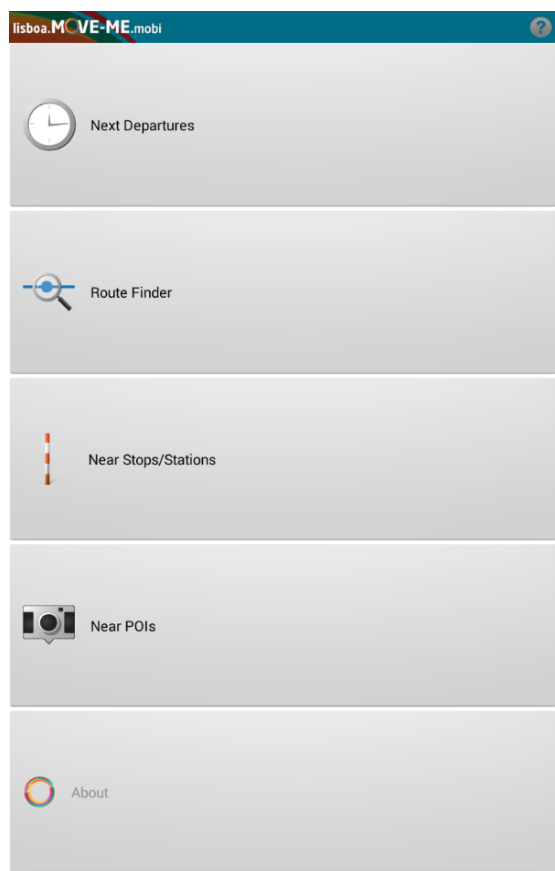
O interface do Sapo Transportes ²⁹ é apresentado na Figura 2.11, foi criado especificamente para a zona metropolitana de Lisboa, e disponibiliza informação acerca dos vários meios de transportes e empresas dessa zona, como autocarros, barcos, comboio e metro. Esta aplicação disponibiliza funcionalidades como:

- Planeamento de rotas;
- Horários de cada paragem e linhas;
- Sistema de favoritos;

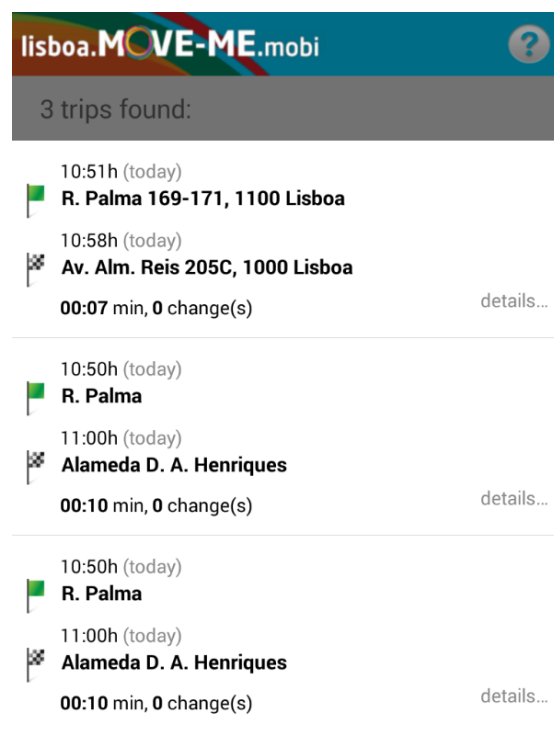
No entanto, a aplicação possui um planeador de rotas que não faz uso do mapa para seleccionar onde estamos ou para onde queremos ir e não possui informação dinâmica nem o estado dos transportes. O Sapo Transportes apenas encontra-se disponível em Android.

²⁹disponível em: <http://m.transportes.sapo.pt/>

2.3.9 Move-Me



(a) Écran principal do Move-Me



(b) Planos de trajetos alternativos do Move-Me

Figura 2.14: Interface da aplicação Move-Me

O Move-Me ³⁰ (Figura 2.12) existe em duas variantes: Move-Me Lisboa e Move-Me Porto. Ambas as versões são a mesma aplicação, com as mesmas funcionalidades, sendo uma dedicada a Lisboa e outra ao Porto. O Move-Me (em ambas as variantes) disponibiliza funcionalidades como a capacidade de verificar os horários das paragens através dos favoritos, um planeador de rotas, com pesquisa por nome ou seleção de agências e linhas para achar as paragens desejadas e a opção de adicionar uma rota para pontos de interesse através de um campo de texto, onde à medida que preenchemos, aparece uma lista de pontos de interesse (semelhante à pesquisa num sistema de GPS). Esta aplicação também fornece a possibilidade de procurar paragens e pontos de interesse num raio definido pelo utilizador. No entanto, o move-me possui um interface pouco moderno, não possui informação dinâmica e o planeamento de rotas é pouco intuitivo e descritivo. O Move-Me encontra-se disponível para as plataformas Android,

³⁰disponível em: <http://www.move-me.mobi/>

iOS e Web Browser.

2.3.10 OpenTripPlanner for Android

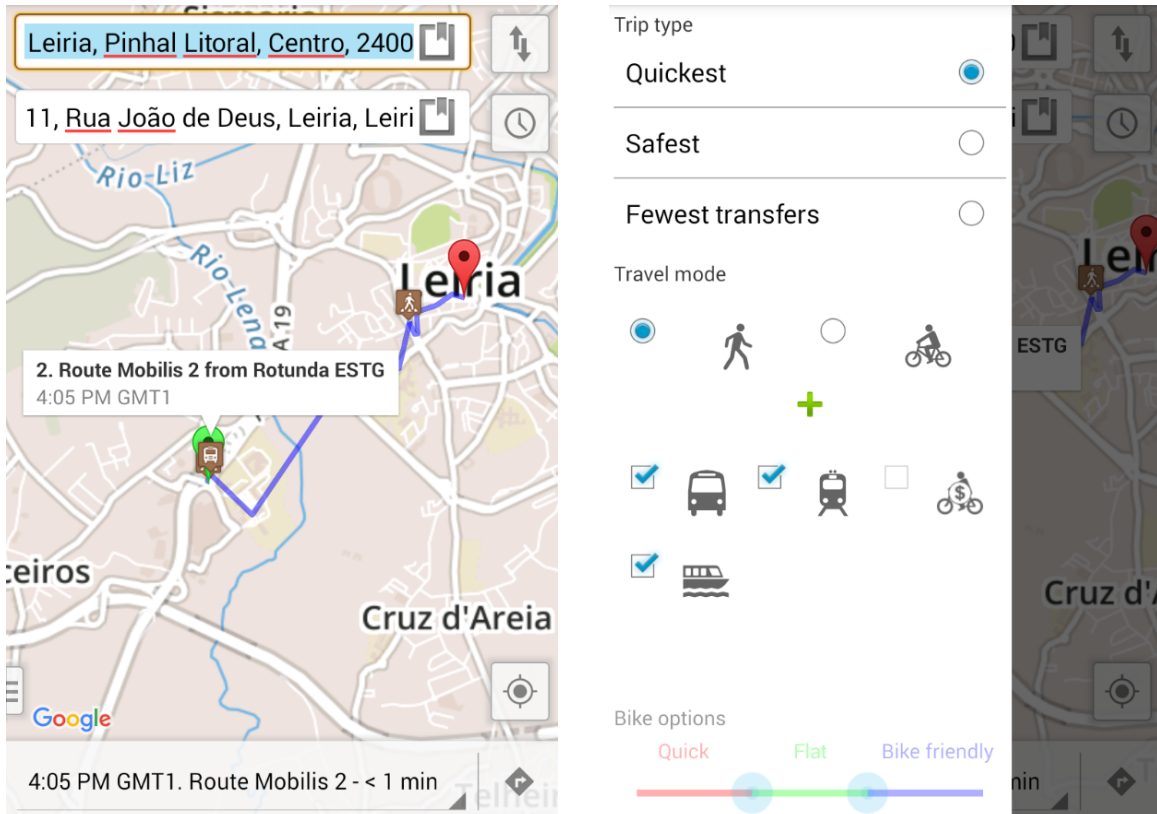


Figura 2.15: Interface da aplicação OpenTripPlanner for Android

O OpenTripPlanner for Android ³¹ tem a interface apresentada na Figura 2.13, é uma aplicação Android desenvolvida pelo Center for Urban Transportation Research da Universidade do Sul da Florida com o objetivo de fornecer uma nova interface à aplicação cliente da plataforma online OpenTripPlanner assim como a capacidade de utilizar GPS e outras características dos dispositivos móveis. Como outras aplicações apresentadas anteriormente, ele é multi-modal (modos apresentados na Figura 2.13-B), com a capacidade de alterar datas/horas e transportes que queremos usar nos planeamentos das rotas desejadas. Para o caso de acionarmos a opção de bicicleta como meio alternativo a andar a pé, são disponibilizados dois perfis pré-definidos como "viagem rápida" ou "viagem segura" e a possibilidade de usar um perfil personalizado onde podemos equili-

³¹disponível em: <https://github.com/CUTR-at-USF/OpenTripPlanner-for-Android/wiki>

brar a nossa viagem entre rápida, plana ou amigável. Infelizmente, o OpenTripPlanner for Android não disponibiliza forma de acedermos aos horários das paragens, rotas dos autocarros e informação em tempo real acerca dos transportes. No entanto, o OpenTripPlanner - Android apresenta algumas funcionalidades únicas como a capacidade de utilizar mapas de diferentes fornecedores (possui vários mapas do OpenStreetMap e Google Maps) e de se ligar a novos servidores não disponibilizados na lista pré-definida de stock, sendo para isso necessário fornecer o endereço do servidor OpenTripPlanner a que nos desejamos ligar. Isto permite, por exemplo, que um servidor OpenTripPlanner criado especificamente para os transportes duma pequena localidade, possa fornecer facilmente a sua informação e de forma adequada a um dispositivo móvel Android. O OpenTripPlanner - Android pode ser utilizado em várias cidades de todo o mundo. No entanto, esta aplicação não possui informação dinâmica, não permite mostrar os horários das paragens e possui um interface adaptado a antigos Android, não fazendo uso da Action Bar. O OpenTripPlanner for Android encontra-se disponível apenas para dispositivos Android.

2.3.11 StarMetro

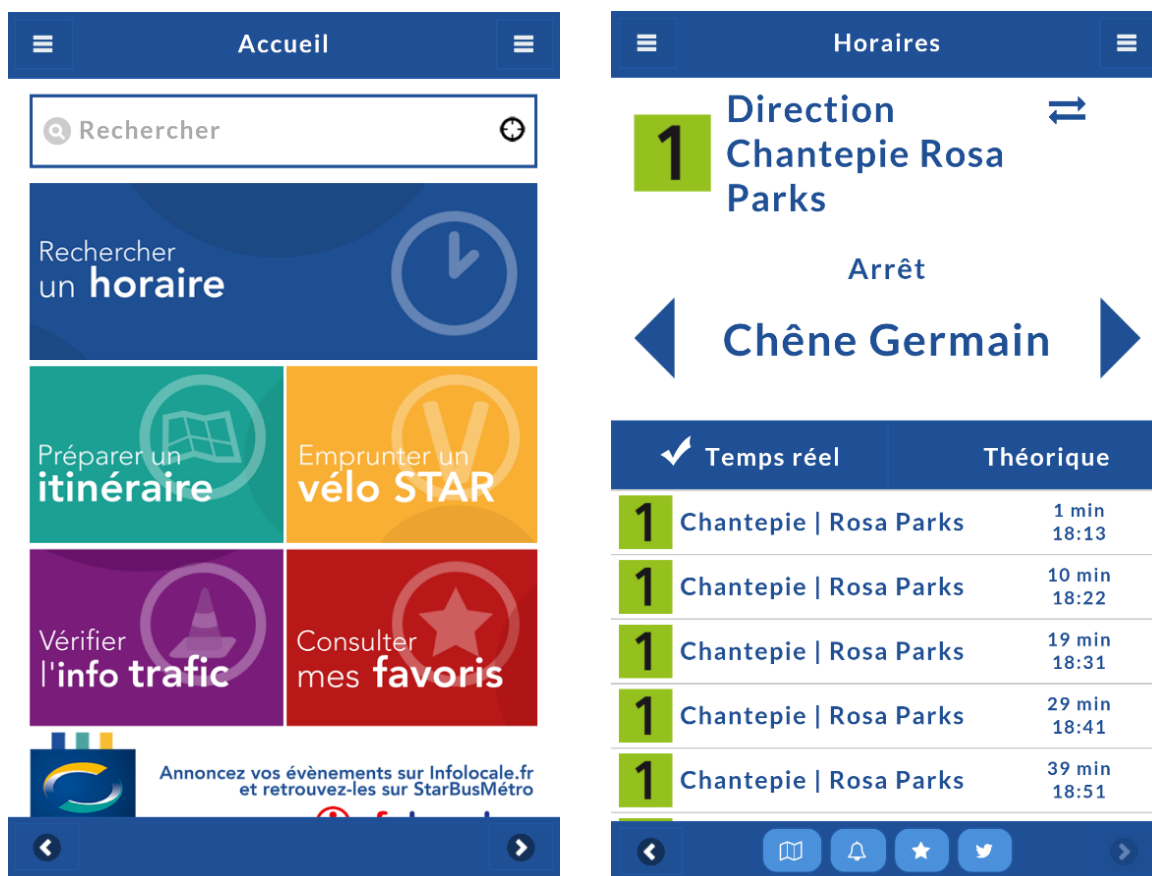


Figura 2.16: Interface da aplicação StarMetro

O Star Bus Metro ³² (Figura 2.10) consiste numa aplicação Android com o objetivo de fornecer informação acerca dos transportes Star da cidade de Rennes.

Esta aplicação possui funcionalidades como:

- Planeamento de rotas;
- Horários de cada paragem e rotas que passam por elas;
- Sistema de favoritos;
- Posição dos autocarros e chegada destes às paragens;
- Verificação do estado dos serviços de transporte para cada linha.

³²disponível em: <https://play.google.com/store/apps/details?id=com.bookbeo.starbusmetro>

O planeamento de trajetos para o Star Bus Metro é semelhante a algumas aplicações Android anteriormente apresentadas, sendo apenas possível fazer a pesquisa através de um endereço ou posição GPS, sendo esta pesquisa como nos sistemas de pesquisa de um GPS automóvel, em que ao final de algumas letras aparecem sugestões. Posteriormente, são apresentados vários trajetos alternativos, onde ao clicarmos num é possível ver uma descrição e um mapa do trajeto. Como noutras aplicações Android, esta aplicação também disponibiliza um sistema de favoritos e uma lista de estados dos transportes, capaz de apresentar um alerta caso uma linha esteja indisponível, assim como um texto descritivo do problema. Uma das funcionalidades diferenciadoras é a capacidade de mostrar os tempos reais estimados de chegada a uma paragem por um dado transporte assim como o tempo teórico previsto (como nos horários disponibilizados numa paragem) e em alguns casos mostrar se um autocarro já chegou a uma paragem ou se ainda está em deslocação. No entanto, apesar desta lista bastante comum de funcionalidades ao longo dos clientes de informação apresentados anteriormente, o Star Bus Metro oferece um interface único mas ligeiramente menos intuitivo. Por exemplo, em vez de selecionarmos uma paragem para verificarmos os seus horários, a sua procura é feita através da seleção de uma linha de transportes, sendo mostrado posteriormente o horário em pormenor para cada paragem, o que torna a pesquisa algo confusa. Ao mesmo tempo, para o planeamento de rotas, poderia fazer uma demonstração da rota num mapa em vez de ser apenas descritiva dos passos a fazer. Outro problema desta aplicação, é estar unicamente disponível em Francês, o que pode trazer problema a quem não conheça a língua. A aplicação Star Bus Metro encontra-se disponível para Android e iOS.

2.3.12 Resumo de funcionalidades

Dado o grande numero de plataformas de clientes de informação disponíveis e de forma a tornar mais fácil a sua comparação, apresenta-se de seguida uma tabela com algumas das características mais importantes e diferenciadores entre as aplicações:

	Moov.it	One Bus Away	Bus Checker	Next Bus - Porto	TransitApp	CityMapper	Sapo Transportes	Move-Me	OpenTripPlanner for Android	StarMetro	Transitr
Planeamento de Trajetos	X	X	X	X	X	X	X	X	X	X	X
Guia de Trajetos	X	X	X	X	X	X				X	
Paragens e horários	X	X	X	X	X	X	X	X		X	X
Transportes em tempo-real	X	X	X	X	X	X				X	X
Suporte a outros servidores									X		

Tabela 2.2: Características mais importantes dos clientes de informação acerca dos transportes públicos

Como é possível observar na tabela acima, existem algumas funcionalidades em comum entre as aplicações. As funcionalidades de planeamento de rota e guia de rota são bastante interessantes em conjunto para quando pretendemos efetuar um trajeto que não costumamos fazer, indicando as horas e transportes que devemos apanhar e ao mesmo tempo guiando-nos pelo planeamento até aos locais desejados. As paragens e horários também são bastante importantes visto podermos facilmente localizar as paragens com precisão e saber que autocarros passam por elas e a que horas, sem que tenhamos de nos deslocarmos até elas. A informação dos transportes em tempo real também é interessante para sabermos onde estão os veículos e se estão atrasados. O suporte a outros servidores consiste na possibilidade de nos ligarmos a um servidor com informação acerca dos transportes públicos desejados para além dos definidos de origem. Todas estas funcionalidades, para além de serem as mais comuns, também são as mais importantes para este trabalho, já que permitem dar a conhecer ao utilizador que transportes utilizar nos seus deslocamentos e ao mesmo tempo, reduzir os tempos de espera numa paragem. A última funcionalidade, apesar de ser utilizada apenas por uma aplicação, é muito importante no contexto deste trabalho, visto permitir que no caso de ser implementada uma plataforma de difusão de dados, esta tenha interface adequada a um dispositivo móvel Android. Tendo em conta estas características, é possível notar que o Moov.it, Bus Checker, TransitApp, CityMapper e Star Metro, desde que possuam a cidade que deseja, são todas elas excelentes soluções, apesar de não ser possível adicionarmos nova informação acerca de uma nova cidade sem que tenhamos de registar a companhia de transportes. Já outras aplicações como o OneBusAway, Next Bus - Porto e o Move-Me que possuem menos funcionalidades, são aplicações especializadas em algumas localidades e sem a possibilidade de podermos

adicionar outras cidades à aplicação. Já a aplicação transitr não pode ser tida como uma solução viável para este projeto visto não existir qualquer informação de como testar o seu planeador de rotas. A aplicação OpenTripPlanner for Android, apesar de ser das soluções com menos funcionalidades, possui a capacidade de ligar-se a um servidor OpenTripPlanner criado por qualquer pessoa. Isto possibilita que quando em conjunto com o OpenTripPlanner, formem uma plataforma de difusão e cliente de informação acerca dos transportes públicos, capaz de disponibilizar informação através de API, cliente Web e cliente Android. É graças a estas capacidades que o cliente de informação escolhido para este trabalho é o OpenTripPlanner for Android.

2.4 Conclusão

Com os transportes públicos a crescerem cada vez mais como uma alternativa aos automóveis, existe também a necessidade dos seus serviços serem alargados e melhorados.[11][3][6] Nesse sentido, muitas companhias de transportes públicos e cidades de grandes dimensões têm optado por difundir os seus transportes em várias plataformas online em vez de simples brochuras em papel ou online. Desta forma, para além de ser disponibilizada *online* informação acerca dos horários dos transportes, é possível saber onde são as paragens, quais as linhas que estão em funcionamento diariamente e até mesmo a posição geográfica dos transportes entre outras funcionalidades anteriormente apresentadas. Para além do desenvolvimento de várias aplicações móveis para este efeito, também têm sido criadas plataformas e API com o objetivo de fornecer informação acerca dos transportes a aplicações terceiras (entre elas, aplicações móveis). No entanto, apesar do crescimento na área, é possível notar que grande parte das plataformas de Difusão e clientes de informação são incompatíveis entre si. Ao mesmo tempo, grande parte das soluções que permitem colocar dados nas suas plataformas, obriga ao registo por parte das companhias de transportes, o que impossibilita a utilização por parte das cidades e outros interessados (a não ser, claro, que se façam passar por companhias de transportes e usem os dados destas). Ainda que hajam algumas soluções que possam permitir a implementação das suas plataformas em servidores privados, nenhuma possui forma de auxiliar as companhias de transportes, cidades ou outros interessados a desenvolver os dados que vão suportar estas plataformas. É neste contexto de incompatibilidades entre plataformas de difusão e clientes de informação, ausência de ajuda à criação dos dados dos transportes e falta de condições para aplicação destas plataformas em cidades, companhias de transportes ou outros interessados de pequenas e médias dimensões que surge a solução apresentada neste trabalho.

Capítulo 3

Proposta de Solução

Neste capítulo é apresentada a proposta de solução para este trabalho, com o objetivo auxiliar cidades pequenas e médias, companhias de transportes e outros interessados a desenvolverem inicialmente a sua informação acerca dos transportes públicos e posteriormente difundi-los numa plataforma online. Para isso, vão ser analisados que objetivos e requisitos são desejados para este trabalho, a arquitetura do sistema resultante e que decisões tiveram de ser tomadas em relação às suas características. Por fim, será apresentada um resumo e conclusão deste capítulo.

3.1 Objetivos da solução proposta

Como é possível verificar pelo estado da arte, para criar uma plataforma capaz de apresentar informação dos transportes públicos de forma livre e que não dependa das companhias de transporte não é simples, visto grande parte da disponibilidade da informação depender das companhias de transporte. Ao mesmo tempo, é desejável que esta plataforma esteja disponível em sistemas fixos e móveis.

Tendo em conta os objetivos definidos no capítulo de introdução, seria interessante esta plataforma possuir um planeador de rotas capaz de mostrar que transportes utilizar para chegar ao seu destino, a capacidade de visualizar os horários dos transportes e saber se estes estão atrasados ou a horas. Ao mesmo tempo, visto a necessidade de responder à atualização da informação dos transportes Mobilis da cidade de Leiria em 2016 e dada a inexistência de soluções para criação/atualização de informação em formato GTFS, também foi desenvolvido um processo que permita a criação de nova informação acerca dos transportes públicos de forma simplificada. Mais tarde, depois da apresentação deste trabalho junto da rodoviária do lis de forma a disponibilizar

esta solução aos utilizadores dos transportes públicos de Leiria, surgiu também a necessidade de uma aplicação que traduzi-se os dados que a rodoviária do lis já possui para um formato GTFS, de forma a reduzir a replicação de informação e a quantidades de erros que poderiam surgir com a utilização do Template Excel criado neste trabalho. Assim, através deste conjunto de plataformas e aplicações é possível apresentar uma solução que permite criar/atualizar informação acerca dos transportes públicos e posteriormente disponibilizar de forma organizada, simples e útil aos utilizadores finais.

3.2 Requisitos do Sistema

É através dos requisitos de sistema que definimos uma lista de funcionalidades desejadas assim como algumas limitações. Desta forma podemos obter uma linha de orientação para o que é respeitado e desejado ao longo do trabalho. Para isso, será apresentada uma tabela com os sistemas e atores deste trabalho seguida das tabelas de requisitos funcionais e não-funcionais. Com isto, é possível demonstrar quem utiliza cada um dos sistemas e que funcionalidades estes desejam.

3.2.1 Atores e sistemas envolvidos neste trabalho

De forma a definir que atores e sistemas estão envolvidos neste trabalho e com isto simplificar a leitura dos requisitos de sistema, é apresentada a seguinte tabela com a descrição de cada um deles:

Nome do ator/sistema	Descrição
Planeador de rotas	Plataforma responsável pela disponibilização de informação acerca dos transportes públicos em ambiente Web
Planeador Android	Aplicação responsável pela disponibilização de informação acerca dos transportes públicos em Android
Template	Ficheiro responsável pela criação/atualização de nova informação em formato GTFS
Servidor de informação dinâmica	Plataforma responsável pela difusão de informação dinâmica ao planeador de rotas
Tradutor GTFS Mobilis	Aplicação responsável pela tradução da informação disponibilizada pela rodoviária do lis para formato GTFS
Utilizador final	Pessoa que utiliza o planeador de rotas ou o planeador Android para se deslocar nos transportes públicos
Administrador	Pessoa responsável por gerir o planeador de rotas, planeador android, template e o tradutor GTFS Mobilis. Podem ser várias pessoas.

Tabela 3.1: Tabela de atores e sistemas

Como se pode verificar na tabela acima, cada um dos sistemas acima envolve uma componente desenvolvida neste trabalho e que em algumas delas, não faz sentido envolver um dado utilizador, como é o caso do Tradutor GTFS do Mobilis, já que consiste num sistema para criar horários em formato GTFS através de dados vindos dos servidores da rodoviária do lis. Ao mesmo tempo, através desta tabela é possível prever que componentes iram compor a arquitetura do sistema.

3.2.2 Lista de requisitos funcionais

De seguida apresentamos a lista de requisitos funcionais do sistema. É através desta que definimos que características são desejadas do ponto de vista dos utilizadores. Estes requisitos (funcionais e não-funcionais) foram definidas ao longo das várias reuniões tidas ao longo deste trabalho junto dos orientadores e também são resultantes da análise das várias aplicações do estado da arte e por necessidade de resposta a alguns acontecimentos durante este trabalho.

#	Descrição do requisito	Prioridade
RF1	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de definir uma origem e destino para planear uma rota	Alta
RF2	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de definir os transportes públicos que pretende usar	Alta
RF3	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de escolher várias rotas alternativas ao executar o planeamento de uma rota	Alta
RF4	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de definir a data e hora do planeamento da uma rota	Alta
RF5	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de descrever os passos resultantes do planeamento de uma rota	Alta
RF6	O planeador de rotas deverá mostrar ao utilizador final a localização das paragens	Alta
RF7	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de verificar os horários de uma paragem	Alta
RF8	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de verificar se um transporte está atrasado	Média
RF9	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de definir uma rota como favorita	Baixa
RF10	O planeador de rotas deverá disponibilizar ao utilizador final a capacidade de visualizar diferentes mapas da mesma zona	Baixa

Tabela 3.2: Requisitos funcionais do planeador de rotas do ponto de vista do utilizador

Na tabela acima são apresentados os vários requisitos do ponto de vista do utilizador final, onde é possível destacar a utilização de um planeador de rotas e opções associadas, ótimo para quem está a descobrir uma nova localidade e não conhece os transportes públicos disponíveis, a capacidade de ver o local das paragens e seus horários assim como a capacidade de saber se um veículo está atrasado. Ao mesmo tempo, apesar de achar-se menos relevante, também seria interessante o planeador de rotas suportar rotas favoritas, algo comum nas soluções apresentadas no estado da arte e a capacidade de visualizar diferentes mapas da mesma zona.

RF11	O planeador de rotas deverá disponibilizar ao administrador a capacidade de definir informação estática (horários, linhas, paragens...)	Alta
RF12	O planeador de rotas deverá disponibilizar ao administrador a capacidade de definir informação dinâmica (localização dos transportes, eventos, atrasos...).	Alta
RF13	O planeador de rotas deverá disponibilizar a capacidade de utilizar o mapa definido pelo administrador para mostrar informação	Baixo

Tabela 3.3: Requisitos funcionais do planeador de rotas do ponto de vista do administrador

Do ponto de vista do administrador de um planeador de rotas, seria interessante este poder obter informação acerca dos transportes públicos através de REST API e ao mesmo tempo, esta mesma informação poder ser acedida por soluções terceiras. Também é do interesse do administrador poder adicionar a sua informação (estática e/ou dinâmica) acerca dos transportes públicos sem que para isso precise de permissão da companhia de transportes. Por fim, a utilização de mapas personalizados pelo administrador, apesar de menos importante que as funcionalidades restantes, pode ser interessante visto este poder seleccionar que mapa mais se adapta à sua implementação.

#	Descrição do requisito	Prioridade
RF14	O planeador Android deverá disponibilizar ao utilizador final a capacidade de definir uma origem e destino para planear uma rota	Alta
RF15	O planeador Android deverá disponibilizar ao utilizador final a capacidade de definir os transportes públicos que pretende usar	Alta
RF16	O planeador Android deverá disponibilizar ao utilizador final a capacidade de escolher várias rotas alternativa ao executar o planeamento de uma rota	Alta
RF17	O planeador Android deverá disponibilizar ao utilizador final a capacidade de definir a data e hora de um planeamento da rota	Alta
RF18	O planeador Android deverá disponibilizar ao utilizador final a capacidade de descrever os passos resultantes do planeamento de uma rota	Alta
RF19	O planeador Android deverá mostrar ao utilizador final a localização das paragens	Alta
RF20	O planeador Android deverá disponibilizar ao utilizador final a capacidade de verificar os horários de uma paragem	Alta
RF21	O planeador Android deverá utilizar a informação estática disponibilizada pela plataforma OpenTripPlanner	Alta
RF22	O planeador Android deverá disponibilizar ao utilizador final a capacidade de verificar se um transporte está atrasado	Média
RF23	O planeador Android deverá disponibilizar ao utilizador final a capacidade de definir uma rota como favorita	Baixa
RF24	O planeador Android deverá disponibilizar ao utilizador final a capacidade de visualizar diferentes mapas da mesma zona	Baixa
RF25	O planeador Android deverá disponibilizar ao utilizador final a capacidade de marcar a sua posição atual usando o GPS	Baixa

Tabela 3.4: Requisitos funcionais do Planeador Android

Na tabela acima são apresentados os vários requisitos para a aplicação Android. Como é possível verificar, grande parte dos seus requisitos são semelhantes ao planeador de rotas com a exceção da utilização do GPS (visto ser uma característica inerente dos dispositivos móveis).

#	Descrição do requisito	Prioridade
RF26	O template deverá disponibilizar ao administrador a capacidade de criar o ficheiro <code>agency.txt</code> em formato GTFS	Alta
RF27	O template deverá disponibilizar ao administrador a capacidade de atualizar o ficheiro <code>agency.txt</code> em formato GTFS	Alta
RF28	O template deverá disponibilizar ao administrador a capacidade de criar o ficheiro <code>calendar.txt</code> em formato GTFS	Alta
RF29	O template deverá disponibilizar ao administrador a capacidade de atualizar o ficheiro <code>calendar.txt</code> em formato GTFS	Alta
RF30	O template deverá disponibilizar ao administrador a capacidade de criar o ficheiro <code>routes.txt</code> em formato GTFS	Alta
RF31	O template deverá disponibilizar ao administrador a capacidade de atualizar o ficheiro <code>routes.txt</code> em formato GTFS	Alta
RF32	O template deverá disponibilizar ao administrador a capacidade de criar o ficheiro <code>stops.txt</code> em formato GTFS	Alta
RF33	O template deverá disponibilizar ao administrador a capacidade de atualizar o ficheiro <code>stops.txt</code> em formato GTFS	Alta
RF34	O template deverá disponibilizar ao administrador a capacidade de criar o ficheiro <code>stop_times.txt</code> em formato GTFS	Alta
RF35	O template deverá disponibilizar ao administrador a capacidade de atualizar o ficheiro <code>stop_times.txt</code> em formato GTFS	Alta
RF36	O template deverá disponibilizar ao administrador a capacidade de criar o ficheiro <code>trips.txt</code> em formato GTFS	Alta
RF37	O template deverá disponibilizar ao administrador a capacidade de atualizar o ficheiro <code>trips.txt</code> em formato GTFS	Alta

Tabela 3.5: Requisitos funcionais do Template

Na tabela acima são apresentados os vários requisitos para o Template, havendo a opção para criar e atualizar cada um dos ficheiros GTFS normalmente utilizados neste trabalho.

RF38	O template deverá disponibilizar ao administrador a capacidade de criar um <code>calendar_dates.txt</code> em formato GTFS	Média
RF39	O template deverá disponibilizar ao administrador a capacidade de criar horários intermédios no <code>stop_times.txt</code> de forma simples	Média
RF40	O template deverá disponibilizar ao administrador a capacidade de atualizar um <code>calendar_dates.txt</code> em formato GTFS	Média
RF41	O template deverá disponibilizar ao administrador a capacidade de criar um <code>fare_attributes.txt</code> em formato GTFS	Baixo
RF42	O template deverá disponibilizar ao administrador a capacidade de atualizar um <code>fare_attributes.txt</code> em formato GTFS	Baixo

Tabela 3.6: Requisitos funcionais do Template para os ficheiros GTFS opcionais

Na tabela acima são apresentados os requisitos que definem a criação/atualização dos ficheiros opcionais GTFS utilizados neste trabalho pelo administrador, referente às tarifas de transporte e dos feriados. Também é possível notar a capacidade de calcular os horários intermédios para o caso de não serem disponibilizados os horários por paragem.

#	Descrição do requisito	Prioridade
RF43	O tradutor GTFS Mobilis deverá disponibilizar ao administrador a capacidade de selecionar os ficheiros necessários a traduzir para GTFS	Alta
RF44	O tradutor GTFS Mobilis deverá disponibilizar ao administrador a capacidade de definir uma pasta para colocar os dados resultantes	Alta
RF45	O tradutor GTFS Mobilis deverá disponibilizar ao administrador a capacidade de gerar automaticamente os ficheiros GTFS	Alta
RF46	O tradutor GTFS Mobilis deverá disponibilizar ao administrador a capacidade de definir os dados da companhia de transportes para gerar em GTFS	Média
RF47	O tradutor GTFS Mobilis deverá disponibilizar ao administrador a capacidade de definir o custo dos bilhetes básicos	Média
RF48	O tradutor GTFS Mobilis deverá disponibilizar ao administrador a capacidade de definir o ano a que os dados pertencem	Média

Tabela 3.7: Requisitos funcionais do Tradutor GTFS Mobilis

Na tabela acima são apresentados os vários requisitos para o tradutor GTFS do Mobilis de Leiria. É possível verificar características como a leitura dos ficheiros base e a escrita da tradução destes, tudo automaticamente. Ao mesmo tempo, visto a falta de informação estruturada acerca dos dados da companhia de transportes e o custo do bilhetes, foi necessário especificar a definição destes campos. Visto todos os anos serem lançados novos horários, também foi definido um requisito para isso.

#	Descrição do requisito	Prioridade
RF49	O servidor de informação dinâmica deverá disponibilizar ao administrador a capacidade de atualizar constantemente a localização dos transportes públicos	Alta
RF50	O servidor de informação dinâmica deverá disponibilizar ao administrador a capacidade de atualizar constantemente o estado de um evento relacionado com uma dada viagem	Alta
RF51	O servidor de informação dinâmica deverá disponibilizar ao administrador a capacidade de atualizar constantemente o estado de uma rota (se está atrasada ou a horas)	Alta

Tabela 3.8: Requisitos funcionais do Servidor de informação dinâmica

Na tabela acima são apresentados os vários requisitos para o servidor de informação dinâmica. Este servidor tem como objetivo fornecer o planeador de rotas com informação dinâmica como a localização dos transportes, estado de uma viagem e eventos/problemas em relação a uma rota. Para isso, é necessário que o servidor disponibilize forma de o administrador colocar a sua informação, daí os seus requisitos.

3.2.3 Lista de requisitos não-funcionais

Apresenta-se de seguida a lista de requisitos não-funcionais do sistema. É através destes que definimos que limitações/restrições precisam de ser respeitadas para que o sistema possa ser utilizável.

#	Descrição do requisito
RNF1	O planeador de rotas deverá ser suportado por vários sistemas operativos
RNF2	O planeador de rotas deverá ser multi-modal
RNF3	O planeador de rotas deverá possuir pontos de interesse
RNF4	A informação do planeador de rotas deverá estar em formato GTFS
RNF5	O planeador Android deverá suportar a versão Android 4.4 e superiores
RNF6	O planeador Android deverá ser multi-modal
RNF7	O planeador Android deverá possuir pontos de interesse
RNF8	O planeador Android deverá fazer uso dos paradigmas de interface do Android 4.4
RNF9	O planeador Android deverá suportar diferentes resoluções de écran
RNF10	O planeador Android deverá fazer uso da API do planeador de rotas para obter informação acerca dos transportes
RNF11	O template deverá simplificar o processo de preenchimento do GTFS
RNF12	O template deverá facilitar o processo de geração do GTFS
RNF13	O template deverá fazer os cálculos dos campos GTFS de forma automatizada
RNF14	O template deverá gerar pelo menos os ficheiros GTFS obrigatórios
RNF15	O tradutor GTFS Mobilis deverá gerar automaticamente os ficheiros GTFS
RNF16	O tradutor GTFS Mobilis deverá gerar os ficheiros obrigatórios do GTFS
RNF17	O servidor de informação dinâmica deverá utilizar GTFS-Realtime nos seus dados

Tabela 3.9: Requisitos não funcionais do sistema

Na tabela acima são apresentados os vários requisitos não funcionais. É possível verificar que o planeador de rotas e planeador Android possuem requisitos bastante semelhantes, visto ambos serem planeadores de rotas. No entanto, no planeador Android foi definido uma versão base de 4.4 porque de acordo com os dashboards Android¹, grande parte dos dispositivos móveis Android possuem o Android 4.4 e superiores. Ao mesmo tempo, os paradigmas de interface do Android 4.4 é bastante semelhante às

¹<https://developer.android.com/about/dashboards/index.html>

versões superiores. De forma a reduzir a carga sobre o dispositivo Android, foi também decidido que este deveria obter a sua informação através da API do planeador de rotas. Em termos do template, havia interesse em que os seus dados GTFS pudessem ser calculados automaticamente e de forma simplificada, visto a complexidade do GTFS. Ao mesmo tempo, também foi definido que o template deveria gerar pelo menos os ficheiros essenciais. Em relação ao tradutor GTFS Mobilis, à semelhança do Template, foi definido que os ficheiros GTFS deveriam ser gerados automaticamente e também gerar pelo menos os ficheiros essenciais.

3.3 Arquitetura do sistema

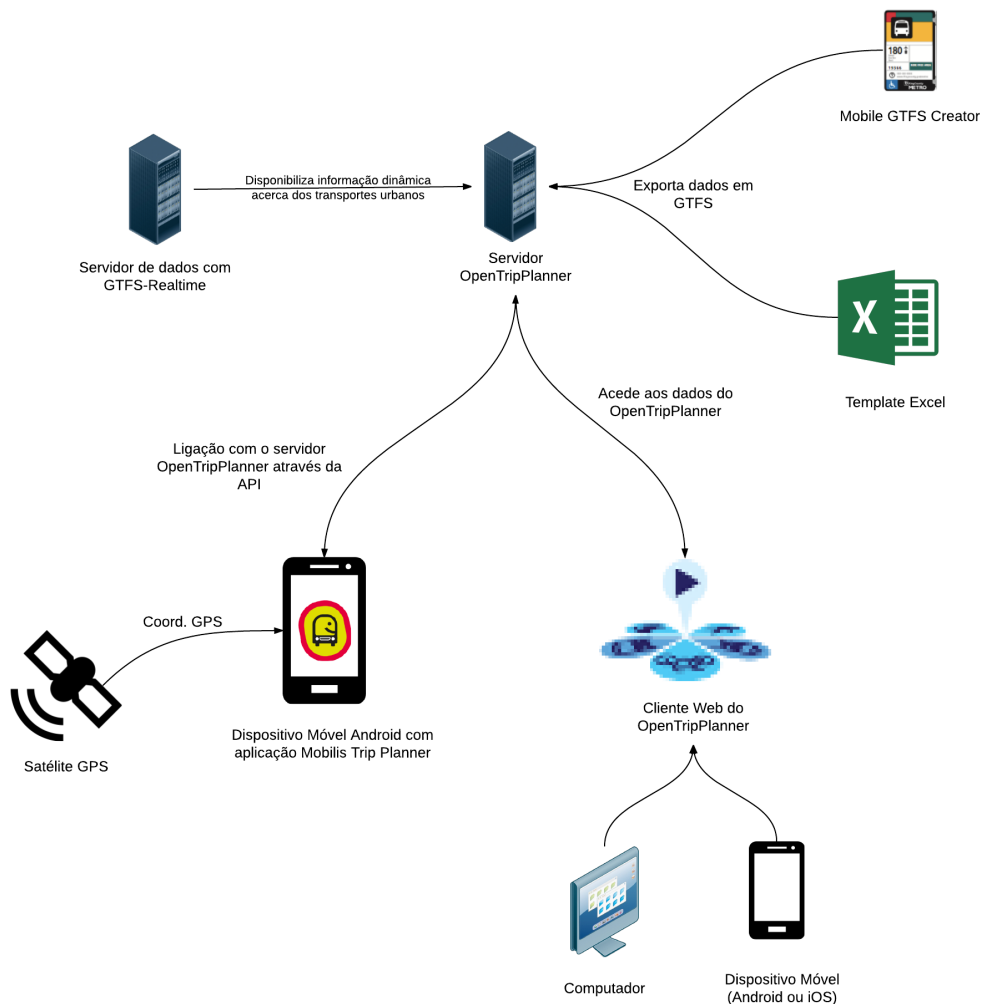


Figura 3.1: Arquitetura do sistema

De forma a dar resposta aos requisitos funcionais e não-funcionais, foi criada a arquitetura de sistema acima apresentada (figura 3.1), podendo ser dividida em 3 partes

essenciais:

- Criação
- Difusão
- Apresentação

A parte de criação é composta por:

- Mobile GTFS Creator
- Template Excel

O Mobile GTFS Creator (que nos requisitos corresponde ao Tradutor GTFS Mobilis) tem como objetivo traduzir os dados dos transportes da rodoviária do lis para formato GTFS. O Template Excel (nos requisitos equivale ao Template) tem como objetivo criar dados dos transportes de raiz.

A parte de difusão é composta por:

- OpenTripPlanner
- Servidor GTFS-Realtime

O OpenTripPlanner (que nos requisitos corresponde ao planeador de rotas) tem como objetivo ler os dados vindos da parte de criação e do servidor de dados GTFS-Realtime e posteriormente difundi-los para a parte de apresentação. O servidor de GTFS-Realtime (que nos requisitos corresponde ao servidor de informação dinâmica) tem como objetivo difundir dados dinâmicos como a localização dos transportes ao OpenTripPlanner.

A parte de apresentação é composta por:

- Computador e um dispositivo móvel geral
- Mobilis Trip Planner

O computador e dispositivos móveis gerais neste caso fazem uso do cliente Web já incluído no OpenTripPlanner. A aplicação Android Mobilis Trip Planner (que nos

requisitos corresponde ao planeador Android) consiste num planeador de rotas desenhado para Android e faz uso da API do OpenTripPlanner para apresentar a informação estática dos transportes públicos. Através desta arquitetura de sistema é possível apresentar uma solução bastante completa, capaz de criar os seus dados e depois posteriormente apresentá-los junto dos utentes dos transportes públicos.

3.4 Expansão do Mobilis

Durante o decorrer deste projeto, foi-nos informado, através de um prospeto, de que iria ser alterado o Mobilis da cidade de Leiria com novas linhas, paragens e horários. Visto estarmos a utilizar os dados do Mobilis para um melhor desenvolvimento deste trabalho, achou-se interessante desenvolver forma de facilmente podermos atualizar os dados GTFS com esta nova informação. Para isso, foi necessário fazer um estudo sobre as características e pormenores desta expansão para depois implementá-los em formato GTFS. Assim, nas próximas secções vamos então apresentar algumas das diferenças introduzidos com a expansão do Mobilis, que implicações e impacto isto trás aos ficheiros em formato GTFS.

3.4.1 Características introduzidas com a atualização do Mobilis

Com alteração do serviços do Mobilis introduzidos em 2016, poucas áreas mantiveram-se inalteradas. Existem novas linhas, novas paragens e novos horários.

Do ponto de vista das novas linhas, passou-se de 2 linhas para 10. Mantêm-se as duas primeiras linhas (linha verde e vermelha) com algumas alterações em termos de paragens e conseqüentemente horários, e outras 8 novas linhas. Ao mesmo tempo, algumas das paragens funcionam em conjunto com transportes interurbanos, também disponibilizados pela rodoviária do lis, obrigando a que o Mobilis viaje para sítios exteriores à zona central de Leiria para fazer algumas paragens em comum. Outra das novidades em termos de linhas, será a criação de uma linha dedicada ao turismo.

Com as novas paragens, surgem novos horários. O Mobilis 1 e 2 possui um novo trajeto (com mais paragens) e passa a haver um autocarro a cada 20 minutos em vez de 17. Ao mesmo tempo, o serviço noturno que anteriormente era da responsabilidade do Mobilis 1 e 2, é agora assumido pelo mobilis 1, também com um horário diferente

apesar de se manter a passagem a cada 60 minutos. Ao sábado também passou a haver horários diferentes, passando para 60 minutos em vez de 40 minutos as passagens dos autocarros. As restantes linhas são criadas de raiz e algumas delas (linha 3 por exemplo) têm paragens em comum com as interurbanas.

Uma grande diferença no entanto, é em termos de serviços (período de tempo em que determinado horário funciona). Anteriormente, os horários mantinham-se fixos ao longo do ano. Atualmente, algumas das linhas mudam o horário consoante a época do ano. Com isto, há linhas, como o Mobilis 1 e 2, que possuem um horário para o período escolar e outro para o período não-escolar, o Mobilis turístico que utiliza um horário diferente durante o Verão e a linha 9 que funciona todo o ano exceto entre os meses de Junho e Setembro. Ao mesmo tempo, existem horários que funcionam com menos paragens em alguns dias da semana e que dependem se estão num período escolar ou não, como no Mobilis 3.

Nota: para além destas alterações por volta de Abril de 2016, os preços dos bilhetes do Mobilis com a passagem para o ano de 2017 sofreram atualizações.

3.4.2 Implicações a nível do GTFS

Como seria de esperar, a crescente complexidade nos novos horários adiciona mais complexidade ao novo GTFS. Apesar das alterações, é possível manter a agência de transportes e preçários (os preços foram alterados com a passagem para o ano de 2017), visto nesse campo não ter havido mudanças. No entanto, como apontado anteriormente, os serviços, linhas, paragens, e horários obriga a grandes mudanças. Convém no entanto referir, que em relação aos ficheiros disponibilizados no início deste trabalho que utiliza os dados anteriores à expansão do Mobilis, foi necessário mudar os dados da companhia de transportes de "RDV" e "Rodotejo" para "RDL" e "Rodoviária do Lis".

Os serviços no `calendar.txt` que funcionam todos os Sábados, dias da semana e dias da semana à noite (Mobilis noturno) podem ser mantidos, mas será necessário criar novos serviços. Isto porque os períodos deixam em grande parte de serem todo o ano, para passarem a ser apenas parte do ano. Por exemplo, serão necessários novos serviços para os dias da semana no período escolar e não-escolar (considerado pelo Mobilis o período entre 30 de Junho e 1 de Setembro).

Para o `calendar_dates.txt`, para além de registarmos os dias de feriado em que o

Mobilis não funciona, como já acontecia anteriormente, será também necessário registrar os dias de funcionamento extra para os serviços da linha turística, visto estes trabalharem todos os feriados.

No caso do `routes.txt` e `stops.txt`, será necessário adicionar as novas linhas e paragens respetivamente. Para este último será no entanto uma tarefa mais morosa dado o grande numero de novas paragens.

Como seria de esperar, o `stop_times.txt` irá ser ainda maior e complexo, dado o grande número de horários que vão resultar dos novos e diferentes serviços implementados. Vai ser necessário implementar horários que nuns casos funcionam para o ano inteiro e noutros casos para períodos escolares e não-escolares, inclusive casos em que funcionam para Sábados do ano inteiro e noutros (Mobilis 1 e 2) que funciona apenas para um período escolar. Ao mesmo tempo, será necessário algum cuidado na criação dos horários de algumas linhas/serviços, porque anteriormente a esta atualização no Mobilis, grande parte dos horários agora não funcionam em períodos de tempo regulares, como acontecia para o Mobilis 1 e 2 que funcionavam de 17 em 17 minutos.

Em relação ao `trip.txt`, dado o grande numero de serviços e horários que advêm da expansão do Mobilis, será necessário tomar atenção os horários que associamos a um serviço.

Para o `calendar_dates.txt`, será necessário replicar os feriados pelos novos serviços. Já para o `fare_attributes.txt` e `fare_rules.txt`, não foi necessário atualizar (na altura que começou a ser feita atualização dos dados para lidar com esta alteração no Mobilis, o custo dos bilhetes era o mesmo).

3.4.3 Especificação de um formato e estrutura para um template auxiliar de criação/atualização de dados GTFS

Durante o desenvolvimento deste trabalho, com a expansão do Mobilis, surgiu a oportunidade de realizar um template que pudesse ajudar na criação/atualização dos ficheiros GTFS para depois serem aplicados no OpenTripPlanner. Para tal, seria necessário definir que formato este template iria ter de forma a simplificar a natureza complexa dos dados em GTFS. Com isto, surgiu dois possíveis formatos:

- Aplicação

- Template em Excel

Apesar da aplicação poder trazer outro grau de interação e simplicidade na criação dos novos ficheiros GTFS, decidimos optar pelo Excel visto ser uma ferramenta bastante comum e que grande das pessoas já têm conhecimentos.

Em termos de estrutura, este Template em Excel pode ser dividido em 3 partes:

- Leiname
- Folhas a preencher
- Folhas resultantes

Como o próprio nome indica, a secção de "Leiname" consiste numa folha que descreve e explica o funcionamento do Template assim como os passos finais para gerar os ficheiros GTFS.

As folhas a preencher, caracterizadas por possuírem no seu nome de folha o numero de ordem seguido do respetivo nome do ficheiro GTFS que representam (por exemplo "0- Agency"), são folhas de estrutura simples e *user friendly*, onde iram ser colocados os dados para posterior tradução em GTFS.

As folhas resultantes, caracterizadas por usar o nome "GTFS Nome-do-ficheiro", possuem todos os dados das "folhas a preencher" estruturados em formato GTFS. É através da exportação das folhas resultantes para CSV e posterior mudança de extensão dos ficheiros para TXT, que surgem os ficheiros GTFS para serem utilizados no OpenTripPlanner. Todo este processo é explicado na secção de "Leiname" mais em detalhe.

3.5 Conclusão

Com a crescente necessidade de melhor difundir a informação acerca dos transporte públicos, torna-se necessário não só desenvolver uma solução que dê resposta aos utilizadores dos transportes públicos mas também aos administradores de sistemas que necessitam de criar nova informação ou atualizar esta. Ao mesmo tempo, há interesse

que qualquer uma das plataformas ou aplicações que comuniquem com utilizadores possuam uma interface simples e eficaz. Para conseguir estes objetivos, foram definidos vários requisitos de sistema onde estão detalhados as funcionalidades desejadas para as várias plataformas e aplicações deste trabalho assim como as limitações que devem ser respeitadas para este poder ser utilizável. De forma a ligar todos estas plataformas e aplicações, foi especificada uma arquitetura de sistema, onde é detalhada cada uma das suas componentes. Por fim, visto a expansão dos transportes públicos Mobilis realizada em 2016, foi feita uma análise às alterações realizadas por estes e de que forma isto influenciaria a informação utilizada neste trabalho. Com tudo isto, foi possível definir que plataformas e aplicações utilizar nesta solução que não só respeitassem os objetivos de criar e disponibilizar informação acerca dos transportes, como também fosse possível melhorar o conhecimento dos utilizadores acerca dos transportes públicos e reduzir os tempos de espera nas paragens.

Capítulo 4

Implementação

Neste capítulo é apresentada a implementação deste trabalho, onde são discutidos os pormenores de desenvolvimento dos vários componentes da arquitetura do sistema, assim como os diversos desafios que surgiram da sua implementação. Por fim, será apresentada um resumo e conclusão deste capítulo.

4.1 GTFS

Nesta subsecção serão discutidos os vários desafios e detalhes das componentes que estão mais ligadas ao formato GTFS, como a estrutura dos ficheiros GTFS criados inicialmente para teste anteriores à expansão dos transportes Mobilis em 2016 e a forma como foi implementado o Template Excel.

4.1.1 Correção e reestruturação dos ficheiros em formato GTFS

Um dos desafios iniciais neste projeto e talvez um dos que mais influenciou o seu desenvolvimento, foi a correção e reestruturação inicial dos vários ficheiros em formato GTFS disponibilizados no início deste projeto. Apresenta-se de seguida os vários desafios que necessitaram de ser ultrapassados de forma a assegurar o correto funcionamento destes dados na plataforma OpenTripPlanner

GTFS base

O primeiro ficheiro GTFS utilizado para teste no início deste trabalho baseou-se nos mesmos desenvolvidos por Pedro José de Sousa Ferreira durante o seu trabalho [13]. No entanto, estes dados encontravam-se dentro de um ficheiro .PDF e com alguns dos seus caracteres alterados. Isto fez com que ao se exportar os dados para um ficheiro de texto, se tornassem incompatíveis com o OpenTripPlanner.

Para facilitar a tarefa de encontrar os vários erros, foi utilizada a ferramenta FeedValidator¹ capaz de executar várias validações sobre os ficheiros GTFS e no final obter uma lista de erros.

De forma a criar o primeiro conjunto de ficheiros respeitando o formato GTFS, foram necessárias as seguintes correções:

- Eliminar vários parágrafos em excesso no ficheiro routes.txt
- Trocar o carácter de duas virgulas seguidas por duas virgulas em separado
- Correção no ficheiro de stops.txt das coordenadas estarem no parágrafo abaixo da linha a que se referem
- Correção de vários parágrafos indevidos no ficheiro stop_times.txt
- Correção de vários parágrafos indevidos no ficheiro calendar.txt
- Alteração das datas de início e fim de serviço no ficheiro calendar.txt de forma a puderem serem utilizadas durante o ano de 2016.
- Alteração da coluna drop_off_time para drop_off_type

Depois de todas estas correções, o número de erros apresentados pelo FeedValidator foi reduzido e permitiu que estes dados pudessem ser utilizados no OpenTripPlanner sem problemas.

Tempos exatos para os horários

Ao testarmos algumas das funcionalidades do OpenTripPlanner, foi possível verificar que os tempos de chegada de um autocarro a uma paragem variavam em vez de serem

¹<https://github.com/google/transitfeed>

fixos. Por outras palavras, imaginemos que planeamos uma rota às 10:30 e que o Mobilis 1 chega às 10:35 à paragem ESTG. Se de seguida alterássemos a hora a que planeamos a rota para 10:32, a hora de chegada do Mobilis 1 à paragem ESTG deveria manter-se nas 10:35. No entanto, com este problema, ele dizia que a hora de chegada do Mobilis 1 seria às 10:37, o que não faz sentido.

Isto acontecia independentemente do tempo a mais ou a menos que programássemos as horas. Se quiséssemos uma viagem 21 minutos mais tarde, o Mobilis 1 passaria a chegar 21 minutos mais tarde à paragem, em vez de seleccionar outro Mobilis.

Para verificar se o problema estava nos dados GTFS ou na plataforma OpenTripPlanner, foi comparado o funcionamento desta com os dados do Mobilis e com os dados dos transportes de Coimbra e de Portland. Com isto, foi possível verificar que de facto o problema estava nos dados GTFS do Mobilis, já que os autocarros de Coimbra e de Portland chegavam a horas fixas.

Para solucionar este problema na altura, foi alterado o valor do campo `exact_times` no ficheiro `frequencies.txt` para "1" ao longo das várias linhas. Esta versão foi mantida durante muito pouco tempo em favor da reestruturação apresentada a seguir.

Horários por paragem

Durante os testes ao funcionamento do OpenTripPlanner com os dados de Coimbra e Portland, para além dos horários dos transportes serem fixos ao fazer o planeamento de uma rota, foi possível verificar que o OpenTripPlanner disponibilizava os horários para cada uma das paragens, algo que não acontecia com os dados GTFS do Mobilis.

Para este problema, haviam duas soluções possíveis na altura: adaptar uma solução semelhante à de Coimbra, fazendo uso do campo `exact_times` no ficheiros `frequencies.txt` e apenas dotar algumas das paragens com horas e minutos no ficheiro `stop_times`, ou então adaptar a solução de Portland, que não fazia uso do ficheiro `frequencies.txt` e colocava todos os horários no ficheiro `stop_times`.

Destas duas soluções, decidiu-se adaptar a estrutura original aplicada pela empresa Trimet para a cidade de Portland visto a adaptação da estrutura de Coimbra não ter tido o efeito desejado. Para isso, foi necessário deixar de utilizar o ficheiro `frequencies.txt` e reconstruir todos os seus horários no ficheiro `stop_times`. Isto obriga praticamente a recriar um horário completo no ficheiro `stop_times` para cada linha do

Mobilis.

Desta forma, tomando como exemplo o Mobilis 1, podemos representar o primeiro horário da manhã da seguinte forma:

```
MOB1D,7:00:00,7:00:00,1_LJL,1,,,,
MOB1D,7:01:00,7:01:00,2_AHA,2,,,,
MOB1D,7:03:00,7:03:00,3_MM,1,3,,,,
MOB1D,7:05:00,7:05:00,4_NL,1,4,,,,
MOB1D,7:07:00,7:07:00,5_AG,1,5,,,,
MOB1D,7:11:00,7:11:00,6_BA,6,,,,
MOB1D,7:14:00,7:14:00,7_EMP,1,7,,,,
MOB1D,7:15:00,7:15:00,8_PIS,1,8,,,,
MOB1D,7:16:00,7:16:00,9_A25A,1,9,,,,
MOB1D,7:19:00,7:19:00,10_ESDS,1,10,,,,
MOB1D,7:22:00,7:22:00,11_BC,1,11,,,,
MOB1D,7:23:00,7:23:00,12_CML,1,12,,,,
MOB1D,7:24:00,7:24:00,13_PM,1,13,,,,
MOB1D,7:26:00,7:26:00,14_ESTG,14,,,,
MOB1D,7:27:00,7:27:00,15_RESTG,15,,,,
MOB1D,7:31:00,7:31:00,16_PE,16,,,,
MOB1D,7:33:00,7:33:00,17_SAP,1,17,,,,
MOB1D,7:34:00,7:34:00,18_BS,1,18,,,,
MOB1D,7:36:00,7:36:00,19_AMP,1,19,,,,
MOB1D,7:37:00,7:37:00,20_ASE,1,20,,,,
MOB1D,7:38:00,7:38:00,21_ANSF,2,21,,,,
MOB1D,7:40:00,7:40:00,22_GUI,1,22,,,,
MOB1D,7:42:00,7:42:00,23_EPL,23,,,,
MOB1D,7:44:00,7:44:00,24_SR1,24,,,,
MOB1D,7:45:00,7:45:00,25_SR2_RF,25,,,,
MOB1D,7:46:00,7:46:00,26_SR3_CETL,26,,,,
MOB1D,7:48:00,7:48:00,27_HPVI,27,,,,
MOB1D,7:49:00,7:49:00,28_HPCE,28,,,,
MOB1D,7:50:00,7:50:00,29_ECM1,29,,,,
MOB1D,7:52:00,7:52:00,30_BM1,30,,,,
MOB1D,7:53:00,7:53:00,31_PHR1,31,,,,
MOB1D,7:55:00,7:55:00,32_JLC1,32,,,,
```

Figura 4.1: Novo formato dos horários para o antigo mobilis 1

Como se pode verificar, a primeira viagem do dia no Mobilis 1 é constituída por 32 paragens, com um horário para cada uma destas paragens e um ID comum para esse conjunto de paragens (ou *trip*/viagem). Agora, para dias da semana existe um Mobilis 1 a cada 17 minutos, logo, seria necessário replicar aquele conjunto de paragens para cada um desses 17 minutos de diferença ao longo do dia. No entanto, existem horários diurnos, noturnos e ao Sábado para o Mobilis 1 e 2. Isto representa no final 43 trips (conjunto de paragens) para o regime diurno, 4 para o Noturno e 9 para o Sábado, só para o Mobilis 1. Para o Mobilis 2 são o mesmo número de trips: 43 para o diurno, 4 para o noturno e 9 para o Sábado. Ao mesmo tempo, cada conjunto de paragens ou *trip*, precisa de um identificador único.

No final de resolvido o stop_times.txt, é necessário definir cada uma destas "*trips*" no

ficheiro trips.txt.

Como se pode concluir pela descrição anterior, esta reestruturação trouxe uma grande complexidade ao GTFS, principalmente no ficheiro stop_times.txt. No entanto, com isto foi possível habilitar no OpenTripPlanner a capacidade de apresentar os horários por paragens e assim habilitar uma funcionalidade bastante importante já que grande parte das aplicações de planeamento de rotas a possuem.

Convém também referir, que sem esta reestruturação final, não seria possível aplicar os horários com o tempo variável entre trips existentes na expansão do transportes Mobilis.

No final de esta reestruturação, foi feita a verificação dos dados através do FeedValidator e apesar de haver alguns warnings, estes não representam nenhuns problemas graves.

4.1.2 Template Excel

No capítulo anterior (Proposta de Solução) foram apresentadas as necessidades e implicações trazidas pela expansão do Mobilis em 2016. De forma a simplificar a tarefa de criação/atualização de dados em formato GTFS referentes a esta expansão, foi proposto o desenvolvimento de um Template em Excel que pudesse auxiliar os utilizadores a resolverem este problema. Nesta secção serão apresentados os vários desafios e detalhes referentes à criação deste Template.

Escolha do formato do template

Um dos desafios iniciais na criação de um Template com o objetivo de criar os dados dos transportes públicos em formato GTFS, foi a escolha da ferramenta para esse fim, visto ser possível criar estes ficheiros através de um Template desenvolvido em Excel, de uma aplicação desenvolvida para o caso, ou criando-os manualmente. Esta última opção, apesar de exigir algum trabalho, consistia na solução mais rápida e simples a ser implementada, visto podermos inserir dados diretamente nos ficheiros GTFS e no final criar um ficheiro compactado com todos estes ficheiros GTFS a serem colocados na plataforma OpenTripPlanner. No entanto, apesar desta forma permitir simplificar em muito o trabalho de difundir os dados do Mobilis inicialmente, complica

em muito o trabalho de atualização ou correção destes no futuro, visto a necessidade de conhecimentos dos ficheiros GTFS. Ao mesmo tempo, estes dados só poderiam ser usados no Mobilis de Leiria, o que não traria nada de novo caso alguém quisesse desenvolver os seus dados para outra localidade. Outra solução interessante para este problema, seria a criação de uma aplicação para o caso. No entanto, isto implicaria dar formação às pessoas sobre como usar a ferramenta assim como a necessidade de compatibilidade dos sistemas operativos com a aplicação. Desta forma, optou-se pela criação de um Template em Excel, visto grande parte das pessoas possuir conhecimentos sobre esta ferramenta e já se encontra disponível em grande dos computadores. Ao mesmo tempo, o Excel fornece de origem a capacidade de facilmente gerar ficheiros em formato CSV, na qual são estruturados os dados em GTFS.

Estruturação dos dados

Estruturar os dados GTFS num ou em vários ficheiros Excel, apresenta dois desafios em termos de estruturação:

- Atribuição de um único ou vários ficheiros Excel para colocar os dados GTFS
- Definição da representação dos dados nos ficheiros

A utilização de um ficheiro Excel para cada ficheiro em GTFS trazia uma maior organização na separação de tarefas, já que cada um dos ficheiro trataria dum tarefa e ao mesmo tempo, gerava o seu GTFS. No entanto, os ficheiros GTFS trabalham em conjunto, havendo alguns dados em comum entre eles em que uma atualização num ficheiro obriga à correção nos outros ficheiros. Ao mesmo tempo, o ato de saltar entre ficheiros para tratar dos dados podia tornar-se um problema. Por essa razão, decidiu-se optar por utilizar um único ficheiro Excel, visto o salto entre ficheiros GTFS e a atualização dos dados ficarem facilitados.

Em relação ao segundo desafio, existia a necessidade de simplificação da estrutura dos ficheiros GTFS visto que alguns dos ficheiros possuem campos que são pouco representativos das suas funções e por vezes existem demasiados campos que não são utilizados.

Para solucionar este problema, cada ficheiro GTFS possui uma folha "a preencher" e uma folha de "GTFS". Na folha "a preencher" foi feita a simplificação da estrutura

GTFS, passando a utilizar um número inferior de campos e ao mesmo tempo, com nome mais representativos. Posteriormente, os dados desta folha de interface iriam ser traduzidos para a respetiva folha GTFS, onde existe uma estrutura integra de todos os dados em formato GTFS.

Apesar da simplicidade de estrutura da folha "a preencher" em relação à dos ficheiros GTFS, foi criada uma folha de ajuda, onde são explicados os vários campos de cada uma das folhas "a preencher".

Criação de um primeiro template Excel para a criação de dados em formato GTFS

Inicialmente, foi desenvolvido um ficheiro excel por ficheiro de GTFS essencial, logo, haveriam ficheiros como `calendar.xls`, `calendar_dates.xls`, entre outros. Cada ficheiro seria então constituído por uma folha de "leiamos" com instruções para o seu preenchimento, uma folha de cálculos, onde estaria uma versão simplificada do GTFS e uma folha com toda a estrutura em GTFS gerada, para depois ser gravada em formato CSV.

Para preencher automaticamente as folhas de GTFS, foram desenvolvidas várias fórmulas que permitiam ir buscar às respetivas secções de "folhas a preencher" os valores pretendidos e se necessário, ainda executar alguns cálculos. Caso fosse preciso colocar dados novos nas folhas de GTFS, era necessário replicar as fórmulas pelas células necessárias.

Reestruturação dos vários templates num só ficheiro e aperfeiçoamento das folhas de preenchimento do template Excel

Para a versão 2, desde logo, foi decidido juntar todos os Templates num só, passando a ser usado um "leiamos" geral dividido por várias secções, onde é explicado como preencher cada uma das folhas. De forma a diferenciar as folhas "a preencher" das folhas de "GTFS", as folhas "a preencher" possuem unicamente o nome dos ficheiros (`calendar`, `calendar_dates`, `stops` e `restantes`) e as folhas GTFS possuíam o termo "GTFS" antes do nome da folha (`GTFS Calendar`, `GTFS calendar_dates`, `GTFS stops` e `as restantes`). Ao mesmo tempo, cada uma das folhas "a preencher" foram numeradas e ordenadas, permitindo assim no "Leiamos" indicar que para adicionar uma nova linha, precisa alterar da folha 4 à 6, por exemplo. Para confundir menos o utilizador, as folhas a

preencher foram movidas para o início, logo a seguir à folha "Leiname" e as folhas de GTFS" estão disponíveis depois de todas as folhas que precisem de ser preenchidas

Para simplificar o processo de preenchimento do Templates, algumas células foram alteradas de forma a disponibilizarem um dropdown menu onde podemos selecionar o valor desejado. Por exemplo, na folha "2-calendar_dates" na coluna "Nome Período de Tempo" existe um menu para selecionar os serviços que foram adicionados na folha "1 - Calendar". Esta funcionalidade também foi adicionada nas folhas de stop_times e trips.

A folha de stop_times, visto a sua complexidade, foi simplificada, passando vários campos para gerar um id aleatório, num simples campo preenchido pelo utilizador.

Na secção de stop_times, foi criada uma folha chamada "5 - Entre-Tempos" para resolver o problema da inexistência das horas para algumas das paragens. Isto foi possível através da diferença de horas disponibilizadas nos horários e o número de paragens entre elas e com isto fazer uma média de tempo entre paragens. A intenção principal com esta folha, seria ajudar na criação dum primeiro horário base, que depois podia ser usado na página principal do stop_times para gerar os restantes horários ao longo do dia.

Adaptação de macros e alteração das folhas de preenchimento

Para esta última versão, foi novamente mudada a forma de trabalhar dos Templates. Até agora, as folhas GTFS eram geradas através de fórmulas que existiam nestas. Com isto, existia a desvantagem de ao criarmos novas linhas com dados nas folhas "a preencher", fosse necessário replicar as fórmulas nas folhas "GTFS" para estas novas linhas. Ao mesmo tempo, a folha stop_times continuava a ser demasiado complexa para gerar os horários.

Para resolver estes problemas, foi adaptado o uso de macros. Ainda assim, todos os templates mantiveram praticamente a sua estrutura em relação à versão 2 com exceção do stop_times, onde foi diminuído em muito a quantidade de dados necessários ao seu preenchimento. Assim, as folhas "1-calendar", "2-calendar_dates", "3-stops" e "4-routes", continuam com a sua estrutura inalterada e foram adicionados dois botões: um para gerar os dados GTFS e outro para limpar os dados GTFS. Ainda que mantendo a estrutura anterior, a folha "a preencher" trips possui mais botões que os templates anteriores, neste caso, um para gerar os valores na folha "6-trips" com base

nos dados presentes no "GTFS" do stop_times (já que normalmente preenche-se primeiro os horários no stop_times), um botão para limpar os valores da folha "trips", um botão para gerar os valores no "GTFS trips" e outro para limpar a respectiva folha GTFS.

Apesar da utilização de macros na folha trips ajudar em muito a preencher os dados nesta, é no stop_times onde residem as maiores vantagens. Em termos de utilização, deverá ser usada uma nova folha por cada linha. Por exemplo, este template possui desde já uma folha stop_times para o Mobilis 5 e para o Mobilis 9. Caso seja desejado criar uma nova linha de Mobilis posteriormente, deverá ser copiada uma destas folhas e depois alterados os valores para os desejados.

Cada uma destas folhas possui 4 colunas que são necessárias preencher:

- Abreviatura da Linha: Abreviatura do nome de uma linha do Mobilis. Exemplo: MOB9 para o Mobilis 9
- Nome da paragem (com menu para escolher o nome da rua). Exemplo: Largo José Lúcio da Silva
- Hora: Hora de chegada do autocarro à paragem. Caso não haja hora, deixa-se a vazio
- Horas início de viagem: Horas iniciais para cada partida, ou conjunto de partidas. Primeiro valor de uma coluna num horário.

No final de preenchida uma folha, existem botões que nos permite adicionar ou remover aquela linha de transportes nas folhas GTFS.

No entanto, convém ter atenção que os horários colocados nestas folhas são sempre gerados completamente. Por exemplo, para o Mobilis 3, existem trips (ou conjuntos de paragens) que não utilizam todas as paragens. Desta forma, para cada um dos horários que possua paragens diferentes, será necessário criar uma nova folha. Ao mesmo tempo, as macros para o stop_times foram implementadas com isto em mente, logo, se os diferentes horários estiverem associados à mesma abreviatura de linha (E.G MOB9), estes são adicionados à lista já existente em vez de sobrepostos. Com isto, podemos adicionar ao GTFS do stop_times os horários completos para o MOB3 presentes numa folha, e depois adicionar os horários incompletos do MOB3 de outra folha.

Com o gerar e remover de algumas linhas de transportes, é possível que fiquem células vazias nas folhas de GTFS, o que pode gerar valores incorretos ao gerar o ficheiro stop_times.csv. Para isso, na folha "Leiname", na secção dos stop_times, encontra-se o botão "Limpar horários GTFS" para fazer uma limpeza geral ao GTFS do stop_times. No entanto, isto apagará todos os dados existentes. Logo, deverá depois percorrer todas as folhas de horários das várias linhas dos Mobilis existentes para adicionar novamente os valores.

Passos finais do template

No final de preenchidos todos as folhas do Template e gerados todos os valores das folhas de GTFS, é necessário passar estes para ficheiros de extensão ".csv", verificar se é usada a vírgula em vez de ponto-e-virgula ao exportar (explicado mais em pormenor de seguida) e mudar a extensão de ".csv" para ".txt". Para isso, para cada folha GTFS do Excel, será necessário exportá-la/guardá-la com extensão ".csv" com separador por vírgulas.

Desta forma, cada folha de GTFS no Excel deverá ser guardada como:

- GTFS Agency → agency.csv
- GTFS Calendar → calendar.csv
- GTFS Calendar_Dates → calendar_dates.csv
- GTFS Routes → routes.csv
- GTFS Stops → stops.csv
- GTFS Stop_Times → stop_times.csv
- GTFS Trips → trips.csv
- GTFS Fare_Attributes → fare_attributes.csv
- GTFS Fare_Rules → fare_rules.csv

No entanto, devido às definições de origem do sistema operativo Windows, não são colocadas virgulas mas sim ponto-e-virgula. Para isso, será necessário abrir os vários

ficheiros ".csv" numa aplicação como o "Bloco de Notas" ou "Notepad++"² e substituir o ";" pela ",". De seguida, na própria aplicação, podemos "Guardar como..." o ficheiro com extensão ".txt" (mantendo o nome do ficheiro) e depois eliminar os ficheiros ".csv". Outra solução será no explorador do windows (desde que este apresente o nome das extensões dos ficheiros), alterar as extensões de ".csv" para ".txt".

No final de passar todos os ficheiros a ".txt", deverá possuir os seguintes ficheiros numa pasta:

- agency.txt
- calendar.txt
- calendar_dates.txt
- routes.txt
- stops.txt
- stop_times.txt
- trips.txt
- fare_attributes.txt
- fare_rules.txt

Para finalizar o processo de criação de um ficheiro GTFS preparado para ser usado num servidor OpenTripPlanner, deverá compactar estes ficheiros num ficheiro ".zip". Poderá usar uma aplicação como o Winzip, Winrar ou 7-Zip para isso.

4.2 OpenTripPlanner

Nesta subsecção serão discutidos os vários desafios e detalhes das componentes que estão ligadas com o OpenTripPlanner, como a própria plataforma, o mapa de Leiria para depois ser aplicado na plataforma, tradução para Português do OpenTripPlanner e por fim a implementação do GTFS-Realtime nesta.

²disponível em: <https://notepad-plus-plus.org/>

4.2.1 Desafios e requisitos do OpenTripPlanner

Como é possível verificar pela arquitetura de sistema disponibilizada no capítulo anterior, o OpenTripPlanner consiste numa das componentes principais deste projeto através da qual disponibilizamos os dados a outras aplicações.

Um dos desafios iniciais do OpenTripPlanner no início deste trabalho, foi a questão da versão a escolher. Isto acontecia porque havia duas documentações diferentes disponíveis: uma no Website oficial para versões mais recentes (≥ 0.11)³ e outra para versões mais antigas (< 0.11)⁴ no Github. Inicialmente, optou-se por usar uma versão antiga, visto possuir uma documentação mais completa. No entanto, visto as versões antigas serem incompatíveis com várias versões do Java que testámos (Java 6, 7 e 8) e IDEs (Netbeans e Eclipse), decidiu-se apostar na segurança e utilizar o OpenTripPlanner 0.18 já pré-compilado. Ao mesmo tempo, com esta versão era possível manter a compatibilidade com a aplicação OpenTripPlanner for Android, que havia sido descoberta na altura.

Mais tarde, devido a uma atualização no MapQuest do OpenTripPlanner (passando a ser uma solução completamente na *Cloud*), houve a necessidade de atualizar esta plataforma visto não aparecer no ecrã principal da aplicação o mapa da zona. Após algum estudo da situação, decidiu-se implementar o OpenTripPlanner versão "1.1.0-SNAPSHOT", vinda diretamente dos repositórios do Github. Como seria de esperar, a atualização do OpenTripPlanner para versões mais recentes quebrou a compatibilidade com o OpenTripPlanner for Android. No entanto, este problema viria a ser resolvido mais tarde através de várias correções no OpenTripPlanner for Android.

Para alimentar um servidor OpenTripPlanner como o disponibilizado neste trabalho são necessários ficheiros como:

- Ficheiro .jar do OpenTripPlanner (foi desenvolvida uma versão personalizada no decorrer deste trabalho)
- Ficheiro de route-config.json configurado para a utilização do GTFS-Realtime
- Mapa da zona (com extensão .osm ou .pbf)
- Ficheiro .zip com os ficheiros em formato GTFS

³<http://docs.opentripplanner.org/en/latest/>

⁴<https://github.com/opentripplanner/OpenTripPlanner/wiki>

Em termos de requisitos para executar o servidor OpenTripPlanner foi utilizado um sistema operativo compatível com o Java 8 e um computador com 4 GB de RAM visto ser necessário dedicar 1 GB de memória RAM ao servidor OpenTripPlanner (mapas maiores poderão precisar de mais memória RAM).

Encontra-se no anexo C uma lista de passos para a instalação e execução de um servidor OpenTripPlanner como o utilizado neste trabalho.

4.2.2 Mapa para a zona de Leiria

Um dos requisitos para o OpenTripPlanner funcionar, como indicado anteriormente, é um mapa da zona. Este mapa necessita de estar em extensão .pbf ou .osm e é através dele que representamos os locais das paragens. Inicialmente, foi feita uma pesquisa por mapas para a zona de Leiria, principalmente na página geofabrik ⁵, no entanto sem sucesso. Visto só estar disponibilizado o mapa de Portugal, foi decidido utilizar este inicialmente para efeitos de teste. No entanto, a utilização de um mapa para Portugal inteiro trazia grandes problemas de performance, visto o sistema demorar vários minutos a fazer o arranque do servidor e necessitar de pelo menos 4 GB de memória RAM dedicada ao servidor OpenTripPlanner. Uma das soluções para este problema seria cortar uma secção do mapa de Portugal para criar um novo mapa apenas com a cidade de Leiria. No entanto, devido a falta de documentação, não foi aplicada esta solução. Para solucionar este problema, foi decidido criar um novo mapa de raiz, usando a função de exportar disponível no Website do OpenStreetMap para criar uma mapa dedicado à zona de Leiria em extensão ".osm".

Com esta alteração foi possível passar-mos para um mapa bem mais pequeno, através do qual conseguimos um arranque do servidor em menos de um minuto e a necessitar apenas 1GB de memória RAM dedicada ao servidor OpenTripPlanner.

4.2.3 Tradução para Português do OpenTripPlanner

Apesar do processo de tradução do cliente web do OpenTripPlanner ser relativamente simples, apresentou alguns desafios ao longo deste projeto. A informação relativa a traduções é mínima, visto a secção "Wiki" disponibilizada no Github do OpenTripPlanner ser referente a versões antigas (inferior à 0.11) e na página oficial do OpenTripPlanner,

⁵<http://www.geofabrik.de/>

não é dada muita ênfase a esta informação. Após alguma pesquisa, foi possível verificar que existe um ficheiro *readme* acerca da localização, especificando que ficheiros seriam necessários alterar e traduzir. No entanto, enquanto se desenvolvia os ficheiros de tradução, não era referido neste como gerar o *pt.json*, apesar de dar a entender que deveria ser gerado automaticamente⁶. No entanto, ao tentar gerar os ficheiros *.jar* sem o *pt.json*, era apresentado um erro no teste de criação da interface da parte cliente e não era criado qualquer *pt.json*. No caso de ser criado o *pt.json* à mão, a compilação passava sem problemas e eram gerados os ficheiros *.jar* necessários para executar o *OpenTripPlanner*. Visto o ficheiro não ser criado automaticamente pelo *OpenTripPlanner*, foram testadas outras soluções como o *localize.biz*⁷, que permite traduzir um ficheiro *.po* para *.json* selecionando a tradução desejada. No entanto, nenhuma das traduções resultava num ficheiro com a estrutura utilizada pelo *OpenTripPlanner*. De forma a resolver este problema, foi criada uma cópia do ficheiro *en.json* e posteriormente traduzido para português e renomeado para *pt.json*. Com isto, já era possível obter uma tradução do *OpenTripPlanner* em Português.

Ultrapassados os desafios, para traduzir o *OpenTripPlanner* para Português, foram necessárias várias alterações como:

- Criação do ficheiro *Portuguese.js* - ficheiro com as afinações para Portugal (se usa distância em milhas ou quilómetros por exemplo);
- Criação do ficheiro *pt.json* - ficheiro em *i18next* com as traduções para Portugal;
- Criação do ficheiro *pt.po* - ficheiro gerado através do *template.pot* com traduções em *pt*;
- Alteração do *config.js* - necessário adicionar a língua portuguesa à lista de línguas;
- Adição do ficheiro *jquery.ui.datepicker-pt.js* - adquirível no github do *jquery*⁸;
- Alteração do ficheiro *index.html* - necessário adicionar o *Portuguese.js* à lista de línguas assim como o ficheiro *jquery.ui.datepicker-pt.js*;
- Alterar o ficheiro *makefile* - necessário adicionar *pt* à variável *LANGS*;

Em termos de afinações sobre estes ficheiros, o *Portuguese.js* foi criado a partir do ficheiro *English.js* e foi necessário definir ao longo deste a língua como "Portuguese" e a

⁶Na secção de "Updating translations" do ficheiro de localização, de facto é referido que estes ficheiros deverão ser criados automaticamente, mas devido à utilização do Babel(<http://babel.pocoo.org/> e ao *i18next-conv* (<https://github.com/jamuhl/i18next-gettext-converter>))

⁷<https://localise.biz/free/converter/po-to-json>

⁸<https://github.com/jquery/jquery-ui/tree/1-9-stable/ui/i18n>

sua abreviatura como "pt". Convém referir que para este caso, tem de ser mesmo "pt" e não "pt_PT" como é referenciado no `readme_localization.md` do OpenTripPlanner⁹. Ao mesmo tempo, também é necessário definir que usamos o sistema métrico em vez de imperial e o formato das horas e datas de Portugal (esquema de 24 horas e formato de datas dd/MM/aaaa). O ficheiro `pt.json`, contém as várias traduções em Português para o OpenTripPlanner na parte cliente em formato i18next. O ficheiro `pt.po` é gerado inicialmente através de uma aplicação de tradução em gettext¹⁰, que usa ficheiro `message.pot` disponibilizado pelo OpenTripPlanner como `emplate` para gerar um ficheiro `.po`. Depois de ser gerado o `pt.po`, a aplicação Poedit também permite a fácil tradução das várias frases/palavras para Português, apresentando o valor do campo original em Inglês e um campo para tradução. Ao mesmo tempo, o Poedit permite mostrar várias *hints* vindas do `message.pot` a indicar de que ficheiro vem aquela frase ou palavra e assim ajudar na contextualização da tradução no momento. No ficheiro `config.js`, é necessário adicionar a língua Portuguesa à lista de línguas disponíveis a serem utilizadas na parte cliente do OpenTripPlanner, sendo necessário adicionar neste caso o nome da língua (Portuguese) e a sua abreviatura (pt). De forma a utilizar um `datepicker` personalizado para Portugal na parte cliente do OpenTripPlanner, é também necessário adicionar o ficheiro `jquery.ui.datepicker-pt.js`. No ficheiro `index.html` tem de ser referido o caminho para o ficheiro `Portuguese.js` à semelhança das outras línguas assim como o ficheiro `datepicker` para "pt", de forma a poder ser usada na parte cliente. Por fim, foi adicionado ao ficheiro `makefile` a abreviatura "pt" à variável `LANGS`. Visto o OpenTripPlanner oficial não possuir a língua Portuguesa, foi decidido submeter esta junto da comunidade desenvolvedora, sendo para isso necessário criar uma instância do servidor do Github num computador, alterar os ficheiros necessários e no final fazer `commit` e `pull-request`. Todos estes passos ocorreram sem problemas e a tradução para Português está atualmente disponível na versão oficial 1.1.0 lançada no dia 17 de Março de 2017. No entanto, para a plataforma OpenTripPlanner desenvolvida neste trabalho e dedicada aos transportes públicos Mobilis, ainda foi necessário adicionar alguma informação nas janelas de "acerca" e "contactos" e alterar as definições originais do OpenTripPlanner para utilizar o sistema métrico em vez de imperial, esquema de 24 horas e datas no formato "dd/MM/aaaa" de forma a evitar que ele use os formatos americanos, definidos como origem no OpenTripPlanner, quando acedemos ao cliente Web. Convém também referir, que mesmo usando as definições de origem do OpenTripPlanner (formato americano), quando é escolhida a língua Portuguesa no menu das línguas, toda a página é traduzida para Português e com o formato de distâncias, horas e datas de Portugal.

⁹https://github.com/opentripplanner/OpenTripPlanner/blob/master/README_LOCALIZATION.md

¹⁰<https://poedit.net/>

4.2.4 GTFS-Realtime

Ao contrário do GTFS, onde os dados estão em simples ficheiros .txt, o GTFS-Realtime obriga a que estes estejam estruturados num ficheiro respeitando o Protocol Buffer Message¹¹. Para isso, foi necessário fazer o download da ferramenta protoc (compilador para Protocol Buffer Message)¹² e instalar no sistema operativo em que iremos serializar os dados. Visto que o Protocol Buffer Message obriga a que os dados sejam estruturados e depois serializados usando o Protoc, é necessário escolher uma linguagem de programação através da qual possamos criar um *script* de forma a gerar um ficheiro serializado. Das várias linguagens disponíveis, decidimos optar pela linguagem Python¹³, visto a sua simplicidade.

Apesar de existirem vários exemplos simples de como estruturar os dados para depois serem serializados usando o Protocol Buffer Message, a estruturação dos dados em GTFS-Realtime não foi direta. A sua estrutura mais complexa e a necessidade de criar funções para cada uma das suas sub-estruturas não era direta nem era possível deduzir através doutros exemplos.

De forma a verificar a correta estruturação dos dados em GTFS-Realtime, para além da criação do script para adicionar os dados, foi criado um script para verificar se os dados tinham sido inseridos corretamente através da impressão da estrutura deste no ecrã.

Desenvolvido o ficheiro GTFS-Realtime em Protocol Buffer Message, foi necessário criar o ficheiro router-config.json, através do qual informamos o OpenTripPlanner da localização do ficheiro com os dados dinâmicos. Visto este ficheiro não ser gerado naturalmente pelo servidor, foi necessário pesquisar por um ficheiro template e adicionar o caminho para o ficheiro GTFS-Realtime¹⁴. Visto o OpenTripPlanner necessitar que o ficheiro fosse disponibilizado online com acesso direto ao seu conteúdo, este foi adicionado a um repositório no GitHub¹⁵ para depois ser acedido.

Desta forma, para implementar um servidor GTFS-Realtime como o utilizado neste projeto é necessário:

¹¹<https://developers.google.com/protocol-buffers/docs/overview>

¹²<https://developers.google.com/protocol-buffers/docs/downloads>

¹³<https://www.python.org/>

¹⁴template disponível em: <https://github.com/opentripplanner/OpenTripPlanner/blob/master/docs/Configuration.m>

¹⁵<https://github.com/>

- Fazer download do compilador para Protocol Buffer Message (na altura, foi feito o download do proto2, havendo atualmente a versão proto3 lançada a 27 de Julho de 2016 ¹⁶)
- Escolher uma linguagem de programação suportada
- Criar os scripts desejados já com a serialização
- Colocar os dados num servidor

Encontra-se no anexo D uma lista de passos para instalação e configuração do GTFS-Realtime em Linux (ao longo deste projeto foi usado o Xubuntu 16.04 LTS¹⁷ mas deverá ser compatível com outras distribuições baseadas em Ubuntu 16.04 LTS).

4.3 Componentes dedicadas aos transportes Mobilis

Nesta secção iremos discutir a implementação e os desafios dos componentes que foram criados com o objetivo de melhorar os transportes Mobilis, como a aplicação Mobilis Trip Planner para dispositivos Android, o Mobilis GTFS Creator e a adaptação da plataforma OpenTripPlanner aos transportes Mobilis.

4.3.1 Mobilis Trip Planner

O Mobilis Trip Planner foi criado com o objetivo de disponibilizar uma aplicação em Android capaz de planear rotas e fornecer informação acerca dos transportes mobilis de Leiria. Esta baseia-se na aplicação OpenTripPlanner for Android¹⁸, capaz de realizar o planeamento de tarefas e por sua vez adiciona algumas características em falta como a capacidade de visualizar os horários das paragens assim como um interface visual mais moderno e semelhante a aplicações Android mais recentes. Um dos problemas iniciais do OpenTripPlanner for Android, consistia na sua incompatibilidade com versões do OpenTripPlanner mais recentes que a versão 0.18. Na altura, pensou-se que este problema resultasse da diferença de resultados entre a versão 0.18 e 0.19 ao pedir os *routers* à API do OpenTripPlanner (imagens dos resultados disponíveis no anexo B). Visto a

¹⁶<https://developers.google.com/protocol-buffers/docs/downloads>

¹⁷<https://xubuntu.org/>

¹⁸<https://github.com/CUTR-at-USF/OpenTripPlanner-for-Android/wiki>

versão 0.19 não trazer nada de novo na altura, decidiu-se manter a compatibilidade com o OpenTripPlanner for Android e apostar na versão 0.18.

Ultrapassado este problema de incompatibilidade com o OpenTripPlanner (viriam a surgir problemas mais tarde no entanto), passou-se ao problema do seu interface estar aperfeiçoado para dispositivos Android versão 2.3.3, fazendo uso do *button menu* em vez da Action Bar. Para resolver este problema, foi feito o *download* do OpenTripPlanner for Android disponibilizado no Github¹⁹ e adaptado o uso de action bar. De forma a navegar facilmente entre écrãs, foi considerada inicialmente a utilização de um Title View e fragmentos de forma a obter um menu semelhante ao usado na Google Play Store. No entanto, após alguns testes foi verificado que apenas era permitido uma instância de MapFragment por atividade. Visto o OpenTripPlanner funcionar com mapas de diferentes fornecedores (Google Maps e OpenStreetMap), decidiu-se assim usar uma Recycler View como substituto do Title View e uma atividade dedicada ao planeamento de rotas vindo do OpenTripPlanner for Android e outra atividade criada de raiz para a apresentação das paragens e os seus horários fazendo uso do mapa do Google Maps. Mais tarde, com a mudança na implementação do MapQuest do OpenStreetMap (passando totalmente a *Cloud*), houve a necessidade de alterar a versão do OpenTripPlanner para 1.0.0 ou superior. Com isto, o Mobilis Trip Planner deixaria de se conseguir conectar à plataforma OpenTripPlanner, devido ao desconhecimento dos limites dos mapas. Após alguma pesquisa, foi descoberto que este problema devia-se à alteração do caminho da função "metadata" da API do OpenTripPlanner, de "/metadata" para simplesmente "/". Corrigido este problema, já deveria ser possível ao Mobilis Trip Planner saber quais eram os limites do mapa e assim poder usar o planeador de rotas. No entanto, ao pedir o planeamento de uma rota, este apresentava um erro na aplicação a notificar que a área estava indisponível, independentemente da origem e destino do planeamento de rota. No entanto, através do Android Monitor do Android Studio foi possível verificar que ao ser feito o planeamento de uma rota, acontecia uma exceção relacionada com os transportes "trainish". Visto o Mobilis de Leiria utilizar exclusivamente autocarros, foi retirado o valor "trainish" do planeamento de rotas para quando estavam selecionados todos os veículos (valor de origem). Posteriormente, visto o interesse da rodoviária do lis neste projeto, decidiu-se alterar a aplicação OpenTripPlanner for Android de forma a torná-la exclusivo aos transportes Mobilis. Para isso, foi necessário desativar o menu de seleção de servidor/cidade e desabilitar a opção de se ligar a outros servidores. Por fim, foi mudado o nome da aplicação para "Mobilis Trip Planner" e mudado o esquema de cores da aplicação de forma a obter uma interface mais agradável visualmente e caracterizada para os transportes mobilis.

¹⁹<https://github.com/CUTR-at-USF/OpenTripPlanner-for-Android/wiki>

4.3.2 Mobilis GTFS Creator

Durante a primeira reunião com a rodoviária do lis a apresentar este projeto, foi proposto como forma de criarem os seus primeiros dados dos transportes Mobilis em formato GTFS o preenchimento do template em Excel anteriormente desenvolvido durante este trabalho. No entanto, foi verificado que esta poderia não ser a melhor solução, dado que a rodoviária do lis já possuía os seus dados em formato digital e a duplicação de dados poderia originar bastantes erros. Por estas razões foi decidido criar uma aplicação que pudesse cumprir este novo objetivo. Para isso, foi desenvolvido o Mobilis GTFS Creator, capaz de fazer a leitura dos vários ficheiros disponibilizados pela rodoviária do lis (como os períodos, feriados, horários, paragens e rotas) e de definir alguns valores que não se encontram disponibilizados nos ficheiro (como alguma informação acerca da companhia de transportes e os custos das viagens) para depois serem traduzidos e escritos para uma pasta de destino em formato GTFS. De forma a suportar todas estas funcionalidades, o Mobilis GTFS Creator foi criado em C#, como uma Windows Application. Com isto, é possível dotar a aplicação de aspeto gráfico e simplificar a tarefa de interação com o utilizador. Em termos de desenvolvimento da aplicação, a utilização do C# permitiu o fácil desenvolvimento gráfico da aplicação e da parte de técnica, visto a facilidade em trabalhar com ficheiros CSV, XML, utilização de listas, dicionários, boa ferramenta de *debugging* e por fim a escrita dos resultados num ficheiro ZIP. Através destas características, é possível ao utilizador da aplicação seleccionar facilmente os ficheiros da rodoviária do lis que pretende usar, escolher a pasta de destino, definir alguns dados extra (se desejar) como o custos de bilhetes entre outros e por fim mandar gerar os dados em formato GTFS, sendo colocado na pasta de destino os respetivos ficheiros txt e zip em formato GTFS. Visto o grande número de desafios, decidiu-se repartir estes por várias subsecções e assim melhorar a compreensão destes.

Estrutura de dados da Rodoviária do Lis e do GTFS

Como apresentado anteriormente, o GTFS é constituído por 6 ficheiros essenciais:

- `agency.txt` - possui dados acerca da companhia de transportes, como nome e URL do website da companhia;
- `calendar.txt` - possui dados acerca da regularidade dos serviços. Indica por exemplo, que uma dada linha de transporte funciona todos os dias da semana (segunda a sexta) entre de 1 de Janeiro de 2016 até 31 de Julho de 2016;

- routes.txt - possui os dados acerca das linhas, como o nome curto e nome;
- stops.txt - possui os dados acerca das paragens, como o nome da paragem e sua posição geoespacial (coordenadas GPS);
- stop_times.txt - possui os dados dos horários, como a hora de partida, chegada e nome da paragem a que se refere;
- trips.txt - tem como objetivo ligar os horários com as linhas e serviços. Interliga os ficheiros stop_times.txt, calendar.txt e routes.txt.

Também é importante referir que existem outros ficheiros opcionais úteis, como por exemplo:

- calendar_dates.txt - possui os dias de funcionamento extra ou de não-funcionamento dos serviços. Por exemplo, todos os feriados em que um dado serviço não funciona, deverão ser aqui registados. Para isso, é necessário indicar o serviço a que se refere, data e tipo de exceção (se é uma dia de funcionamento ou não)
- fare_attributes - caracteriza várias classes de pagamento, possuindo dados como a quantia a pagar e qual a moeda utilizada.
- fare_rules.txt - associa uma classe de pagamento a uma linha de transportes

Já os dados da rodoviária do lis são caracterizados pelos seguintes ficheiros:

- desc_paragens.csv : ficheiro com o código, número e nome da paragem usado para usar no aspeto visual
- feriados.csv : ficheiro com os feriados portugueses
- feriadosle.csv : ficheiro com os feriados municipais
- frequenciasecra.csv : ficheiro com o código para uma frequência semanal (Segunda a sexta, só às quartas, Segundas e sextas...)
- masterstagelist.xml : ficheiro com os números e nomes das paragens em formato abreviatura e as suas coordenadas GPS.
- periodosle.csv : ficheiro com o nome do período de funcionamento de um serviço (funcionando como código), data de início e data de fim desse período de funcionamento

- `servicos.xml` : Código das linhas dos autocarros, nome (Mobilis 1, Mobilis 2...), lista de paragens para cada linha com número e nome da paragem em formato abreviatura
- `internetmobilis.csv` : Ficheiro com os horários. Junta o código do `desc_paragens` para indicar paragem de origem e de destino, horário por paragens (também para origem e destino), código das frequências e períodos. Existem mais valores, mas sem grande importância.

Como é possível verificar, os dados são em grande parte semelhantes/traduzíveis para o formato GTFS e é possível estabelecer ligações entre os vários ficheiros disponíveis. No entanto, para gerar completamente os ficheiros em GTFS sem necessitar de campos extras na aplicação Mobilis GTFS Creator, seria necessário um ficheiro com os dados da rodoviária do lis (pelo menos nome da empresa, nome curto, página web, número de telefone) e outro ficheiro com o custo das tarifas. De forma a lidar com todos os estes dados, foram criadas listas para cada um dos ficheiros da rodoviária e selecionado apenas os campos mais importantes. Para evitar a escrita para os ficheiros GTFS sempre que se cria uma nova linha de informação, todos os dados são escritos primeiramente numa lista, dicionário ou hashset e posteriormente passados para um ficheiro de texto em formato GTFS. Convém também referir que como referenciado pela rodoviária do lis, nem todos os ficheiros são relacionáveis através de simples identificadores numéricos visto serem usados por diferentes sistemas (não foi indicado pela rodoviária do lis em detalhe que sistema é usado para gerir os dados dos transportes, para além de que é um servidor IBM AS400²⁰). No entanto, para conseguir a tradução para GTFS dos dados da rodoviária do lis, seriam necessárias várias ligações, pesquisas e tratamento de dados da rodoviária do lis, sendo apresentado de seguida o trabalho feito sobre estes para conseguir a sua tradução para GTFS.

Calendar.txt

Para conseguir criar o `calendar.txt` em GTFS, é necessário criar um ficheiro com a seguinte estrutura:

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
FULLW,1,1,1,1,1,0,0,20140101,20171231
WE,0,0,0,0,0,1,0,20140101,20171231
```

Figura 4.2: Estrutura do ficheiro `calendar.txt`

²⁰<http://search400.techtarget.com/definition/AS-400>

Para conseguir esta estrutura em GTFS, foi necessário utilizar o ficheiro frequenciasecra_20160913204201553.csv, com a seguinte estrutura:

```
0;Diariamente;SEGUNDA-FEIRA;SIM;;;
0;Diariamente;TERÇA-FEIRA;SIM;;;
0;Diariamente;QUARTA-FEIRA;SIM;;;
0;Diariamente;QUINTA-FEIRA;SIM;;;
0;Diariamente;SEXTA-FEIRA;SIM;;;
0;Diariamente;SÁBADO;SIM;;;
0;Diariamente;DOMINGO;SIM;;;
1;Domingos;DOMINGO;SIM;;;
2;AOS DOMINGOS (OU 2*S FEIRAS SE FERIADO);DOMINGO;Não;1;;;
3;AOS DOMINGOS E 2*S FEIRAS SE FERIADO;DOMINGO;Sim;1;;
3;AOS DOMINGOS E 2*S FEIRAS SE FERIADO;SEGUNDA-FEIRA;sim;1;;
```

Figura 4.3: Estrutura do ficheiro frequenciasecra_20160913204201553.csv

O ficheiro periodosle.csv, com a estrutura:

```
periodo;dataini;datafim;descweb
01/06 15/09;01/jun;15/set;01/06 a 15/09
01/06 30/09;01/jun;30/set;01/06 a 30/09
01/07 15/09;01/jul;15/set;01/07 a 15/09
01/07 31/08;01/jul;31/ago;01/07 a 31/08
01/07a11/09;01/jul;11/set;01/07 a 11/09
01/07a12/09;01/jul;12/set;01/07 a 12/09
01/07a13/09;01/jul;13/set;01/07 a 13/09
01/07a14/09;01/jul;14/set;01/07 a 14/09
01/07a15/09;01/jul;15/set;01/07 a 15/09
```

Figura 4.4: Estrutura do ficheiro periodosle.csv

E o ficheiro internetMobilis.csv, com a estrutura:

```
"101001","4242","4300","001","009","3","000","0.00","8.00","8.25","106","02","ida","01/09a30/06","000","190-LG.J.LÚCIO/GARE","157-CAMPUS 2 IPL"
"101001","4300","4242","009","019","3","000","0.00","8.25","8.40","106","02","ida","01/09a30/06","000","157-CAMPUS 2 IPL","190-LG.J.LÚCIO/GARE"
"101002","4242","4300","001","009","3","000","0.00","8.40","9.05","106","02","ida","01/09a30/06","000","190-LG.J.LÚCIO/GARE","157-CAMPUS 2 IPL"
"101002","4300","4242","009","019","3","000","0.00","9.05","9.20","106","02","ida","01/09a30/06","000","157-CAMPUS 2 IPL","190-LG.J.LÚCIO/GARE"
"101003","4242","4300","001","009","3","000","0.00","9.20","9.45","106","02","ida","01/09a30/06","000","190-LG.J.LÚCIO/GARE","157-CAMPUS 2 IPL"
"101003","4300","4242","009","019","3","000","0.00","9.45","10.00","106","02","ida","01/09a30/06","000","157-CAMPUS 2 IPL","190-LG.J.LÚCIO/GARE"
"101004","4242","4300","001","009","3","000","0.00","12.00","12.25","106","02","ida","01/09a30/06","000","190-LG.J.LÚCIO/GARE","157-CAMPUS 2 IPL"
"101004","4300","4242","009","019","3","000","0.00","12.25","12.40","106","02","ida","01/09a30/06","000","157-CAMPUS 2 IPL","190-LG.J.LÚCIO/GARE"
"101005","4242","4300","001","009","3","000","0.00","12.40","13.05","106","02","ida","01/09a30/06","000","190-LG.J.LÚCIO/GARE","157-CAMPUS 2 IPL"
"101005","4300","4242","009","019","3","000","0.00","13.05","13.20","106","02","ida","01/09a30/06","000","157-CAMPUS 2 IPL","190-LG.J.LÚCIO/GARE"
```

Figura 4.5: Estrutura do ficheiro internetMobilis.csv

Como é possível notar, os dados dos ficheiros períodos e frequências são algo semelhantes aos usados pelo GTFS e possíveis de serem traduzidos. Já o ficheiro internet-Mobilis tem todas as combinações de períodos e frequências utilizadas pelo Mobilis, logo cada uma destas deverá resultar pelo menos num serviço.

No entanto, existiu a necessidade de resolver alguns problemas, como

- Definição de um ID base para cada um dos serviços do calendar.txt

Verificando as estruturas acima, é possível reparar que tanto os ficheiros "periodosle.csv" como "frequenciasecra_20160913204201553.csv" não têm um ID em comum. A única relação entre estes dois ficheiros consiste no facto do ID de cada um deles aparecer no ficheiro internet_mobilis.csv. Isto significa que ao longo do internet_mobilis.csv vão surgindo várias combinações entre os IDs dos períodos e os IDs das frequências e cada uma dessas combinações irá consistir em pelo menos um serviço no calendar.txt. Visto que muitas destas combinações repetem-se ao longo do inter_mobilis.csv, foi usada uma hashset que descarta automaticamente valores repetidos. Para criar um ID para cada uma destas combinações no calendar.txt, foi usado o ID da frequência (por exemplo "106"), concatenado com o ID do período (por exemplo "01/09a30/06") e usando um separador com o valor "VV" entre eles, resultando num ID base com o nome "106VV01/09a30/06". No entanto, existem períodos com até quatro intervalos de data, logo, este simples ID não chegaria para identificar cada um dos serviços.

- ID de serviços semelhantes para diferentes intervalos de data.

No formato GTFS, todos os serviços têm apenas uma data de início e de fim. No entanto, como é possível verificar no ficheiro de periodosle.csv disponibilizado pela rodoviária do lis, há períodos que funcionam num intervalo de data, outros em dois intervalos de data e outros que funcionam em quatro intervalos de data. Para lidar com isto, o Mobilis GTFS Creator cria um serviço no calendar.txt para cada um destes intervalos, atribuindo a cada um deles um ID base (acima explicado como foi criado) e no final adiciona um separador "VV" seguido de um número para cada um dos intervalos. Isto resulta, em que para um dos serviços que use, por exemplo, o período "Período Esc", sejam criados os serviços no calendar.txt com os IDs "106VVFérias Esc.VV0", "106VVFérias Esc.VV1", "106VVFérias Esc.VV2", "106VVFérias Esc.VV3". Para o caso de serem dois intervalos, ele cria dois serviços a acabarem em "VV0" e "VV1".

- Tradução de frequência de dias em várias linhas, para um formato CSV

Para lidar com as frequências por dia da semana para um dado serviço, é guardado num dicionário cada um dos códigos do ficheiro frequenciasecra.csv e um array de 7 elementos representando cada um dos dias da semana. Este array é preenchido com 1 para dias da semana em que o serviço funciona e 0 nos dias em que não funciona. Visto no ficheiro de frequências poder ser usado tanto a designação "Segunda" como "2", o GTFS Mobilis Creator foi adaptado para lidar com ambas as situações.

- Dia 1 de Janeiro em vez de 31 de Dezembro

Como é possível verificar no ficheiro de períodos da rodoviária do lis, há referência a 1 de Janeiro como último dia de trabalho para o serviço com o código "Férias Esc.". Para lidar com esta situação, é feito a tradução de "01/jan" para "31/dez".

- Diferença de formatos de datas entre o GTFS e os dados da rodoviária do lis

Visto o GTFS utilizar um formato "aaaaMMdd" para o calendar.txt, foi necessário traduzir o formato utilizado nos períodos e ainda complementar com o ano a que se referem aqueles dados (o Mobilis GTFS Creator assume por defeito o ano corrente como o ano dos dados). Visto não existir um formato em português para os meses do ano como apresentado no ficheiro períodos, foi criada estrutura Switch que traduz os meses para números. Posteriormente, bastou criar uma string que concatenasse os 4 dígitos do ano, número do mês e por fim o dia do ano para serem usados no calendar.txt.

Calendar _dates.txt

Para conseguir criar o calendar _dates.txt em GTFS, é necessário criar um ficheiro com a seguinte estrutura:

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
FULLW,1,1,1,1,1,0,0,20140101,20171231
WE,0,0,0,0,0,1,0,20140101,20171231
```

Figura 4.6: Estrutura do ficheiro calendar _dates.txt

Para conseguir esta estrutura em GTFS, foi necessário utilizar o ficheiro feriados.csv, com a seguinte estrutura:

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
FULLW,1,1,1,1,1,0,0,20140101,20171231
WE,0,0,0,0,0,1,0,20140101,20171231
```

Figura 4.7: Estrutura do ficheiro feriados.csv

E o ficheiro feriadosle.csv, com a seguinte estrutura:

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
FULLW,1,1,1,1,1,0,0,20140101,20171231
WE,0,0,0,0,0,1,0,20140101,20171231
```

Figura 4.8: Estrutura do ficheiro feriadosle.csv

Visto a aplicação já possuir nesta altura os dados do `calendar.txt` estruturados numa lista, não houve a necessidade de leitura doutros ficheiros.

Como é possível notar, existe semelhanças nas estruturas mas há a necessidade de realizar alguns tratamentos nos dados de forma a conseguir os dados em formato GTFS.

Como tal, existiu a necessidade de resolver alguns desafios, como

- Ligação entre dos serviços do `calendar.txt` com os dias dos feriados

Para se conseguir criar uma linha no `calendar_dates.txt` é necessário saber que serviços são afetados pelo feriado assim como a sua data. Em relação às datas dos feriados, grande parte do problema já havia sido resolvido no desenvolvimento do ficheiro `calendar.txt`, podendo traduzir datas como "01/jan" para "20170101". Ao mesmo tempo, apesar dos ficheiros dos feriados também utilizarem datas em formato "dd/MM/aaaa", isto não representa um problema, visto ser um formato padrão. Através destes dois formatos de datas ("aaaaMMdd" e "dd/MM/aaaa") é possível através de várias funções C# conseguir comparar datas facilmente.

Desta forma, foi implementado um ciclo para cada um dos ficheiros dos feriados (`feriados.csv` e `feriadosle.csv`), que verifica se a data de um feriado se encontra entre a data de início e a data de fim de um dos serviços do ficheiro `calendar.txt`. No caso de isso se verificar é guardada numa lista o nome do serviço do `calendar.txt` em que o feriado se encontra e a data do feriado em formato "aaaaMdd". Posteriormente, os dados dessa lista são escritos no ficheiro `calendar_dates.txt`

Stops.txt

Para conseguir criar o `stops.txt` em GTFS, é necessário criar um ficheiro com a seguinte estrutura:

```

stop_id,stop_name,stop_desc,stop_lat,stop_lon,zone_id,stop_url
1_LJL,Largo José Lúcio da Silva,,39.746458,-8.804027,,
2_AHA,Avenida Heróis de Angola,,39.747928,-8.804115,,
3_MM_1,Mercado Municipal/Av.Cidade Maringá,,39.749772,-8.806574,,
4_NL_1,Nova Leiria,,39.752397,-8.808659,,
5_AG_1,Almoinha Grande,,39.754286,-8.811902,,
6_BA,Bairro das Almoínhas,,39.754148,-8.819742,,
7_EMP_1,Estádio Magalhães Pessoa,,39.750760,-8.812511,,

```

Figura 4.9: Estrutura do ficheiro stops.txt

Para conseguir esta estrutura em GTFS, foi necessário utilizar o ficheiro MasterStageList.xml, com a seguinte estrutura:

```

<ParagemInfo>
  <Nome>192-LG.J.LÚCIO/GARE</Nome>
  <Abrigo />
  <Postalete />
  <Coroa />
  <CoordenadasList>
    <Coordenada>
      <ID>0</ID>
      <Latitude>39.74680</Latitude>
      <Longitude>-8.80326</Longitude>
      <TolLat>18</TolLat>
      <TolLong>18</TolLong>
      <Sentido>IDA_E_VOLTA</Sentido>
    </Coordenada>
  </CoordenadasList>
  <EditDate>05-11-2015 12:26:30</EditDate>
</ParagemInfo>

```

Figura 4.10: Estrutura do ficheiro MasterStageList.csv

E o ficheiro desc_paragens2.csv, com a seguinte estrutura:

```

404;0 - Terminal Rodoviário
3306;4 - Marinheiros: Estrada dos Marinheiros
3307;5 - Marinheiros: Estrada dos Marinheiros
3308;6 - Marinheiros: R. 20 Junho/Igreja Marinheiros
3309;7 - Marrazes: R. das Oliveiras/X R. 20 Junho
3310;8 - Marrazes: Estrada de S. Tiago
3311;9 - Marrazes: Estrada de S. Tiago

```

Figura 4.11: Estrutura do ficheiro desc_paragens2.csv

Como é possível notar, existem os dados necessários à criação do stops.txt, mas será necessário tratamento dos dados para conseguir haver ligação entre os ficheiros masterstagelist.xml e desc_paragens2.csv.

Como tal, existiu a necessidade de resolver alguns desafios, como:

- Interligação dos ficheiros desc_paragens2.csv e MasterStageList.xml

Como é possível notar, não existe uma ligação direta entre o ficheiro desc_paragens2.csv e o MasterStageList.xml, visto este último não utilizar os códigos das paragens vindas do desc_paragens2.csv. Ao mesmo tempo, os nomes das paragens não são exatamente iguais, visto o desc_paragens2.csv utilizar o nome da paragem a ser mostrada ao utente dos transportes públicos e o MasterStageList.xml usar um nome abreviado dessa mesma paragem. No entanto, depois de confirmar junto da rodoviária do lis que podia ser feita interligação dos ficheiros desta forma, foi utilizado o código numérico presente em ambos os nome das paragens do desc_paragens2.csv e mastestagelist.xml como forma de ligação. Isto significa, por exemplo, que a paragem do desc_paragens2.csv "18 - Gândara: Esc. Afonso Lopes Vieira" corresponde à paragem "18-GÂNDARA ESC.A.LOPES" no masterstagelist.xml, graças ao código numérico "18". Ainda assim, foi necessário algum tratamento de dados neste campo, visto no desc_paragens2.csv ser usado um espaço em branco entre o número e o hífen e no masterstagelist.xml. De forma assegurar que o número lido do nome da paragem é o ID de ligação entre os ficheiros, foi mantido o hífen junto com o número (por exemplo, "18-"), visto este diferenciar de qualquer outro valor que possa estar presente no nome da paragem.

- Definição das coordenadas GPS

Com o problema da interligação dos dados resolvido, foi aberta a possibilidade de ligar as coordenadas GPS às paragens, sendo este o objetivo principal da ligação dos ficheiros desc_paragens2.csv e MasterStageList.xml. Para isso, foi criado um ciclo em que é verificado se existe cada um dos códigos numéricos vindos do desc_paragens2.csv, como por exemplo "192-", no campo "nome" dos vários "ParagemInfo" presentes no ficheiro masterstagelist.xml. No caso de haver o código numérico no campo "nome" do masterstagelist.xml, é verificado se ele possui coordenadas GPS e no caso de ter, estas são adicionadas à lista com os dados a serem exportados para o stop.txt. No caso de não ser possível encontrar determinada paragem no masterstagelist.xml ou no caso de a paragem não possuir as coordenadas GPS, são enviadas as coordenadas GPS com o valor "0" para a longitude e latitude.

- Diferenças entre os código do desc_paragens2.csv e do internetmobilis.csv

Apesar de à primeira vista os código serem iguais, no ficheiro desc_paragens2.csv, códigos inferiores a 1000 não possuem um 0 à esquerda mas no internetmobilis.csv possuem. Visto a lista através da qual o stops.txt é criado vir a ser utilizada na criação do ficheiro stop_times.txt, foi necessário corrigir esta situação, adicionando o

"0" à esquerda nos códigos das paragens do desc_paragens2.txt para valores abaixo dos "1000", passando assim paragens como a "404 a "0404".

Routes.txt

Para conseguir criar o routes.txt em GTFS, é necessário criar um ficheiro com a seguinte estrutura:

```
route_id,agency_id,route_short_name,route_long_name,route_desc,route_type,route_url,route_color,route_text_color
MOB1,RDT,Mobilis 1,Mobilis 1 - Linha Verde,,3,,,
MOB2,RDT,Mobilis 2,Mobilis 2 - Linha Vermelha,,3,,,
```

Figura 4.12: Estrutura do ficheiro routes.txt

Para conseguir esta estrutura em GTFS, foi necessário utilizar o ficheiro servicos.xml, com a seguinte estrutura:

```
<Servico>
  <Number>1011</Number>
  <Name>MOBILIS 1: Circular Leiria sentido Estação</Name>
  <LogicalRoute />
  <Dop>LEIRIA</Dop>
  <editDate>15-04-2016 16:35:39</editDate>
  <Stages>
    <Paragem>
      <Order>1</Order>
      <zoneOrder>1</zoneOrder>
      <Name>190-LG. J. LÚCIO/GARE</Name>
      <Urbana />
      <Km />
      <Pzona>SIM</Pzona>
    </Paragem>
    <Paragem>
      <Order>2</Order>
      <zoneOrder>2</zoneOrder>
      <Name>195-AV. HERÓIS ANGOLA</Name>
      <Urbana />
      <Km />
      <Pzona>SIM</Pzona>
    </Paragem>
    <Paragem>
      <Order>3</Order>
      <zoneOrder>3</zoneOrder>
      <Name>196-MERC.MUNICIPAL</Name>
      <Urbana>SIM</Urbana>
      <Km />
      <Pzona>SIM</Pzona>
    </Paragem>
    <Paragem>
      <Order>4</Order>
      <zoneOrder>4</zoneOrder>
      <Name>198-ARRABALDE PONTE</Name>
      <Urbana />
      <Km />
      <Pzona>SIM</Pzona>
    </Paragem>
  </Stages>
</Servico>
```

Figura 4.13: Estrutura do ficheiro servicos.xml

Como é possível notar, existem os dados necessários à criação do routes.txt, mas será necessário fazer a filtragem dos dados referentes ao Mobilis de forma a manter o ficheiro final apenas com a informação essencial.

Como é possível verificar acima, para criar um routes.txt será necessário o código de uma linha, um nome curto e um nome longo para a linha . Para isso, foram pesquisados todos os nós do ficheiro servicos.xml que possuam no nome a substring "MOB" e da informação disponível de cada desses nós encontrados, foi armazenado o número da linha que serviria como ID de linha no GTFS (por exemplo "1011") e o nome completo da linha (para o 1011 seria "MOBILIS 1: Circular Leiria sentido Estação") num dicionário.

Depois de completo o dicionário com todos os IDs e nomes de linha do ficheiro servicos.xml, bastou criar um ciclo onde para cada um dos valores do dicionário seria criada uma linha no ficheiro routes.txt, usando o ID de linha sem qualquer modificação ("1011"), criar e usar o nome curto da linha concatenando as primeiras três letras do nome completo da linha ("MOB") e a letra anterior aos dois pontos (":"), originando nomes curtos como por exemplo "MOB1", "MOB2" e "MOBR" e por fim, o nome completo para uma linha foi o mesmo que o nome do serviço no ficheiro servico.xml.

Stop_times.txt

Para conseguir criar o stop_times.txt em GTFS, é necessário criar um ficheiro com a seguinte estrutura:

```

trip_id,arrival_time,departure_time,stop_id,stop_sequence,stop_headsign,pickup_type,drop_off_type,shape_dist_traveled
MOB1D,7:00:00,7:00:00,1_LJL,1,,,,
MOB1D,7:01:00,7:01:00,2_AHA,2,,,,
MOB1D,7:03:00,7:03:00,3_MM_1,3,,,,
MOB1D,7:05:00,7:05:00,4_NL_1,4,,,,
MOB1D,7:07:00,7:07:00,5_AG_1,5,,,,
MOB1D,7:11:00,7:11:00,6_BA,6,,,,
MOB1D,7:14:00,7:14:00,7_EMP_1,7,,,,
MOB1D,7:15:00,7:15:00,8_PIS_1,8,,,,
MOB1D,7:16:00,7:16:00,9_A25A_1,9,,,,
MOB1D,7:19:00,7:19:00,10_ESDS_1,10,,,,
MOB1D,7:22:00,7:22:00,11_BC_1,11,,,,
MOB1D,7:23:00,7:23:00,12_CML_1,12,,,,
MOB1D,7:24:00,7:24:00,13_PM_1,13,,,,
MOB1D,7:26:00,7:26:00,14_ESTG,14,,,,
MOB1D,7:27:00,7:27:00,15_RESTG,15,,,,
MOB1D,7:31:00,7:31:00,16_PE,16,,,,
MOB1D,7:33:00,7:33:00,17_SAP_1,17,,,,
MOB1D,7:34:00,7:34:00,18_BS_1,18,,,,
MOB1D,7:36:00,7:36:00,19_AMP_1,19,,,,
MOB1D,7:37:00,7:37:00,20_ASE_1,20,,,,
MOB1D,7:38:00,7:38:00,21_ANSF_2,21,,,,
MOB1D,7:40:00,7:40:00,22_GUI_1,22,,,,
MOB1D,7:42:00,7:42:00,23_EPL,23,,,,
MOB1D,7:44:00,7:44:00,24_SR1,24,,,,
MOB1D,7:45:00,7:45:00,25_SR2_RF,25,,,,
MOB1D,7:46:00,7:46:00,26_SR3_CETL,26,,,,
MOB1D,7:48:00,7:48:00,27_HPVI,27,,,,
MOB1D,7:49:00,7:49:00,28_HPCE,28,,,,
MOB1D,7:50:00,7:50:00,29_ECML,29,,,,
MOB1D,7:52:00,7:52:00,30_BM1,30,,,,
MOB1D,7:53:00,7:53:00,31_PHR1,31,,,,
MOB1D,7:55:00,7:55:00,32_ULC1,32,,,,

```

Figura 4.14: Estrutura do ficheiro stop_times.txt

Para conseguir esta estrutura em GTFS, não foi necessário fazer a leitura de quaisquer ficheiros adicionais visto as estruturas essenciais já se encontrarem na aplicação.

Sendo este um dos ficheiros mais importantes para a criação dos dados GTFS, visto ser onde estão concentrados todos os horários, é naturalmente o mais complexo e complicado de traduzido. Logo, é na criação deste ficheiro que se concentram grande parte dos desafios do Mobilis GTFS Creator e seguramente onde se investiu mais recursos. Por essa razão, iremos discutir os vários desafios individualmente e como estes foram ultrapassados.

- Diferenças de um horário representado no sistema da rodoviário do lis e no GTFS

De facto, existe uma grande diferença na quantidade de informação necessária para definir os horários para um conjunto de paragens entre estes dois formatos, como se pode verificar comparando os valores da rodoviária do lis:

```

"101001","4242","4300","001","009","3","000"," 0.00"," 8.00"," 8.25","106","02","ida","01/09a30/06","000","190-LG.J.LÚCIO/GARE","157-CAMPUS 2 IPL"
"101001","4300","4242","009","019","3","000"," 0.00"," 8.25"," 8.40","106","02","ida","01/09a30/06","000","157-CAMPUS 2 IPL","190-LG.J.LÚCIO/GARE"

```

Figura 4.15: Estrutura do ficheiro internet_mobilis.csv para a *trip* 101001

Com os valores do GTFS:

```

101001106VV01/09a30/06VV0,8:00:00,8:00:00,4242,0,,,,
101001106VV01/09a30/06VV1,8:00:00,8:00:00,4242,0,,,,
101001106VV01/09a30/06VV0,,,3430,1,,,,
101001106VV01/09a30/06VV1,,,3430,1,,,,
101001106VV01/09a30/06VV0,,,3428,2,,,,
101001106VV01/09a30/06VV1,,,3428,2,,,,
101001106VV01/09a30/06VV0,,,3463,3,,,,
101001106VV01/09a30/06VV1,,,3463,3,,,,
101001106VV01/09a30/06VV0,,,3456,4,,,,
101001106VV01/09a30/06VV1,,,3456,4,,,,
101001106VV01/09a30/06VV0,,,3448,5,,,,
101001106VV01/09a30/06VV1,,,3448,5,,,,
101001106VV01/09a30/06VV0,,,3450,6,,,,
101001106VV01/09a30/06VV1,,,3450,6,,,,
101001106VV01/09a30/06VV0,,,3517,7,,,,
101001106VV01/09a30/06VV1,,,3517,7,,,,
101001106VV01/09a30/06VV0,8:25:00,8:25:00,4300,8,,,,
101001106VV01/09a30/06VV1,8:25:00,8:25:00,4300,8,,,,
101001106VV01/09a30/06VV0,,,3472,9,,,,
101001106VV01/09a30/06VV1,,,3472,9,,,,
101001106VV01/09a30/06VV0,,,3461,10,,,,
101001106VV01/09a30/06VV1,,,3461,10,,,,
101001106VV01/09a30/06VV0,,,3462,11,,,,
101001106VV01/09a30/06VV1,,,3462,11,,,,
101001106VV01/09a30/06VV0,,,3466,12,,,,
101001106VV01/09a30/06VV1,,,3466,12,,,,
101001106VV01/09a30/06VV0,,,3467,13,,,,
101001106VV01/09a30/06VV1,,,3467,13,,,,
101001106VV01/09a30/06VV0,,,3469,14,,,,
101001106VV01/09a30/06VV1,,,3469,14,,,,
101001106VV01/09a30/06VV0,,,3432,15,,,,
101001106VV01/09a30/06VV1,,,3432,15,,,,
101001106VV01/09a30/06VV0,,,3431,16,,,,
101001106VV01/09a30/06VV1,,,3431,16,,,,
101001106VV01/09a30/06VV0,,,3430,17,,,,
101001106VV01/09a30/06VV1,,,3430,17,,,,
101001106VV01/09a30/06VV0,8:40:00,8:40:00,4242,18,,,,
101001106VV01/09a30/06VV1,8:40:00,8:40:00,4242,18,,,,

```

Figura 4.16: Estrutura do ficheiro stop_times.txt para a *trip* 101001

Como se pode verificar, o ficheiro stop_times.txt apesar de não ser tão complexo por linha, possui um número de linhas muito superior. Ao mesmo tempo, o que à primeira vista pode parecer repetição de informação no stop_times.txt, consiste em horários para dois intervalos de datas diferentes (os IDs possuem um valor diferente no final). Este grande aumento no número de linhas também acontece porque o GTFS precisa de saber para cada *trip id* quais as paragens todas onde este passa. Convém também referir que apesar do stop_times.txt possuir paragens sem horários, isto não representa um problema para o OpenTripPlanner visto este calcular um horário médio para todas as paragens sem horário.

- Preenchimento de um horário

Como se pode verificar pela diferença de estruturas entre os horários da rodoviária do lis e o GTFS, o preenchimento dos horários não seria algo direto ou simples. Para resolver este problema, foi construído uma estrutura (neste caso dicionário) que armazenaria desde logo todas as paragens para um dado serviço (por exemplo, o 1010) e depois à medida que ia lendo as horas dos horários da rodoviária do lis, ia preenchendo os horários nessa estrutura. Posteriormente, quando se mudava de viagem/trip, todos

os dados deste dicionário eram passados para estrutura que ia ter todos os dados do `stop_times.txt`. Para os casos em que a viagem/*trip* era um circuito (havia uma paragem que se repetia) foi necessário diferenciar o ID da paragem que se repetia de forma a não gerar um erro de repetição de ID e ao mesmo tempo, indicasse que era uma paragem repetida porque era normal haver um horário definido para essa última paragem. A solução encontrada para este problema foi adicionar alguns caracteres extra ao ID da paragem repetida e para os casos em que era preciso definir um horário para esta segunda paragem, antes era verificado se o horário da primeira estava já preenchido. No caso de estar, o horário era definido para a segunda paragem.

- Construção de um ID por conjunto de paragens

Apesar de o ficheiro `internet_mobilis.csv` da rodoviária do lis já possuir um ID, este não poderia ser usado diretamente no `stop_times.txt`. Isto porque para além de existirem períodos que funcionam em vários intervalos de datas (cada intervalo precisa de ter um *Trip ID* diferente), também existem situações no `internet_mobilis.txt` em que o mesmo ID possui frequências diferentes (caso do 101307 com frequências 106 e 107). Para resolver este problema, cada `tripID` é constituído pelo ID do `internet_mobilis.csv`, ID da frequência, ID do período e por um valor a identificar o intervalo de data. Para separar os vários IDs, foram usado os caracteres "VV" como separadores visto ser um conjunto de letras que naturalmente não se iriam repetir.

- Horas para lá da meia-noite

No caso de os horários passarem da meia-noite, o estrutura GTFS define que horário deverá passar a ser 24 horas mais o número de desejadas (2 da manhã neste caso seria 26 horas). Visto o Mobilis não possuir horários a partirem antes da 5 manhã, foi definido que se as horas de um horário for menor que 5, deverá ser somado a esse valor 24 horas.

- Conjuntos de paragens com frequências diferentes mas com o mesmo ID base de horário

Apesar de não ser algo muito comum, os `internet_mobilis` possui alguns horários em que para o mesmo ID, existem duas frequências diferentes, como se pode verificar na figura seguinte:

```

"101307", "4337", "3335", "022", "028", "3", "000", " 0.00", "12.15", "12.22", "106", "05", "ida", "", "000", "191-LG. J. LÚCIO/GARE", "35-C. AREIA COL. C.M. I"
"101307", "4337", "3335", "022", "028", "3", "000", " 0.00", "12.15", "12.22", "107", "05", "ida", "", "000", "191-LG. J. LÚCIO/GARE", "35-C. AREIA COL. C.M. I"
"101307", "4337", "3349", "022", "035", "3", "000", " 0.00", "12.15", "12.30", "106", "05", "ida", "", "000", "191-LG. J. LÚCIO/GARE", "49-TELHEIRO R. IM.127"
"101307", "4337", "3349", "022", "035", "3", "000", " 0.00", "12.15", "12.30", "107", "05", "ida", "", "000", "191-LG. J. LÚCIO/GARE", "49-TELHEIRO R. IM.127"
"101307", "3335", "3349", "028", "035", "3", "000", " 0.00", "12.22", "12.30", "106", "05", "ida", "", "000", "35-C. AREIA COL. C.M. I", "49-TELHEIRO R. IM.127"
"101307", "3335", "3349", "028", "035", "3", "000", " 0.00", "12.22", "12.30", "107", "05", "ida", "", "000", "35-C. AREIA COL. C.M. I", "49-TELHEIRO R. IM.127"

```

Figura 4.17: Horários com o mesmo ID mas frequência diferentes

Visto o ID criado para o stop_times.txt já conseguir lidar com esta situação, o processo de tradução do horário para GTFS não, porque o Mobilis GTFS Creator assumia que todas as linhas no ficheiro internet_mobilis.csv com o mesmo ID pertenciam ao mesmo horário com a mesma frequência e período. Para resolver este problema, foi criada uma lista que guarda nela as frequências que ocorriam num conjunto de linhas com o mesmo ID. Para o exemplo da imagem acima, seriam guardadas as frequências "106" e "107". Como nestes casos os horários são os mesmos, depois de ser escrito o horário para a frequência "106", era escrita a "107".

- Partidas do mesmo autocarro para múltiplos destinos ao mesmo tempo na mesma linha

Outro dos desafios no stop_times.txt consistia em horários para o mesmo autocarro (mesmo ID) com a mesma origem e destinos diferentes, como se pode verificar na figura seguinte:

```

"101301", "3331", "3330", "001", "002", "3", "000", " 0.00", " 6.40", " 6.42", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "29-PTE. PEDRA N109 (X)"
"101301", "3331", "3327", "001", "003", "3", "000", " 0.00", " 6.40", " 6.44", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "26-GÂNDARA X CEMITÉR"
"101301", "3331", "3325", "001", "004", "3", "000", " 0.00", " 6.40", " 6.46", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "24-GÂNDARA CAPELA GD"
"101301", "3331", "3324", "001", "005", "3", "000", " 0.00", " 6.40", " 6.48", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "23-GÂNDARA (X) AMOR"
"101301", "3331", "3322", "001", "006", "3", "000", " 0.00", " 6.40", " 6.50", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "21-GÂNDARA (X) BAQUEL."
"101301", "3331", "3320", "001", "007", "3", "000", " 0.00", " 6.40", " 6.52", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "19-GÂNDARA ESC.AF.LP"
"101301", "3331", "3319", "001", "008", "3", "000", " 0.00", " 6.40", " 6.54", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "17-MARRAZES R. S. TIRO"
"101301", "3331", "3317", "001", "009", "3", "000", " 0.00", " 6.40", " 6.55", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "15-MARRAZES B. S. CARN"
"101301", "3331", "3496", "001", "010", "3", "000", " 0.00", " 6.40", " 6.56", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "215-MARRAZES CT. SAÚD"
"101301", "3331", "3315", "001", "011", "3", "000", " 0.00", " 6.40", " 6.57", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "13-MARRAZES JUNTA F."
"101301", "3331", "3312", "001", "012", "3", "000", " 0.00", " 6.40", " 6.58", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "10-MARRAZES X CEMIT."
"101301", "3331", "3311", "001", "013", "3", "000", " 0.00", " 6.40", " 6.59", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "9-MARRAZES ES. S. TIAG"
"101301", "3331", "3332", "001", "014", "3", "000", " 0.00", " 6.40", " 7.00", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "31-MARRAZES R. OLIVEI"
"101301", "3331", "3308", "001", "015", "3", "000", " 0.00", " 6.40", " 7.02", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "6-MARINHEIROS IGREJA"
"101301", "3331", "3307", "001", "016", "3", "000", " 0.00", " 6.40", " 7.03", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "5-MARINHEIROS ES.MA."
"101301", "3331", "3399", "001", "017", "3", "000", " 0.00", " 6.40", " 7.05", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "103-Q. BISPO E.MA/T.O"
"101301", "3331", "3397", "001", "018", "3", "000", " 0.00", " 6.40", " 7.06", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "101-Q. BISPO E.MA/ISS"
"101301", "3331", "3333", "001", "019", "3", "000", " 0.00", " 6.40", " 7.07", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "32-Q. BISPO RUA MATIN"
"101301", "3331", "3431", "001", "020", "3", "000", " 0.00", " 6.40", " 7.08", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "138-R. CAF. MOUZ. ALBUQ"
"101301", "3331", "3430", "001", "021", "3", "000", " 0.00", " 6.40", " 7.09", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "137-LG. CÔNEG. MAIA/SÉ"
"101301", "3331", "4337", "001", "022", "3", "000", " 0.00", " 6.40", " 7.10", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "191-LG. J. LÚCIO/GARE"
"101301", "3331", "3428", "001", "024", "3", "000", " 0.00", " 6.40", " 7.13", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "135-MERCADO SANT'ANA"
"101301", "3331", "3463", "001", "025", "3", "000", " 0.00", " 6.40", " 7.14", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "177-LG. DA REPÚBLICA"
"101301", "3331", "3457", "001", "026", "3", "000", " 0.00", " 6.40", " 7.15", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "169-BAIRRO SARGENTOS"
"101301", "3331", "4299", "001", "027", "3", "000", " 0.00", " 6.40", " 7.16", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "33-FRISAÇÃO ESCOLA 1"
"101301", "3331", "3335", "001", "028", "3", "000", " 0.00", " 6.40", " 7.17", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "35-C. AREIA COL. C.M. I"
"101301", "3331", "3337", "001", "029", "3", "000", " 0.00", " 6.40", " 7.18", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "37-C. AREIA R. D. JOSÉ"
"101301", "3331", "3339", "001", "030", "3", "000", " 0.00", " 6.40", " 7.19", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "39-C. AREIA R. LINO A."
"101301", "3331", "3341", "001", "031", "3", "000", " 0.00", " 6.40", " 7.20", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "41-C. AREIA R. LIN. ANT"
"101301", "3331", "3343", "001", "032", "3", "000", " 0.00", " 6.40", " 7.22", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "43-TELHEIRO QUARTEL"
"101301", "3331", "3345", "001", "033", "3", "000", " 0.00", " 6.40", " 7.23", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "45-TELHEIRO ESC. PRIM"
"101301", "3331", "3348", "001", "034", "3", "000", " 0.00", " 6.40", " 7.24", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "48-TELHEIRO (X) R. PAZ"
"101301", "3331", "3349", "001", "035", "3", "000", " 0.00", " 6.40", " 7.25", "106", "01", "ida", "", "000", "30-PTE. PEDRA X MATOE", "49-TELHEIRO R. IM.127"

```

Figura 4.18: Horários com o mesmo ID partidas iguais e chegadas diferentes

Visto ser algo estranho, esta questão foi verificada junto da rodoviária do lis e visto o sistema deles necessitar destas linhas para funcionar, foi assumido (pela rodoviária do

lis) que o Mobilis GTFS Creator também deveria conseguir lidar com o problema. Para resolver este desafio, foi então implementado um sistema que lia e armazenava o destino da primeira linha no ficheiro `internet_mobilis.csv` para determinado ID e comparava à origem das linhas seguinte. No caso de a linha ter uma origem igual ao valor de destino anteriormente gravado, significava que essa linha no `internet_mobilis.csv` era a que devia ser interpretada. No caso de ser diferente, ele continuava para a linha seguinte do ficheiro `internet_mobilis.csv`.

Trips.txt

Para conseguir criar o `trips.txt` em GTFS, é necessário criar um ficheiro com a seguinte estrutura:

```
route_id,service_id,trip_id,trip_headsign,direction_id,block_id,shape_id
MOB1,FULLW,MOB1D,,0,,
MOB1,FULLW,MOB1D2,,0,,
MOB1,FULLW,MOB1D3,,0,,
MOB1,FULLW,MOB1D4,,0,,
MOB1,FULLW,MOB1D5,,0,,
MOB1,FULLW,MOB1D6,,0,,
MOB1,FULLW,MOB1D7,,0,,
MOB1,FULLW,MOB1D8,,0,,
MOB1,FULLW,MOB1D9,,0,,
MOB1,FULLW,MOB1D10,,0,,
MOB1,FULLW,MOB1D11,,0,,
MOB1,FULLW,MOB1D12,,0,,
MOB1,FULLW,MOB1D13,,0,,
MOB1,FULLW,MOB1D14,,0,,
```

Figura 4.19: Estrutura do ficheiro `trips.txt`

Para conseguir esta estrutura em GTFS, não foi necessário fazer a leitura de quaisquer ficheiros adicionais visto as estruturas essenciais já se encontrarem na aplicação.

Como descrito anteriormente, este ficheiro tem como objetivo ligar os horários criados no `stop_times.txt` com as frequências de um serviço disponibilizado no ficheiro `calendar.txt` e com uma linha do `routes.txt`. Para isso foi utilizada a lista de horários anteriormente criada a partir do ficheiro `internet_mobilis.txt` e o hashset com todos os dados estruturados do ficheiro `calendar.txt`. Desta forma, utilizando estas duas estruturas, era possível interligar os ficheiros para criar o ficheiro `trips.txt`.

Como é possível verificar na estrutura GTFS, serão precisos dados como o ID da linha, o ID do serviço no `calendar` e o ID Trip que já há-de estar disponível no ficheiro `stop_times.txt`.

Para isso foram criados dois ciclos, um para a lista de horários do `internet_mobilis.txt`

e outra para o hashset do calendar.txt. Através destes, é possível definir que serviços no calendar.txt estão associados a cada um dos valores da lista de horários e assim preencher um hashset organizado com todos os valores que irão ser utilizados no ficheiro GTFS trips.txt. Visto o ID da linha estar representado tanto no ID do calendar.txt como no ID do internet_mobilis.txt, isto não representou qualquer problema.

Posteriormente, estes valores são escrito no ficheiro trips.txt assim como alguns outros valores constantes ao longo das várias linhas. Tudo isto resulta num ficheiro com uma estrutura e valores como estes:

```
1010,106VV01/09a30/06VV0,101001106VV01/09a30/06VV0,,0,,
1010,106VV01/09a30/06VV1,101001106VV01/09a30/06VV1,,0,,
1010,106VV01/09a30/06VV0,101002106VV01/09a30/06VV0,,0,,
1010,106VV01/09a30/06VV1,101002106VV01/09a30/06VV1,,0,,
1010,106VV01/09a30/06VV0,101003106VV01/09a30/06VV0,,0,,
1010,106VV01/09a30/06VV1,101003106VV01/09a30/06VV1,,0,,
1010,106VV01/09a30/06VV0,101004106VV01/09a30/06VV0,,0,,
1010,106VV01/09a30/06VV1,101004106VV01/09a30/06VV1,,0,,
```

Figura 4.20: Estrutura do ficheiro stops.txt

Agency.txt

Para conseguir criar o agency.txt em GTFS, é necessário criar um ficheiro com a seguinte estrutura:

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
FULLW,1,1,1,1,1,0,0,20140101,20171231
WE,0,0,0,0,0,1,0,20140101,20171231
```

Figura 4.21: Estrutura do ficheiro stops.txt

Para conseguir esta estrutura em GTFS, não foi necessário fazer a leitura de quaisquer ficheiros adicionais visto estes dados não nos ter sido fornecido na altura pela rodoviária do lis.

Para lidar com este problema, o Mobilis GTFS Creator oferece a possibilidade de se preencher todos os dados necessários na janela "Companhia de transportes..." acessível ao clicar no botão "Companhia de transportes..." na janela principal da aplicação. Visto estes dados serem em grande parte constantes, no caso de o utilizador não os preencher no interface, o Mobilis GTFS Creator irá criar o ficheiro GTFS agency.txt com os seguintes valores:

```
agency_id,agency_name,agency_url,agency_timezone,agency_phone
RDL,Rodoviária do Lis,http://www.mobilis.pt/, Europe/Lisbon,244811507
```

Figura 4.22: Estrutura resultante de origem para o Agency.txt

Fare_rules.txt e fare_attributes.txt

Para conseguir criar o ficheiro fare_rules.txt, neste caso bastará colocar apenas o cabeçalho do GTFS.

E o ficheiro fare_attributes.txt com a seguinte estrutura:

```
fare_id,price,currency_type,payment_method,transfers,transfer_duration
p,4.80,EUR,1,0,
u,1.25,EUR,0,0,
d,2.00,EUR,0,0,
s,7.00,EUR,1,0,
m,15.55,EUR,1,0,
r,11.65,EUR,1,0,
```

Figura 4.23: Estrutura do ficheiro fare_attributes.txt

Para conseguir esta estrutura em GTFS, não foi necessário fazer a leitura de quaisquer ficheiros adicionais visto estes dados não nos ter sido fornecido na altura pela rodoviária do lis.

Para lidar com este problema, o Mobilis GTFS Creator oferece a possibilidade de se preencher todos os dados necessários na janela "Outros" acessível ao clicar no botão "Outros..." na janela principal da aplicação. Visto estes dados serem em grande parte constantes, no caso de o utilizador não os preencher no interface, o Mobilis GTFS Creator irá criar os ficheiros fare_rules.txt (apenas cabeçalho) e fare_attributes.txt com os seguintes valores:

```
fare_id,price,currency_type,payment_method,transfers,transfer_duration
m,1.30,EUR,0,0,
```

Figura 4.24: Estrutura resultante de origem para o Fare_attributes.txt

4.4 Conclusão

Como é possível verificar ao longo deste capítulo, foram vários os desafios de implementação e integração das várias plataformas. No entanto, também é possível notar o processo de evolução das plataformas e aplicações à medida que foram sendo resolvidos os vários problemas. Inicialmente, apesar de haverem dados em formato GTFS que

conseguiam ser lidos e utilizados pelo OpenTripPlanner, estes eram praticamente inúteis numa situação real visto a hora de chegada de um autocarro a uma paragem variar consoante a hora a que era feito o planeamento e não estarem disponíveis os horários por paragem. Só depois da reestruturação dos ficheiros GTFS executada durante este trabalho, foi possível a plataforma OpenTripPlanner funcionar corretamente e ao mesmo tempo, suportar horários com tempos entre conjuntos de viagens variável, resultantes da expansão do mobilis em 2016. No entanto, também a plataforma OpenTripPlanner necessitou de ser atualizada ao longo do projeto para suportar as alterações feitas sobre o MapQuest do OpenStreetMap a meio deste trabalho e ainda traduzida para Português de forma a ser usada por uma maior quantidade de pessoas em Portugal. A aplicação OpenTripPlanner for Android também foi outra das aplicações que foi evoluindo bastante ao longo deste projeto à medida que foram sendo resolvidos os vários desafios, começando inicialmente por ser uma aplicação que não suportava as versões do OpenTripPlanner mais recentes e com um interface e forma de trabalhar vinda do Android 2.3.3. No final, esta aplicação veio a tornar-se no Mobilis Trip Planner, capaz de se conectar a versões do OpenTripPlanner mais recentes, possuir um visual e forma de trabalhar mais semelhante a aplicação Android mais recentes, assim como ainda foi possível adicionar a capacidade de visualizar os horários por paragem, uma funcionalidade bastante comum e útil disponibilizada por outras aplicações desta área. Não menos importante, foi o desenvolvimento do Template Excel que permite criar e atualizar com maior facilidade a variada informação acerca dos transportes públicos em formato GTFS, algo bastante útil para qualquer administrador que tenha sido encarregue de criar e manter os dados de uma companhia de transportes. Já Mobilis GTFS Creator foi criado a pensar na rodoviária do lis, com o objetivo de traduzir para GTFS os dados em formato digital que estes já possuíam nos seus servidores. Naturalmente, foi uma aplicação que apresentou bastantes desafios e de natureza complexa mas que permite em poucos minutos gerar um ficheiro compactado com toda a informação dos horários, paragens e linhas do Mobilis pronto a ser utilizado no OpenTripPlanner quase sem ser necessária interação humana. Por fim, é interessante verificar que todas estas aplicações e plataformas trabalham em conjunto para serem capazes de fornecer uma solução capaz de não só auxiliar à criação e atualização de informação acerca dos transportes públicos em formato digital, assim como depois os difunde pela *Internet* e ajuda os vários utentes destes transportes a tirar o melhor partido deles, cumprindo assim em termos de funcionalidades e características os objetivos apontados inicialmente na introdução.

Capítulo 5

Avaliação e Testes

Neste capítulo são apresentados os testes da aplicação Mobilis Trip Planner realizados junto de possíveis utilizadores, definição das tarefas lhes foram atribuídas, resultados e correções. Por fim, na secção de "Conclusão", será feita uma análise geral sobre esta fase e que conclusões se puderam retirar.

5.1 Fases de teste

Para testar a aplicação Mobilis Trip Planner, foram feitas duas fases de testes:

- Testes preliminares;
- Testes com utilizadores.

Os testes preliminares foram realizados com o objetivo de perceber que problemas o Mobilis Trip Planner poderia apresentar, que funcionalidades seriam interessantes de serem exploradas num teste final e se possível, fazer algumas correções antes dos testes principais. Para isso, o número de pessoas envolvidas neste teste foi baixa (3 pessoas), não foi utilizado um guião onde poderíamos de forma organizada anotar os resultados e também não foi bem definido o grupo de tarefas a serem testadas (consistia mais num grupos de funcionalidades a testar) Para os testes com os utilizadores, aumentou-se o número de utilizadores (para 8) junto dos quais foi testada a aplicação Mobilis Trip Planner, já havia um grupo de tarefas bem definidas que seriam interessantes de testar e um guião que permitia anotar os tempos e erros realizados pelos utilizadores ao testar a aplicação assim como as suas sugestões e observações.

5.2 Testes preliminares

Em geral, todos os utilizadores reconheceram que esta aplicação seria uma mais valia para o Mobilis e para Leiria em termos de utilização dos transportes, visto através do planeador conseguirem obter resposta de que autocarros e paragens utilizar para chegarem a determinado destino e ao mesmo tempo, verificar quais os horários de uma determinada paragem. Convém também referir que, no geral, a funcionalidade recebida com mais entusiasmo foi o planeamento de rotas. Apesar de não terem sido definidas um grupo de tarefas a executar pelo utilizador, foram vistas funcionalidades como:

- Ver os horários das paragens;
- Planeamento de rotas e alteração das horas e datas a que estes planos eram realizados;
- Verificar a descrição do planeamento de uma rota.

Em termos de características da utilização dada à aplicação por parte dos utilizadores, foi possível verificar que:

- Visto a aplicação arrancar na página de planeamento, ao ser pedido para ver os horários das paragens, foi possível notar que houve alguma hesitação inicial a conseguir perceber onde conseguiam aceder estas mas mesmo assim, em pouco tempo perceberam que tinham de ir para a página das "paragens" através do menu
- O símbolo do Mobilis como indicador de paragem não apresentou dificuldade para os utilizadores, com estes a clicarem nele em pouco tempo para depois verem os horários da paragem
- Em relação à forma de pedir o planeamento de rota, foi possível verificar que os utilizadores fizeram um maior uso das textboxes e GPS que do mapa. Dos testes realizados, se o primeiro planeamento fosse usado o GPS, ao ser pedido um segundo planeamento sem usar GPS, utilizavam apenas as textboxes. Ao mesmo tempo, o contrário também se aplicava: se no primeiro usavam apenas textboxes, no segundo usavam GPS;
- Alguma dificuldade em achar o botão para alterar a data/hora de um planeamento de rota.

No final de se verificar que o utilizador havia testado o grupo de funcionalidades apontadas acima, era feita a anotação de várias sugestões/observações e ao mesmo tempo, o utilizador poderia testar a aplicação à vontade. Por acaso, nesta última fase foi sempre pedido aos utilizadores para ver se conseguiam visualizar a descrição do planeamento de uma rota. Ao fazer este pedido, foi possível verificar que havia um período inicial em que os utilizadores ficavam sem saber onde clicar (apesar de em pouco tempo conseguirem chegar à resposta) e também foi possível que grande parte das visualizações da descrição da rota foi feita através do clique da janela por cima do planeamento da rota. Também foi possível verificar nesta fase de exploração livre que os utilizadores tinham tendência para clicar para ver as rotas alternativas a um plano. Em termos de sugestões/observações da aplicação, foi possível verificar e até confirmado pelos utilizadores, que o botão para mudança de data/hora no planeamento era pouco claro em termos visuais da sua funcionalidade devido em parte pelo ícone e sítio onde este se encontrava. Um dos utilizadores inclusive, veio clicar neste botão quando foi pedido para ver os horários de uma paragem (contando assim como erro) mas que por acaso, veio acelerar depois o processo de mudar as datas/horas do planeamento para esse mesmo utilizador. Outra tendência interessante foi ao entrar na atividade que tinha as "paragens", onde na altura estavam todas as paragens a serem colocadas no ecrã, ouvir os utilizadores comentarem "Tanta paragem!". Em termos de sugestões para a aplicação, um dos utilizador achou que ao se clicar numa paragem da atividade "paragens", em vez de aparecer no fundo do ecrã o nome da paragem e o botão para ver as horas, poderia aparecer essa informação num pequeno balão ao clicar sobre a paragem, assim como qual era o próximo autocarro a parar naquela paragem e poder definir a mesma como local de origem ou destino na parte de planeamento de rotas. Por fim, ele também indicou que acharia mais correto a atividade "planeamento" e "paragens" serem a mesma, concentrando todas as funcionalidades no mesmo mapa.

Nesta fase de testes preliminares, a aplicação foi testada com duas pessoas do sexo masculino abaixo dos 18 anos e outra entre os 30 e os 50 anos também do sexo masculino, todos utilizadores do Mobilis. Os testes foram feitos utilizando um *smartphone* Samsung Galaxy S5 New, durante a tarde e foi testada perto da igreja dos Marrazes com um dos utilizadores com menos de 18 anos e com o utilizador na escala dos [30,50] anos. Com o outro utilizador, abaixo dos 18 anos, o teste foi feito no Largo José Lúcio em Leiria, também durante a tarde.

5.2.1 Correção do Mobilis Trip Planner

Infelizmente, o tempo disponível para correções depois dos testes preliminares e os principais foi limitado, não permitindo grandes alterações na aplicação. Ainda assim, na atividade "paragens", deixaram de ser colocadas todas as paragens, para serem apenas apresentadas as do Mobilis 1. Outra alteração feita na aplicação, por iniciativa própria, foi o fecho da atividade "paragens" quando se volta à página "planeamento" em vez de se estar a criar uma nova atividade. Desta forma, é possível evitar que se esteja constantemente a criar atividades "planeamento" e manter acessível algum planeamento de rota realizado nesta.

5.3 Testes com utilizadores

Há semelhança com a fase de testes preliminares, os utilizadores com que testámos o Mobilis Trip Planner continuam a achar uma mais valia para o Mobilis e para Leiria. Todas as tarefas foram cumpridas pelos utilizadores e mesmo em casos que a aplicação apresentasse algum erro devido a uma rota mal planeada, os utilizadores percebiam o erro e eram capazes de corrigir o problema. Como nos testes preliminares, as pessoas demonstraram mais entusiasmo pelo planeador de rotas em grande parte por facilmente descobrirem como ir para determinado local e a que horas. Para esta fase de testes, já com um guião de tarefas melhor definido, foi pedido para:

- Ver os horários de duas paragens diferentes;
- Mudar de linha a ser mostrada na atividade "paragens";
- Fazer dois planeamentos de rota diferentes e um terceiro planeamento com horas/data diferentes da atual;
- Verificar a descrição do planeamento de uma rota;

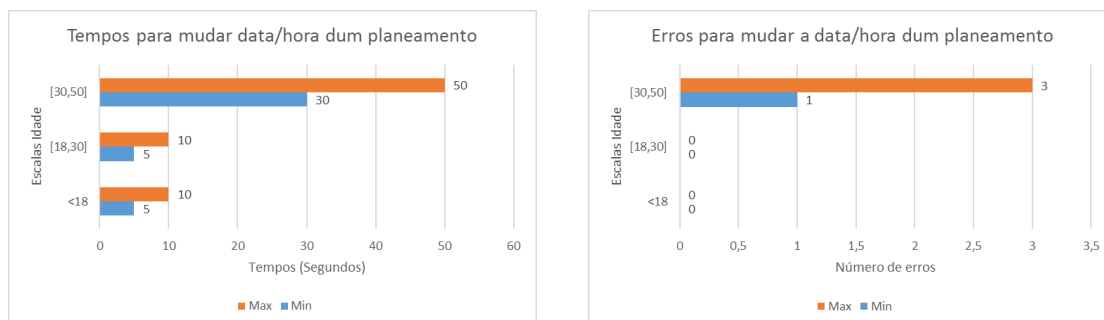
No final destas tarefas, era dada a possibilidade de o utilizar explorar à sua vontade a aplicação enquanto eram discutidas as observações/sugestões da aplicação. Estão disponibilizados no anexo E os resultados de tempos e erros para uma destas funcionalidades.

Durante a execução das tarefas acima por parte dos utilizadores, foi possível verificar que:

- Da mesma forma que nos testes preliminares, ao ser pedido para ver os horários das paragens, foi possível notar que houve alguma hesitação inicial a conseguir perceber onde conseguiam aceder estas mas mesmo assim, em pouco tempo perceberam que tinham de ir para a página das "paragens" através do menu;
- O símbolo do Mobilis como indicador de paragem continuou a não apresentar uma grande dificuldade para os utilizadores, com estes a clicarem nele em pouco tempo para depois verem os horários da paragem
- Em termos de forma de pedir o planeamento de rota, foi possível verificar que os utilizadores fizeram um maior uso das textboxes e GPS que do mapa. Dos testes realizados, se o primeiro planeamento fosse usado o GPS, ao ser pedido o segundo planeamento usando outra forma de o planear, a escolha seguinte seria usar apenas textboxes. Ao mesmo tempo, o contrário também se aplicava: se no primeiro usavam apenas textboxes, no segundo usavam GPS.
- Visto ter-se deixado de apresentar todas as paragens ao ser pedido para ver os horários destas, os utilizadores já não fizeram o comentário de haverem paragens a mais no ecrã.
- O botão para mudar a data/hora continuou a ser problemático;
- O spinner/dropdown menu para a mudança de linha na atividade de "paragens" não estava muito perceptível.
- A forma mais utilizada para aceder à descrição da rota continua a ser clicar no balão descritivo do planeamento da rota

Como se pode notar, grande parte dos problemas dos testes preliminares mantêm-se nesta fase e surgiram outros. Algo que pode ser visto com naturalidade dado o maior número de pessoas a testarem a aplicação. Em termos de observações/sugestões feitas pelos utilizadores, algumas delas mantêm-se, como é o caso do botão de mudança de data/hora do planeamento de rota continuar a ser pouco representativo da sua funcionalidade havendo por exemplo, um utilizador que pensasse que este consistia num histórico de planeamentos de rota executados e outra utilizador que pensou que de alguma forma este botão estivesse relacionado com o campo de destino no planeamento de uma rota visto este se encontrar à frente da textbox de destino. Ao mesmo tempo,

também é possível verificar isto através dos tempos e erros que os utilizadores fizeram para cumprir esta tarefa, como se pode demonstrar nos gráficos seguintes:

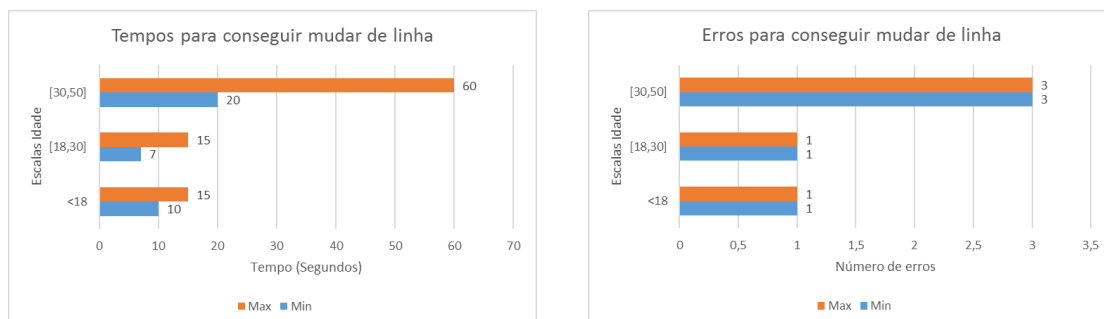


(a) Tempos que os utilizadores demoraram para mudar a hora de um planeamento

(b) Quantidade de erros cometidos a mudar a hora de um planeamento

Figura 5.1: Estatísticas da mudança de horário de um planeamento

Em relação à mudança de linha na atividade "Paragens", foi possível verificar que não era algo que eles encontrassem muito rapidamente e descrevem como sendo pouco perceptível (não sobressaía muito devido ao menu). Há semelhança com o que acontece com a mudança de horários dum planeamento, é possível verificar isto através dos tempos e erros feitos pelos utilizadores:



(a) Tempos que os utilizadores demoraram a mudar a linha das paragens

(b) Quantidade de erros cometidos a mudar a linha das paragens

Figura 5.2: Estatística da mudança de linha na atividade "Paragens"

Como é possível verificar pelos gráficos apresentados acima e inclusive noutros testes realizados (disponíveis no anexo E), os utilizadores na escala de idades [30,50] possuem mais erros e demoram mais tempo para conseguir qualquer uma das tarefas definidas em relação a utilizadores mais novos. Grande parte destes erros consistiu em saltos entre a atividade "planeamento" e "paragens", ocasionalmente um clique no botão voltar do *smartphone*, que os fazia sair da aplicação quando estavam na atividade "planeamento" e noutras vezes um clique nas opções das textboxes (onde podiam usar o GPS

para definir a origem/destino, local dum contacto e ponto no mapa). Em relação aos tempos para cumprir determinada tarefa, foi possível verificar que estes utilizadores gastavam mais tempo a analisar a situação e se alguma das funcionalidades não fosse clara, procuravam outras soluções. No entanto, também foi possível verificar que as restantes idades (<30) estavam mais à vontade a lidar com aplicações novas e que usavam uma forma de navegar na aplicação mais característica de sistemas Android mais recentes (uso da Action Bar em vez do botão voltar do *smartphone* por exemplo).

Em relação aos testes preliminares, mantém-se também a sugestão de que as paragens disponibilizadas na atividade "Paragens" deveriam ser passadas para a atividade "Planeamento", passando utilizar o mesmo mapa. Uma sugestão nova de um utilizador mais atencioso foi a mudança da cor de fundo por detrás dos botões de menu, sugerindo passar a usar a mesma cor que a Action Bar como cor de fundo. Também foi possível verificar ao longo dos vários testes junto dos utilizadores que estes continuavam a utilizar mais as textboxes e GPS em vez do mapa, como se pode verificar nos gráficos seguintes, demonstrando quais os métodos usados no primeiro planeamento (no segundo planeamento, ninguém quem antes tinha usado textbox, passou a usar GPS):

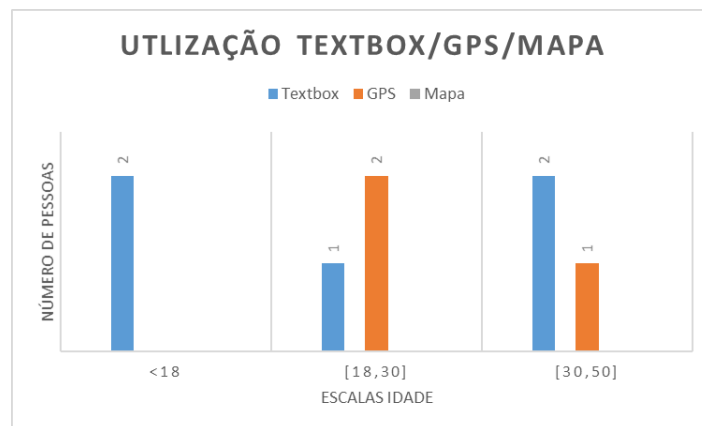


Figura 5.3: Estatísticas de utilização das textboxes, GPS e mapas no planeamento de uma rota

Houve no entanto um utilizador que, no final da lista de tarefas, a explorar a aplicação, tentou usar o mapa e conseguiu inclusive criar o planeamento de uma rota mas só ao final de algumas tentativas e recorrendo a bastantes toques no ecrã. Convém no entanto referir que grande parte destes toques consistiram em toques duplos (que resultam em *zoom in*) em vez de toques longos. Também foi possível verificar junto dos utilizadores que costumam usar o Mobilis, que um planeamento de uma rota com destino à rodoviária de Leiria não era algo simples porque o nome de "rodoviária" não era sugerido na textbox. No entanto, foi possível verificar que alguns dos utilizadores

usavam outros pontos conhecidos nas redondezas para chegar à paragem desejada. Outra sugestão interessante por parte de um dos utilizadores foi a notificação de chegada à paragem de destino quando estamos a circular de autocarro. Em relação à tarefa de verificar a descrição do planeamento de um trajeto, como se demonstra no gráfico a seguir, grande parte dos utilizadores continuou a clicar no balão/infowidget do planeamento do trajeto para obter a descrição deste, apesar de ter havido quem clicasse no respetivo botão de descrição:

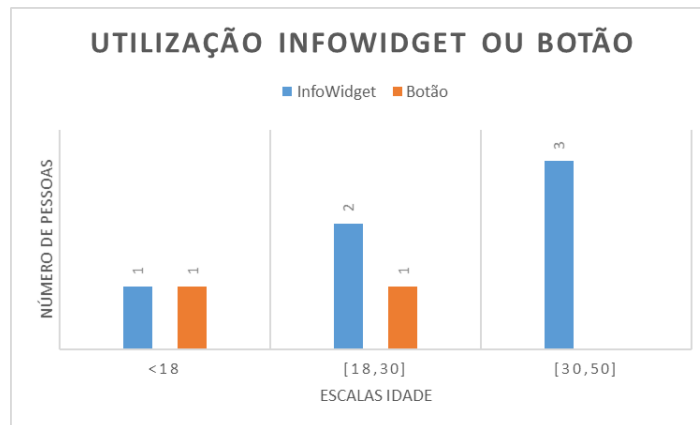


Figura 5.4: Estatísticas de utilização da Infowidget VS Botão de descrição no planeamento de um trajeto

Durante esta fase de testes também houve a possibilidade de testar o cliente Web do OpenTripPlanner junto de um utilizador que possuía um Iphone 6 e assim verificar quais as dificuldades destes na sua utilização. O pedido de um planeamento de rota, visto ter sido feito através das textboxes não revelou problemas mas houve alguma sobreposição de janelas. Em relação a visualizar os horários de uma paragem no cliente Web, o utilizador ainda tentou explorar melhor a aplicação, mas acabou por desistir no fim. Em parte isto deve-se à dificuldade de se navegar pela página web num dispositivo móvel assim como ao nome pouco comum para visualizar os horários de uma paragem (Stop Viewer).

Para esta fase de testes com os utilizadores, a aplicação foi testada com 7 pessoas do sexo masculino e uma pessoa do sexo feminino. No entanto, não foi possível verificar uma mudança de desempenho fora do normal em relação a pessoas do sexo oposto. Das pessoas que participaram neste teste, 4 eram utilizadores do Mobilis, 1 era utilizadora de outros transportes públicos e as restantes não utilizavam os transportes públicos. Em termos de idades, foi possível fazer os testes junto de duas pessoas abaixo dos 18, três entre os 18 e os 30 e outras 3 com mais de 30 anos. Os testes foram feitos utilizando um *smartphone* Samsung Galaxy S5 New, ao final da tarde/noite e grande parte perto das piscinas municipais de Leiria.

5.3.1 Correção do Mobilis Trip Planner depois dos testes principais

Depois dos testes principais decidiu-se corrigir alguns dos problemas mais comuns e simples de serem resolvidos. Por isso, uma das primeiras alterações a serem realizadas foi a mudança do ícone para alterar a data/hora do planeamento para um calendário em vez de um relógio. Visto que o Mobilis Trip Planner tem definido o design para dispositivos móveis de vários tamanhos, foi necessário alterar o ícone neles os todos. Outro problema bastante comum foi a falta de perceção da existência do spinner com as várias linhas na atividade "paragens". Visto a sua aparência ser bastante minimalista e de facto parecer algo integrado com o menu da aplicação, foi mudado o seu aspeto fazendo-o parecer mais como um dropdown menu comum de *desktop*. Ao mesmo tempo, foi seguido o conselho de mudar a cor de fundo do menu onde estão os botões de navegação da aplicação para a da action bar. Para além de conferir um melhor aspeto à aplicação, esta alteração também fez sobressair mais a existência da spinner existente na atividade "paragens". Apesar de não ter sido pedido aos utilizadores para explorar esta funcionalidade, foi possível verificar que muito poucos utilizadores descobriu que a atividade "planeamento" possuía uma navigation drawer onde era possível escolher que transportes públicos usar (Mobilis só possui autocarros no entanto) e escolher se fariam o deslocamento final até ao destino a pé ou de bicicleta. Para resolver este problema ainda vindo do OpenTripPlanner for Android, foram usadas as mesmas especificações de tamanho e aspeto geral do botão de GPS de forma a manter a consistência da aplicação e visto o ícone do botão não ser o mais adequado, foi alterado. Outro problema que foi possível verificar, que por acaso nenhum utilizador reparou foi o nome das colunas na atividade dos horários estarem trocados, tendo sido depois colocado corretamente. Infelizmente, por questões de complexidade de temporais, não foi possível implementar correções.

5.4 Conclusão

Geralmente, pode-se apontar esta fase de testes como positiva. Durante os testes do Mobilis Trip Planner foi possível pelos comentários dos utilizadores dos transportes Mobilis que seria uma aplicação que os podia ajudar a melhor utilizarem os transportes como forma de deslocamento assim como seria uma mais valia para a cidade de Leiria. Mesmo os utilizadores que não utilizam os transportes públicos ou que usam outros os transportes de outras companhias, reconhecem que seria uma aplicação interessante

de estar disponibilizada para a cidade de Leiria. Como seria de esperar, foi possível verificar vários problemas de *design* e até de implementação presentes na aplicação ao longo dos testes preliminares e principais mas mesmo assim, a aplicação cumpriu as suas funções e todos os utilizadores conseguiram completar as tarefas que lhes foram apresentadas.

Capítulo 6

Conclusão

O problema do crescente número de automóveis é um dos grandes problemas modernos já à alguns anos. A sua crescente utilização tem trazido bastantes problemas ambientais e torna bastante complicada a circulação e estacionamento nas grandes cidades devido à falta de espaço. Também do ponto de vista económico, este torna-se mais caro em termos económicos em relação a outras soluções existentes, como os transportes públicos [11][3][6]. No entanto, o problema dos atrasos e a necessidade de conhecimento dos transportes públicos continuam a ser dois grandes problemas hoje em dia [9][4][5].

É neste contexto que surge a solução apresentada neste trabalho, capaz de tanto ajudar as companhias de transportes públicos e/ou cidades a criarem, atualizarem e difundirem os seus dados dos transportes em formato digital, assim como aumenta o bem-estar dos utilizadores destes transportes disponibilizando-lhes ferramentas com a capacidade de ver os horários das paragens *online* e planear um trajeto para determinado destino utilizando os transportes públicos.

Para tal, esta solução implementa plataformas como o OpenTripPlanner e o GTFS-Realtime, capazes de interpretar dados em formato GTFS e GTFS-Realtime para depois difundi-los pela *internet* de forma a serem acessíveis a qualquer hora e lugar. Visto a grande utilização dos dispositivos móveis, também foi desenvolvida a aplicação Mobilis Trip Planner dedicada aos transportes Mobilis da cidade de Leiria e capaz de oferecer funcionalidades como o planeamento de rota e a verificação dos horários das paragens. Ao mesmo tempo, visto haver interesse nesta solução disponibilizar a sua informação em Português, foi também feita a tradução do OpenTripPlanner para Português e posteriormente disponibilizada à comunidade de desenvolvedora, estando de momento implementada no OpenTripPlanner versão 1.1.0 oficial.

No entanto, também é importante conseguir alimentar todas estas plataformas com informação. Para tal, no início foram desenvolvidos vários ficheiros de teste em formato GTFS para serem aplicados na plataforma OpenTripPlanner e posteriormente melhorados de forma a habilitarem todas as funcionalidades desta plataforma. No entanto, foi verificado ao longo do desenvolvimento destes dados que não era uma tarefa simples e que podia se tornar algo bastante complexo, principalmente para alguém não especializado. Para tal, foi desenvolvido um Template em Excel capaz de oferecer a qualquer pessoa ou interessado a possibilidade de facilmente criar e atualizar os vários ficheiros em formato GTFS. Mais tarde, depois de uma reunião com a Câmara Municipal de Leiria e a Rodoviária do Lis acerca da utilização desta solução nos transportes Mobilis e visto que o Template Excel poderia não ser a melhor solução para este caso visto já existirem dados em formato digital, foi criado o Mobilis GTFS Creator, capaz de traduzir para GTFS os dados já existentes.

Com tudo isto, foi possível criar uma solução que para além de ser capaz de cumprir os objetivos propostos para este trabalho, adiciona valor à plataforma OpenTripPlanner oficial através da tradução deste para Português mas também aos transportes Mobilis visto esta solução estar a difundir a sua informação neste momento através de um servidor virtual no IPLeiria e o Mobilis Trip Planner ter sido recebido com interesse por parte dos seus utilizadores.

Apesar desta solução cumprir os seus objetivos, existem várias correções ou novas funcionalidades que podiam ser implementadas. Em termos de plataformas, o cliente Web do OpenTripPlanner poderia ser caracterizado tendo em vista os transportes Mobilis, já que ele possui uma personalização bastante generalista e poderia ser criada uma aplicação para lidar com os dados do GTFS-Realtime mais facilmente, já que neste momento está tudo definido em vários *scripts* que precisam de ser executados manualmente. Em relação ao Template Excel, poderia ser desenvolvida uma macro que automaticamente criasse os vários ficheiros em formato GTFS e possivelmente os compactasse, já que neste momento essa tarefa está ao encargo do utilizador. Em relação ao Mobilis GTFS Creator, seria necessário corrigir os vários *warnings* que acontecem ao executar o FeedValidator e trabalhar junto com a rodoviária do lis para resolver estes problemas nos dados e ao mesmo tempo, poderia ser revisto o código deste de forma a melhorar a sua performance, dado que é uma aplicação que pode demorar alguns minutos para traduzir os ficheiros para GTFS. Em relação à aplicação Mobilis Trip Planner, deveria ser revisto o autocomplete das suas textboxes na atividade "planeamento", já que em algumas situações, não aparecem as paragens desejadas ou os locais corretos e também facilitar a utilização do GPS para definir pontos de origem ou destino, já que também foi outra das funcionalidades mais usadas. Em relação à

atividade "paragens", esta poderia ser adaptada à atividade "planeamento", passando a disponibilizar as suas funcionalidades no mapa desta. Por fim, poderia ser adicionada a funcionalidade de receber informação dinâmica como a localização dos transportes, atrasos das viagens e eventos (por exemplo, uma reparação de via que dificulte a circulação durante alguns dias) e a capacidade de ver o estado dos transportes (se estão ativos ou por alguma razão foram cancelados).

Bibliografia

- [1] David Alves, Luis M. Martinez e Jose M. Viegas. «Retrieving real-time information to users in public transport networks: an application to the Lisbon bus system». Em: *15th meeting of the EURO Working Group on Transportation* (out. de 2012). Acedido: 2016-09-15. URL: <http://www.sciencedirect.com/science/article/pii/S1877042812042279>.
- [2] Sean J Barbeau et al. «Smart Phones and Smart Travelers». Em: *53rd Annual Transportation Research Forum, Tampa, Florida, March 15-17, 2012*. 207217. Transportation Research Forum. 2012.
- [3] Gabriela Beirao e J.A. Sarsfield Cabral. *Understanding attitudes towards public transport and private car: A qualitative study*. Estudo. FEUP, jun. de 2007.
- [4] Janet E. Dickinson et al. «Tourism and the smartphone app: capabilities, emerging practice and scope in the travel domain». Em: *Current Issues in Tourism* (2014).
- [5] Trinh Minh Tri Do, Jan Blom e Daniel Gatica-Perez. «Smartphone usage in the wild: a large-scale analysis of applications and context». Em: *Proceedings of the 13th international conference on multimodal interfaces*. ACM. 2011, pp. 353–360.
- [6] Ricardo Miguel Pinto Ferreira. «Aplicação móvel sensível ao contexto para transportes públicos». Tese de doutoramento. 2012.
- [7] Werner Gronau e Andreas Kagermeier. «Key factors for successful leisure and tourism public transport provision». Em: *Journal of Transport Geography* 15.2 (2007), pp. 127–135.
- [8] Mineta Transportation Institute. «Using Bicycles for the First and Last Mile of a Commute». Em: Acedido: 2016-09-14. Set. de 2009. URL: <http://transweb.sjsu.edu/MTIportal/research/publications/documents/BikeCommute.pdf>.
- [9] Jerald Jariyasunant et al. «Mobile Transit Trip Planning with Realtime Data». Em: *Transportation Research Board 89th Annual Meeting* (nov. de 2009). Acedido: 2016-09-14. URL: <http://bayen.eecs.berkeley.edu/sites/default/files/conferences/trb10c.pdf>.

- [10] Zhili Liu, Xudong Jia e Wen Cheng. «Solving the last mile problem: Ensure the success of public bicycle system in Beijing». Em: *Procedia-Social and Behavioral Sciences* 43 (2012), pp. 73–78.
- [11] David Roberts. *Personal rapid transit: The future of public transportation, maybe*. Ed. por Grist. Acedido: 2016-09-13. Jan. de 2015. URL: <http://grist.org/business-technology/personal-rapid-transit-the-future-of-public-transportation-maybe/>.
- [12] Wade Roush. «Welcome to Google Transit: How (and Why) the Search Giant is Remapping Public Transportation». Em: Acedido: 2016-10-25. Fev. de 2012. URL: <http://bit.ly/2eFFGSF>.
- [13] Pedro Jose de Sousa Ferreira. *Integracao de tecnologias moveis nos transportes publicos*. report. IPL - ESTG, mar. de 2015.
- [14] Pieter Vansteenwegen et al. «The city trip planner: an expert system for tourists». Em: *Expert Systems with Applications* 38.6 (2011), pp. 6540–6546.
- [15] Hai Wang. *Approximating the Performance of a Last Mile Transportation System*. Relatorio. Tsinghua University, 2009.

Anexos

Anexo A

Protótipos da aplicação Mobilis Trip Planner

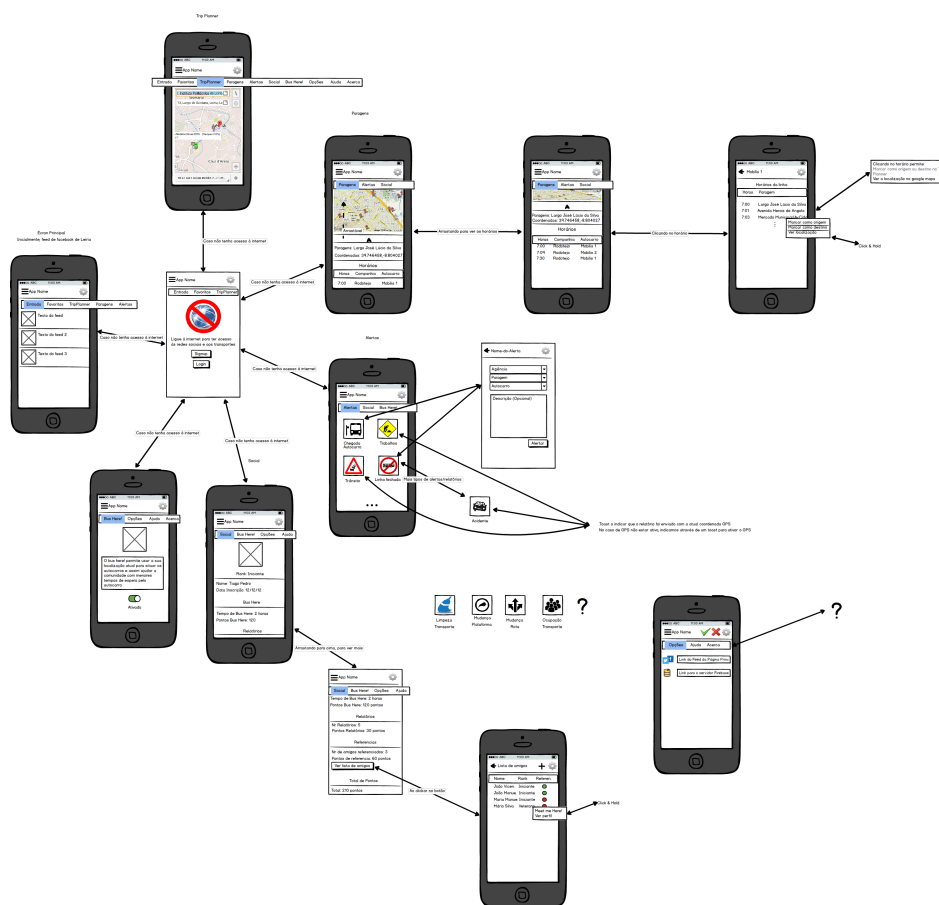


Figura 6.1: Protótipos iniciais do Mobilis Trip Planner


```

- <RouterList>
- <routerInfo>
- <routerInfo>
  <routerId/>
- <polygon>
  <type>Polygon</type>
  <coordinates>-8.7887216</coordinates>
  <coordinates>39.6995287</coordinates>
  <coordinates>-8.8712561</coordinates>
  <coordinates>39.702025</coordinates>
  <coordinates>-8.866231</coordinates>
  <coordinates>39.746388</coordinates>
  <coordinates>-8.82986</coordinates>
  <coordinates>39.7904362</coordinates>
  <coordinates>-8.7813137</coordinates>
  <coordinates>39.7751561</coordinates>
  <coordinates>-8.776647</coordinates>
  <coordinates>39.7727</coordinates>
  <coordinates>-8.7139548</coordinates>
  <coordinates>39.718597800000005</coordinates>
  <coordinates>-8.7887216</coordinates>
  <coordinates>39.6995287</coordinates>
</polygon>
<buildTime>1468247147720</buildTime>
- <transitModes>
  <transitModes>BUS</transitModes>
</transitModes>
<centerLatitude>39.7417969</centerLatitude>
<centerLongitude>-8.806154500000002</centerLongitude>
<hasParkRide>>false</hasParkRide>
- <travelOptions>
- <travelOptions>
  <value>TRANSIT,WALK</value>
  <name>TRANSIT</name>
</travelOptions>
- <travelOptions>
  <value>BUS,WALK</value>
  <name>BUS</name>
</travelOptions>
- <travelOptions>
  <value>WALK</value>
  <name>WALK</name>
</travelOptions>

```

Figura 6.4: Resposta do route no OTP 19

Anexo C

Passos para instalação do servidor OpenTripPlanner

1. Instalar/Possuir um sistema operativo Linux (utilizado neste projeto: distribuição Xubuntu 16.04. Deverá ser compatível com outras distribuições baseadas em Ubuntu 16.04 LTS) ou um sistema operativo Windows (utilizado neste projeto: Windows 7 e 10)
2. Instalar o JDK 8 no sistema.
3. Criar uma pasta com o nome OpenTripPlanner
4. Colocar o conteúdo da pasta: OTP_Desenvolvimento\OTP_Final

5. Executar numa linha de comandos (DOS ou Terminal Linux) dentro da pasta OTP_Final: `java -Xmx1G -jar otp-1.1.0-PT.jar -build .\GTFS_PBF -inMemory -port 8080`

Anexo D

Passos para instalação do servidor GTFS-Realtime

1. Instalar/Possuir um sistema operativo linux (utilizado neste projeto: distribuição Xubuntu 16.04. Deverá ser compatível com outra distribuições baseadas em Ubuntu 16.04 LTS)
2. Fazer download do ficheiro compactado indicado para o sistema operativo em: <https://developers.google.com/protocol-buffers/docs/downloads>
3. Para o Linux deverá ser necessário instalar um compilador c++ como o g++ através de: `sudo apt-get install g++`
4. Seguir os passos indicados no Readme que vem dentro do ficheiro compactado (Para Linux, fazer `./configure`, `make` e é preferível fazer `sudo make install`) (Para Windows, bastará executar o `protoc.exe`)
5. Para o Linux deverá ser executado o seguinte comando: `sudo ldconfig`
6. Instalar o pip do Python através de: `sudo apt-get install python-pip`
7. Executar de seguida o comando: `sudo pip install protobuf`
8. Ir à pasta com os ficheiros do GTFS-Realtime
9. Correr o comando "make python"
10. Para criar o ficheiro com os alertas e posição de um autocarro, executar o seguinte comando:
`./add_alert_python`
Isto irá gerar um ficheiro com o nome "teste" com a informação em formato binário
11. Colocar online através de um servidor web ou noutra lugar que permita aceder em formato raw ao ficheiro "teste". Um exemplo é o colocar no GitHub e depois aceder em formato raw ao ficheiro, devolvendo um link como:

<https://raw.githubusercontent.com/XtremeDS/BinTransp/master/teste>

12. Para dotar o OpenTripPlanner deste trabalho com a informação dinâmica do GTFS-Realtime, é necessário criar um ficheiro `route_config.json` na pasta "GTFS" anteriormente criada com o seguinte conteúdo:

```
{
  routingDefaults: {},
  updaters: [
    { type: "real-time-alerts", frequencySec: 30, url: "https://raw.githubusercontent.com/XtremeDS/BinTransp/master/teste",
      feedId: "RDT" }
  ]
}
```

O URL poderá ou deverá ser trocado caso use o ficheiro de GTFS-Realtime noutra endereço.

Anexo E

Estatísticas dos testes principais

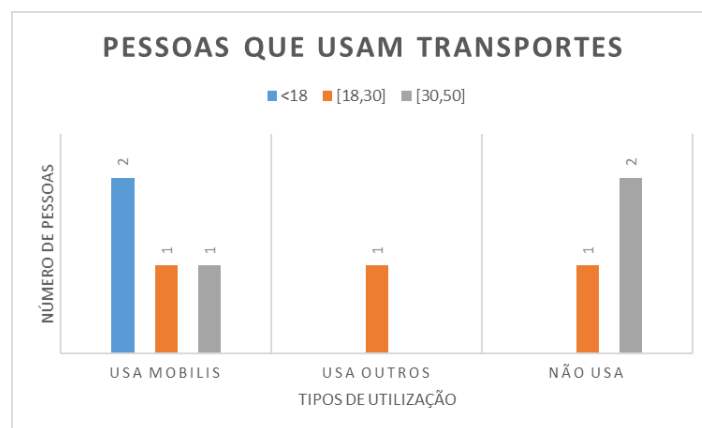


Figura 6.5: Divisão das pessoas nos teste principal

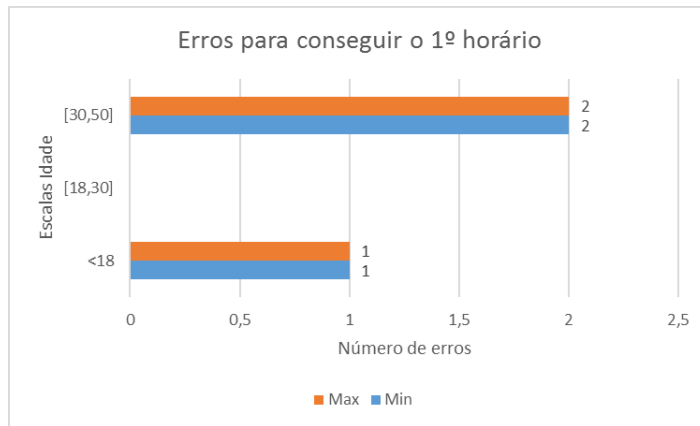


Figura 6.6: Quantidade de erros para conseguir o primeiro horário

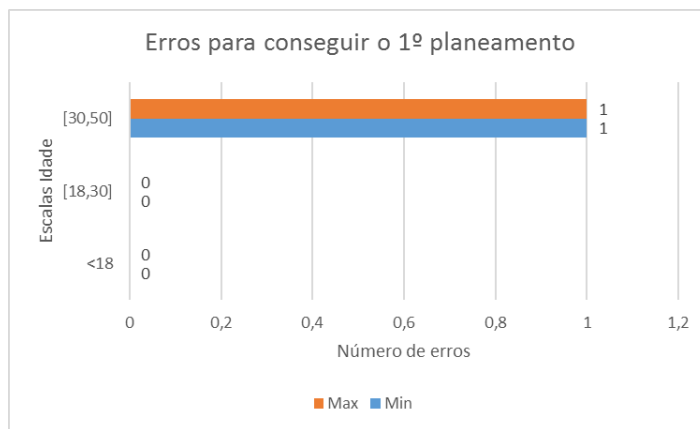


Figura 6.7: Quantidade de erros para conseguir o primeiro planeamento

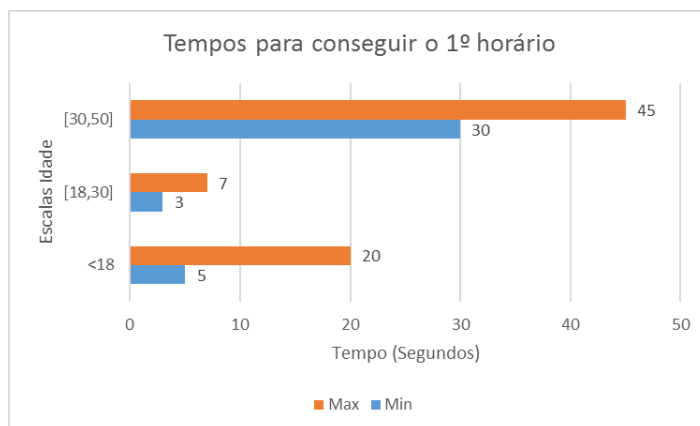


Figura 6.8: Tempo para conseguir o primeiro horário

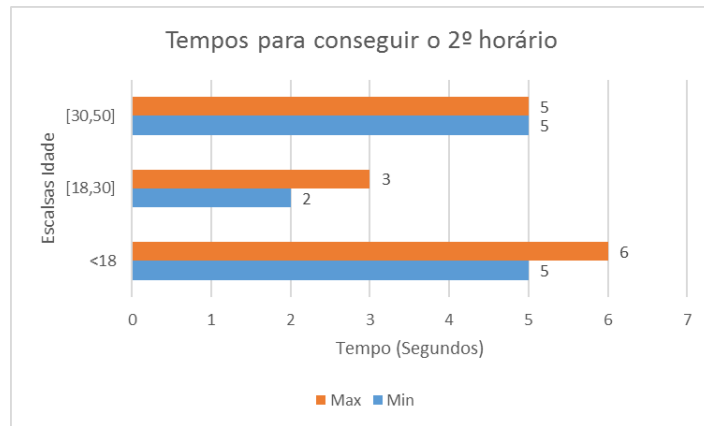


Figura 6.9: Tempo para conseguir o segundo horário

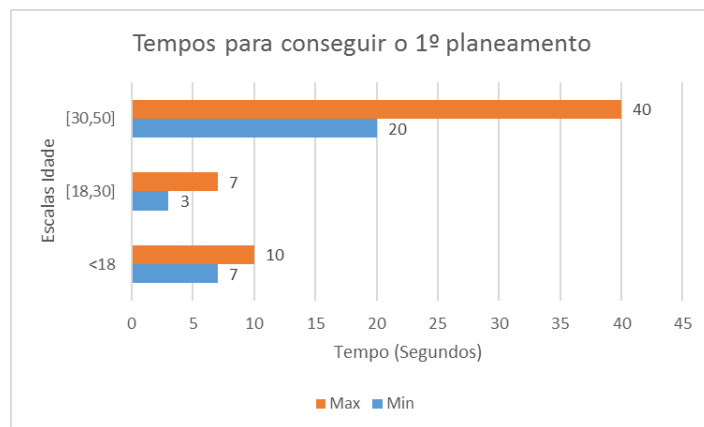


Figura 6.10: Tempo para conseguir o primeiro planeamento

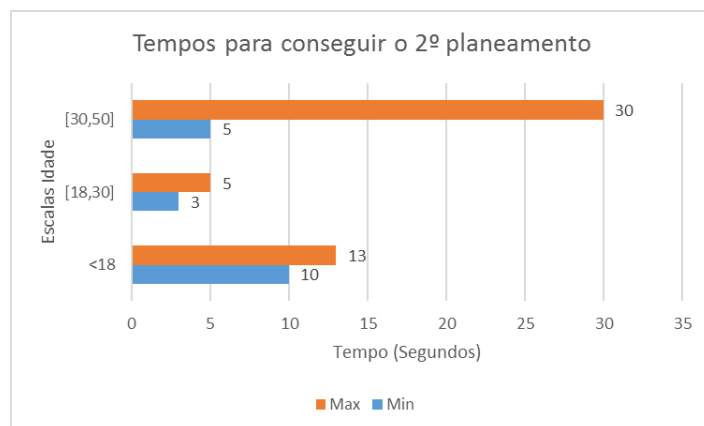


Figura 6.11: Tempo para conseguir o segundo planeamento

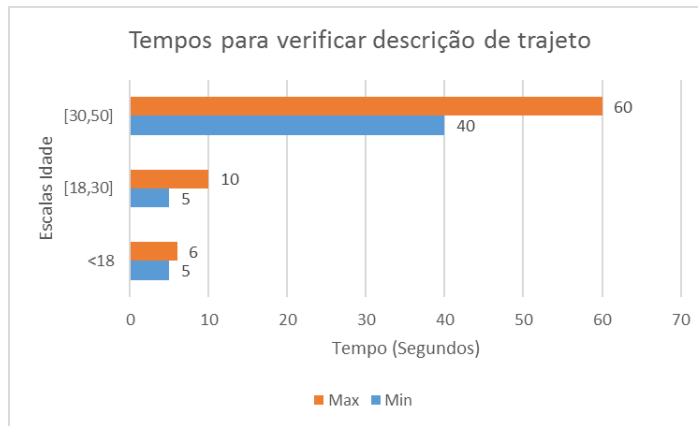


Figura 6.12: Tempo para ver a descrição dum planeamento

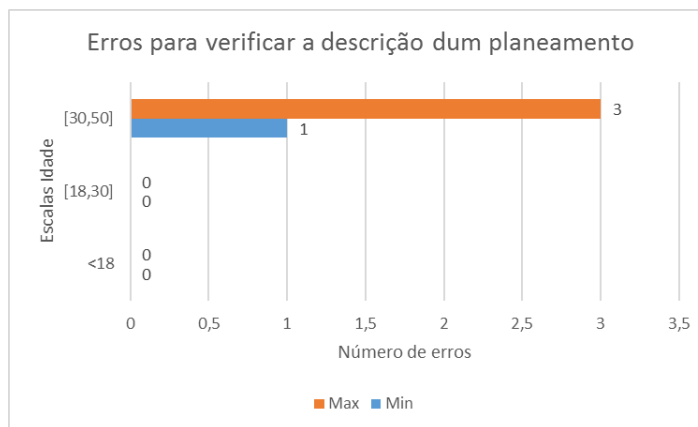


Figura 6.13: Tempo para ver a descrição dum planeamento