

FXBus – Uma Aplicação Móvel para Combater a Ansiedade de Perder o Autocarro

DISSERTAÇÃO DE MESTRADO

João Bernardo Fernandes Nunes Luís

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

setembro | 2013

FXBus – Uma Aplicação Móvel para Combater a Ansiedade de Perder o Autocarro

DISSERTAÇÃO DE MESTRADO

João Bernardo Fernandes Nunes Luís

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADOR

Leonel Domingos Telo Nóbrega

CO-ORIENTADOR

Luís Armando de Aguiar Oliveira Gomes

Índice

Índice.....	i
Índice de Tabelas	iii
Índice de Gráficos	iv
Índice de Figuras.....	v
Agradecimentos	ix
Resumo.....	xi
Abstract.....	xiii
Lista de Acrónimos.....	xv
1. Introdução	1
1.1. Motivação.....	2
1.2. Problema.....	2
1.3. Contribuições	3
1.4. Metodologia.....	4
1.5. Organização da Dissertação	5
2. Trabalhos Relacionados.....	8
2.1. <i>Field Trial of Tiramisu: Crowd-Sourcing Bus Arrival Times to Spur Co-Design</i>	8
2.2. <i>Location-Aware Tools for Improving Public Transit Usability</i>	9
2.3. <i>Bus Catcher: A Context Sensitive Prototype System for Public Transportation Users</i>	9
2.4. Avaliação da FXBus e de outras aplicações	10
3. Tecnologias Utilizadas	16
3.1. <i>Smartphones e Tablets</i>	16
3.2. Plataforma Android.....	18
3.3. Aplicações <i>Web</i> vs Nativas	20
3.4. Ambiente de Desenvolvimento.....	22
3.4.1. Eclipse	22
3.4.2. <i>Android SDK e AVD</i>	24
3.4.3. <i>Plugin ADT</i>	26
3.5. Outras Tecnologias Utilizadas.....	26
3.5.1. <i>Google Calendar API</i>	27

3.5.2.	<i>Google Maps Android SDK V2</i>	27
3.5.3.	<i>Google Places API</i>	29
3.5.4.	<i>MySQL Server, PHP e JSON</i>	30
3.5.5.	<i>Base de Dados SQLite</i>	31
4.	Desenho do Sistema	32
4.1.	Casos de Utilização	32
4.1.1.	Modelo de Tarefas	34
4.2.	Requisitos.....	40
4.2.1.	Requisitos Funcionais	40
4.2.2.	Requisitos Não-Funcionais	42
4.3.	<i>Wireframes/Mockups</i>	44
4.3.1.	<i>Versão Smartphone</i>	44
4.3.2.	<i>Versão Tablets</i>	45
4.4.	Arquitetura	46
4.4.1.	Arquitetura MVC.....	46
4.4.2.	Arquitetura Cliente-Servidor (<i>Broker</i>)	48
4.5.	Padrões de Desenho	50
4.5.1.	Padrão <i>State</i>	50
4.5.2.	Padrão <i>Adapter</i>	52
4.5.3.	Padrão <i>Observer</i>	52
4.5.4.	Padrão <i>Proxy</i>	53
5.	Implementação	54
5.1.	Desenho da interface.....	54
5.1.1.	<i>Versão para Smartphone</i>	54
5.1.2.	<i>Versão para Tablet</i>	56
5.2.	Integração de múltiplas línguas	56
5.3.	Funcionalidades da aplicação.....	57
5.3.1.	Menu	57
5.3.2.	Atividade <i>Próximas Carreiras</i>	58
5.3.3.	Atividade <i>Alarmes</i>	60
5.3.3.1.	<i>Adicionar Alarme</i>	61
5.3.3.2.	<i>Editar Alarme</i>	63
5.3.3.3.	<i>Remover Alarme</i>	63

5.3.4.	Atividade Horários	64
5.3.5.	Atividade Mapa	65
5.3.5.1.	Rota da Carreira.....	66
5.3.5.2.	Horário na Paragem.....	67
5.3.5.3.	Pesquisa por Nome.....	68
5.3.6.	Atividade Itinerários	69
5.3.7.	Atividade Contatos.....	71
5.3.8.	Serviço de Sincronização	72
5.3.9.	Serviço de Alerta Início de Viagem.....	73
5.3.10.	Serviço de Alerta de Tempo Restante até à Paragem.....	74
6.	Testes e Resultados	75
6.1.	Testes de Usabilidade.....	75
6.2.	Estudo de Utilização.....	76
6.3.	Avaliação do Utilizador	79
6.4.	Discussão.....	81
7.	Lançamento	85
7.1.	Google <i>Play</i>	85
7.1.1.	Correção de Erros e Falhas.....	88
7.1.2.	Estudo de utilização.....	88
8.	Conclusões	91
8.1.	Trabalhos Futuros	92
	Referências	95
Anexo A	– Inquérito de avaliação do teste de usabilidade.....	99
Anexo B	– Primeiro cenário de usabilidade.....	105
Anexo C	– Segundo cenário de usabilidade.....	107
Anexo D	– Implementação na versão para <i>Tablet</i>	109

Índice de Tabelas

Tabela 2.1 – Resumo das aplicações avaliadas	10
Tabela 3.1 – Mercado global de sistemas operativos para dispositivos móveis [10].	17

Índice de Gráficos

Gráfico 2.1 – Avaliação da aplicação Tiramissu	12
Gráfico 2.2 – Avaliação da aplicação Metro Lisboa	13
Gráfico 2.3 – Avaliação da aplicação Madrid Metro	13
Gráfico 2.4 – Avaliação da aplicação Lisboa Move-me	13
Gráfico 2.5 – Avaliação da aplicação Moobly	13
Gráfico 2.6 – Avaliação da aplicação SMTUC Mobile	14
Gráfico 2.7 – Avaliação da aplicação TUB Mobile	14
Gráfico 2.8 – Avaliação da aplicação One Bus Away	14
Gráfico 2.9 – Avaliação da aplicação Izi Carris.....	14
Gráfico 2.10 – Avaliação da aplicação desta dissertação (FXBus).....	15
Gráfico 3.1 – Distribuição dos Sistemas Operativos dos utilizadores dos Horários do Funchal com dispositivos móveis do tipo <i>Smartphone</i>	18
Gráfico 3.2 – Cota de mercado dos dispositivos Android por versão do OS	19
Gráfico 3.3 – Cota de mercado dos dispositivos Android por tamanho do ecrã	19
Gráfico 3.4 – Cota de mercado dos dispositivos Android por densidade do ecrã.....	20
Gráfico 6.1 – Distribuição da utilização das atividades	77
Gráfico 6.2 – Distribuição da utilização pelas funcionalidades	78
Gráfico 6.3 – Avaliação da utilidade da aplicação	79
Gráfico 6.4 – Avaliação do <i>layout</i> de visualização	80
Gráfico 6.5 – Avaliação da terminologia utilizada	80
Gráfico 6.6 – Avaliação das capacidades do sistema.....	81
Gráfico 6.7 – Avaliação da aprendizagem do sistema.....	81

Gráfico 6.8 – Avaliação da reacção geral à aplicação.....	82
---	----

Índice de Figuras

Figura 3.1 – <i>Home page</i> da aplicação <i>web</i> móvel dos Horários do Funchal.....	21
Figura 3.2 – Ambiente de trabalho do eclipse	23
Figura 3.3 – Janela de criação de um novo projecto da Android.....	23
Figura 3.4 – Ambiente gráfico do <i>SDK Manager</i>	24
Figura 3.5 – Ambiente gráfico do <i>AVD Manager</i>	25
Figura 3.6 – Ambiente gráfico de um <i>AVD</i> como exemplo	25
Figura 3.7 – Ambiente gráfico do <i>plugin ADT</i>	26
Figura 3.8 – Importação da livreria do <i>Google Play Services</i> para o Eclipse	28
Figura 3.9 – Adicionar livreria ao projecto no Eclipse.....	29
Figura 3.10 – Base de Dados <i>SQLite</i> utilizado no dispositivo Android.....	31
Figura 4.1 – Diagrama de casos de utilização da aplicação desta dissertação	33
Figura 4.2 – Modelo de tarefa para “Consultar Agendamentos”.....	34
Figura 4.3 – Modelo de tarefa para “Editar agendamento”	35
Figura 4.4 – Modelo de tarefa para “Apagar agendamento”	35
Figura 4.5 – Modelo de tarefa para “Visualizar rota no mapa”	36
Figura 4.6 – Modelo de tarefa para “Visualizar paragens no mapa”.....	36
Figura 4.7 – Modelo de tarefa para “Visualizar horários”	37
Figura 4.8 – Modelo de tarefa para “Pesquisa de itinerário”	37
Figura 4.9 – Modelo de tarefa para “Resultados da pesquisa de itinerário”	38
Figura 4.10 – Modelo de tarefa para “Sincronização”	39
Figura 4.11 – Modelo de tarefa para “Alerta de primeira paragem”.....	39

Figura 4.12 – Modelo de tarefa para “Alerta tempo restante para chegar à paragem”	40
Figura 4.13 – Exemplo dos <i>wireframes</i> para <i>smartphone</i>	44
Figura 4.14 – Formulário utilizado para registar e classificar erros do utilizador ao usar a aplicação [23].....	45
Figura 4.15 – Exemplo dos <i>wireframes</i> para <i>Tablet</i>	46
Figura 4.16 – Desenho do estilo arquitectural MVC utilizado na aplicação android .	47
Figura 4.17 – Desenho do estilo arquitectural Cliente-Servidor por meio de um <i>Broker</i>	49
Figura 4.18 – Ciclo de vida da Actividade [29].....	51
Figura 5.1 – Opções disponíveis na Actividade <i>Maps</i>	55
Figura 5.2 – Exemplo do resultado após ser seleccionada uma Carreira	55
Figura 5.3 – Opções e resultados disponíveis na activade <i>Maps</i> , na versão para <i>Tablet</i>	56
Figura 5.4 – Menu da aplicação, versão em Inglês	57
Figura 5.5 – Menu da aplicação, versão em Português.....	57
Figura 5.6 – Menu aberto com as opções de todas actividades disponíveis da versão para <i>Tablet</i>	58
Figura 5.7 – Lista de todas as Carreiras com a hora do próximo autocarro	59
Figura 5.8 – Lista de Carreiras/Variantes favoritas.....	59
Figura 5.9 – Submenu com opções referentes às Carreiras/Variantes	59
Figura 5.10 – Pesquisa de Carreiras/Variantes através da barra de pesquisa	60
Figura 5.11 – <i>Popup</i> com as opções de agendar um alarme na actividade Próximas Carreiras.....	61
Figura 5.12 – Alarme agendado através da actividade Próxima Carreiras.....	61
Figura 5.13 – Alarme agendado através da actividade Horários.....	62

Figura 5.14 – Agendar viagem na atividade Alarmes.....	63
Figura 5.15 – <i>Popup</i> de confirmação para apagar agendamento	64
Figura 5.16 – Lista de todas as Carreiras/Variantes na atividade Horários.....	65
Figura 5.17 – Horários diários da carreira selecionada	65
Figura 5.18 – Lista de todas as Carreiras/Variantes na atividade Mapa	66
Figura 5.19 – Vista de pesquisa livre no mapa com todas as paragens visíveis.....	66
Figura 5.20 – Rota da carreira selecionada, com as respetivas paragens e posição do autocarro em tempo real	67
Figura 5.21 – Informações referentes a paragem X, como o nome, o código e as Carreiras que passam por ali	67
Figura 5.22 – Horário em tempo real para a paragem X	68
Figura 5.23 – Pesquisa de lugares.....	68
Figura 5.24 – Histórico de itinerários.....	69
Figura 5.25 – Pesquisar novo itinerário.....	70
Figura 5.26 – Pesquisar novo itinerário através do modo de texto.....	70
Figura 5.27 – Pesquisa de Itinerário através de pontos previamente guardados.....	71
Figura 5.28 – Pesquisa de itinerário através de escolha do ponto no mapa	71
Figura 5.29 – Sincronização da base de dados <i>online</i> com a base de dados local ..	72
Figura 7.1 – Consola do programador, aba de relatórios de erro.....	86
Figura 7.2 – Consola do programador, aba de classificações e comentários	86
Figura 7.3 – Mercado Google Play, visualização da classificação e dos comentários	87
Figura 7.4 – Consolado programador, aba de estatísticas.....	87
Figura 7.5 – Distribuição da utilização das atividades (segundo estudo).....	89
Figura 7.6 – Distribuição da utilização pelas funcionalidades (segundo estudo)	90
Figura. A.1 – Inquérito de avaliação do teste de usabilidade (1/7).....	99

Figura. A.2 – Inquérito de avaliação do teste de usabilidade (2/7)	100
Figura. A.3 – Inquérito de avaliação do teste de usabilidade (3/7)	101
Figura. A.4 – Inquérito de avaliação do teste de usabilidade (4/7)	102
Figura. A.5 – Inquérito de avaliação do teste de usabilidade (5/7)	102
Figura. A.6 – Inquérito de avaliação do teste de usabilidade (6/7)	103
Figura. A.7 – Inquérito de avaliação do teste de usabilidade (7/7)	104
Figura. D.1 – <i>Layout</i> da atividade Próximas Carreiras na versão para <i>Tablet</i>	109
Figura. D.2 – <i>Layout</i> da atividade Alarmes na versão para <i>Tablet</i>	110
Figura. D.3 – <i>Layout</i> da atividade Mapa na versão para <i>Tablet</i>	110
Figura. D.4 – <i>Layout</i> da atividade Horários na versão para <i>Tablet</i>	111
Figura. D.5 – <i>Layout</i> da atividade Itinerários (pesquisa) na versão para <i>Tablet</i>	111
Figura. D.6 – <i>Layout</i> da atividade Itinerários (resultados) na versão para <i>Tablet</i> ...	112

Agradecimentos

Gostaria de agradecer em primeiro lugar ao meu avô, José António Fernandes Luís, que me motivou para continuar a estudar e me financiou para que este momento fosse possível.

A todos os meus familiares que me deram apoio e suporte durante todos estes anos.

À minha namorada Ana Camacho que esteve sempre presente, até mesmo quando não podia.

Ao meu amigo e colega de trabalho Pedro Miguel Nogueira que contribuiu com o seu conhecimento e talento para o enriquecimento deste trabalho.

Aos meus orientadores Leonel Nóbrega e Luís Gomes que estiveram presentes ao longo do desenvolvimento deste trabalho e que também me auxiliaram com os seus preciosos conhecimentos.

Ao pessoal da empresa Horários do Funchal, Pedro Vieira, Marco Lobo e Jonathan Vieira que contribuíram e colaboraram para este projeto. Também aos outros colegas da empresa que contribuíram com o seu conhecimento em diversas áreas e receberam-me com o maior carinho.

Às pessoas que testaram a aplicação resultante desta dissertação e contribuíram para o resultado final deste trabalho.

Aos meus colegas de curso com quem passei bons e maus momentos e sempre me apoiaram.

Muito Obrigado!

Resumo

Os utilizadores de transportes públicos enfrentam dificuldades no seu dia-a-dia quando usam o autocarro para deslocar-se. Essas dificuldades passam por atrasos pessoais, por enganos na consulta dos horários das carreiras, entre outros, resultando na perda do autocarro.

Este trabalho consistiu no desenvolvimento de uma aplicação móvel para a plataforma Android, tendo como principal objectivo fornecer ao utilizador uma ferramenta leve, rápida e útil para utilizar quando se encontra em movimento. O público-alvo desta aplicação são os utilizadores de transportes públicos da empresa Horários do Funchal, que necessitam de realizar deslocações frequentes, e não têm acesso constante à *internet*, dentro e fora de casa.

Realizou-se uma pesquisa, através de um inquérito divulgado nas redes sociais, aos utilizadores de transportes públicos e, posteriormente, utilizou-se essa pesquisa no processo de *brainstorming* para construir as ideias para as funcionalidades da aplicação. Também se pesquisaram por outras aplicações com objectivos semelhantes e compararam-se essas com a aplicação desta dissertação.

Posteriormente, implementou-se o primeiro protótipo e realizaram-se mais testes com novos utilizadores, que usaram a aplicação no seu dia-a-dia. Após se terem corrigido os erros encontrados pelos utilizadores na aplicação, lançou-se uma versão *beta online* no mercado da Google Play.

Os resultados obtidos com esta aplicação comprovam não só que existe a necessidade de uma aplicação destas no mercado, mas também permitiram ainda recolher dados sobre melhoramentos que podem ser feitos na mesma.

Ao desenvolver esta aplicação pôde-se compreender as necessidades e as limitações que são impostas pelo cliente, como também colocar em prática os conhecimentos adquiridos ao longo do curso.

Palavras-chave: Android, Aplicação Móvel, Tablet, Smartphone

Abstract

The public transport users suffer difficulties in their daily life when using the bus to move themselves. These difficulties came from personal delays, mistakes when consulting the bus schedule, among others, resulting in losing the bus.

This work was about the development of a mobile application for the Android platform, with the primary objective of providing the user a fast and lightweight tool to use when moving, without access to sources of information such as the Internet. The target audience for this application are the users of Horários do Funchal, who need to make frequent journeys, and do not have constant access to the internet, inside and outside their home.

We conducted a survey aimed at public transport users, launched through social networks, on users of public transport, and subsequently we used in this research process to build brainstorm ideas. We also researched other applications with similar objectives and compared these with the application we developed.

Later we implemented the first prototype and carried out more tests with new users, who used the application in their daily life. After having corrected errors found by the users in the application, a beta version was released online, in the Google Play market.

The results obtained with this application demonstrate not only that there is space in the market for an application like the one developed in this dissertation. We also collected data about improvements that can be made in it.

While developing this application we could understand the needs and limitations that are imposed by the client, but also put into practice the knowledge acquired during the degree.

Keywords: Android, Aplicação Móvel, Tablet, Smartphone

Lista de Acrónimos

ADT – *Android Development Tools*

API – *Application Programming Interface*

AVD – *Android Virtual Devices*

DP – *Density-independent Pixel*

DPI – *Dots Per Inch*

GPS – *Global Positioning System*

HDPI – *High Dots Per Inch*

HTTP – *Hyper Text Transfer Protocol*

HTTPS – *Hyper Text Transfer Protocol Secure*

JSON – *JavaScript Object Notation*

LDPI – *Low Dots Per Inch*

MDPI – *Medium Dots Per Inch*

OS – *Operative System*

PDF – *Portable Document Format*

PHP – *Hypertext Preprocessor*

REST – *Representational State Transfer*

SDK – *Software Development Kit*

SMS – *Short Message Service*

SQL – *Structured Query Language*

TV DPI – *Television Dots Per Inch*

XHDPI – *Extra High Dots Per Inch*

XML – *Extensible Markup Language*

XXHDPI – *Extra Extra High Dots Per Inch*

1. Introdução

“Information technology and business are becoming inextricably interwoven. I don't think anybody can talk meaningfully about one without the talking about the other.”

(Bill Gates).

Os transportes públicos têm um grande impacto na sociedade, na economia, no ambiente e também na qualidade de vida dos cidadãos. É nesta última área que este trabalho se focou, e para tal procuraram-se diferentes formas de proporcionar um aumento da qualidade de vida aos utilizadores [1].

Cada vez mais, os utentes de transportes públicos procuram o máximo de informação, de forma a poderem, por exemplo, escolher a melhor paragem para apanhar o autocarro e poder chegar ao seu destino a tempo. Assim, proporcionar um meio através do qual o utente possa procurar e organizar as suas viagens, de uma forma simples e rápida ajudará a tornar o serviço de transportes públicos mais atrativo.

Actualmente, um dos meios que os utentes dos transportes públicos utilizam são os dispositivos móveis, como os *Tablets* e os *Smartphones*, para consultarem as informações. Utilizando um destes dispositivos, o utente pode ainda escolher como aceder às informações, através de um portal *online*, de um serviço de SMS, de ficheiros previamente descarregados ou ainda, através de uma aplicação nativa [2]. Existem vários exemplos dessas aplicações nativas, para *Android* e/ou *iOS*, como a aplicação Tiramissu [3] ou a OneBusAway [4], que têm um grande número de utilizadores satisfeitos e com boas classificações nos respetivos mercados.

A empresa Horários do Funchal mostrou interesse nestas novas tecnologias e, sendo responsável pela maior parte da rede de transportes públicos da Região Autónoma da Madeira, necessita de mais e novos sistemas que fomentem a utilização dos seus serviços.

Tendo em conta que os utilizadores cada vez mais têm dispositivos *Smartphone* ao seu alcance, devido à constante queda nos preços, existem condições para contribuir com uma aplicação que auxilie os utentes da empresa Horários do Funchal nas suas deslocações do dia-a-dia.

1.1. Motivação

A sociedade atual está dependente dos meios de transporte para progredir o que faz com que os meios de transporte públicos sejam essenciais nesse progresso. O uso dos transportes públicos também tem um forte impacto positivo no meio ambiente e na qualidade de vida de um cidadão. Desta forma, é importante garantir o bom funcionamento deste serviço, planeando devidamente os horários e os percursos para que o sistema de transportes públicos não seja negativamente afectado, com filas de espera enormes, atrasos e outros problemas com origem num deficiente planeamento [1].

Este trabalho teve como principal objetivo providenciar aos utentes de transportes públicos, nomeadamente aos utentes dos Horários do Funchal, um meio alternativo ao portal *online* existente, que disponibilizasse algumas das informações mais importantes para o utente, em conjunto com novos serviços. Estes serviços podem acrescentar mobilidade aos utentes, dando acesso à informação quando os utilizadores estão fora de casa ou fora do trabalho sem acesso ao computador e à *internet*. Ao mesmo tempo, fornecer mais informação à empresa dos Horários do Funchal, de modo a que esta possa melhor organizar as viagens.

1.2. Problema

Um dos problemas que os utilizadores de transportes públicos enfrentam é o de receber a informação necessária, na altura certa, de uma forma simples e rápida. Neste momento, a empresa Horários do Funchal tem alguns serviços que auxiliam os passageiros com essas informações, como o portal *online* e o serviço de horários em tempo real por SMS. Mas, estes serviços requerem *internet* ou atribuem um custo para o passageiro, como o serviço por SMS, que estão limitados em relação à quantidade de informação que pode fornecer ao utilizador.

A empresa também é afetada por este problema, uma vez que é conveniente para a empresa fornecer esta informação, na altura certa, à maior quantidade de passageiros. Assim, terá mais utilizadores a usufruir dos seus serviços e também poderá ajudar a reduzir os custos que a empresa tem com a manutenção do *website*

para dar resposta à grande quantidade de utilizadores e o custo dos SMS, sendo que o utilizador paga metade e a empresa paga a outra metade.

Além do referido, a empresa gostaria de conhecer ao máximo as necessidades dos seus utentes, pois assim poderá melhor gerir os seus recursos e dar uma resposta mais eficaz.

1.3. Contribuições

Esta dissertação apresenta uma aplicação móvel que pode ser utilizada em *Smartphones* e *Tablets* com o sistema operativo Android, com o intuito de ajudar os seus utilizadores, tendo como público-alvo os utentes frequentes de transportes públicos jovens e adultos. Naturalmente, não se exclui a possibilidade de ser utilizada por outro tipo de utilizadores, como turistas entre outros, que necessitem de informações relativamente aos transportes públicos dos Horários do Funchal.

A aplicação será disponibilizada gratuitamente, sem nenhum encargo na sua aquisição, nem com a sua utilização. Porém, se o utilizador quiser utilizar a aplicação num local sem *internet* terá que ter realizado previamente uma sincronização com o servidor, através de uma ligação Wi-Fi ou, então, terá que ter uma ligação à *internet* através de um plano de dados.

Resumindo, o utilizador será capaz de realizar com esta aplicação as seguintes funcionalidades:

- Visualizar os horários de todas as carreiras existentes na rede;
- Visualizar a hora de saída do próximo autocarro;
- Visualizar o tempo restante até o autocarro chegar a paragem selecionada;
- Visualizar a rota e posição do(s) autocarro(s) de uma determinada carreira selecionada;
- Agendar, Editar ou Remover alarmes para avisar quando o autocarro sai;
- Pesquisar por locais utilizando o mapa da API do Google;
- Pesquisar por percursos da origem até o destino;
- Adicionar/Remover carreiras dos favoritos;

- Ser notificado quando o autocarro sai da primeira paragem.

1.4. Metodologia

De modo a alcançar os objetivos propostos, começou-se por realizar um estudo de satisfação do utente de transportes públicos, focando os utentes dos Horários do Funchal. Através desse estudo, tentou-se compreender quais as necessidades dos utentes locais, tendo-se também realizado um *brainstorming* com alguns destes utentes.

Realizado o *brainstorming*, construiu-se um conjunto de *user stories*, os quais foram entregues a utentes dos Horários do Funchal e foi pedido que fossem ordenados do mais importante para o de menor importância. Posteriormente, foi atribuído uma cotação, sendo a maior para o primeiro cartão escolhido e a menor para o último cartão. Somada a cotação atribuída por cada utente, constataram-se quais as funcionalidades mais relevantes.

Com as funcionalidades definidas, pesquisaram-se outras aplicações que focassem algumas destas funcionalidades e analisaram-se. Feito isto, desenharam-se os *mockups* e realizaram-se testes de usabilidade, de modo a corrigir erros de interação.

Com base nos *mockups* finais, procedeu-se à implementação de um primeiro protótipo e realizaram-se novos testes de usabilidade. O método de desenvolvimento utilizado consistiu em falar com o cliente, desenvolver pequenos protótipos e testá-los consecutivamente, aperfeiçoando-os. Adaptou-se algumas técnicas utilizadas no *Agile Development*, como por exemplo os *Sprints*, cujo significado é rápido, e com esta abordagem é possível conceber um protótipo do sistema rapidamente, testá-lo e voltar a redesenhar se não estiver bom o suficiente, sem que se perca muito do trabalho realizado [5].

Por fim, realizou-se um estudo de utilização em relação à aplicação final, com os utentes que a utilizaram no seu dia-a-dia. Depois, procedeu-se às correcções finais, e ao lançamento nos mercados de aplicações *online*, como o Google Play e a Samsung Apps.

1.5. Organização da Dissertação

Esta dissertação encontra-se dividida em oito capítulos que descrevem todos os procedimentos utilizados na realização desta dissertação. Estes capítulos são:

- **Introdução** – Na introdução encontram-se tópicos como o contexto em que esta dissertação se insere, a motivação, o problema e a metodologia utilizada.
- **Tecnologias Utilizadas** – Este capítulo aborda todas as tecnologias utilizadas para a realização desta dissertação.
- **Desenho do Sistema** – Contém todos os passos e diagramas que descrevem a arquitectura e o desenho do sistema.
- **Implementação** – Descreve como todas as funcionalidades do sistema foram implementadas e interligadas na aplicação.

- Testes e Resultados – Este capítulo descreve os testes realizados e os resultados obtidos ao longo do desenvolvimento da aplicação.
- Lançamento – Neste tópico são descritos quais os mercados onde a aplicação foi lançada e quais os resultados obtidos.
- Conclusões – Por fim, são apresentadas as conclusões obtidas com esta dissertação.

2. Trabalhos Relacionados

“The true sign of intelligence is not knowledge but imagination.”

(Albert Einstein).

De modo a compreender-se as necessidades de um utente de transportes públicos, compilou-se diversos artigos referentes a aplicações e métodos utilizados nesta área.

2.1. *Field Trial of Tiramisu: Crowd-Sourcing Bus Arrival Times to Spur Co-Design*

O projeto *Tiramisu* é baseado numa aplicação móvel desenvolvida para um estudo, onde os seus autores tiveram como objetivo utilizar tecnologia *crowd-sourcing* para melhorar os transportes públicos, nomeadamente, os autocarros [6]. Tendo por base este objetivo, os autores criaram um sistema que permite, através de um *smartphone*, partilhar a localização GPS do utilizador quando este está no autocarro e também o estado do autocarro, (se está vazio, meio-cheio ou lotado). Com estas informações e com alguns cálculos de computador, foi possível fornecer a outro utilizador informações muito fiáveis sobre o estado e localização do autocarro.

Os resultados obtidos neste projeto foram, no geral, muito bons. Criaram-se três grupos de utilizadores, os estudantes, os trabalhadores a tempo-parcial e os trabalhadores a tempo-inteiro, sendo este último grupo o que mais utilizou a aplicação, registando acessos em 68% das suas viagens. O que menos acessos, registou foi o grupo de estudantes, registando apenas 48% das suas viagens, levando a concluir que todos os utilizadores registaram pelo menos metade das suas viagens. Quando questionados sobre as funcionalidades da aplicação, os participantes na sua maioria deram *feedback* positivo, sendo a informação em tempo real uma das funcionalidades que mais gostaram, quando esta estava disponível. Refira-se que os dados em tempo real nem sempre estavam disponíveis, devido à quantidade de participantes, já que esta informação era partilhada pelos utilizadores e estes nem sempre a partilhavam. Mas, se existir uma grande adesão ao sistema, o *crowd-sourcing* pode funcionar como meio de fornecer informação num sistema de transportes públicos.

2.2. *Location-Aware Tools for Improving Public Transit Usability*

Em “*Location-Aware Tools for Improving Public Transit Usability*” é apresentada uma proposta para incentivar as pessoas a utilizarem mais os transportes públicos através de uma aplicação móvel, designada *OneBusAway*, que utiliza informações como a localização de paragens e os horários em tempo real para mostrar ao utilizador qual ou quais as opções que ele tem a partir da sua localização [7]. Esta abordagem teve resultados excelentes com 87% dos inquiridos a classificar positivamente a aplicação e com maior número de acessos à funcionalidade de “Guardar Paragens nos Favoritos” e a funcionalidade de procurar “Paragens aqui por Perto”. Além destes resultados, ainda questionaram os utilizadores, quanto às mudanças que esta aplicação trouxe para o seu dia-a-dia e os resultados foram, na sua maioria, muito satisfatórios, com melhorias no tempo de espera pelo autocarro e também, uma menor distância a percorrer.

2.3. *Bus Catcher: A Context Sensitive Prototype System for Public Transportation Users*

O *Bus Catcher* é um projeto destinado a utentes de transportes públicos que utilizem sistemas móveis para comunicarem com a rede de transportes públicos, de forma a obter toda a informação necessária para a viagem do passageiro [8]. Esta aplicação tem como base o sistema GPS da rede de transportes públicos, para que o utente possa controlar a posição no mapa do autocarro que pretende apanhar. Além da principal funcionalidade de monitorização, a aplicação inclui outras funcionalidades, como selecionar o ponto de origem e o ponto de destino, sendo apresentado no mapa qual, ou quais, os autocarros que podem ser escolhidos. Para testar esta aplicação os autores utilizaram uma amostra que representava diversas classes de utilizadores e de diversas idades. Os resultados desta aplicação mostraram uma grande aceitação por parte dos utilizadores que responderam positivamente quando questionados sobre a utilidade da aplicação e a possibilidade de alargar o sistema a mais zonas. O único aspeto negativo mencionado tinha haver com a questão da ligação por dados ser lenta e limitar a resposta do sistema, tornando-o lento.

2.4. Avaliação da FXBus e de outras aplicações

Como é descrito nos capítulos 2.1, 2.2 e 2.3, existem inúmeras aplicações nativas para *Smartphones* e outros dispositivos semelhantes, com o objetivo de melhorar a rede de transportes públicos. Estas aplicações podem ser classificadas quanto ao tipo de serviços que oferecem e ao tipo de transportes públicos (autocarros, comboios ou metro). Realizou-se, então, uma pesquisa de modo a compreender que outras aplicações existiam e que tipo de serviços eram oferecidos em função da rede transportes a que a aplicação se destinava.

A Tabela 2.1 mostra a aplicação que foi realizada neste trabalho e outras nove aplicações que já existem e que têm o mesmo objetivo. A tabela tem o nome das aplicações, respetivas localidades a que se destinam e que tipos de transportes públicos cobrem.

Tabela 2.1 – Resumo das aplicações avaliadas

Aplicação	País, Localidade	Tipo de Transportes Públicos		
		Autocarros	Comboios	Metro
IZI Carris	Portugal, Lisboa	Sim	Não	Não
Lisboa Move-me	Portugal, Lisboa	Sim	Sim	Sim
Metro Lisboa	Portugal, Lisboa	Não	Não	Sim
Madrid Metro Bus Cercanias	Espanha Madrid	Sim	Sim	Sim
Moobly	Brasil, Porto	Sim	Não	Não
SMTUC Mobile	Portugal, Coimbra	Sim	Não	Não
TUB Mobile	Portugal, Braga	Sim	Não	Não
OneBusAway	EUA, Washington, Seattle	Sim	Não	Não
Tiramissu	EUA, Pennsylvania, Pittsburgh	Sim	Não	Não
FXBus	Portugal, Madeira	Sim	Não	Não

De modo a poder-se avaliar as aplicações quanto às suas funcionalidades criaram-se cinco grupos, nos quais são descritas diversas funcionalidades e atribuiu-se cotações

a cada funcionalidade. Estes grupos foram criados de acordo com o tipo de serviço e a cotação foi atribuída segundo o grau de importância para o utilizador, verificado através do *brainstorming* e do inquérito realizado aos utentes.

- Grupo Horários – Este grupo contém todas as funcionalidades possíveis quanto às horas de saída dos transportes públicos e outros mecanismos para auxiliar o utilizador.
 - Horários diários (2 valores);
 - Horários em tempo real (3 valores);
 - Notificações ou alarmes (3 valores);
 - Próximos transportes públicos a sair (2 valores).
- Grupo Localização – O segundo grupo contém todas as funcionalidades referentes a informações representadas no mapa.
 - Posição das paragens de transportes públicos (3 valores);
 - Rota dos transportes públicos (2 valores);
 - Posição dos transportes públicos (2 valores);
 - Posição GPS do utilizador (3 valores).
- Grupo Informações – O grupo três contém funcionalidades referentes a informações úteis ao passageiro de transportes públicos.
 - Adicionar favoritos (1 valor);
 - Lista de carreiras (2 valores);
 - Pesquisar por carreiras (2 valores);
 - Lista de paragens da carreira (2 valores);
 - Pesquisar por paragens da carreira (1 valor);
 - Preçário (1 valor);
 - Outras informações (1 valor).
- Grupo Língua/Dispositivo – Este grupo diz respeito à capacidade da aplicação ser desenvolvida para diferentes línguas e para diferentes dispositivos.

- Língua nativa (1 valor);
 - Uma língua estrangeira (2 valores);
 - Outras Línguas (2 valores);
 - *Smartphone* (3 valores);
 - *Tablet* (2 valores).
- Grupo Itinerários – O último grupo contém funcionalidades relativas ao cálculo de itinerários.
 - Pesquisar por Itinerários (3 valores);
 - Guardar histórico de itinerários pesquisados (2 valores);
 - Guardar lugares favoritos (2 valores);
 - Pesquisar por lugares no mapa (3 valores).

Se se avaliarem os grupos acima descritos, concluir-se-á que as cotações de cada grupo de funcionalidade, somadas, resultarão num total de dez valores.

Usando essa escala pode-se então avaliar as diferentes aplicações e compará-las com a aplicação desta dissertação (FXBus).

Começou-se, então, por avaliar a aplicação *Tiramissu*, que obteve resultados razoáveis como se pode observar no Gráfico 2.1. Esta aplicação não contém qualquer funcionalidade no grupo de Itinerários, mas por outro lado é a única que utiliza o *crowd-sourcing* como parte do sistema de informação.

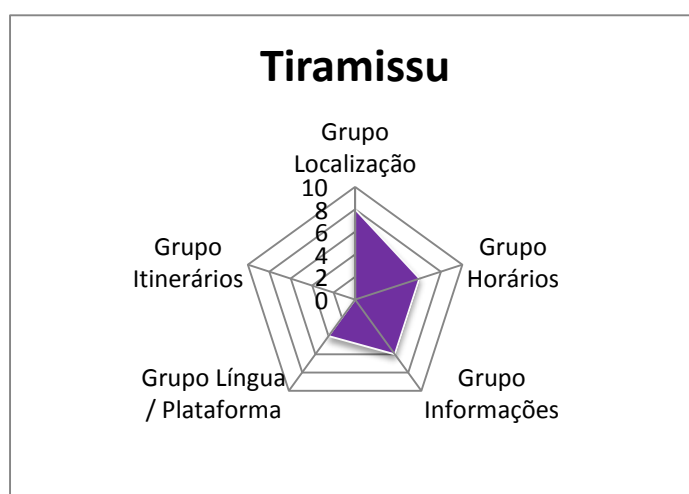


Gráfico 2.1 – Avaliação da aplicação Tiramissu

Observando os gráficos das restantes aplicações, pode-se concluir que as aplicações que obtiveram uma maior área foram o TUB Mobile (Gráfico 2.7), o Metro Lisboa (Gráfico 2.2), Lisboa Move-me (Gráfico 2.4), Moobly (Gráfico 2.5) e o One Bus Away (Gráfico 2.8).

Aplicações como o Madrid Metro (Gráfico 2.3), SMTUC Mobile (Gráfico 2.6) e Izi Carris (Gráfico 2.9), obtiveram áreas muito pequenas, tornando-as aplicações com menos funcionalidades, inclusivamente em alguns grupos, como o grupo de localização e itinerários, não obtiveram pontuação (sem funcionalidades) ou obtiveram baixa pontuação.

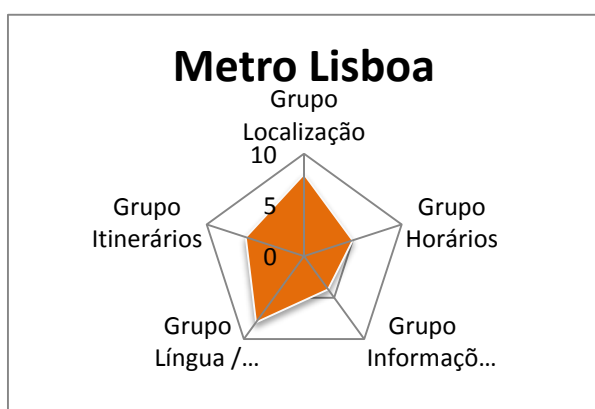


Gráfico 2.2 – Avaliação da aplicação Metro Lisboa

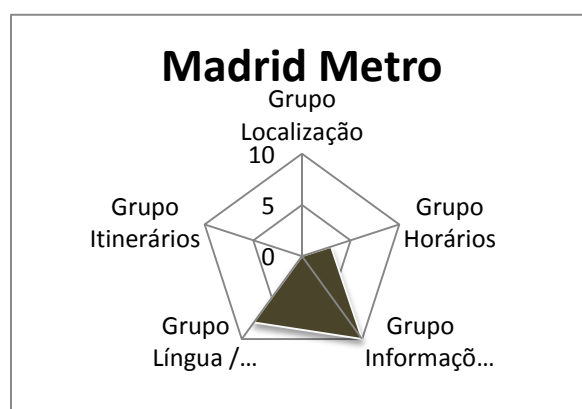


Gráfico 2.3 – Avaliação da aplicação Madrid Metro

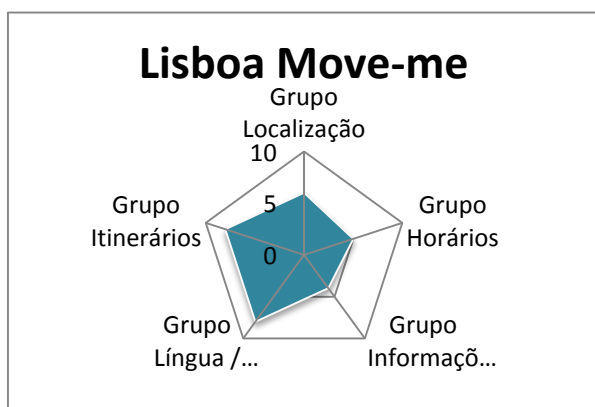


Gráfico 2.4 – Avaliação da aplicação Lisboa Move-me

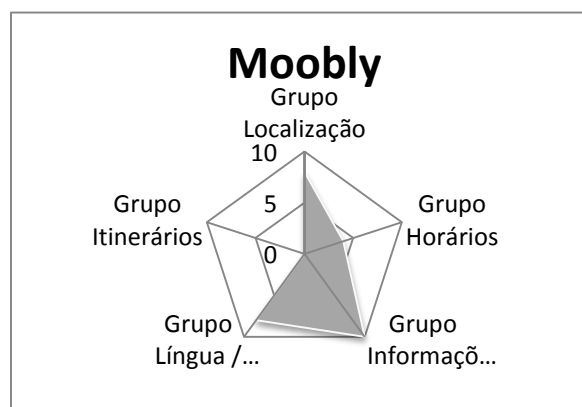


Gráfico 2.5 – Avaliação da aplicação Moobly

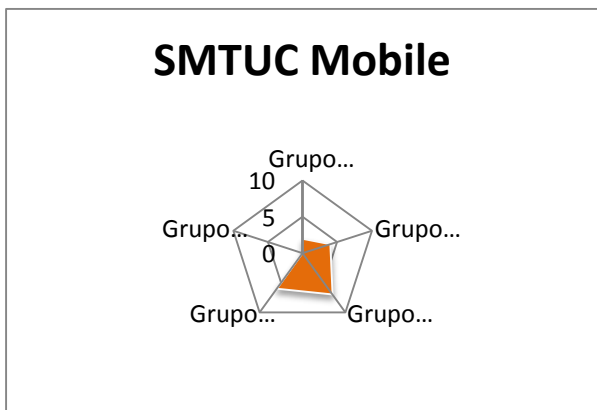


Gráfico 2.6 – Avaliação da aplicação SMTUC Mobile

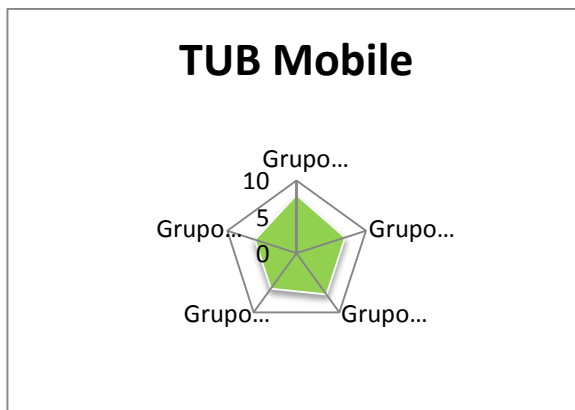


Gráfico 2.7 – Avaliação da aplicação TUB Mobile

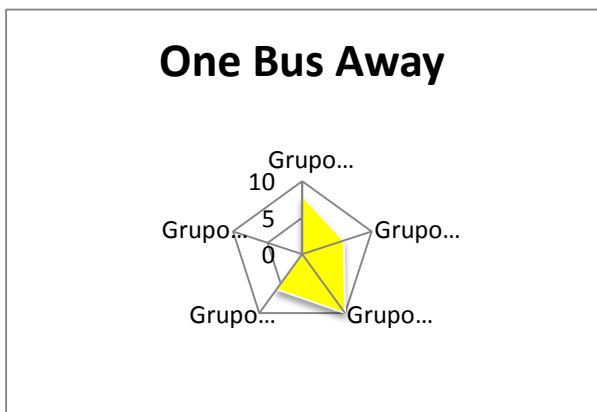


Gráfico 2.8 – Avaliação da aplicação One Bus Away

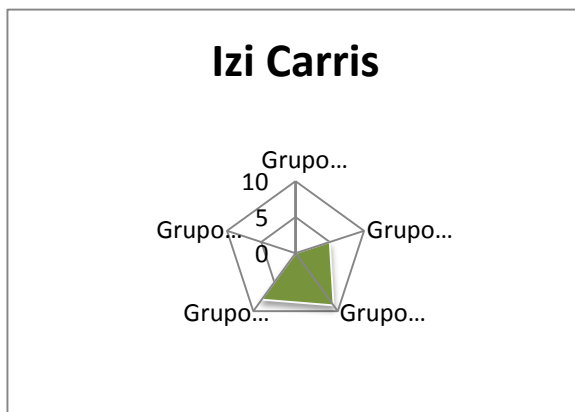


Gráfico 2.9 – Avaliação da aplicação Izi Carris

Utilizando os mesmos critérios para avaliar a aplicação desenvolvida neste trabalho, obtiveram-se resultados satisfatórios, como se pode observar no Gráfico 2.10. Obteve-se a pontuação máxima nos grupos itinerários, localização e horários. Nos grupos língua/dispositivos e informações faltam algumas funcionalidades que poderiam ter sido acrescentadas.

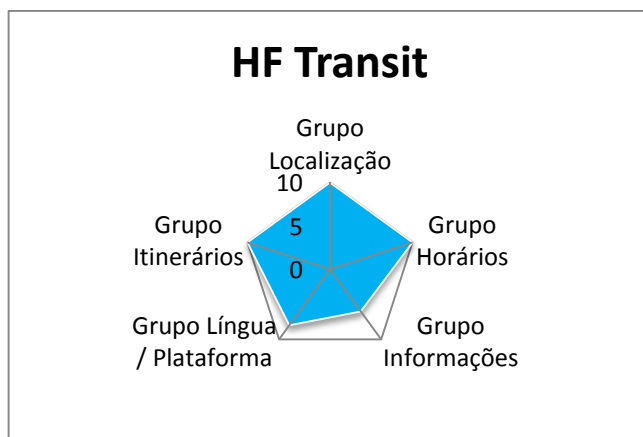


Gráfico 2.10 – Avaliação da aplicação desta dissertação (FXBus)

3. Tecnologias Utilizadas

“Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them.”

(Steve Jobs).

A tecnologia que se escolhe quando se começa um projeto pode definir o rumo e o produto final. Assim, realizar a análise de quais são as melhores tecnologias para resolver o problema é uma decisão crítica para o projetista e devem ser tidos em conta factores como a capacidade de escalabilidade, de flexibilidade para alterar no futuro e de compatibilidade com os dispositivos do público-alvo.

3.1. **Smartphones e Tablets**

Os pequenos dispositivos móveis deixaram de ser tão pequenos em termos de capacidade de processamento. Segundo Mark Weiser (O fundador da computação ubíqua) os dispositivos no futuro tenderão a se tornar cada vez mais ubíquos, ou seja, menos perceptíveis aos seus utilizadores na medida em que a computação está a acontecer e o seu utilizador nem se dá conta [9].

Mark Weiser e os seus colegas defendiam que os dispositivos convergiriam em três tipos, *Tabs*, *Pads* e *Boards*, sendo que hoje em dia já se pode observar que os *Tabs* e *Pads*, conhecidos hoje como *Smartphones* e *Tablets*, já se encontram aos milhões, de diversas marcas conhecidas, como a Apple e a Samsung. Os *Boards* começam agora a surgir com mais frequência nas escolas e nos escritórios, conhecidos como quadros interativos, mas também nas casas, onde os televisores começam a convergir nas novas *Smart TV*.

Os sistemas operativos para estes dispositivos tiveram de ser capazes de albergar uma grande variedade de sensores e de adaptar-se a diferentes resoluções e tamanhos de ecrãs. Entre os mais conhecidos encontram-se os sistemas operativos Android, da Google, iOS, da Apple e o Windows Phone, da Microsoft. Na Tabela 3.1 pode-se observar o crescimento dos sistemas operativos a nível global para o ano 2012 e a previsão para 2016.

Tabela 3.1 – Mercado global de sistemas operativos para dispositivos móveis [10]

Sistema Operativo	Vendas 2012 (milhões)	Cota de Mercado 2012	Cota de Mercado 2016	Crescimento 2012 - 2016
Android	497.1	68.8%	63.8%	16.3%
iOS	135.9	18.8%	19.1%	18.8%
Blackberry OS	32.5	4.5%	4.1%	14.6%
Symbian	23.9	3.3%	N/A	N/A
Windows Phone	17.9	2.5%	11.4%	71.3%
Linux	N/A	2.0%	1.5%	10.5%
Others	15.1	2.1%	0.1%	N/A
Total	722.4	100%	100%	100%

Com base na tabela anterior, pode-se comprovar que o sistema operativo da Android é líder de vendas, com 479.1 milhões de vendas e o iOS em segundo lugar, com 135.9 milhões. O Windows Phone ainda não ocupa um lugar nos primeiros três, mas pode-se observar que a previsão na taxa de crescimento é a maior com 71,3%, tornando-o num candidato aos primeiros lugares nos próximos anos.

Pretendeu-se saber se os utilizadores de transportes públicos dos Horários do Funchal seguem a mesma tendência do mercado global. Com esse fim, realizou-se um estudo para responder a esta e a outras questões. Os resultados obtidos foram dentro do esperado, como se pode verificar através do Gráfico 3.1. O inquérito obteve um total de 138 respostas, dos quais 66 eram utilizadores de transportes públicos e tinham dispositivos móveis. Desses, 77% tinham pelo menos um dispositivo móvel com Android instalado, seguindo do iOS com 11% e do Windows Phone com apenas 4%. Estes resultados mostram claramente que a região da Madeira segue a tendência global e que o SO Android é dominante.

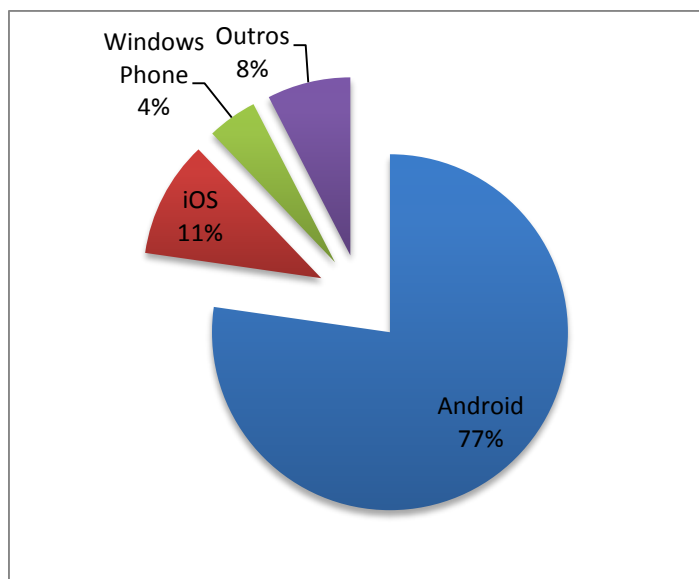


Gráfico 3.1 – Distribuição dos Sistemas Operativos dos utilizadores dos Horários do Funchal com dispositivos móveis do tipo *Smartphone*

3.2. Plataforma Android

O OS Android contém diversas versões, desde a primeira versão (Android 1.0 Base) à última versão atual 4.2 (Jelly Bean) [11].

A primeira versão que impulsionou o Android foi a versão 1.5 (Cupcake) em maio de 2009. Esta versão, embora primitiva, já era dotada de diversas funcionalidades que ainda hoje se podem encontrar nas novas versões, e além disso estava disponível em 27 línguas e já suportava diferentes tamanhos e resoluções de ecrãs [12].

Depois, seguiram-se novas versões, como o Android 1.6 (Donut), 2.1 (Eclair), 2.2 (Froyo), 2.3 (Gingerbread), 3.2 (Honeycomb), 4.0 (Ice Cream Sandwich) e 4.2 (Jelly Bean), entre outras. Estas versões trazem novas funcionalidades em relação às anteriores e cada vez mais suportam novos dispositivos como *Smartphones*, *Tablets* e até televisores, de diferentes dimensões e com diferentes resoluções [11].

Com tantas versões no mercado é normal haver uma segmentação. De todas as versões existem algumas que dominam, albergando a grande maioria dos clientes, e é preciso que uma aplicação suporte o máximo de dispositivos possíveis sem perder funcionalidades ou capacidade. Entre as diferentes versões, a que domina no mercado é o Jelly Bean, com 38% de cota de mercado, logo de seguida a versão Gingerbread com 34% e a Ice Cream Sandwich com 23% [13]. As outras versões têm

cotas muito pequenas e tendência a diminuir. O Gráfico 3.2 representa a segmentação no mercado entre as versões do Android.

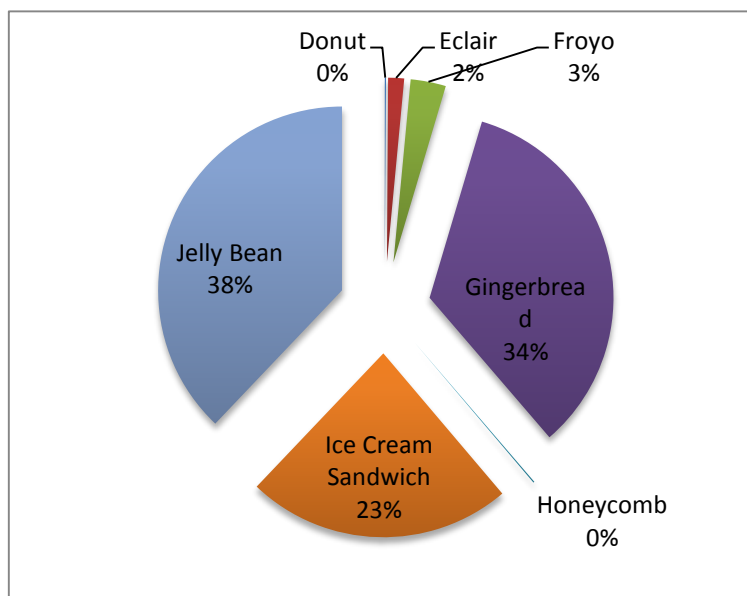


Gráfico 3.2 – Cota de mercado dos dispositivos Android por versão do OS

Além da segmentação por plataforma, pode-se ainda ver qual é a segmentação por dispositivo, de acordo com a respetiva resolução e densidade de ecrã. Neste ponto pode-se observar claramente que a resolução mais utilizada é a da categoria *Normal*, com 80% de cota dos dispositivos, seguido do *Small* com 10%, *Large* com 6% e *Xlarge* com 4%, como pode-se observar no Gráfico 3.3 [13].

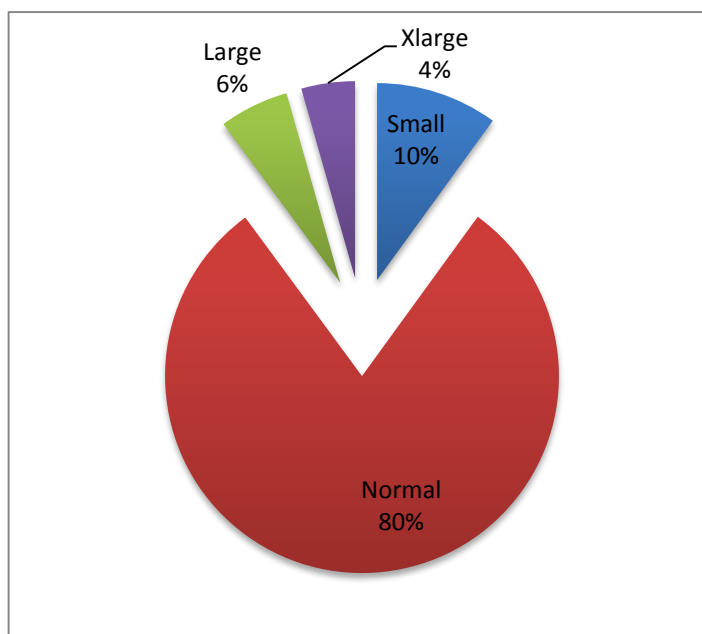


Gráfico 3.3 – Cota de mercado dos dispositivos Android por tamanho do ecrã

A densidade mais comum é a da categoria *HDPI* com 36% de cota, seguido da *XHDPI* com 25%, a *MDPI* com 23% e a *LDPI* com 10%. As restantes densidades, a *XXHDPI* com 5% e a *TVDPI* com 1% têm menos cota, provavelmente por terem aparecido por último no mercado e ainda não existirem muitos dispositivos com estas densidades [13].

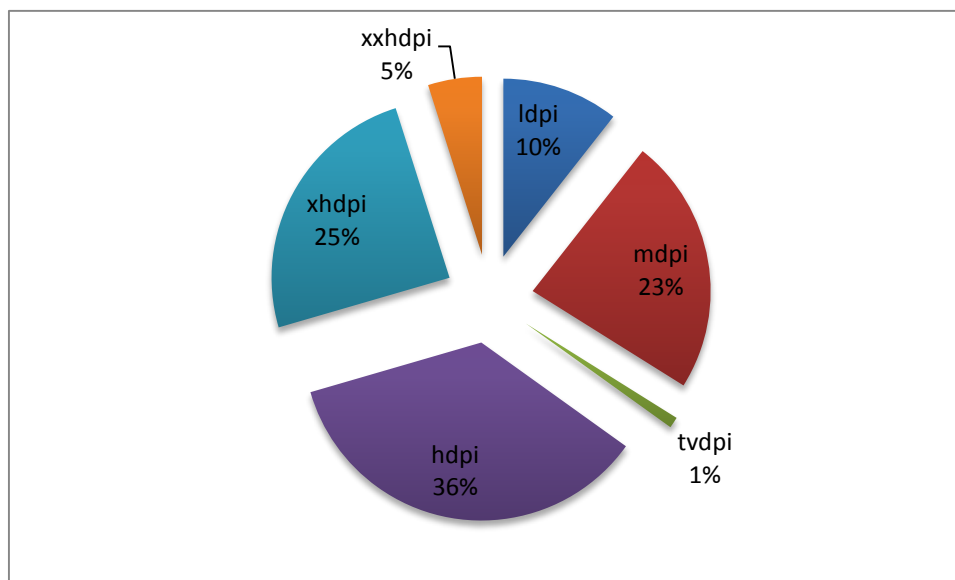


Gráfico 3.4 – Cota de mercado dos dispositivos Android por densidade do ecrã

Se se compararem os três gráficos acima (Gráfico 3.2, Gráfico 3.3 e Gráfico 3.4) vê-se que os dispositivos mais vendidos são os que vêm com o sistema operativo Android 4.2 Jelly Bean, com tamanhos e resoluções pertencentes à categoria Normal e com a densidade *HDPI*.

3.3. Aplicações *Web* vs Nativas

Tanto as aplicações *web* como as aplicações nativas têm vantagens e desvantagens, umas em relação às outras. Por exemplo, ao analisarem-se as vantagens das aplicações *web* em relação às aplicações nativas, encontram-se aspetos como a versatilidade de a mesma aplicação suportar diferentes dispositivos e diferentes sistemas operativos. No entanto, o facto de existirem diferentes *web browsers* constitui uma dor de cabeça para os desenvolvedores, isto porque os diferentes *web browsers* interpretam os estilos CSS de diferentes formas, fazendo com que a mesma aplicação tenha aspetos diferentes, em diferentes *web browsers*. A utilização de *frameworks* tem vindo a combater este problema, tornando-o cada vez menor. Um

exemplo destas *frameworks* é o JQuery (JavaScript library) que trabalha aspetos como a interface, a interação e a programação com base em Javascript [14].

Se se observarem as desvantagens das aplicações *web*, apercebe-se que, quando toca a dispositivos móveis, estas desvantagens são enormes, dado que as aplicações nativas têm acesso a diferentes níveis de memória do dispositivo, sensores e serviços, sendo que uma aplicação *web* está muito limitada neste aspeto, apenas consegue a localização do utilizador e pouco mais. Se se pretender utilizar os sensores do telemóvel para algumas funcionalidades ou para guardar dados localmente, uma aplicação nativa é a melhor alternativa.

No caso desta dissertação, a empresa Horários do Funchal já possui uma aplicação *web* móvel. Contudo, ao analisar-se esta aplicação facilmente apercebe-se que a aplicação está um pouco estática, com as informações em formatos estáticos como o PDF. Na Figura 3.1 pode-se observar a interface da aplicação *web* móvel dos Horários do Funchal.



Figura 3.1 – *Home page* da aplicação *web* móvel dos Horários do Funchal

No inquérito realizado aos utilizadores dos Horários do Funchal (ver Anexo A) identificaram-se diversos problemas, para os quais uma aplicação móvel nativa pode ser a resposta. Por exemplo, 60% dos utilizadores que responderam ao inquérito e utilizavam os Horários do Funchal como meio de transporte público, tinham um dispositivo móvel, mas apenas 44,5% tinham acesso à internet móvel e, mesmo assim, com tarifários limitados.

Tendo em conta que 97% desses utilizadores utilizam o *website* como principal fonte de consulta de informações, e apenas 44,5% tem acesso a internet móvel, faz com que mais de metade esteja privada do acesso a informações através do *website* quando está em movimento.

3.4. Ambiente de Desenvolvimento

Ao pesquisar-se sobre ambientes de desenvolvimento para a plataforma Android encontram-se várias aplicações, o NetBeans, o Eclipse e a recente oferta da Google, o Android Studio. No entanto, na página oficial da Android Developer [15] encontram-se apenas duas aplicações para fazer o *download*, o Eclipse e o Android Studio.

3.4.1. Eclipse

Escolheu-se o Eclipse para programar a aplicação para esta dissertação visto que na altura apenas estava disponível o NetBeans e o Eclipse e este último tem algumas vantagens em relação ao NetBeans, como por exemplo a possibilidade de instalar o *plugin* do *ADT*, que permite editar os *views* da interface gráfica sem ser pelas *tags* XML.

Para instalar o Eclipse, pode-se realizar o *download* no site da *Android Developers*, na secção de *develop, tools* e aí escolhe-se a versão que já traz o Eclipse, o *plugin ADT* e o *Android SDK*, tudo num só pacote. Na Figura 3.2 pode-se observar o aspeto gráfico da aplicação [15].

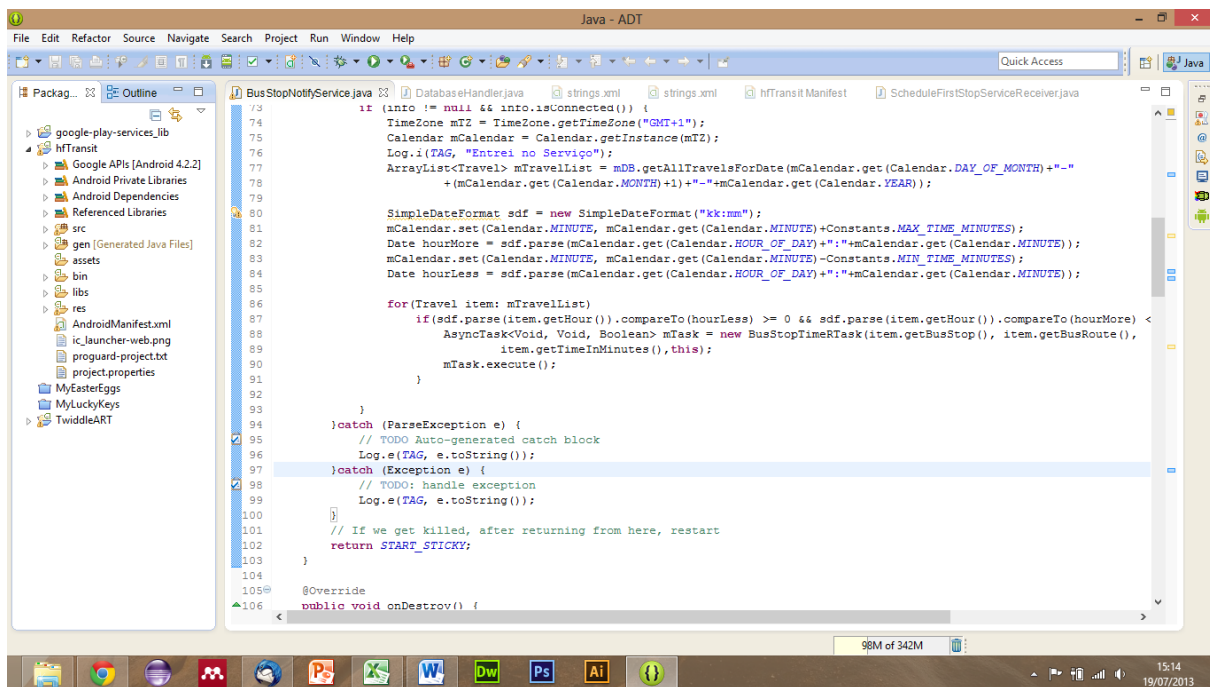


Figura 3.2 – Ambiente de trabalho do eclipse

Para criar um novo projeto, basta ir ao *File, New, Android Application Project*, sendo depois apresentada uma nova janela, onde se tem de escolher o nome para a aplicação, o nome para o projeto e o nome para o *package*. Também é requerido que se escolha qual é o SDK da Android para o qual se pretende compilar e qual é a versão menor a que se pretende dar suporte (ver Figura 3.3). Depois deste ponto, é só clicar seguinte (*Next*) e é gerado um novo projeto com o *template* e algumas configurações predefinidas.

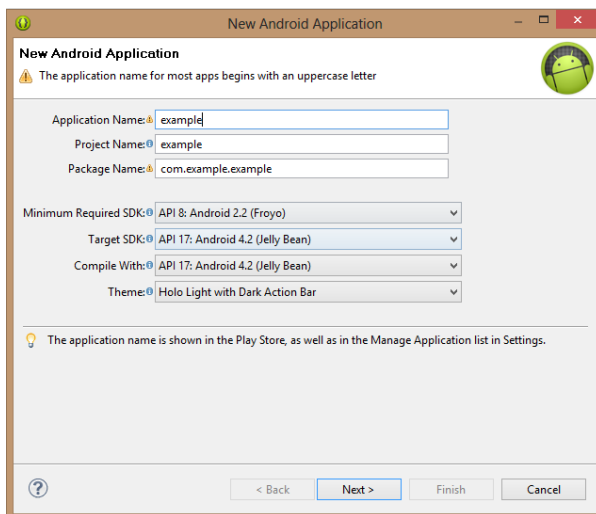


Figura 3.3 – Janela de criação de um novo projecto da Android

3.4.2. Android SDK e AVD

O *SDK da Android* consiste num conjunto de ferramentas e bibliotecas que auxiliam os programadores (*developers*) na sua tarefa de desenvolver aplicações. Como já foi referido, existem diversos *SDK's* e tem-se de saber qual é o nosso público-alvo, pois esse factor é determinante na escolha do *SDK* que, por sua vez, condiciona a programação do programador. Na Figura 3.4 pode-se observar o *SDK Manager*, onde se escolhe quais as versões que se quer fazer o *download* e, posteriormente, instalar [16].

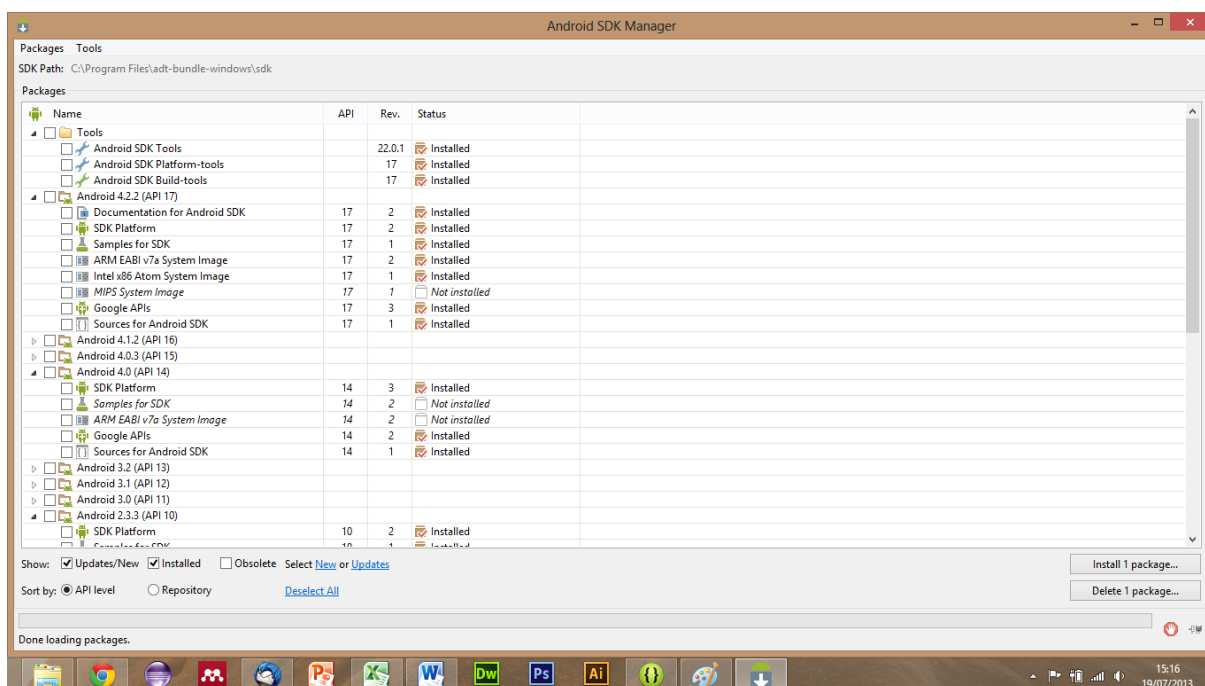


Figura 3.4 – Ambiente gráfico do *SDK Manager*

Também tem-se acesso ao *AVD Manager* a partir do *SDK Manager* (ou a partir do Eclipse). O *AVD* é uma máquina virtual que simula o funcionamento de um dispositivo móvel com o sistema operativo da Android. Estes *AVD* são criados através do *AVD Manager* e têm as funcionalidades que se pretenda, ou seja, escolhe-se uma versão do sistema operativo e também escolhem-se quais os periféricos que se quer que funcionem, como por exemplo *Bluetooth*, *Wi-Fi*, *GPS*, entre outros. Na Figura 3.5 pode-se observar o ambiente gráfico do *AVD Manager*.

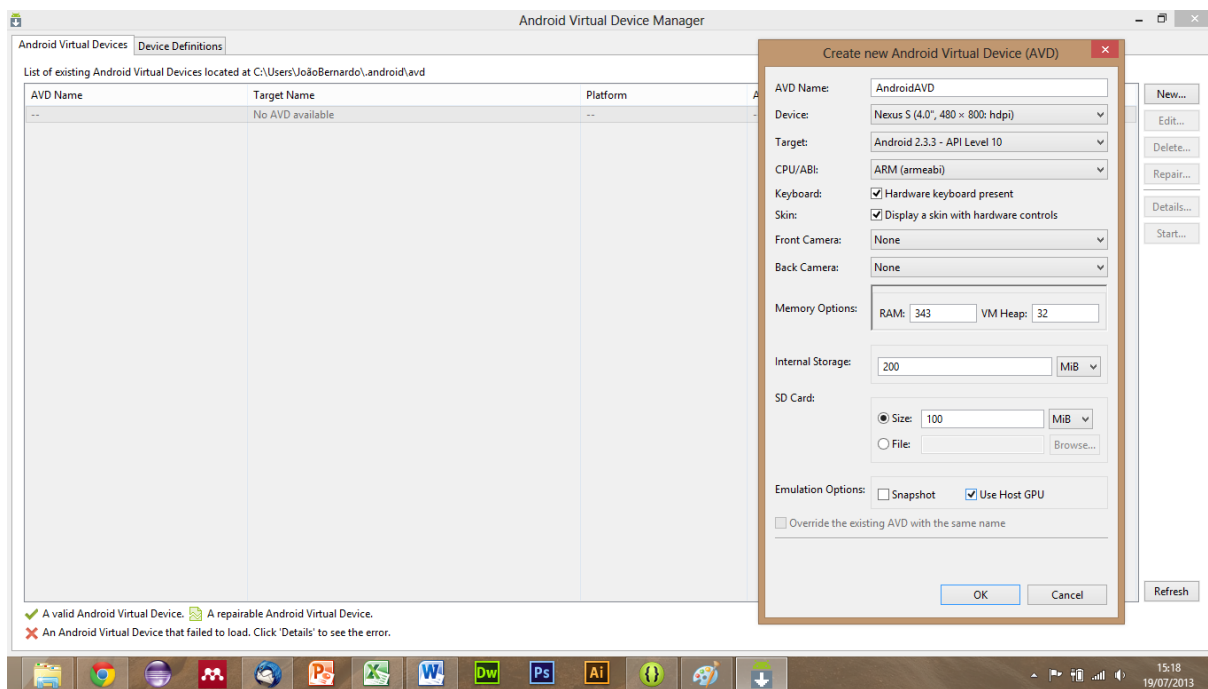


Figura 3.5 – Ambiente gráfico do AVD Manager

Na Figura 3.6 pode-se visualizar um AVD criado em funcionamento.

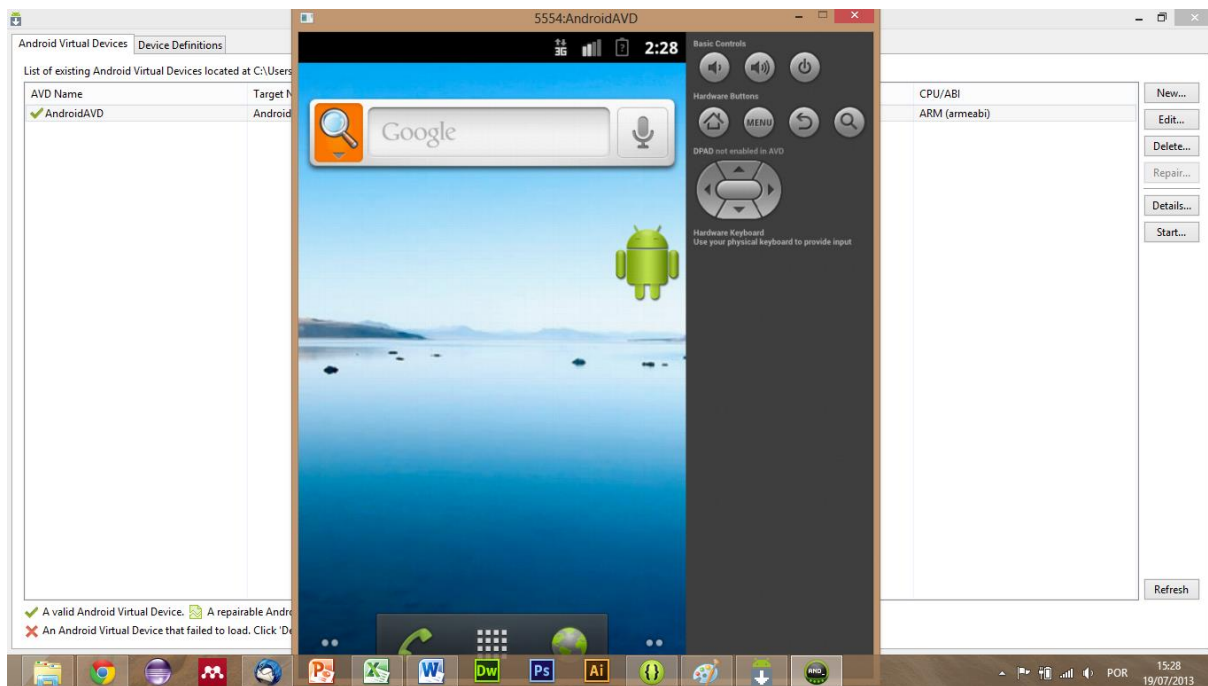


Figura 3.6 – Ambiente gráfico de um AVD como exemplo

3.4.3. Plugin ADT

O *plugin ADT* é uma funcionalidade extra e opcional que se pode instalar no Eclipse, com o intuito de auxiliar no desenho dos *layouts* da aplicação a desenvolver. Este *plugin* é gratuito como todas as outras ferramentas acima descritas neste documento e além de facilitar no desenho do *layout*, ainda permite visualizar o aspeto final para diferentes dispositivos. Pode-se visualizar na Figura 3.7 o ambiente gráfico do *plugin* [17].

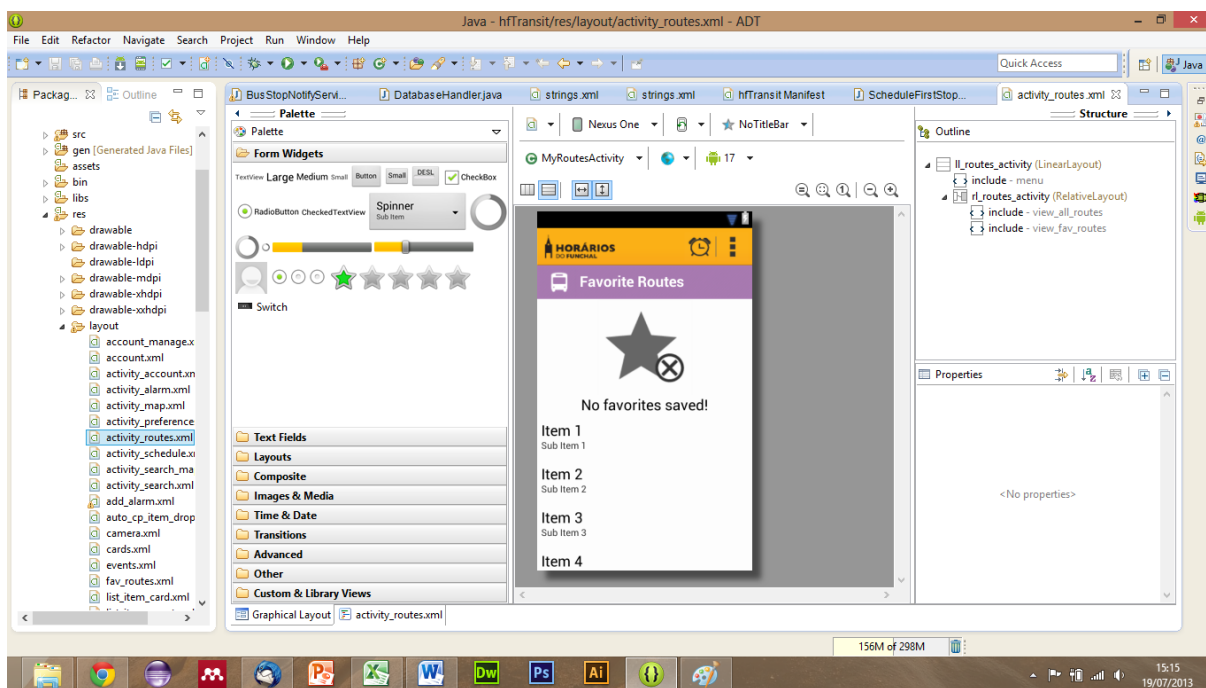


Figura 3.7 – Ambiente gráfico do *plugin ADT*

3.5. Outras Tecnologias Utilizadas

Já se viu que, para começar a programar para Android, é necessária uma série de aplicações e instalações. Mas, se se utilizarem outras tecnologias em conjunto com o Android, ainda é preciso importar bibliotecas para o projeto. Falando do caso específico das bibliotecas e das API's do Google, pode-se obter toda a informação através da consola do Google (sendo necessário criar uma conta) em <https://code.google.com/apis/console/>.

3.5.1. Google Calendar API

De modo a poder-se implementar a funcionalidade de agendamento de viagens utilizou-se o *Google Calendar*. Utilizar o *Google Calendar* para esta funcionalidade traz diversas vantagens, como por exemplo a autenticação que já é feita através do sistema de contas da Android e também todo o processo da sincronização.

Para utilizar o *Google Calendar* existem duas alternativas através da *API* do *Google Calendar* ou através do *Calendar Provider* do *Android SDK*.

Para utilizar a *API* do *Google Calendar* tem-se de ir ao *Google Console* e, na secção serviços, ativar os que se pretendam utilizar, neste caso o serviço *Calendar API*. Este serviço no momento em que foi utilizado era gratuito, até um limite de 10000 pedidos por dia, sendo que a partir desse momento tinha um custo associado [18].

Pelo *Calendar Provider* do *Android SDK* esse limite e custo desaparecem, mas fica-se limitado noutra aspeto, pois só se podem utilizar calendários que já tenham sido criados pelo utilizador. Como a criação de calendários não é uma prioridade para esta funcionalidade, visto que pretende-se só, adicionar as viagens de autocarro à agenda pessoal do utilizador e não, criar um gestor de agendas, optou-se pela utilização desta última opção.

3.5.2. Google Maps Android SDK V2

As funcionalidades da aplicação desta dissertação que envolvam a visualização de pontos *GPS* no mapa foram criadas através da *SDK* do *Google Maps* para Android. Escolheu-se este *SDK*, pois já traz uma série de funcionalidades que auxiliam na visualização destes pontos e é facilmente integrável com a plataforma Android.

Existem outras formas de visualizar pontos *GPS* num mapa, como, por exemplo, o *SDK* da *Bing Maps* para Android, que segundo as críticas é muito bom. Mas, mais uma vez, utilizar todos os sistemas do Google traz vantagens, por exemplo a nível de familiaridade da interface, visto que o *Google Maps* vem de origem no dispositivo Android.

Também poder-se-ia utilizar outras *API's* como o *Sapo Maps* ou o *Yahoo Maps*, mas optou-se pela utilização de um *SDK* para a plataforma que pode guardar os mapas

localmente, sem necessitar de uma ligação à Internet constante, e que traz mais ferramentas, originando menos limitações.

Antes de se começar a utilizar o *Google Maps*, tem-se de executar alguns passos, de forma a ter-se acesso a esta ferramenta. Primeiro, tem-se de ir ao Google Console e ativar o serviço, em seguida tem-se de registar o projeto, o *package* do projeto e o certificado de impressão digital SHA1. Este certificado pode ser encontrado através da linha de comandos, executando o seguinte comando:

- `keytool -list -v -keystore <nome da keystore>.keystore`

Depois de registar o projeto, pode-se importar a biblioteca para a aplicação, e para isso basta no Eclipse ir a *File, Import* e procura na pasta do *SDK* da Android a biblioteca do *Google Play Services*. Antes de clicar para importar, é preciso seleccionar a opção *Copy to Workspace*, de modo a não danificar a biblioteca original (Figura 3.8).

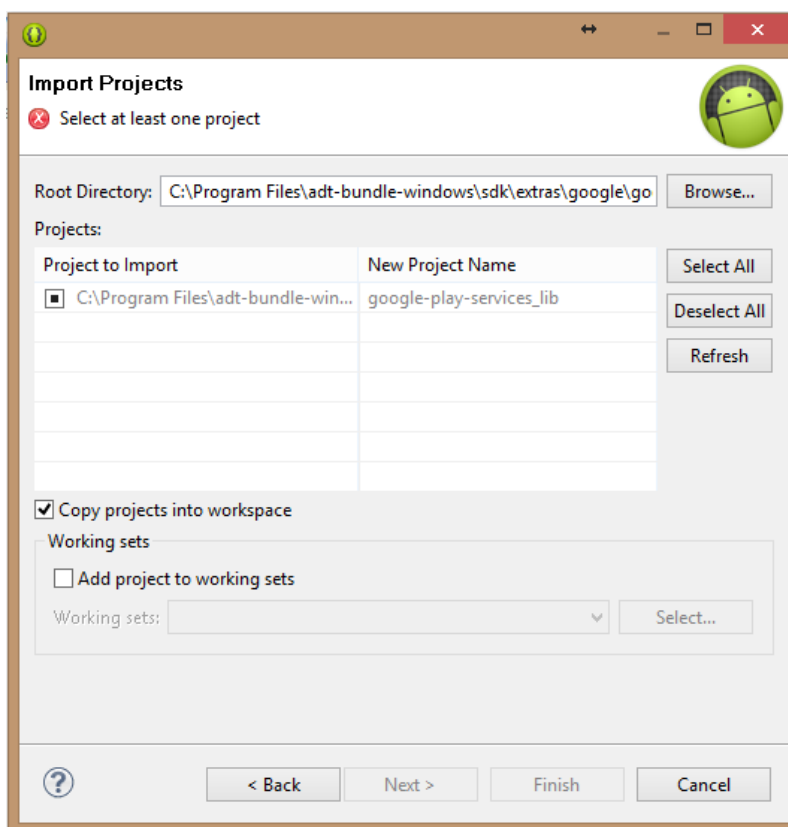


Figura 3.8 – Importação da livreria do *Google Play Services* para o Eclipse

Com a livreria importada para o Eclipse e o projeto criado, falta só interligar os dois. Para isso, basta clicar com o botão direito sobre o projeto criado e, em seguida, seleccionar *Propriedades, Android* e na secção *library* clicar em adicionar (Figura 3.9).

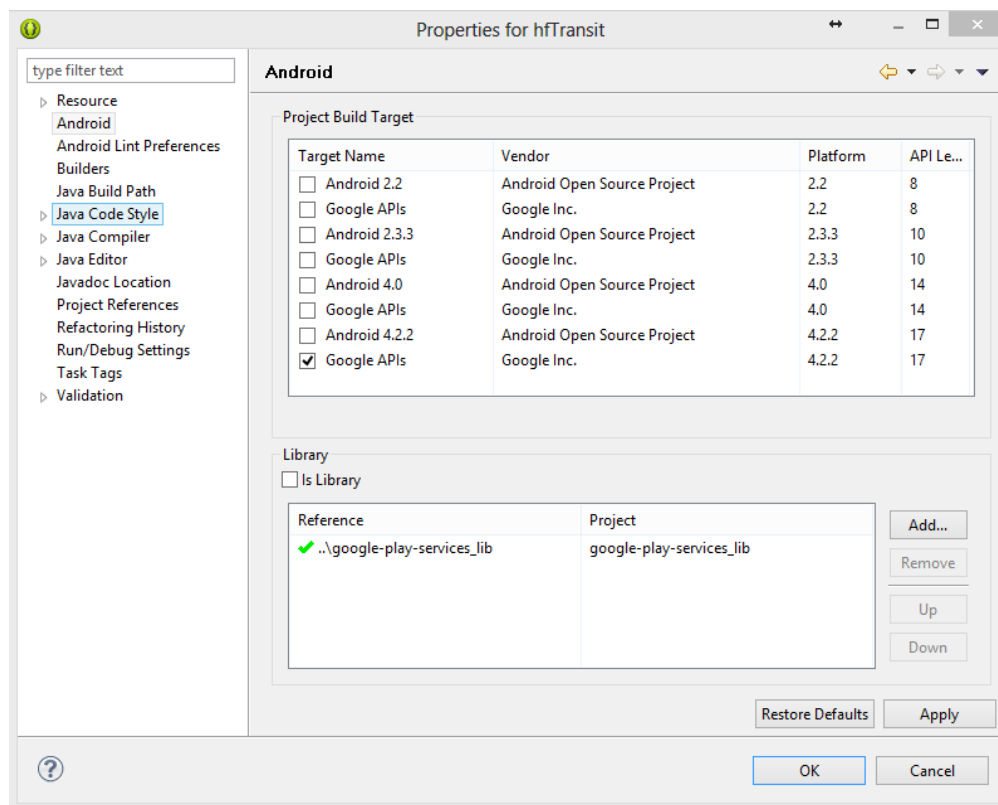


Figura 3.9 – Adicionar livraria ao projecto no Eclipse

A partir deste ponto podem-se utilizar as funcionalidades do *Google Maps* no nosso projeto [19].

3.5.3. Google Places API

Para os utilizadores menos familiarizados com a Madeira, implementou-se uma funcionalidade de pesquisa por lugar. Esta funcionalidade usa dois modos, o *offline* e o *online*, ou seja, quando a aplicação está *offline*, a palavra que o utilizador insere na barra de pesquisa é utilizada para procurar na base de dados local, que contém algumas informações de ruas e edifícios. Quando a aplicação está *online*, decidiu-se utilizar a *API* do *Google Places* para realizar a pesquisa [20].

Como nas outras *API's* utilizadas, no caso da *API* do *Google Places* existem outras opções no mercado, como por exemplo a *Location API* da *Bing*. Ao comparar-se estas duas *API* referidas, verifica-se que as semelhanças são grandes, pois as duas permitem a pesquisa através de uma palavra-chave e a comunicação é feita através de *JSON* por meio de uma ligação *HTTPS*. Mas, utilizando a *API* do *Google Places*,

já existem bibliotecas para fazer esta comunicação encriptada e para decodificar os dados para uma lista de objetos, facilitando assim a tarefa.

A *API* do *Google Places* permite pesquisar por nomes e fotografias de lugares entre outros, em todo o mundo, mas como a aplicação desta dissertação é direccionada para a região da Madeira, limitou-se a pesquisa para esta área e apenas para nomes, optimizando assim a pesquisa.

Esta *API* é gratuita, mas com um limite de 1000 pedidos diários, a partir deste último pedido, a aplicação pesquisa apenas localmente. Para utilizar a *API* sem limites é preciso subscrever ao serviço pago do *Google Places*.

A comunicação entre a *API* e a aplicação foi feita através de uma comunicação *REST*, neste caso através de *JSON*. Como esta *API* utiliza obrigatoriamente uma comunicação segura por *HTTPS*, utilizaram-se livrarias disponibilizadas pelo Google que auxiliam no processamento destes dados (essas livrarias podem ser descarregadas em <https://developers.google.com/resources/api-libraries/download/discovery/v1/java>).

3.5.4. MySQL Server, PHP e JSON

As tecnologias *PHP* e *JSON* foram utilizadas para realizar a comunicação da aplicação com o servidor dos Horários do Funchal. A aplicação necessita de determinadas informações, de forma a funcionar e essas informações estão armazenadas numa base de dados *MySQL*. O problema é que o Android não consegue comunicar directamente com uma base de dados *MySQL*, logo necessita do auxílio de uma página *web* que leia da base de dados e converta para *JSON* que é uma linguagem que o Android já consegue decodificar.

Estas páginas *PHP* estão alojadas no servidor dos Horários do Funchal em <http://www.horariosdofunchal.pt> e para serem acedidas pelos dispositivos Android é necessário fazer um pedido *HTTP* para o link www.horariosdofunchal.pt/integracoes/integra.php e com algumas variáveis *POST* para o servidor saber que tipo de informação o dispositivo requer.

3.5.5. Base de Dados SQLite

De modo a guardar as informações provenientes do servidor dos Horários do Funchal, utilizou-se uma base de dados local no dispositivo. Para isso, implementou-se a base de dados em SQLite, sendo este o procedimento recomendado pelo *website* da *Android Developers* e também porque pode-se optar por seguir a mesma lógica utilizada na base de dados do servidor. Na Figura 3.10 pode-se observar a estrutura da base de dados local.

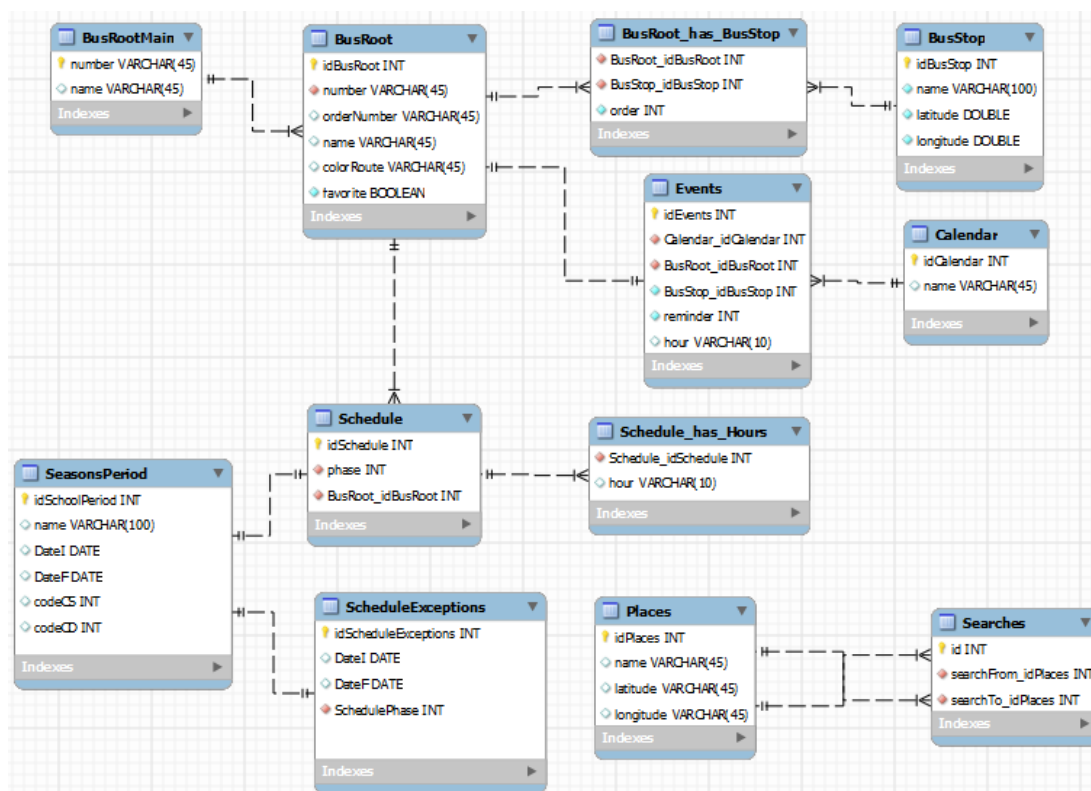


Figura 3.10 – Base de Dados SQLite utilizado no dispositivo Android

4. Desenho do Sistema

“Design is not just what it looks like and feels like. Design is how it works.”

(Steve Jobs).

O desenho do sistema consiste em todos os detalhes da aplicação, como a componente visual, a interacção e a arquitectura. Para que a aplicação seja um sucesso, o desenho deve ser robusto, flexível, apelativo, escalável, entre outras características que o desenho do sistema deve ser.

4.1. Casos de Utilização

Uma das vistas arquitecturais que se utilizou para representar a arquitectura do sistema da aplicação desta dissertação foi a vista de Casos de Utilização (*Use Cases*).

A vista de casos de utilização permite saber quais são as acções que o sistema ou aplicação pode realizar, ou seja, permite visualizar as relações entre as acções e os atores do sistema. Esta vista não especifica a ordem com que as acções são realizadas, nem quantos passos são necessários para concluir a acção. Esta vista tem diversos componentes que se podem utilizar para descrever o sistema, mas na maioria dos casos utilizam-se os 3 principais componentes, o Ator, a Acção e o Sistema [21].

Para esta dissertação identificaram-se dois atores, o gestor e o utente do Horários do Funchal e diversas acções realizadas. Na Figura 4.1 pode-se observar o diagrama final.

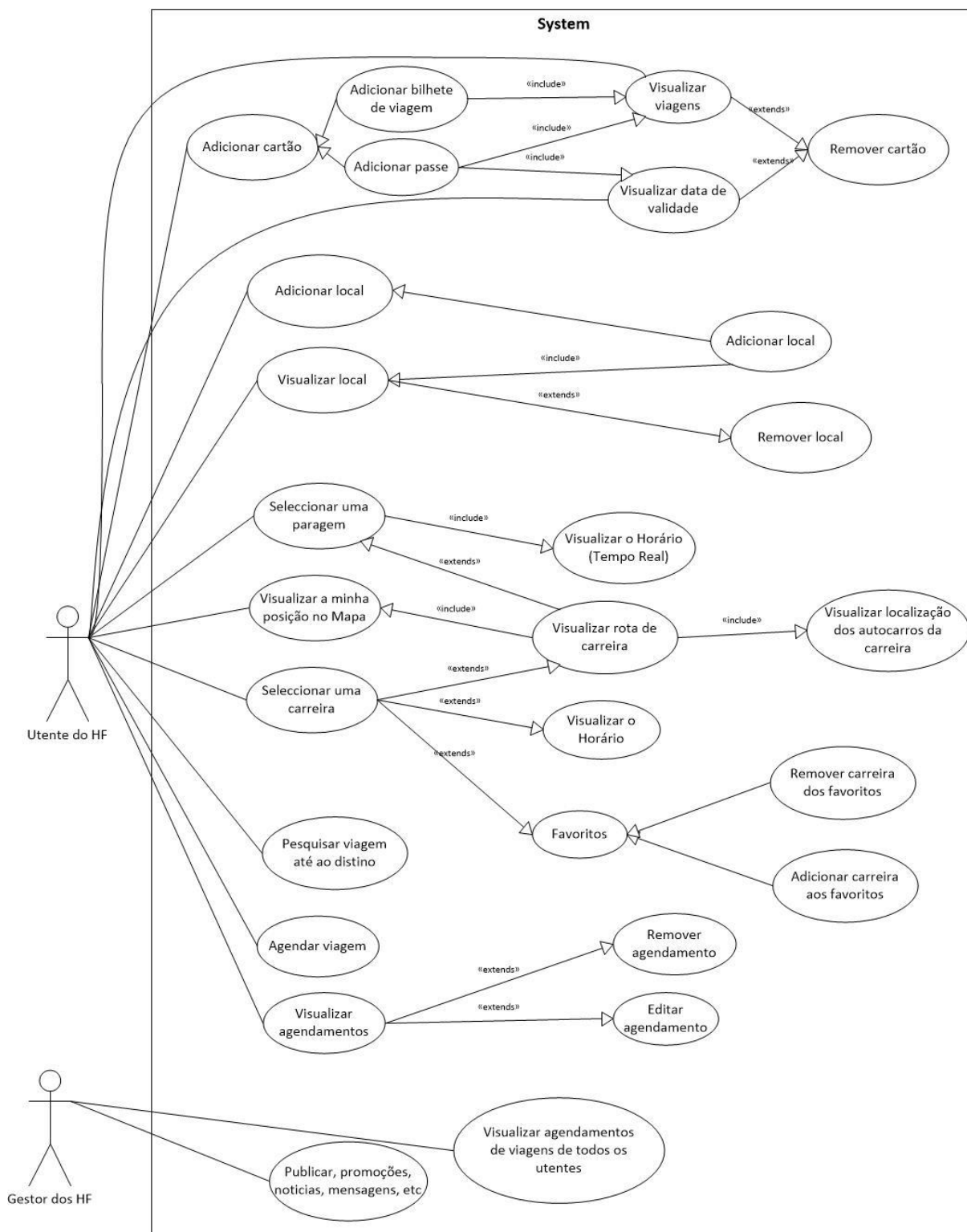


Figura 4.1 – Diagrama de casos de utilização da aplicação desta dissertação

4.1.1. Modelo de Tarefas

Criaram-se alguns modelos de tarefas, de modo a ter-se uma ideia de como o sistema vai interagir com o utilizador. Começou-se por definir o modelo para o caso da visualização de agendamentos e todo o processo em torno desse caso (observar Figura 4.2). Inicialmente, tem-se de entrar na atividade Alarmes e automaticamente o sistema carrega todos os agendamentos existentes que à data ainda não tenham expirado, e apresenta-os ao utilizador. A partir deste momento, o utilizador pode adicionar um novo agendamento, editar ou apagar um agendamento existente.

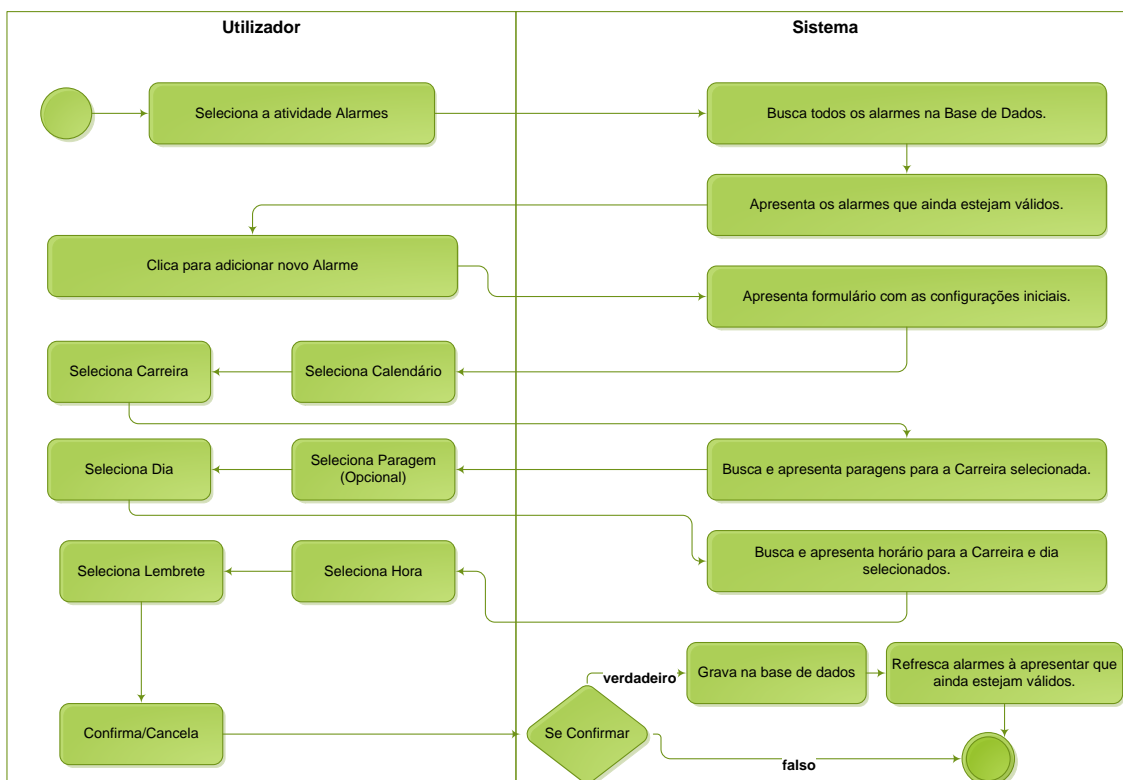


Figura 4.2 – Modelo de tarefa para “Consultar Agendamentos”

Para editar um agendamento o processo é muito semelhante ao processo de adicionar um novo, com a exceção dos valores apresentados no formulário que são os valores do agendamento a editar, como se pode observar na Figura 4.3.

A funcionalidade de visualizar a rota de uma carreira no mapa é descrita no modelo de tarefa que pode ser observado na Figura 4.5.

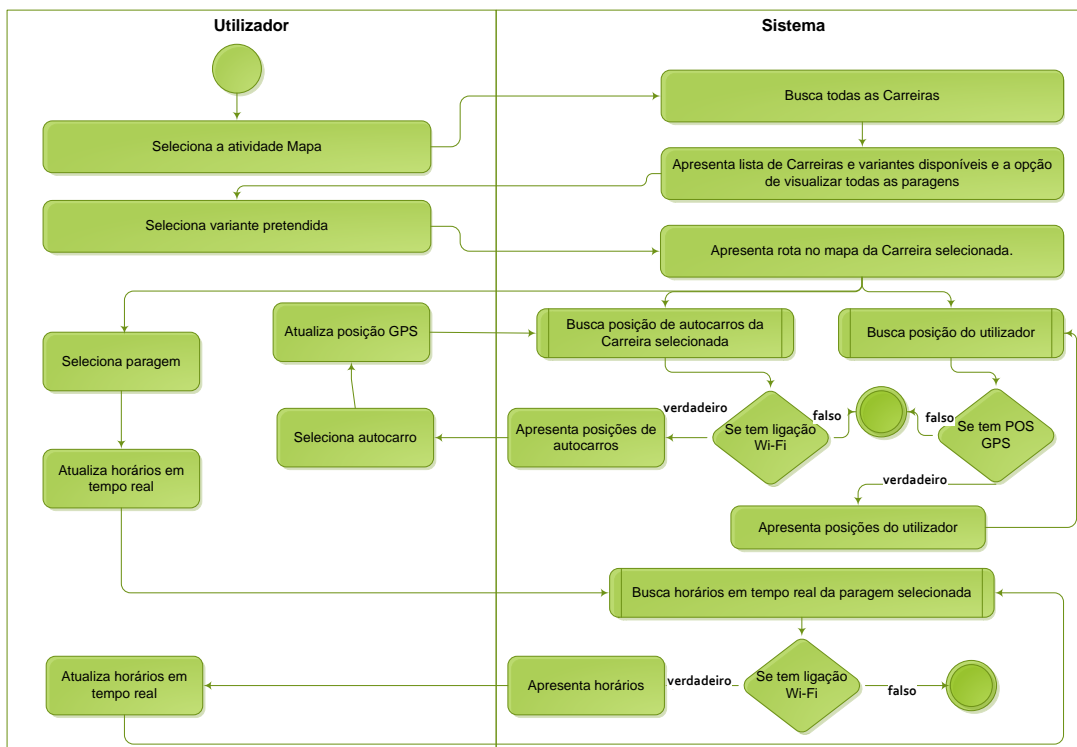


Figura 4.5 – Modelo de tarefa para “Visualizar rota no mapa”

O modelo de tarefa que representa a funcionalidade de visualizar as paragens no mapa e os horários em tempo real é representado na Figura 4.6.

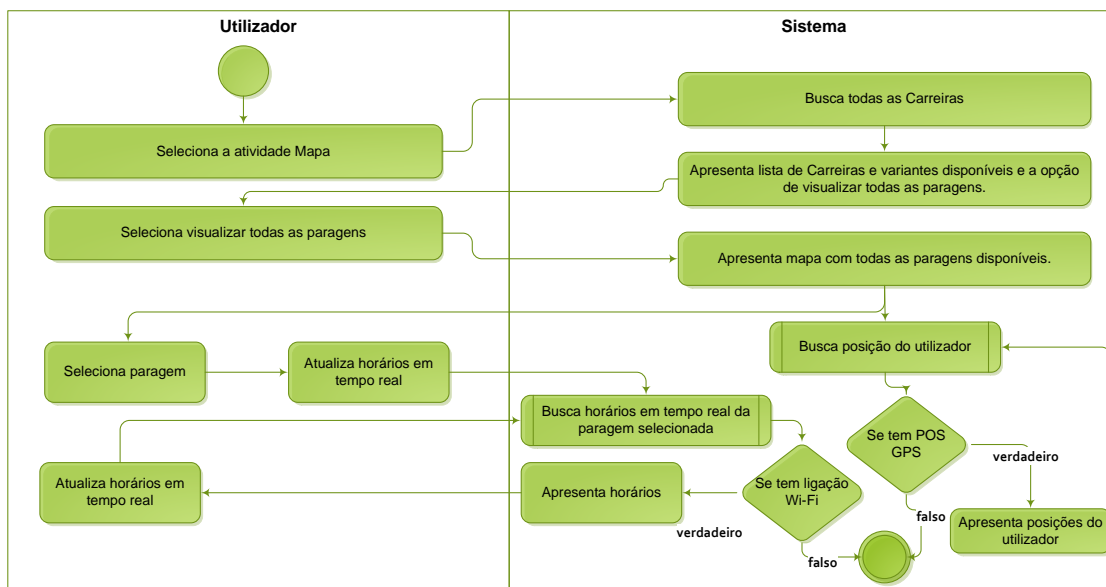


Figura 4.6 – Modelo de tarefa para “Visualizar paragens no mapa”

Para representar a funcionalidade de visualizar os horários das carreiras, utilizou-se o modelo de tarefa desenhado na Figura 4.7.

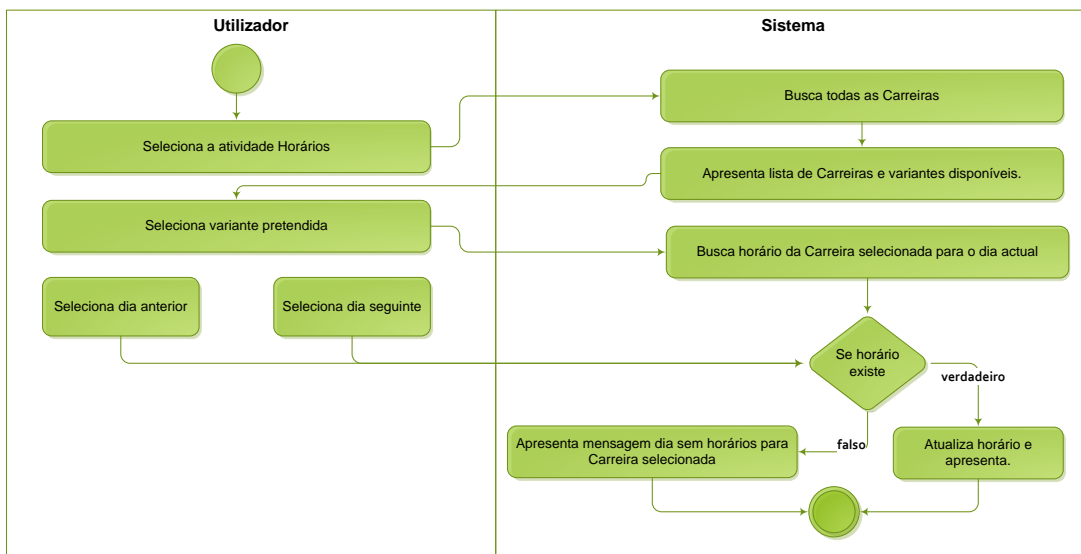


Figura 4.7 – Modelo de tarefa para “Visualizar horários”

As funcionalidades de pesquisa por itinerários e de gravar locais como favoritos estão desenhadas na Figura 4.8.

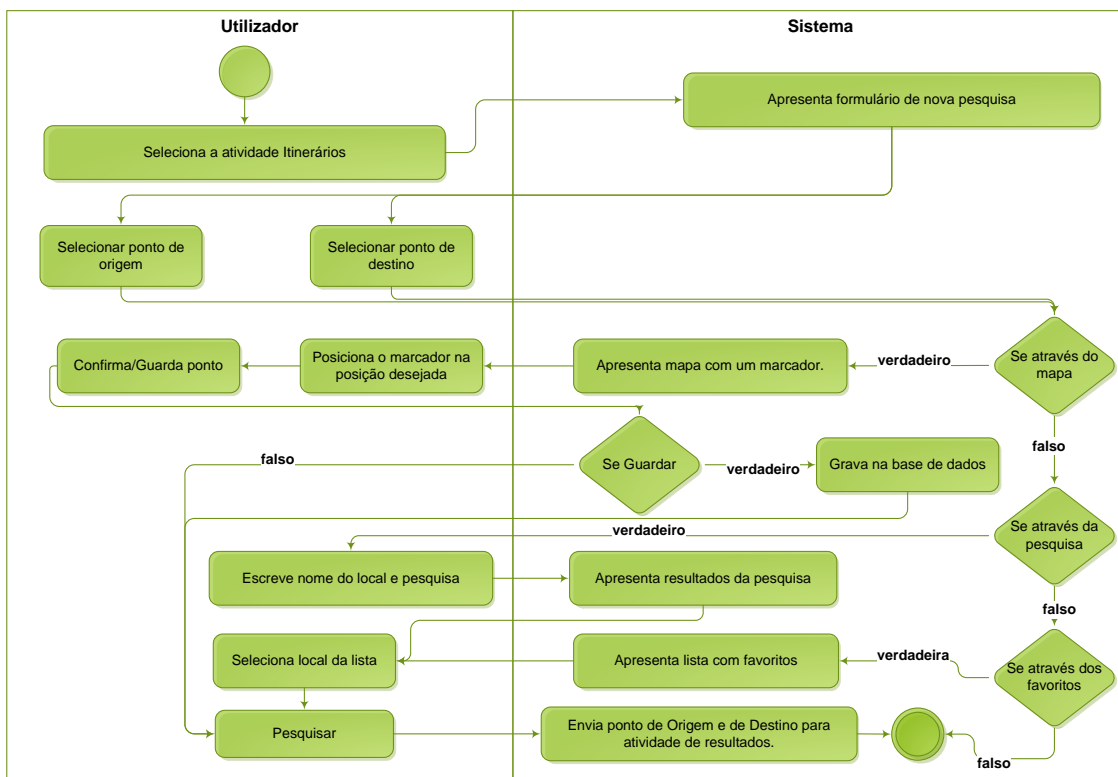


Figura 4.8 – Modelo de tarefa para “Pesquisa de itinerário”

Após realizar a pesquisa do itinerário, são calculadas as opções viáveis e são apresentados os resultados no mapa, onde o utilizador pode navegar entre as várias opções. Pode-se observar o modelo da tarefa desta funcionalidade na Figura 4.9.

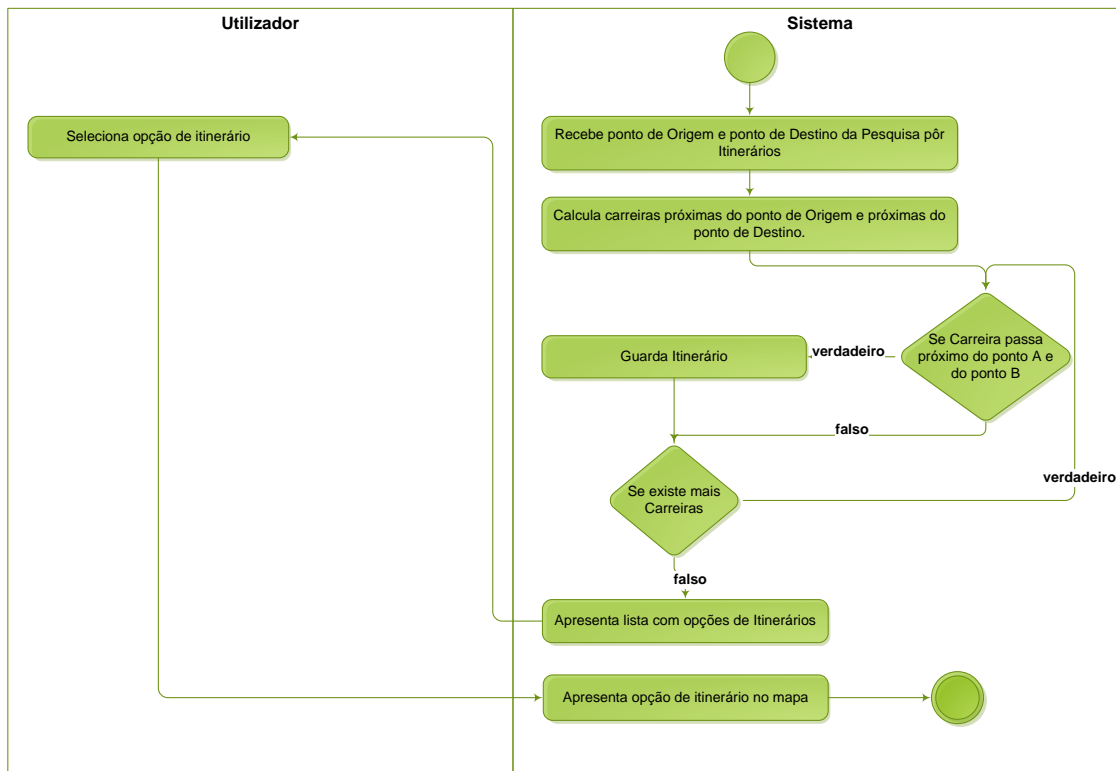


Figura 4.9 – Modelo de tarefa para “Resultados da pesquisa de itinerário”

O utilizador não tem ação direta nas funcionalidades de alertas de agendamentos e de sincronização da aplicação. No entanto, desenhou-se o modelo de tarefas destes, de modo a compreender-se o funcionamento destas funcionalidades. Na Figura 4.10 pode-se observar o modelo de tarefa para a sincronização da aplicação.

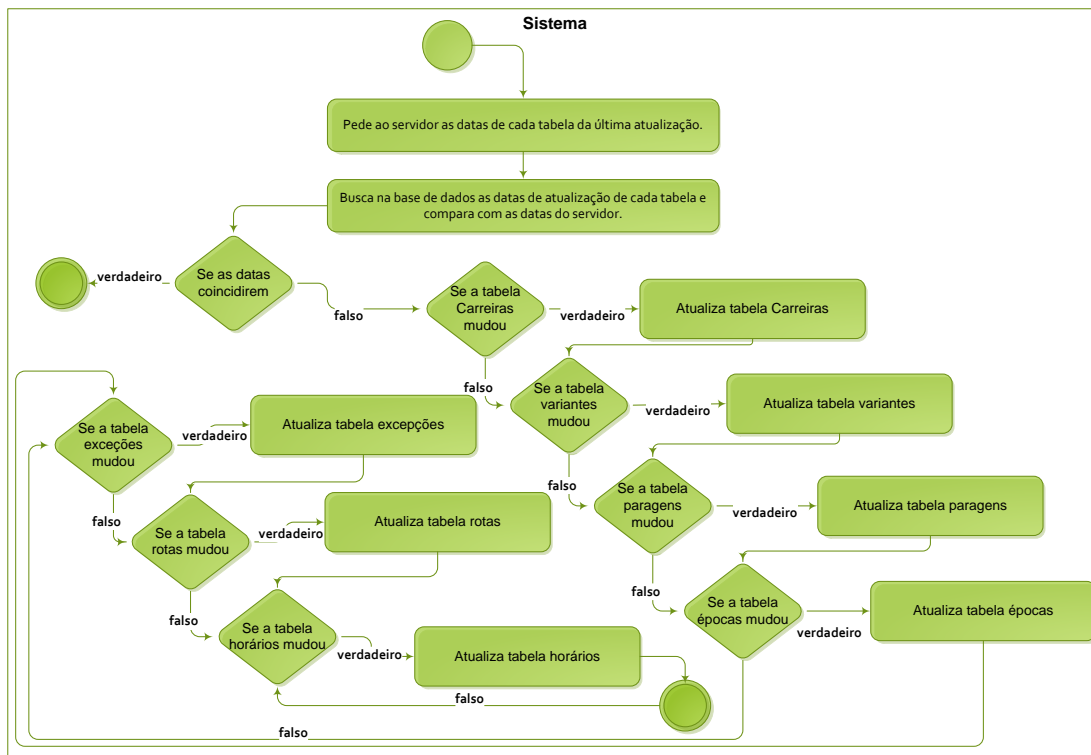


Figura 4.10 – Modelo de tarefa para “Sincronização”

A funcionalidade do sistema para alertar o utilizador quando o autocarro da viagem agendada sai da primeira paragem encontra-se representada na Figura 4.11.

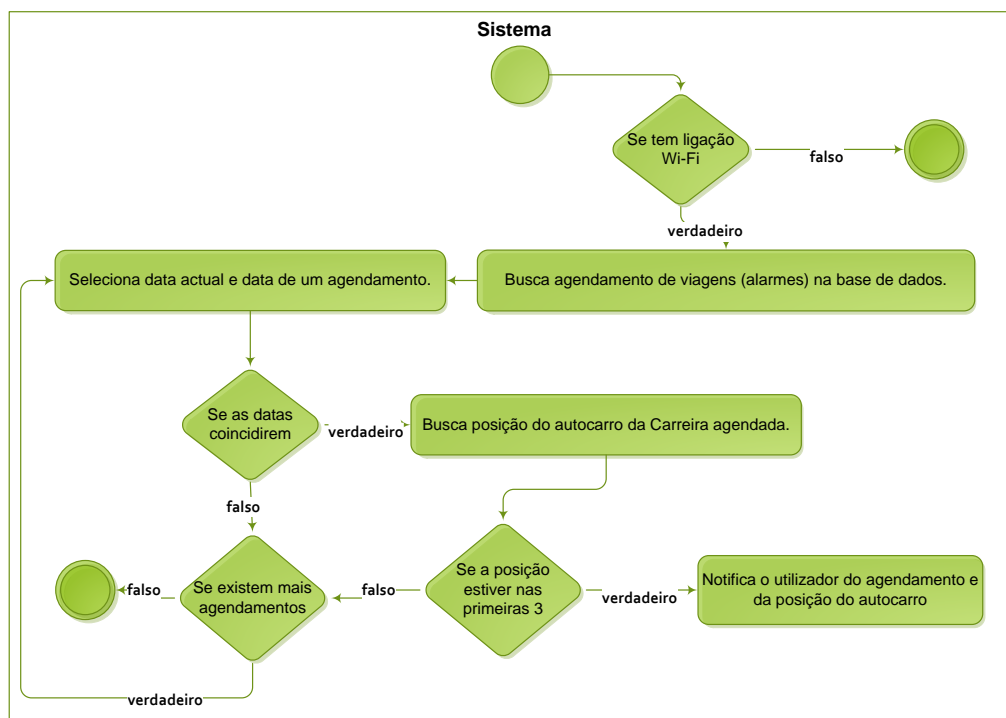


Figura 4.11 – Modelo de tarefa para “Alerta de primeira paragem”

Por último, tem-se o modelo de tarefa da funcionalidade para alertar o utilizador quando o autocarro da viagem que agendou está quase a chegar à paragem que definiu. Pode-se observar este modelo na Figura 4.12.

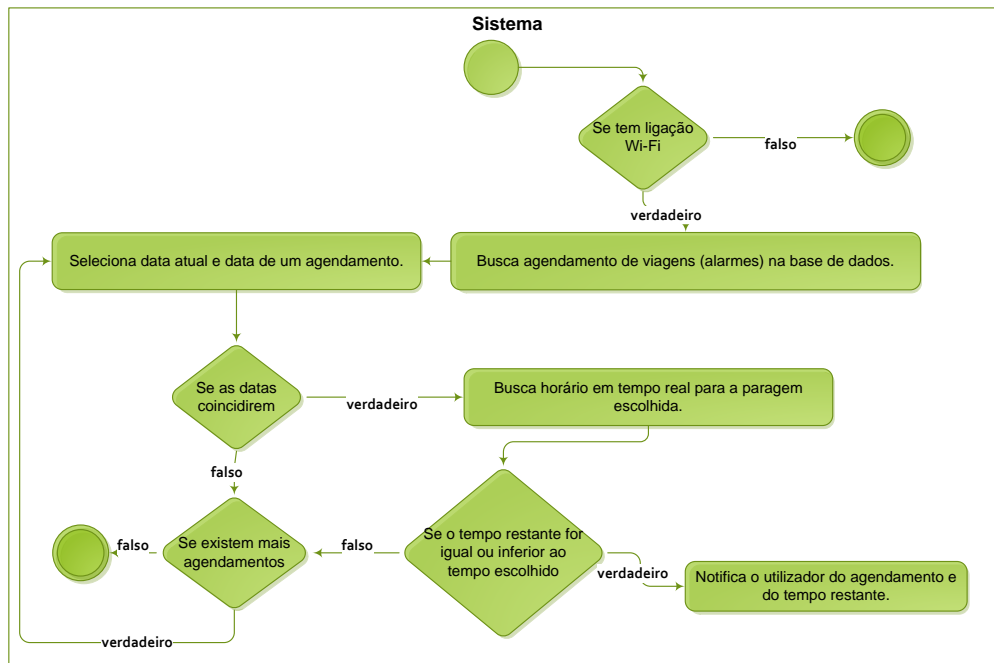


Figura 4.12 – Modelo de tarefa para “Alerta tempo restante para chegar à paragem”

4.2. Requisitos

Através dos *Uses Cases* é possível ter uma ideia mais clara do que o sistema vai ser capaz de realizar e isso ajuda a definir os requisitos do sistema. Os requisitos do sistema são as condições cujas exigências devem ser satisfeitas. Estes requisitos podem ser divididos em dois grupos: os requisitos funcionais e não-funcionais [22].

4.2.1. Requisitos Funcionais

Os requisitos funcionais englobam as funcionalidades do sistema, ou seja, as funções que o sistema permite ao utilizador realizar [22]. Para o caso desta dissertação, as funcionalidades já estão definidas nos *Uses Case*, logo podem-se descrever as seguintes funcionalidades:

- F1. Selecionar item do menu.
- F2. Selecionar item do submenu.

-
- F3. Pesquisar o próximo autocarro a sair de cada Carreira (“Next to Go”).
 - F4. Filtrar Carreiras por nome ou número.
 - F5. Navegar livremente pelo mapa.
 - F6. Selecionar vista de mapa, “normal”, “satélite” ou “hibrido”.
 - F7. Aumentar *zoom* no mapa.
 - F8. Diminuir *zoom* no mapa.
 - F9. Pesquisar pela localização do utilizador no mapa (necessita de *GPS* ou *Wi-Fi*).
 - F10. Pesquisar por cidades, concelhos, ruas e pontos de referência no mapa.
 - F11. Pesquisar pela rota de uma determinada Carreira.
 - F12. Pesquisar por paragens no mapa.
 - F13. Pesquisar os horários em tempo real numa determinada paragem.
 - F14. Agendar viagem no calendário e os respetivos alarmes.
 - F15. Editar viagem no calendário e os respetivos alarmes.
 - F16. Eliminar viagem no calendário e os respetivo alarme.
 - F17. Pesquisar pelos horários de uma determinada Carreira.
 - F18. Ver os horários de uma determinada Carreira para o dia seguinte.
 - F19. Ver os horários de uma determinada Carreira para o dia anterior.
 - F20. Marcar ponto *GPS* no mapa para pesquisar itinerários.
 - F21. Adicionar pontos *GPS* de referência aos favoritos.
 - F22. Pesquisar possíveis percursos entre dois pontos *GPS* (Itinerários).
 - F23. Selecionar pesquisa do histórico dos itinerários para nova pesquisa.
 - F24. Adicionar Carreiras aos favoritos.
 - F25. Remover Carreiras dos favoritos
 - F26. Visualizar localização dos Horários do Funchal.
 - F27. Contatar Horários do Funchal por correio electrónico.

- F28. Contatar Horários do Funchal por telefone.
- F29. Visitar o *website* dos Horários do Funchal.
- F30. Adicionar bilhete ou passe de viagem.
- F31. Editar bilhete ou passe de viagem.
- F32. Eliminar bilhete ou passe de viagem.
- F33. Visualizar informações referentes aos bilhetes ou passes de viagem.

4.2.2. Requisitos Não-Funcionais

Os requisitos não funcionais englobam funcionalidades sobre as quais o utilizador não tenha um controlo direto e também funcionalidades que dizem respeito ao desempenho, robustez, portabilidade, segurança, escalabilidade, entre outras [22].

No caso do sistema, desta dissertação podem-se definir como requisitos não-funcionais os seguintes:

- NF1. Usabilidade – Notificar o utilizador sobre uma viagem agendada para breve.
- NF2. Usabilidade – Notificar o utilizador sobre o início de deslocação do autocarro de uma viagem agendada.
- NF3. Usabilidade – Verificar se existem atualizações no sistema de informação dos Horários do Funchal e descarrega as atualizações, caso necessário (atualizações automáticas).
- NF4. Segurança – Utilização de uma ligação *HTTPS* para comunicar com a API do *Google Places*.
- NF5. Disponibilidade – Utilização de uma ligação *Wi-Fi* ou *Dados* para comunicar com os servidores.
- NF6. Usabilidade – Se os textos forem muito grandes para o espaço predefinido, corem horizontalmente até passarem por completo.
- NF7. Usabilidade – Os *layouts* utilizados têm um *padding* de 10dp.
- NF8. Usabilidade – Os botões usam ícones *standard*, definidos pela *Android Developers*, para as funcionalidades gerais.

-
- NF9. Usabilidade – Os textos têm uma margem de 5dp em relação a outros *widgets*.
- NF10. Usabilidade – Os teclados que aparecem nos campos de inserção de texto são específicas, consoante o tipo de inserção desejado.
- NF11. Usabilidade – Cada actividade tem um cabeçalho de uma determinada cor, predefinida, para ajudar o utilizador a se guiar na aplicação.
- NF12. Usabilidade – O ponto de origem e de destino na pesquisa por itinerários apresenta um X enquanto não for introduzido corretamente.
- NF13. Desempenho – As listas de carreiras com o horário das próximas Carreiras deve ser carregado dinamicamente à medida que o utilizador desliza a lista das Carreiras.
- NF14. Desempenho – As listas de dados ao serem carregadas devem mostrar um *popup* com um *loading* de forma a garantir que o utilizador não interrompe o processo.
- NF15. Desempenho – Utilização do sistema de posicionamento global do dispositivo, através de *GPS* ou *IP* através de *Wi-Fi* ou *GPRS*, por esta ordem.
- NF16. Capacidade/Disponibilidade – Utilização de *SQLite* para guardar dados do utilizador no dispositivo.
- NF17. Adaptabilidade – Suporte para diferentes dispositivos Android (*Smartphones* e *Tablets*), de diferentes tamanhos de ecrã e diferentes resoluções.
- NF18. Adaptabilidade – Suporte para duas línguas (língua portuguesa e língua inglesa).
- NF19. Adaptabilidade – Compatibilidade com dispositivos Android com versões entre 2.2 a 4.2, incluindo estes.
- NF20. Segurança – A aplicação deve exigir *login* para aceder aos dados de utilizador guardados no servidor dos Horários do Funchal.

4.3. Wireframes/Mockups

Construir uma aplicação com as funcionalidades referidas anteriormente, requer alguma prototipagem, pois existem várias formas de as implementar, mas apenas algumas são verdadeiramente boas e práticas. De modo a cometer o menor número de erros possível, seguiram-se as 10 heurísticas de Nielsen [23] e os princípios de *design* propostos pela *Android Developers* [24]. Para poder corrigir erros e de forma rápida, utilizou-se uma ferramenta de prototipagem de baixo nível, visto que um protótipo de alto nível requer muito tempo e este é um recurso limitado.

A aplicação que se utilizou para desenhar o protótipo foi o *Balsamiq Mockups* que permite desenhar os *wireframes* e fazer pequenos testes. Este programa já traz uma série de ferramentas (*tools*) e *widgets* convencionais que permitem uma optimização no desenho. Além disso, permite importar imagens de *widgets* que não contenha e também possibilita realizar pequenos testes com utilizadores, pois este programa permite criar *links* entre os vários *wireframes*, transmitindo assim a sensação de navegação pelas várias funcionalidades da aplicação [25].

4.3.1. Versão Smartphone

Criaram-se, então, os *wireframes* para a versão de *smartphone*, como se pode observar em alguns exemplos na Figura 4.13.

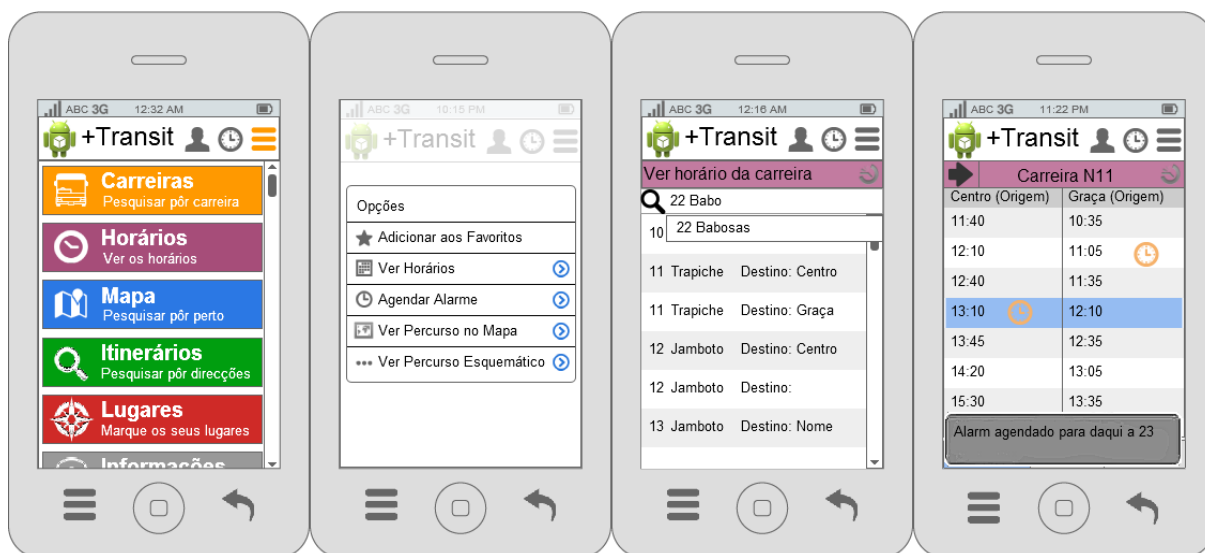


Figura 4.13 – Exemplo dos *wireframes* para *smartphone*

Após se ter uma versão completa dos *wireframes*, pediu-se a utilizadores voluntários dos Horários do Funchal para os testarem. Para esse teste criaram-se diversos cenários de usabilidade (ver Anexo B) e utilizou-se um formulário, representado na Figura 4.14, para registar os erros que os utilizadores cometiam. Este formulário foi adaptado do artigo “Peer Reviews for Usability Users Interface” [23].


Defect # _____	Scenario/Frame _____	Title _____
Location (mark, identify)		Feature: <input type="checkbox"/> button <input type="checkbox"/> graphic <input type="checkbox"/> palette <input type="checkbox"/> chart/graph <input type="checkbox"/> grid <input type="checkbox"/> panel <input type="checkbox"/> checkbox <input type="checkbox"/> header <input type="checkbox"/> pop-up <input type="checkbox"/> color <input type="checkbox"/> icon <input type="checkbox"/> radio b. <input type="checkbox"/> command <input type="checkbox"/> keystroke <input type="checkbox"/> screen <input type="checkbox"/> detail <input type="checkbox"/> label <input type="checkbox"/> symbol <input type="checkbox"/> dialog <input type="checkbox"/> list <input type="checkbox"/> tool <input type="checkbox"/> dropdown <input type="checkbox"/> map <input type="checkbox"/> table <input type="checkbox"/> field <input type="checkbox"/> menu <input type="checkbox"/> text <input type="checkbox"/> function <input type="checkbox"/> menu item <input type="checkbox"/> view <input type="checkbox"/> glyph <input type="checkbox"/> message <input type="checkbox"/> window <input type="checkbox"/> other:
		Problem: <input type="checkbox"/> presentation design <input type="checkbox"/> interaction design <input type="checkbox"/> alignment <input type="checkbox"/> connection <input type="checkbox"/> function missing <input type="checkbox"/> inconvenient <input type="checkbox"/> slow <input type="checkbox"/> awkward <input type="checkbox"/> distracting <input type="checkbox"/> function hidden <input type="checkbox"/> inefficient <input type="checkbox"/> unexpected <input type="checkbox"/> complex <input type="checkbox"/> error handling <input type="checkbox"/> hidden behavior <input type="checkbox"/> nonstandard <input type="checkbox"/> uninformative <input type="checkbox"/> cluttered <input type="checkbox"/> failed <input type="checkbox"/> inconsistent <input type="checkbox"/> position <input type="checkbox"/> unreliable <input type="checkbox"/> confusing <input type="checkbox"/> other:
	Usability principle: <input type="checkbox"/> visibility/availability <input type="checkbox"/> structure/organization <input type="checkbox"/> tolerance/flexibility <input type="checkbox"/> feedback <input type="checkbox"/> reuse/consistency <input type="checkbox"/> simplicity/efficiency	Severity: 4 = critical <input type="checkbox"/> 3 = major 2 = minor 1 = nominal ? = evaluate
NOTES		

Figura 4.14 – Formulário utilizado para registar e classificar erros do utilizador ao usar a aplicação [23]

No final de cada teste com um utilizador, de modo a avaliar a opinião deste, realizou-se um inquérito (ver Anexo A) que foi construído com base num modelo *online* de Gary Perlman's [26].

Os testes, como esperado, revelaram erros de usabilidade na interface da aplicação. Desta forma, apresentaram-se novas propostas para solucionar estes erros e voltou-se a testar até se conseguirem resultados dentro do desejado.

4.3.2. Versão *Tablets*

Na versão para *Tablets* utilizou-se o mesmo procedimento e as mesmas ferramentas. A única diferença foi o *layout* que era apresentado ao utilizador, pois como se tem mais espaço no dispositivo, não se justifica ter o mesmo *layout*. Assim, apresenta-se mais informação ao mesmo tempo, e organizada de forma diferente, como se pode observar em alguns exemplos na Figura 4.15.

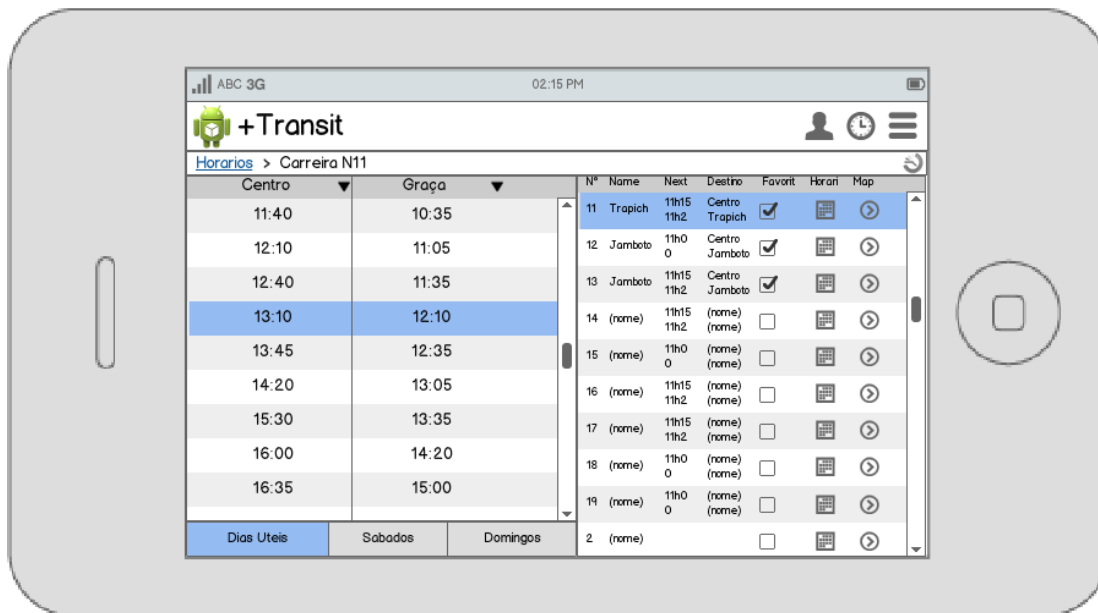


Figura 4.15 – Exemplo dos *wireframes* para *Tablet*

4.4. Arquitetura

A arquitetura de uma aplicação consiste na interligação de vários elementos arquitetais. Estes elementos podem ser bases de dados, servidores, clientes, filtros, entre outros e a forma como estes elementos se interligam define o estilo arquitetural, sendo que uma aplicação pode ser construída apenas com base num estilo ou em vários [27].

Os estilos arquitetais usados na aplicação desta dissertação foram o estilo MVC (*Model, View, Controller*) e o estilo Cliente-Servidor.

4.4.1. Arquitetura MVC

A arquitetura MVC consiste em segmentar a aplicação em três componentes, sendo que cada uma destas têm as suas responsabilidades e elas não interferem com as responsabilidades de outra componente [27]. Estas componentes são:

- *Model* – Define a semântica da aplicação e o seu comportamento. É responsável por tratar os dados e armazená-los.
- *View* – Define o aspeto visual da aplicação. É responsável pela forma como os dados são apresentados ao utilizador.

- *Controller* – Gere as interações do utilizador com o *Model* e as *Views* da aplicação.

Na aplicação desta dissertação utilizou-se este estilo arquitetural visto que o SDK da Android praticamente implementa este estilo em todos os projetos por defeito. Poder-se-ia optar por contrariar este estilo e utilizar outro, uma vez que tem-se alguma liberdade para o fazer, mas como o SDK da Android implementa o MVC de uma forma muito flexível e fácil para o programador, decidiu-se utilizar o MVC. Na Figura 4.16 pode-se observar como ficou a estrutura final MVC da aplicação Android.

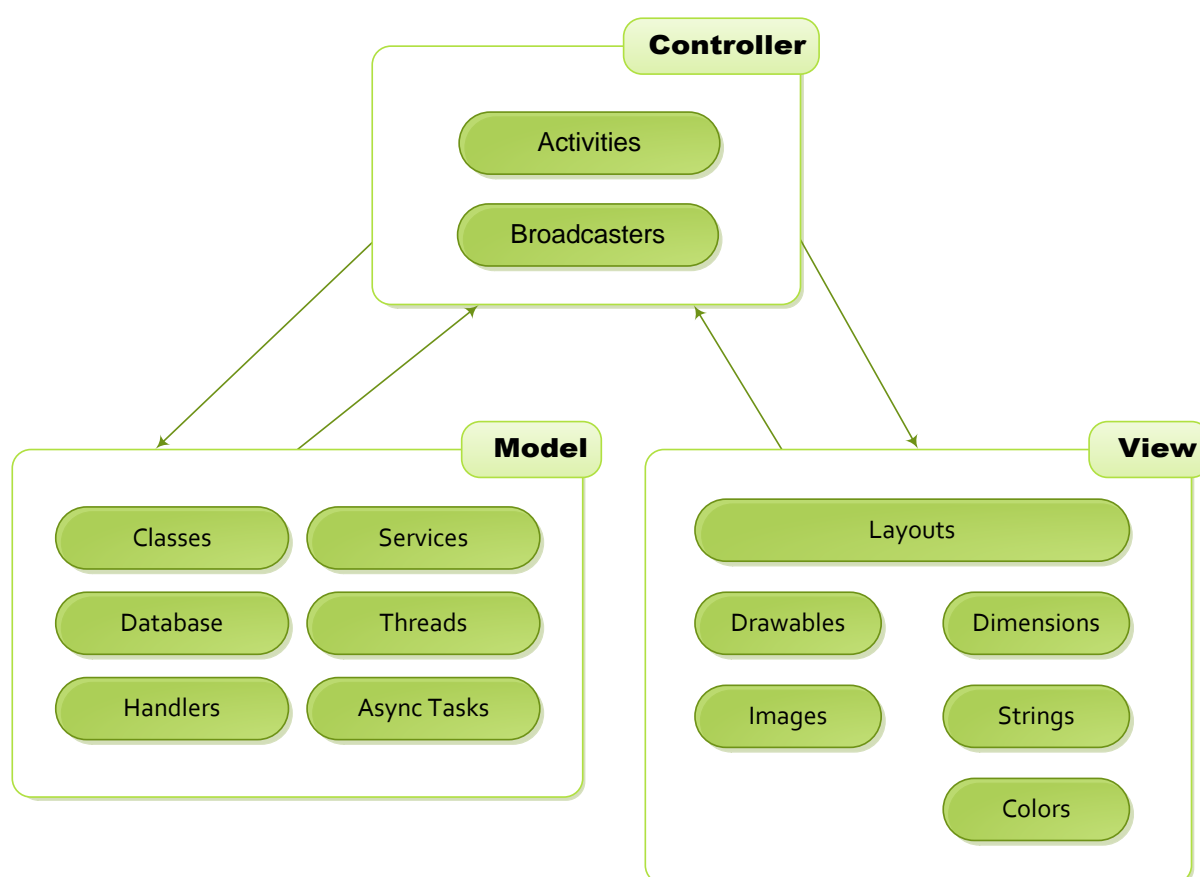


Figura 4.16 – Desenho do estilo arquitetural MVC utilizado na aplicação android

Ao criar um projeto Android, é gerado um projeto com várias pastas e ficheiros. Estas pastas já separam os ficheiros que vão fazer parte do *View* dos ficheiros que fazem parte do *Model* e do *Controller*. As pastas são as seguintes:

- *Src* – Esta pasta contém todos os ficheiros java, entre os quais se encontram as atividades, os *broadcast*, as classes, os serviços, base de dados, *handlers* e os *asynctasks*.
- *Gen* – Esta pasta contém o ficheiro R que faz a ligação entre o código e os recursos XML. Cada vez que é adicionado um *Layout*, uma imagem ou outro tipo de *resource* é gerado um ID inteiro no ficheiro R. O ficheiro R é um ficheiro do sistema e não deve ser alterado manualmente.
- *Res* – Esta pasta contém todos os recursos da aplicação, imagens, *layout*, sons, vídeos, *strings*, entre outros. Também é nesta pasta que se separam os *Layouts* por dispositivo. Por exemplo se se pretender criar *layouts* diferentes para *smartphone* e para *tablet* cria-se uma nova pasta dentro dos *resources* com um nome *layout-large*.
- *Bin* – Esta pasta contém os ficheiros que são gerados quando se compila a aplicação. É uma pasta do sistema e não se necessita alterar nada.
- *Libs* – Nesta pasta inserem-se as bibliotecas ou *SDK* que sejam necessários na aplicação. Basta importar o ficheiro para esta pasta e associar nas propriedades da aplicação.

O ficheiro *manifest* serve para declarar todas as propriedades da aplicação. Entre elas encontram-se as atividades, serviços e *broadcast* que são utilizados. Também se declara aqui quais os periféricos do dispositivo que são utilizados como por exemplo GPS, vibratório, entre outros.

4.4.2. Arquitetura Cliente-Servidor (*Broker*)

A arquitetura Cliente-Servidor é utilizada para passar dados entre o servidor e o cliente ou entre clientes por meio do servidor. Uma forma de implementar esta arquitetura é através do *Broker* [27].

O *Broker* é um componente do servidor, que serve de intermediário entre os clientes e outros servidores, ou seja, transmite o pedido do cliente ao servidor e vice-versa. Desta forma, os clientes não precisam saber de todos os endereços dos servidores de informação para comunicarem e também contribui para o aumento da segurança,

pois se algum cliente mal-intencionado tentar realizar ataques ao servidor, o *Broker* serve de barreira, protegendo os outros servidores.

Para a aplicação desta dissertação é necessário que esta comunique com os servidores dos Horários do Funchal, de modo a descarregar a informação referente às Carreiras, aos horários e outras informações. Como estas informações estão espalhadas por diferentes servidores e alguns são críticos para o bom funcionamento dos serviços dos Horários do Funchal, decidiu-se criar um *Broker*. Pode-se observar o esquema deste na Figura 4.17.

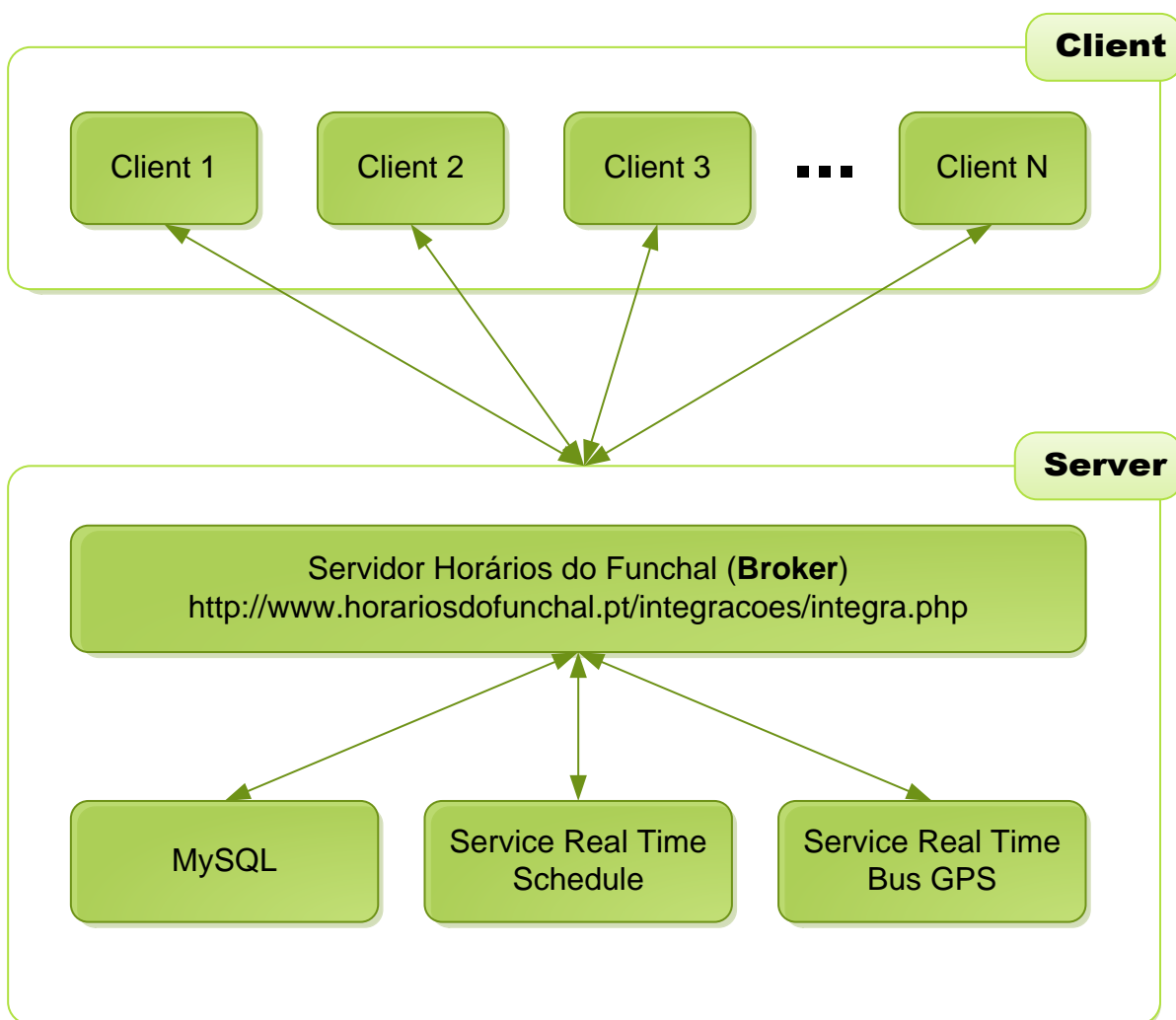


Figura 4.17 – Desenho do estilo arquitetural Cliente-Servidor por meio de um *Broker*

Os clientes representados na Figura 4.17 são os dispositivos Android dos utilizadores dos Horários do Funchal e o servidor que representa o *Broker* é o servidor público. Os outros servidores, nomeadamente o servidor *MySQL*, e os servidores que albergam

os serviços de tempo real, não são visíveis aos utilizadores finais, apenas são visíveis para o *Broker* e o utilizador para aceder a estes, tem de o fazer por meio do *Broker*.

O *Broker* consiste numa página *web*, em *php*, que recebe algumas variáveis por meio de POST e, posteriormente, faz o pedido aos respetivos servidores, onde de facto se encontram as informações. Estes devolvem ao *Broker* e os dispositivos conseguem obter a informação desejada.

4.5. Padrões de Desenho

Os padrões de desenho são métodos convencionais utilizados por diversos programadores quando estes estão a programar. Existem três classes de padrões de desenho, nomeadamente [28]:

- Padrões de Criação (*Creational Patterns*) – Estes padrões servem para lidar com a criação de objetos, de forma que posteriormente estes objetos não causem problemas com outras partes do código. Exemplos: *Factory*, *Singleton*, *Prototype*, etc ...
- Padrões Estruturais (*Structural Patterns*) – Os padrões estruturais são responsáveis por definir relações entre entidades, como objetos e classes, de forma a criar uma estrutura. Exemplos: *Adapter*, *Proxy*, *Composite*, etc ...
- Padrão Comportamentais (*Behavioral Patterns*) – Por último tem-se os padrões comportamentais que identificam padrões de comunicação comuns, ou seja, estes padrões focam-se na comunicação entre objetos cooperativos. Exemplos: *Observer*, *State*, *Strategy*, etc ...

4.5.1. Padrão *State*

O padrão *state* permite ao objeto alterar o seu comportamento quando o seu estado interno muda, ou seja, o objeto tem diversos estados internos e em função do estado em que se encontra, permite realizar diferentes tarefas. Estes estados podem ser mudados durante o tempo de execução.

Na aplicação desenvolvida utiliza-se este padrão por defeito nas classes da atividade. O sistema Android implementa o padrão *state* nas suas atividades, tornando estas, num ciclo de vida da atividade como se pode observar na Figura 4.18.

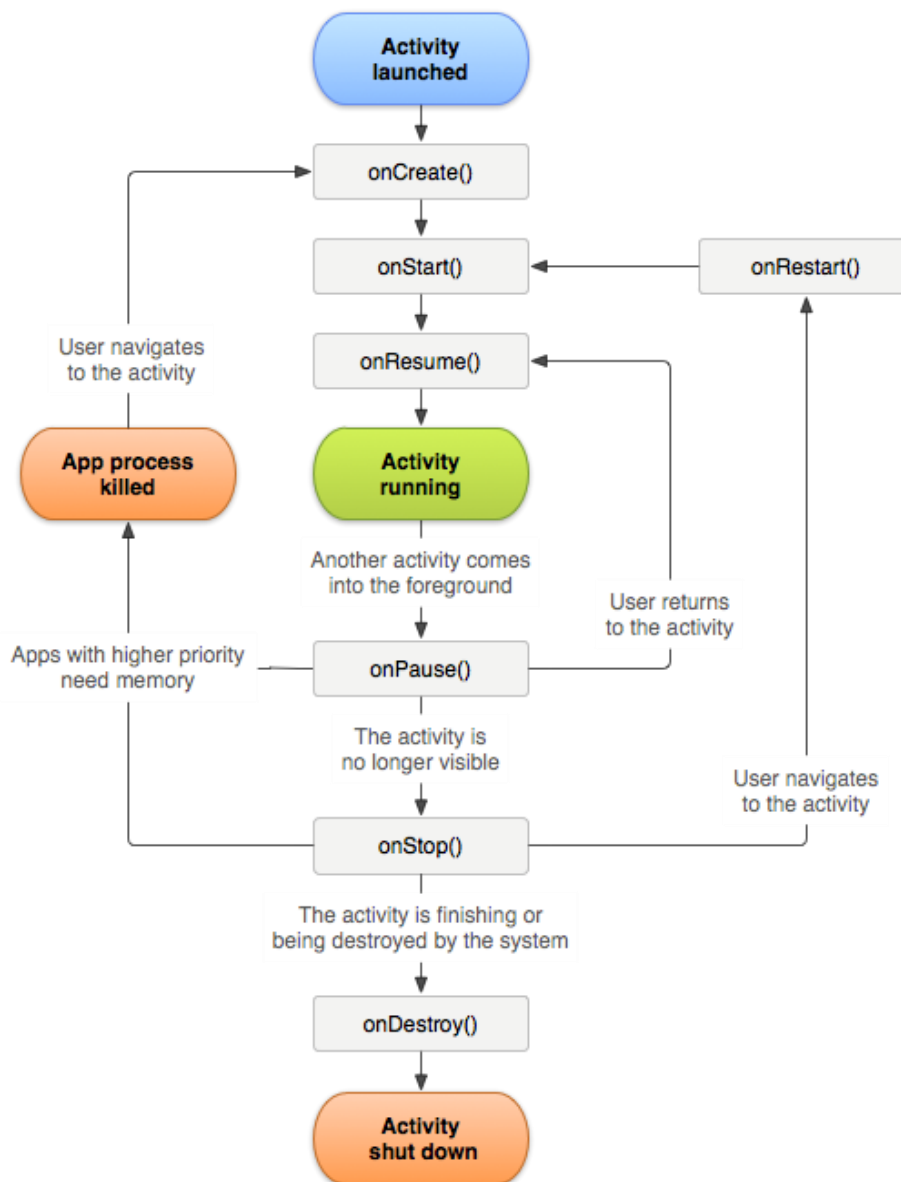


Figura 4.18 – Ciclo de vida da Actividade [29]

Na atividade têm-se vários estados, entre os quais se pode optar por utilizá-los a todos ou apenas alguns. Por exemplo, o estado *onCreate* é o primeiro estado que a atividade executa e é executado apenas no momento de criação, não se repete. Neste estado pode-se, por exemplo, declarar a inicialização de variáveis ou carregar os *views* dos *widgets*.

Os estados *onStart*, *onResume*, *onPause* e *onStop* são estados intermédios que podem ser utilizados quando se cria outra atividade a partir da primeira. Quando isto ocorre, a primeira atividade entra no estado *onPause* e logo em seguida no *onStop*. Posteriormente, quando a atividade que foi criada for destruída, o sistema volta à atividade inicial, executando os estados *onRestart*, *onStart* e *onResume*, por esta ordem.

Também deve-se utilizar o estado *onStop* para destruir objetos desnecessários, de forma a libertar memória ou de forma a evitar que existam *leaks* no código.

4.5.2. Padrão *Adapter*

Este padrão, como o nome indica, serve para converter ou adaptar a interface de um objeto para outro objeto, ou seja, podem-se utilizar objetos diferentes em conjunto, que de outra forma seriam incompatíveis.

Para esta aplicação, o padrão *adapter* foi crucial, pois todas as listas na aplicação utilizavam objetos diferentes e sem o padrão *adapter*, a lista não conseguiria representar estes objetos. Por definição uma lista, apenas representa o objeto *string* e por meio do padrão *adapter*, isto já não é verdade.

4.5.3. Padrão *Observer*

O padrão *observer* é utilizado quando se pretende subscrever uma funcionalidade, ou seja, pretende-se receber uma notificação de quando uma determinada funcionalidade é executada. Por definição deste padrão, não é mencionado o número de subscrições que um objeto pode ter.

Na aplicação utilizou-se este padrão para resolver um problema de inicialização, pois quando se declarava um *view* numa atividade, este *view* fica com o valor nulo durante alguns microssegundos, sendo esse o tempo que o dispositivo leva para carregar o *view*. Neste tempo, não se pode declarar nenhum valor inicial no *view*, pois dessa forma a aplicação dá erro e fecha. Inserindo um *observer* no objeto do *view* pode-se receber a notificação quando o *view* é totalmente carregado e desta forma podem-se declarar os valores iniciais.

4.5.4. Padrão *Proxy*

Este padrão tem como finalidade tomar o lugar de um objeto, de forma a controlá-lo. Desta forma, quando a criação de um objeto exige muito poder de processamento e não se justifica fazer com que o utilizador espere, utilizar o padrão *proxy* é uma boa alternativa.

Nesta aplicação utilizou-se este padrão para realizar longos cálculos de informação, em longas listas de objetos. Por exemplo, quando o utilizador inicia a aplicação é apresentada uma lista com todas as carreiras com o próximo autocarro a sair e esta informação representa muitas pesquisas e cálculos, em dezenas de milhares de registos. Logo, utilizou-se o *proxy* para auxiliar nestas pesquisas, de forma que ao navegar pelas listas, o movimento seja fluído.

5. Implementação

“Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program”

(Linus Torvalds).

Após terem-se definido todos os passos anteriores, estão reunidas todas as condições para passar à fase de implementação. Começou-se, então, por desenhar os *layouts* que já tinham sido esboçados nos *wireframes* e, posteriormente, passou-se para o código, de forma a interligar a componente visual com o código.

5.1. Desenho da interface

Para cada atividade que se utiliza, criou-se um *layout* principal através de um ficheiro *XML* e gravou-se em duas localizações na pasta *layout* (para *Smartphones*) e na pasta *layout-large* (para *Tablets*). A partir deste momento, para evitar repetir mais entradas nos ficheiros *XML*, utilizou-se um componente que se chama *fragment-layout* e, através deste, pode-se importar outros ficheiros *XML* com outros componentes. Isto faz com que se possa criar apenas uma secção do *layout* e inserir nos *layouts* principais, da forma que se pretender. Para além disso, facilita nas modificações, pois se se apagar ou se se alterar algum componente de uma destas secções, as alterações surgem em todos os *layouts* principais.

Também se criaram mais duas pastas, *layout-land* e *layout-large-land*, de modo a adaptar melhor alguns *views* que ficavam distorcidos quando o dispositivo era girado do modo *portrait* para o modo *landscape*.

Desta forma, evita-se repetir código *XML* e poupa-se muito do trabalho e tempo a desenhar para dois dispositivos diferentes.

5.1.1. Versão para *Smartphone*

Na versão para *Smartphone*, como é óbvio, não se podem mostrar todas as funcionalidades disponíveis ao utilizador ao mesmo tempo, uma vez que o espaço é limitado. Por esse motivo, optou-se por mostrar as opções disponíveis na atividade e após o utilizador selecionar a opção, o menu com as opções é escondido e o resultado

é mostrado. Para voltar às opções, basta o utilizador carregar no botão voltar (*return*) e o *layout* com o resultado é escondido e volta a ser mostrado o *layout* com as opções.

Na Figura 5.1 e na Figura 5.2 pode-se ver um exemplo da atividade Mapa (*Maps*) e de como este processo se desenrola na aplicação.

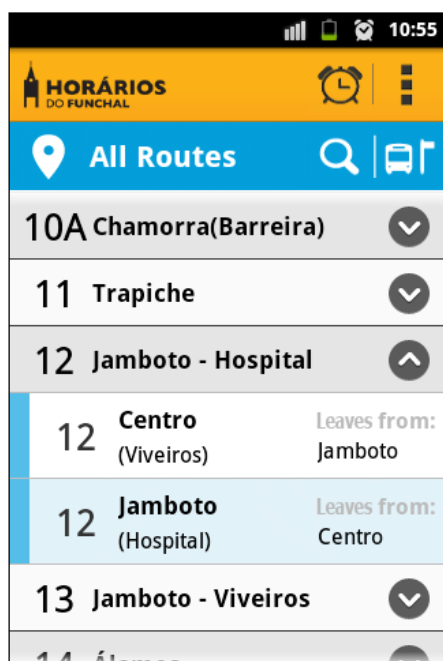


Figura 5.1 – Opções disponíveis na Atividade *Maps*

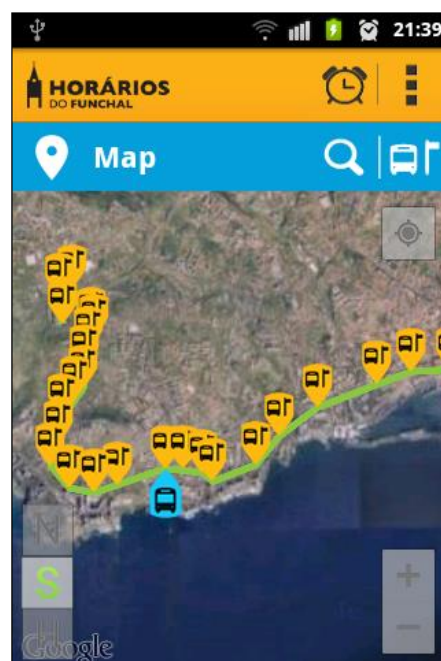


Figura 5.2 – Exemplo do resultado após ser selecionada uma Carreira

Outro aspeto que se teve em consideração foi o facto de diferentes tamanhos de ecrã e diferentes resoluções gerarem diferentes *DPI*. Desta forma, as margens dos componentes ou os tamanhos dos próprios componentes ficam diferentes, dependendo do *DPI* do dispositivo. Para resolver estes problemas teve-se de duplicar as imagens utilizadas para os diferentes tamanhos e teve-se de criar um ficheiro *XML* com as dimensões das margens, dependendo do tamanho do dispositivo. Também se definiram os tamanhos dos componentes na unidade *DP* que é a unidade complexa da Android que tem em consideração a densidade do dispositivo (*DPI*) [30], ou seja:

$$Px = DP * (DPI / 160) \quad [30]$$

5.1.2. Versão para *Tablet*

Na versão para *Tablet* tem-se mais espaço disponível e por isso podem-se mostrar as opções e os resultados ao utilizador no mesmo *layout*. Assim, e utilizando o exemplo acima, na atividade *Maps* pode-se mostrar a lista com as Carreiras e as outras opções do lado direito do monitor e no lado esquerdo o mapa com os resultados (ver Figura 5.3).

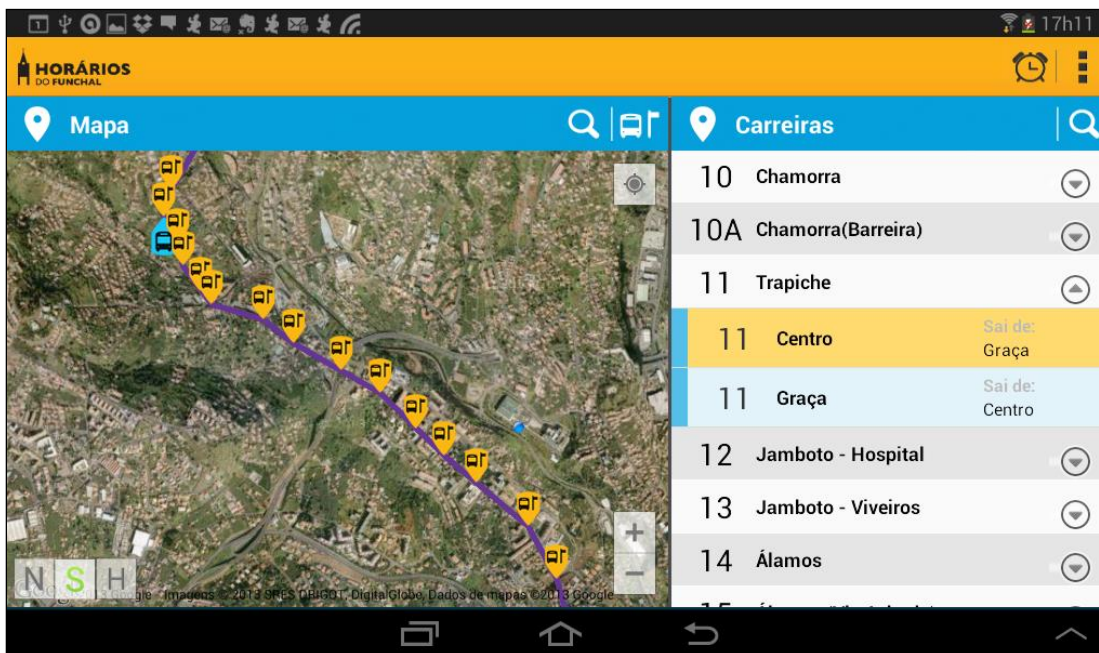


Figura 5.3 – Opções e resultados disponíveis na atividade *Maps*, na versão para *Tablet*

Para as outras atividades seguiu-se o mesmo esquema, tanto na versão de *Smartphone* como na versão de *Tablet*.

5.2. Integração de múltiplas línguas

A implementação de várias línguas num dispositivo pode revelar-se uma tarefa árdua. Felizmente, da mesma forma que os *layouts* e as dimensões podem ser diferentes dependendo do dispositivo, as línguas também o podem ser. Para isso, basta criar uma pasta com o nome *values* e depois do travessão, colocar as iniciais do país. Por exemplo, para a língua portuguesa basta criar uma pasta com o nome *Values-PT* e dentro cria-se um ficheiro XML com as *strings*. O resto do sistema da Android encarrega-se de verificar o ID de cada *string*, de verificar o idioma do dispositivo e de apresentar ao utilizador as palavras corretas (ver Figura 5.4 e Figura 5.5).

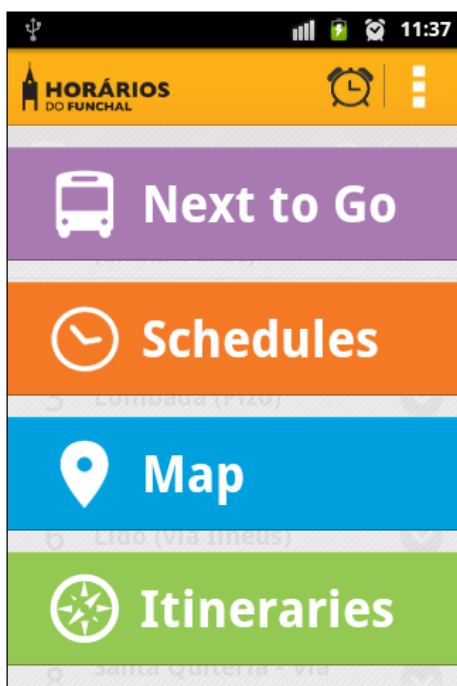


Figura 5.4 – Menu da aplicação, versão em Inglês

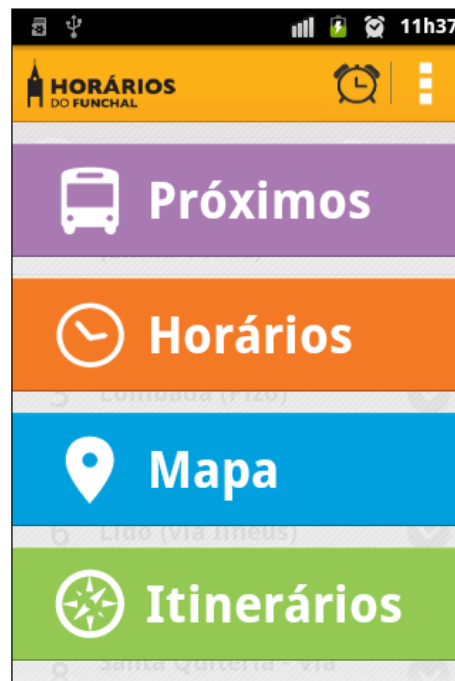


Figura 5.5 – Menu da aplicação, versão em Português

5.3. Funcionalidades da aplicação

Para segmentar as funcionalidades criaram-se várias atividades, sendo que cada uma delas carrega um *layout* principal e diversos outros secundários. Nos pontos seguintes deste documento utilizou-se a versão de *Smartphone* para descrever as funcionalidades. No Anexo D podem-se observar estas mesmas funcionalidades implementadas na versão para *Tablet*.

5.3.1. Menu

De modo a implementar-se o menu (requisito F1), criou-se um *layout* com os botões das atividades da aplicação. Através deste menu, pode-se aceder a todas as opções disponíveis na aplicação e como tal está sempre acessível na aplicação através do botão menu, que se encontra no canto superior direito do *view*.

De modo a evitar declarar os eventos dos botões em todas as atividades, criou-se o objeto *MenuManager* que é responsável por tratar disto. Desta forma, basta declarar este objeto em todas as atividades e enviar o contexto da atividade em que se

encontra. Na Figura 5.6 pode-se observar o botão menu que está a *highlight* e o próprio *layout* do Menu que aparece sobreposto ao *layout* da atividade em que se encontra.

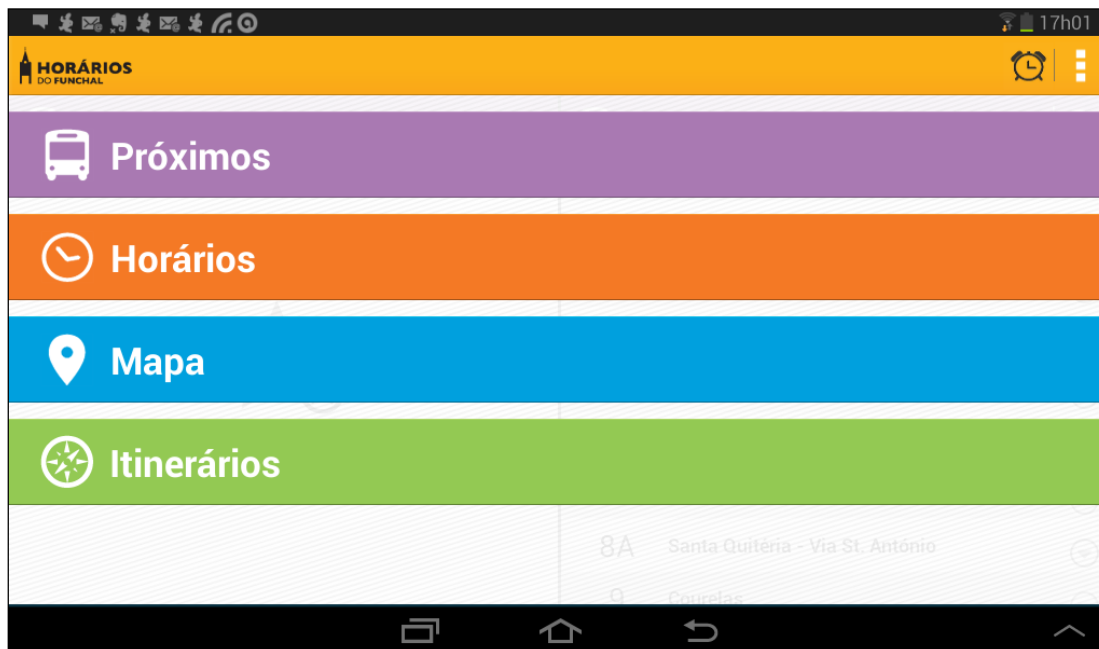


Figura 5.6 – Menu aberto com as opções de todas atividades disponíveis da versão para *Tablet*

5.3.2. Atividade Próximas Carreiras

A atividade Próximas Carreiras é a responsável por mostrar a hora do próximo autocarro a sair em cada Carreira. Nesta atividade também se podem associar carreiras favoritas, de forma a se ter sempre à mão o horário e outras informações das Carreiras que mais se utiliza. Nesta atividade utilizam-se vários requisitos funcionais como o requisito F2, F3, F4, F14, F16, F24 e o F25, e também utiliza-se alguns requisitos não funcionais como o requisito NF6, NF7, NF8, NF9, NF10, NF11, NF13, NF14, NF17, NF18.

Na Figura 5.7 pode-se observar a lista com todas as Carreiras e as respetivas variantes (subidas e descidas de uma Carreira) na versão de *smartphone*.

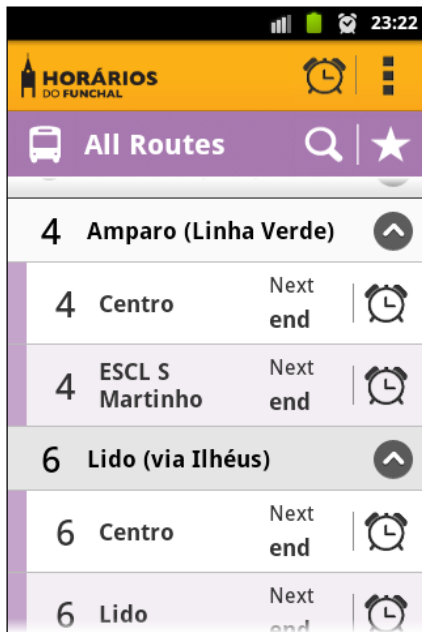


Figura 5.7 – Lista de todas as Carreiras com a hora do próximo autocarro

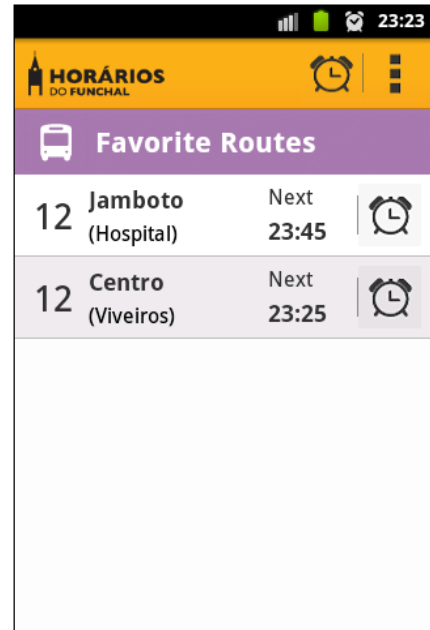


Figura 5.8 – Lista de Carreiras/Variantes favoritas

Como existem cerca de 70 Carreiras e 257 variantes é útil ter-se uma secção com as Carreiras e variantes favoritas (ver Figura 5.8 acima). Desta forma se o utilizador clicar no botão em forma de estrela, aparece uma janela com as Carreiras previamente seleccionadas pelo utilizador como favoritas. Caso pretenda adicionar ou remover a Carreira dos favoritos, basta fazer a acção *long press* sobre o item da Carreira e aparecerá o ecrã da Figura 5.9.

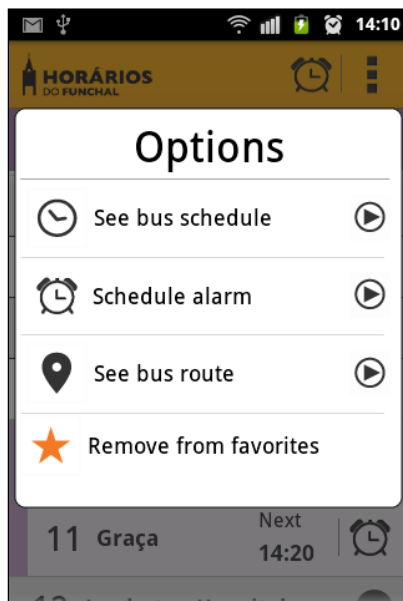


Figura 5.9 – Submenu com opções referentes às Carreiras/Variantes

Ainda se criou outra funcionalidade para facilitar o acesso à Carreira pretendida sem ter de se fazer *scroll*: inseriu-se uma barra de pesquisa, para que o utilizador possa pesquisar a Carreira através do nome ou do número. E esta pesquisa é feita em tempo real, ou seja, cada vez que é inserida uma letra na barra, é realizada uma pesquisa à base de dados local. Pode-se observar o resultado na Figura 5.10.



Figura 5.10 – Pesquisa de Carreiras/Variantes através da barra de pesquisa

5.3.3. Atividade Alarmes

Na atividade Alarmes tem-se uma lista com todas as viagens agendadas e os respectivos lembretes associados. Nesta atividade podem-se agendar novas viagens, editar ou eliminar. Esta funcionalidade tem como principal objetivo contribuir para que as pessoas mais distraídas evitem perder o autocarro.

Nesta atividade cumprem-se os requisitos funcionais: F14, F15 e F16 e também cumprem-se os requisitos não funcionais: NF6, NF7, NF8, NF9, NF10, NF11, NF17, NF18 e NF19.

Esta funcionalidade utiliza o *Google Calendar* para registar as viagens e caso o utilizador tenha acesso à internet é realizada a sincronização dos calendários com outros dispositivos. Também tem associado dois serviços que têm como objetivo auxiliar na notificação do utilizador (No ponto 5.3.9 e no 5.3.10 refere-se o objetivo destes dois serviços).

5.3.3.1. Adicionar Alarme

Apesar de existir uma atividade destinada ao agendamento de viagens, estendeu-se esta funcionalidade a outras atividades como na atividade Próximas Carreiras e na atividade Horários.

Para adicionar um alarme na atividade Próximas Carreiras basta clicar no item de uma Carreira/Variante e aparecerá um *popup* com as opções de selecionar a paragem e o tempo para o lembrete (como se pode observar na Figura 5.11). Se já tiver um alarme agendado, aparecerá o ícone de alarme com a cor amarela, como se pode observar na Figura 5.12.

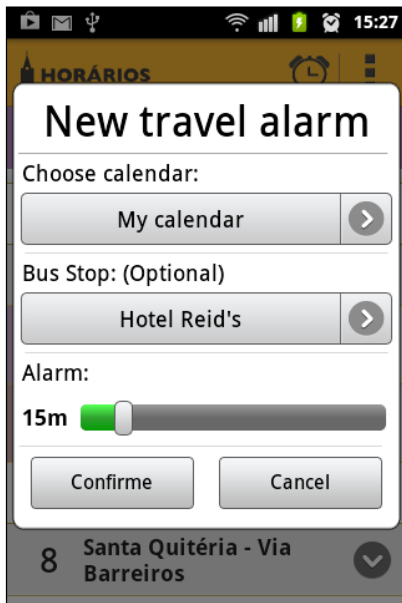


Figura 5.11 – *Popup* com as opções de agendar um alarme na atividade Próximas Carreiras

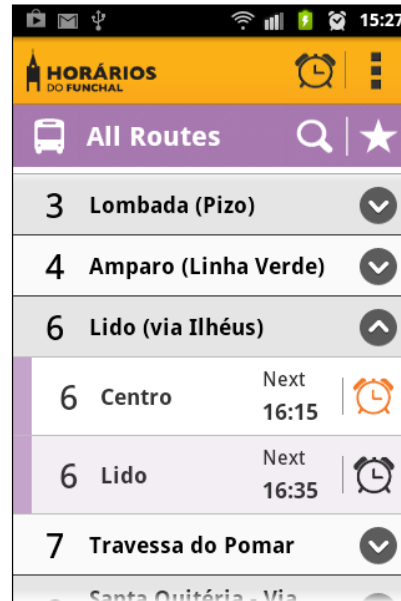


Figura 5.12 – Alarme agendado através da atividade Próximas Carreiras

Na atividade Horários procede-se de forma semelhante, seleciona-se a Carreira e Variante de que se pretende ver os horários e aparece a lista com as horas dos autocarros e um pequeno alarme ao lado de cada hora (observar Figura 5.13). Quando o utilizador clica sobre uma hora do horário aparece o mesmo *popup* que na Figura 5.11.

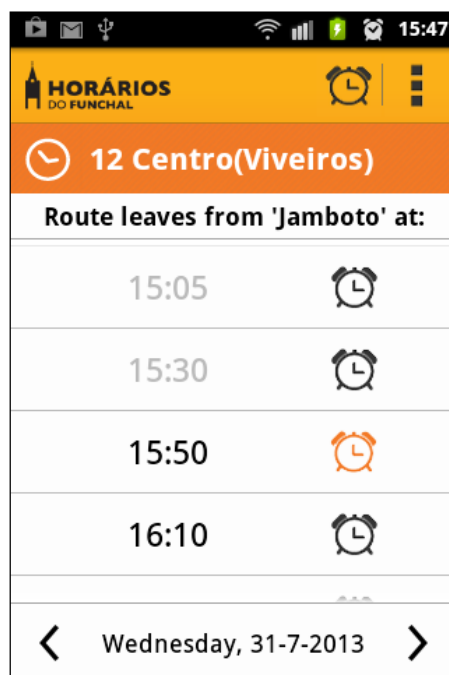


Figura 5.13 – Alarme agendado através da atividade Horários

Para agendar um alarme na atividade Alarme, basta clicar no botão mais (observar Figura 5.14) e aparece uma nova janela com todas as opções que apareciam no *popup* (escolher o calendário, escolher a paragem e escolher o tempo para o lembrete) e com mais duas opções que eram escolhidas por omissão nas outras atividades (a data e a carreira). Nas outras atividades seleciona-se a Carreira/Variante ao clicar na lista e a data era a data do dia actual. Na atividade Próximas Carreiras e na atividade Horários era a data do dia para o qual estava-se a ver o horário.

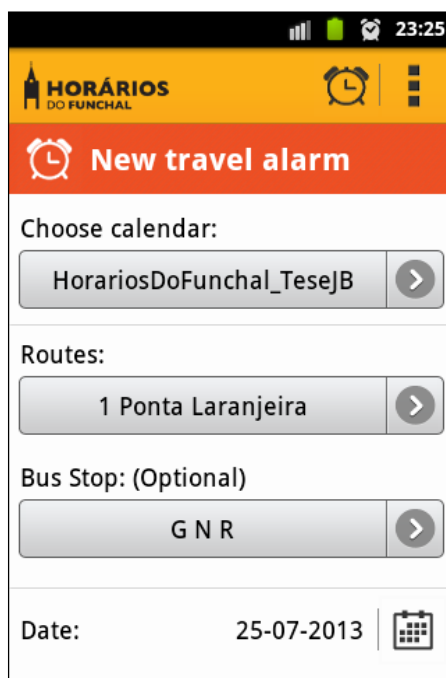


Figura 5.14 – Agendar viagem na atividade Alarmes

5.3.3.2. Editar Alarme

Para editar um agendamento de uma viagem tem-se obrigatoriamente de ir à atividade Alarmes. Isto porque nas outras atividades apenas é possível mudar o calendário onde vai ser agendada a viagem, a paragem e o tempo para o lembrete, deixando as outras opções bloqueadas, visto que são pré-selecionadas.

A diferença entre a funcionalidade de editar alarme e a de adicionar alarme é o nome e em termos de opções, aparece as configurações da viagem que se está a alterar e não a primeira opção de cada uma das listas.

5.3.3.3. Remover Alarme

Já remover um agendamento de uma viagem pode ser feito em qualquer uma das três atividades, visto que o único requisito para apagar um agendamento é o identificador do próprio e a confirmação do utilizador. Como se pode observar na Figura 5.15, aparece um *popup* antes de apagar o agendamento da viagem definitivamente, visto que o utilizador pode clicar sem querer em cima do botão de apagar.

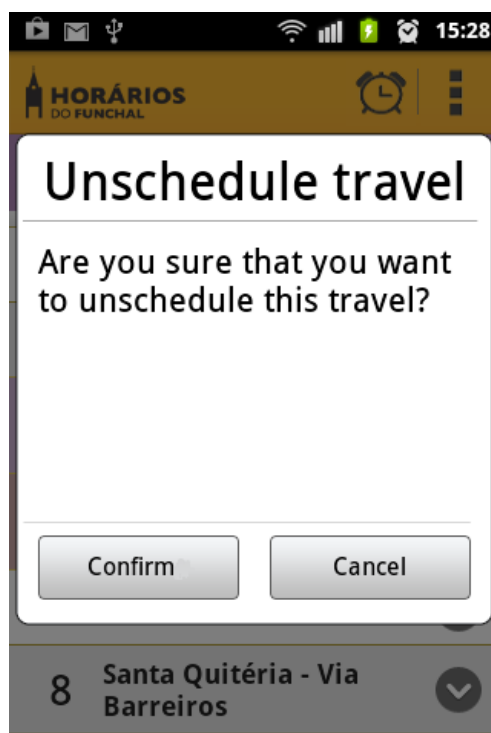


Figura 5.15 – *Popup* de confirmação para apagar agendamento

5.3.4. Atividade Horários

A atividade Horários, como o nome indica, é responsável por tratar dos horários de todas as carreiras durante todo o ano. Esta atividade segue o mesmo padrão de desenho das outras atividades: primeiro mostra uma lista com todas as carreiras (observar Figura 5.16) e posteriormente o utilizador seleciona a carreira pretendida e a janela com a lista desaparece e é apresentada a janela com os horários do dia atual da Carreira/Variante selecionada (observar Figura 5.17).

Esta actividade cumpre os requisitos funcionais: F2, F4, F14, F16, F17, F18 e F19 e também cumpre os requisitos não funcionais: NF6, NF7, NF8, NF9, NF10, NF11, NF14, NF17, NF18 e NF19.

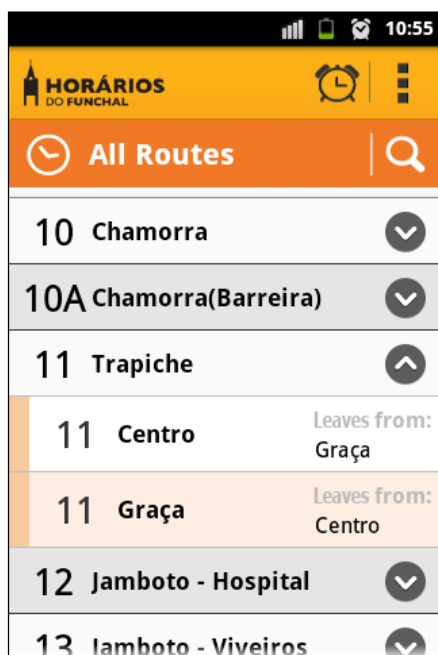


Figura 5.16 – Lista de todas as Carreiras/Variantes na atividade Horários

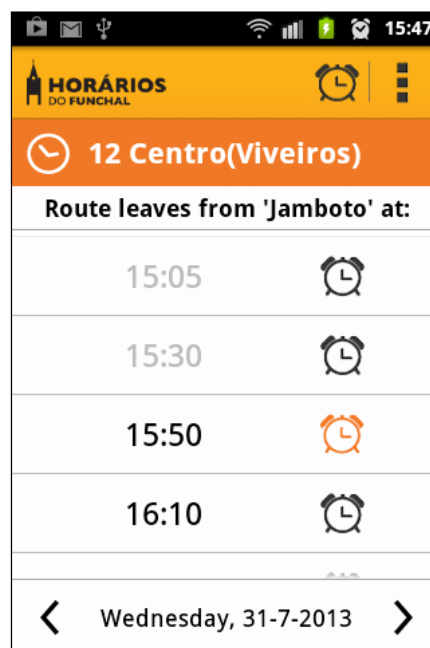


Figura 5.17 – Horários diários da carreira selecionada

Para selecionar o dia seguinte ou o dia anterior, basta clicar no botão com a seta para a direita (mostra o dia seguinte) ou a seta para a esquerda (mostra o dia anterior). Desta forma, o utilizador não precisa de se preocupar se o dia para o qual está a consultar os horários é um dia útil, um sábado ou um feriado/domingo e em que época do ano se encontra, pois a aplicação trata dessa informação pelo utilizador. Através do *website* a única forma de consultar os horários é através de um documento PDF com uma tabela de vários horários que mudam consoante o dia do ano e o utilizador tem de saber se o dia para o qual está a consultar é um feriado/domingo, sábado ou dia de semana.

Utilizaram-se diferentes cores para mostrar ao utilizador quando uma determinada hora do horário já tinha passado, quando tinha uma viagem agendada e ainda quando a viagem estava por se realizar (observar Figura 5.17 acima).

5.3.5. Atividade Mapa

Para cumprir os requisitos funcionais F2, F4, F5, F6, F7, F8, F9, F10, F11, F12 e F13 e os não funcionais NF4, NF6, NF7, NF8, NF9, NF10, NF11, NF14, NF15, NF17, NF18 e NF19 criou-se a atividade Mapa que tem como principal objetivo auxiliar aqueles

utilizadores que não conhecem muito bem a região ou que pretendam procurar por alguma carreira ou paragem que não estejam habituados a utilizar. Esta atividade também segue o padrão definido para a aplicação, mostrando a lista com as Carreiras/Variantes primeiro (observar Figura 5.18), mas com a exceção de possuir um botão extra que serve para saltar a escolha da Carreira/Variante e ver apenas o mapa com todas as paragens disponíveis (observar Figura 5.19).

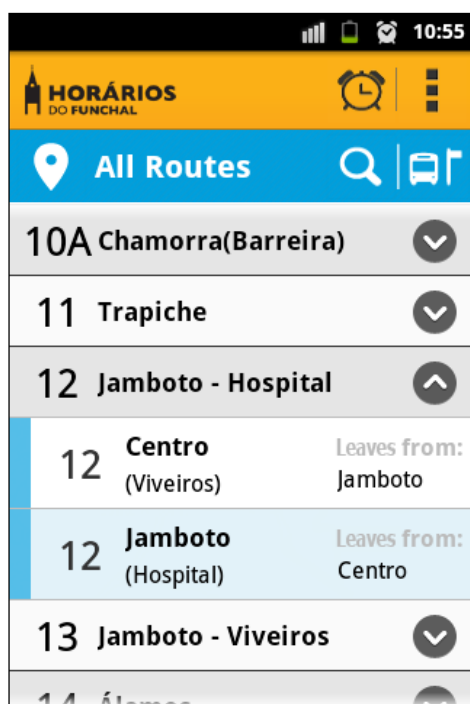


Figura 5.18 – Lista de todas as Carreiras/Variantes na atividade Mapa

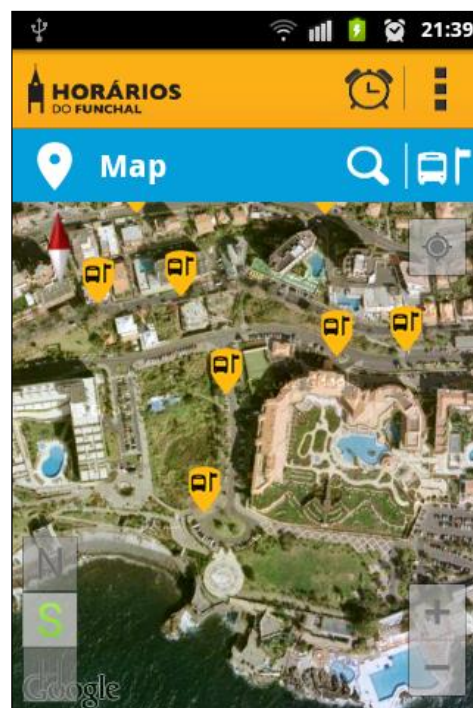


Figura 5.19 – Vista de pesquisa livre no mapa com todas as paragens visíveis

5.3.5.1. Rota da Carreira

Se o utilizador selecionar uma Carreira/Variante da lista é apresentada uma janela com a rota, as respetivas paragens e a posição GPS dos autocarros em tempo real (observar a Figura 5.20). Depois o utilizador ainda pode saber quais são as outras Carreiras que passam pelas paragens daquela rota, como se pode observar na Figura 5.21.

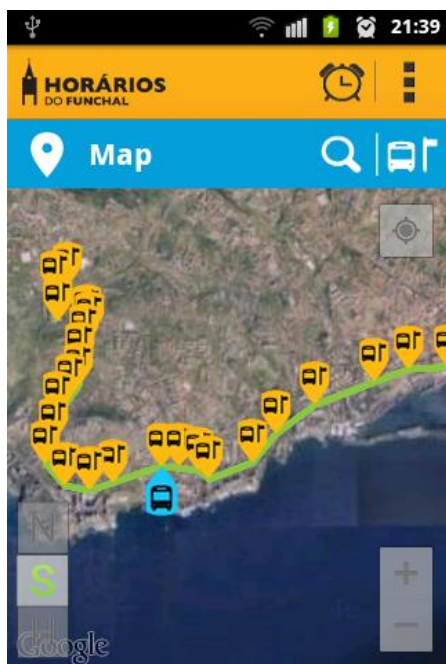


Figura 5.20 – Rota da carreira seleccionada, com as respetivas paragens e posição do autocarro em tempo real



Figura 5.21 – Informações referentes a paragem X, como o nome, o código e as Carreiras que passam por ali

5.3.5.2. Horário na Paragem

O utilizador também pode saber o tempo que falta para o autocarro chegar a uma determinada paragem, para isso basta clicar sobre uma paragem duas vezes e aparecerá um *popup* com os minutos restantes para cada uma das Carreiras que passam na respetiva paragem, como se pode observar na Figura 5.22. Para utilizar esta funcionalidade é necessário o utilizador ter uma ligação à internet por Wi-Fi ou por dados.

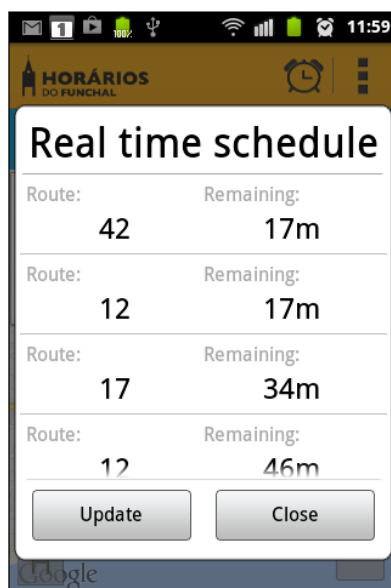


Figura 5.22 – Horário em tempo real para a paragem X

5.3.5.3. Pesquisa por Nome

Na atividade Mapa é útil ter uma pesquisa por nome, visto que o utilizador pode não saber qual ou quais as Carreiras que vão para um determinado destino. Logo inseriu-se uma barra de pesquisa para se poder escrever o nome de um concelho, de uma localidade, de uma rua ou um marco histórico, entre outras (visualizar Figura 5.23).



Figura 5.23 – Pesquisa de lugares

A pesquisa pode ser feita através de dois modos: *online* e *offline*. No modo *online* utiliza-se a *API* do *Google Places* para realizar a pesquisa, sendo que se envia a

palavra para o servidor da Google e posteriormente a Google devolve uma lista de resultados com os pontos *GPS* e a descrição dos lugares. No modo *offline* utilizam-se as informações referentes às paragens para devolver os resultados, visto que os nomes das paragens são na maioria dos casos o nome da rua ou do concelho ou até de um ponto de referência da cidade, como é o caso dos centros comerciais.

5.3.6. Atividade Itinerários

A atividade Itinerários tem como objetivo calcular a rota entre dois pontos *GPS*. Esta rota inclui a deslocação a pé do utilizador desde a origem até uma paragem, a escolha da Carreira certa, a paragem de descida e a deslocação a pé da paragem até ao ponto de destino. Esta funcionalidade ainda não está preparada para cálculos de itinerários com auxílio de mais de uma Carreira.

Esta atividade utiliza os seguintes requisitos funcionais: F20, F21, F22 e F23 e também utiliza os seguintes requisitos não funcionais: NF4, NF6, NF7, NF8, NF9, NF10, NF11, NF17, NF18, NF16 e NF19.

A atividade vai guardar um histórico de itinerários pesquisados anteriormente e pode-se facilmente voltar a ver os resultados, clicando no item da lista do histórico (observar Figura 5.24).

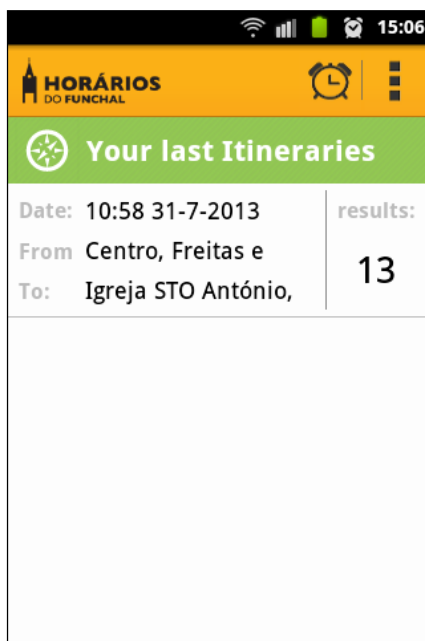


Figura 5.24 – Histórico de itinerários

Para realizar a pesquisa de um novo itinerário, basta preencher o campo de origem e o campo de destino. Isso pode-se fazer de três formas:

1. Marcar ponto no mapa – O botão com a forma de um marcador redireciona para uma janela com o mapa da Google Maps e um marcador que o utilizador pode arrastar livremente pelo mapa, de forma a escolher o ponto pretendido (ver Figura 5.28).
2. Pesquisa por texto – Se o utilizador preferir pode escrever o nome da zona ou local pretendido e é automaticamente feita uma pesquisa pela API do Google Places e posteriormente é apresentada uma lista com os resultados (ver Figura 5.26).
3. Utilizar um ponto guardado – Também existe a opção de utilizar um ponto que tenha sido previamente guardado, selecionando um destes através do *widjet spinner* que aparece junto da caixa de texto (ver Figura 5.27). Ao escolher o ponto no mapa, o utilizador pode optar por guardar o ponto, caso pretenda utiliza-lo para pesquisas futuras.

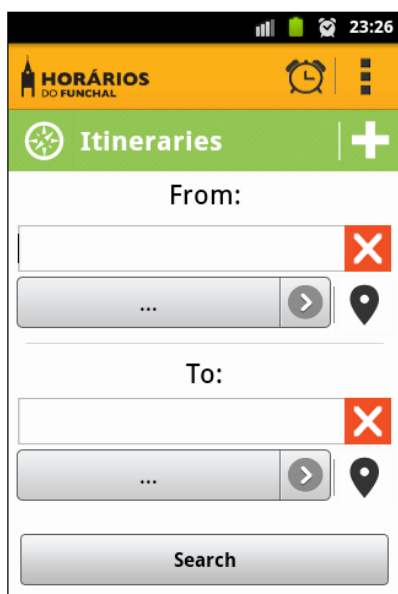


Figura 5.25 – Pesquisar novo itinerário

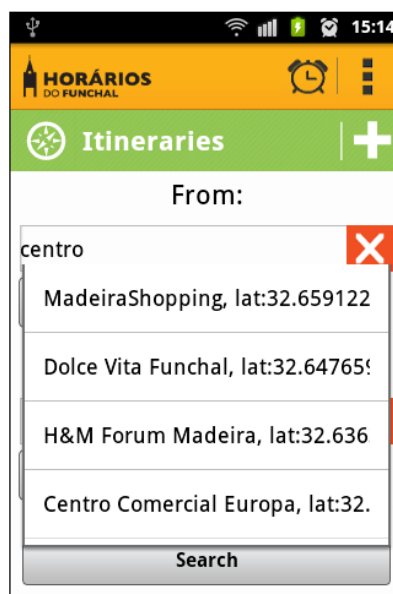


Figura 5.26 – Pesquisar novo itinerário através do modo de texto

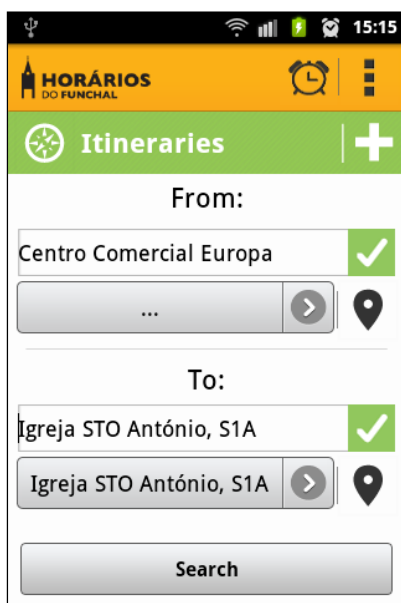


Figura 5.27 – Pesquisa de Itinerário através de pontos previamente guardados

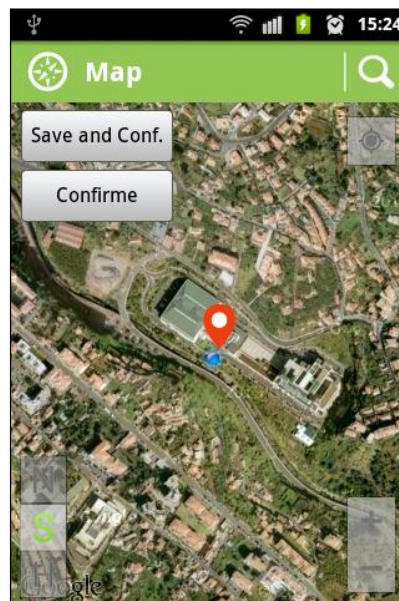


Figura 5.28 – Pesquisa de itinerário através de escolha do ponto no mapa

5.3.7. Atividade Contatos

De modo a que os utilizadores tenham algum suporte caso necessitem de contactar diretamente a empresa Horários do Funchal, criou-se uma secção com o número de telefone, *email*, localização e *link* para o *website*. Além dos contatos da empresa, inseriram-se os contatos do programador caso o utilizador queira contactar diretamente o programador, para algum problema relacionado com a aplicação.

Esta atividade cumpre os requisitos funcionais F26, F27, F28 e F29 e também cumpre com os requisitos não funcionais NF6, NF7, NF8, NF9, NF11, NF17, NF18 e NF19.

5.3.8. Serviço de Sincronização

Para que a aplicação esteja sempre atualizada é necessário verificar periodicamente se existem atualizações nos servidores dos Horários do Funchal. Por esse motivo criou-se um serviço que é executado em segundo plano e que trata de verificar se existem atualizações nos dados a descarregar. Esta verificação ocorre cada vez que o dispositivo deteta uma ligação à internet. Este serviço cumpre com os requisitos não funcionais NF3, NF5 e NF19.

O objetivo deste serviço é que atualize toda a informação, sem que o utilizador se aperceba e, por isso, não apresenta nenhum *layout* com informações. No entanto, na primeira vez que o utilizador usa a aplicação, esta não tem quaisquer dados para mostrar, logo foi necessário criar-se um pequeno *view* com um *loading* para o utilizador não pensar que a aplicação está com algum problema (ver Figura 5.29).

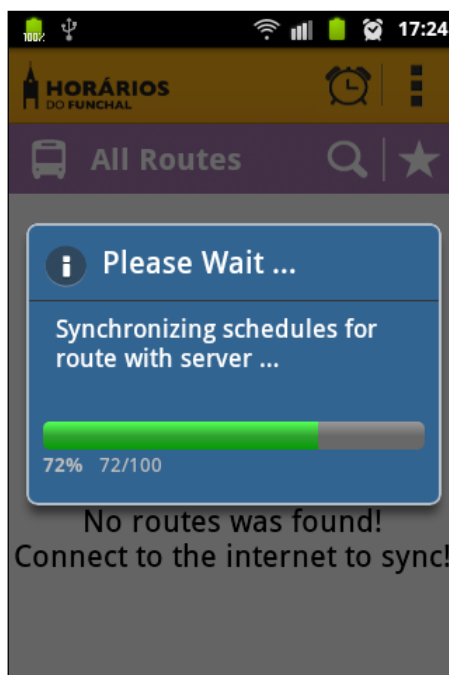


Figura 5.29 – Sincronização da base de dados *online* com a base de dados local

5.3.9. Serviço de Alerta Início de Viagem

O serviço de Alerta Início de Viagem serve para notificar o utilizador quando o autocarro relativo a uma viagem agendada pelo utilizador sai da primeira paragem. Este serviço foi criado para compensar o utilizador nas situações em que o autocarro sai atrasado ou sai mais cedo, visto que os motoristas ocasionalmente saem atrasados ou adiantados, por motivos de força maior.

Este serviço, ao contrário da notificação de agendamento de viagem, só toca no dispositivo em que foi criado o agendamento, ou seja, não é sincronizado com outros dispositivos através do Google Calendar. Também necessita de uma ligação constante à internet para funcionar, caso contrário pode não funcionar. Este serviço cumpre com os requisitos não funcionais NF1, NF5, NF17, NF18 e NF19.

5.3.10. Serviço de Alerta de Tempo Restante até à Paragem

Este serviço tem como objetivo notificar o utilizador de quando tempo falta para o autocarro, correspondente a um agendamento, chegar à paragem desejada. Tal como no Serviço de Alerta Início de Viagem, este serviço só funciona com uma ligação constante à internet e é opcional, pois quando o utilizador agenda uma viagem aparece a opção de seleccionar uma paragem referente à Carreira/Variante seleccionada e, por defeito, fica a primeira paragem da rota seleccionado. Este serviço, como no serviço anterior, cumpre com os requisitos não funcionais NF1, NF5, NF17, NF18 e NF19.

6. Testes e Resultados

“Program testing can be used to show the presence of bugs, but never to show their absence!”

(Edsger Dijkstra).

Como primeiro teste a esta aplicação, procuraram-se cinco voluntários que utilizassem os transportes públicos no seu dia-a-dia. Esta procura foi realizada através de um convite divulgado pelas redes sociais e através do *website* Moodle do Centro de Competências de Ciências Exactas e da Engenharia da Universidade da Madeira.

De vários voluntários, escolheram-se cinco que se pensou representarem melhor a amostra dos utilizadores alvo. Estes cinco utilizadores tinham os seguintes perfis:

- Jorge Lopes – Utilizador dos Horários dos Funchal, 27 anos, frequenta as seguintes Carreiras: 12, 13, 17, 15B, 32, e 42, todos os dias para se deslocar de casa à Universidade.
- David Olim – Utilizador assíduo dos Horários dos Funchal, 23 anos, frequenta as seguintes Carreiras: 13, 15, 15B, 20, 21, 46 e 48.
- Hugo Vieira – Utilizador dos Horários dos Funchal, 25 anos, frequenta as seguintes Carreiras: 1, 2, 4, 12, 13, 17, 28, 42 e 48, todos os dias para se deslocar de casa ao trabalho e vice-versa.
- Cláudia Vieira – Utilizadora assídua dos Horários dos Funchal, 24 anos, frequenta as seguintes Carreiras: 13, 15, 15B, 20, 21, 46 e 48.
- Sara Araújo – Utilizadora assídua dos Horários dos Funchal, 22 anos, frequenta as seguintes Carreiras: 15, 15B, 42, 46 e 49.

6.1. Testes de Usabilidade

Os participantes escolhidos começaram por realizar um teste de usabilidade, onde se utilizou o cenário descrito no Anexo C. Após realizar este teste com todos os utilizadores, analisaram-se os problemas que estes tinham em comum ao utilizar a aplicação e tomou-se nota de algumas preferências por parte destes.

Pôde-se comprovar que quase todas as atividades tinham alguns problemas na interação e alguma falta de informação, de modo a dar *feedback* do que o utilizador estava a visualizar. Enumeraram-se os seguintes problemas detetados nesta fase :

- Na atividade horários quando uma hora tem um agendamento, se se clicar outra vez o alarme é removido sem perguntar primeiro.
- Na atividade horários não apareceu escrito o destino/origem da carreira.
- Na atividade mapa esperava-se que o botão menu abrisse as opções do mapa.
- Na atividade mapa não identificou à primeira os ícones de localização e de posição do autocarro.
- Na atividade mapa não atualizou a posição do autocarro.
- Na atividade mapa não soube como ver as paragens se tivesse pouco *zoom*.
- Na atividade itinerários os *labels* estavam confusos.
- Ao clicar no botão para trás, nem sempre mostrava a atividade anterior.

Com estas informações, pôde-se melhorar a interface e as funcionalidades da aplicação, preparando assim uma versão de teste para os voluntários utilizarem no seu dia-a-dia.

6.2. Estudo de Utilização

Instalou-se uma versão de teste da aplicação nos dispositivos dos participantes, informando-os que podiam utilizar a aplicação no seu dia-a-dia à medida que sentissem necessidade. Além disso, foi pedido ainda que confirmassem os horários dos autocarros que frequentassem, pois a aplicação era uma versão *Beta*, e provavelmente conteria erros que lhes pudesse prejudicar a viagem.

Após uma semana de utilização, reuniu-se novamente com os participantes para recolher os dados (registados na base de dados da aplicação através do uso da mesma) e falar um pouco sobre a experiência de utilizar a aplicação. Estes dados foram recolhidos e convertidos para um ficheiro Excel, de forma a se poder analisar posteriormente.

No Gráfico 6.1 pode-se observar a distribuição da utilização das atividades. Desta forma pôde-se perceber qual foi a atividade mais utilizada.

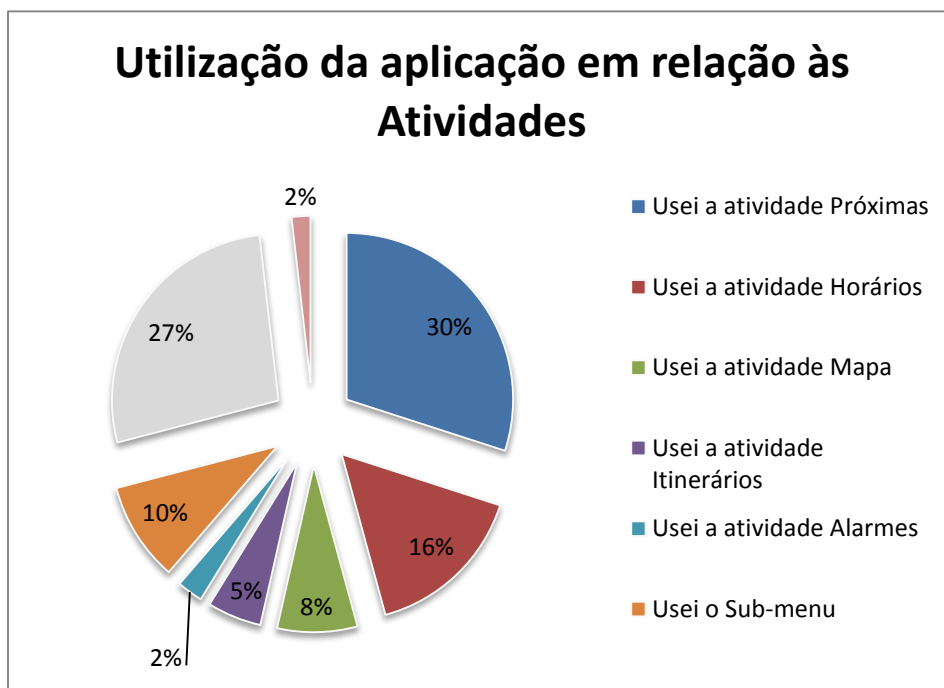


Gráfico 6.1 – Distribuição da utilização das atividades

Após normalizar o gráfico acima, a atividade mais utilizada, com 30%, foi a de Próximas Carreiras. As outras atividades mais utilizadas foram a dos Horários com 16% e a do Mapa com 8%. Este aspeto é o esperado, visto que os utilizadores quando questionados para que o utilizavam a aplicação, respondiam que a usavam para ver o horário das carreiras que utilizavam regularmente (é realizado na atividade Horários) e o horário em tempo real nas paragens (é realizado na atividade Mapa).

Se se analisar a utilização quanto às funcionalidades, pode-se perceber porque motivo as outras atividades (itinerários e alarmes) não foram muito utilizados (observar Gráfico 6.2). Note-se que todas as funcionalidades aparecem. Essas que não aparecem não foram utilizadas uma única vez pelo global dos participantes.

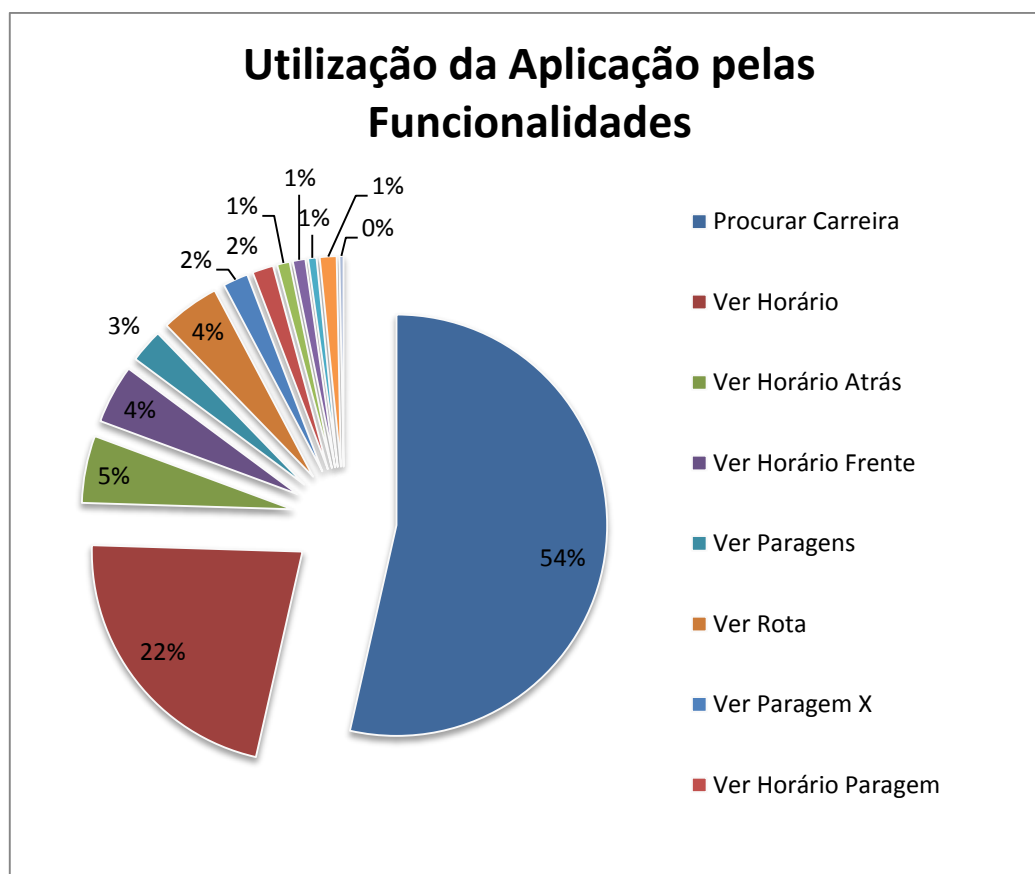


Gráfico 6.2 – Distribuição da utilização pelas funcionalidades

Curiosamente a funcionalidade mais utilizada, com mais de metade das utilizações (54%), é a da barra de pesquisa da aplicação para procurar por Carreiras. A funcionalidade de pesquisa e a de adicionar a Carreira aos favoritos foram criadas com o intuito do utilizador filtrar a lista mais depressa. No entanto, ao invés da funcionalidade de pesquisa, a funcionalidade de adicionar aos favoritos não foi utilizada nem uma única vez. Provavelmente, esta situação ocorreu devido ao facto de o utilizador ter usado a aplicação por pouco tempo e não a ter explorado ao máximo. Quando questionados sobre esta questão, os participantes responderam que devido à barra de pesquisa não sentiram necessidade de adicionar Carreiras aos favoritos.

A segunda funcionalidade mais utilizada foi a de visualizar os horários, com 22% (ver horários e navegar para frente (4%) e para trás (5%) nos horários), o que corresponde com a segunda atividade mais utilizada (Atividade Horários).

De seguida, têm-se as funcionalidades relativas à atividade Mapa, como visualizar as rotas (4%), as paragens (3%) e escolher uma paragem específica (2%).

As outras funcionalidades, como já se tinha notado anteriormente, não foram muito utilizadas. É o caso da funcionalidade de adicionar alarme e da de pesquisar por itinerários que, mais uma vez, possivelmente devido ao facto de os utilizadores não terem tido muito tempo para explorar a aplicação, e também por serem funcionalidades menos comuns para um utilizador habitual dos transportes públicos.

6.3. Avaliação do Utilizador

Após a entrega dos dados foi requerido que os participantes respondessem a um pequeno inquérito de satisfação, de forma a se registar a opinião dos participantes. Utilizou-se o inquérito representado no Anexo A para este fim.

O Gráfico 6.3 mostra os resultados da opinião dos participantes em relação à utilidade da aplicação.

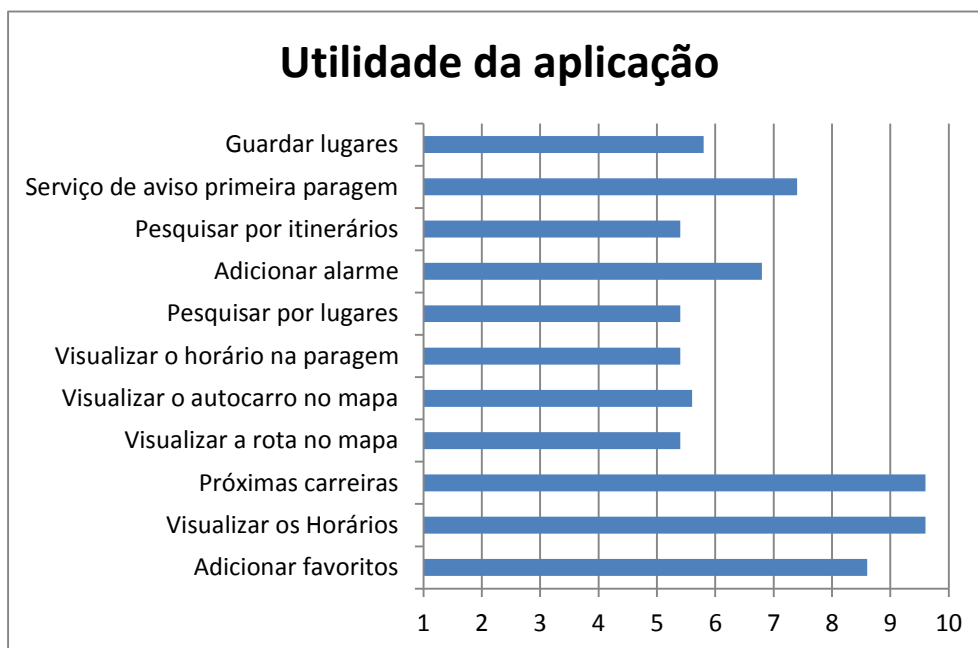


Gráfico 6.3 – Avaliação da utilidade da aplicação

Como se pode observar a funcionalidade mais popular é a de visualizar as próximas carreiras e a de visualizar os horários, com 9 valores, numa escala de 0 a 10. Logo de seguida tem-se a funcionalidade de adicionar favoritos, com 8,5 valores, revelando uma contradição, pois esta funcionalidade não foi utilizada por nenhum dos participantes durante o tempo de teste.

Todas as outras funcionalidades registaram uma média superior a 5 valores, o que significa que todas as funcionalidades são importantes, em certa medida, para os utilizadores.

Analisando a aplicação quanto ao *layout* da própria, registou-se uma avaliação muito positiva por parte dos participantes, obtendo no mínimo 8 valores na categoria de animações das tarefas, numa escala de 0 a 10. No Gráfico 6.4 pode-se observar a média da votação dos cinco participantes.

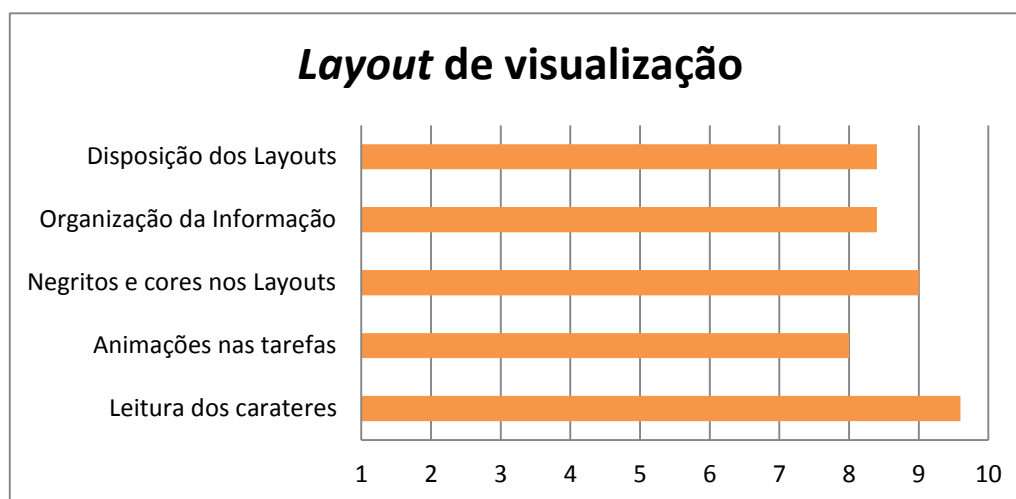


Gráfico 6.4 – Avaliação do *layout* de visualização

Pediu-se também aos participantes para avaliarem a aplicação em relação à terminologia da aplicação. Os resultados também foram muito positivos, obtendo valores de 7 para cima, como se pode observar no Gráfico 6.5.

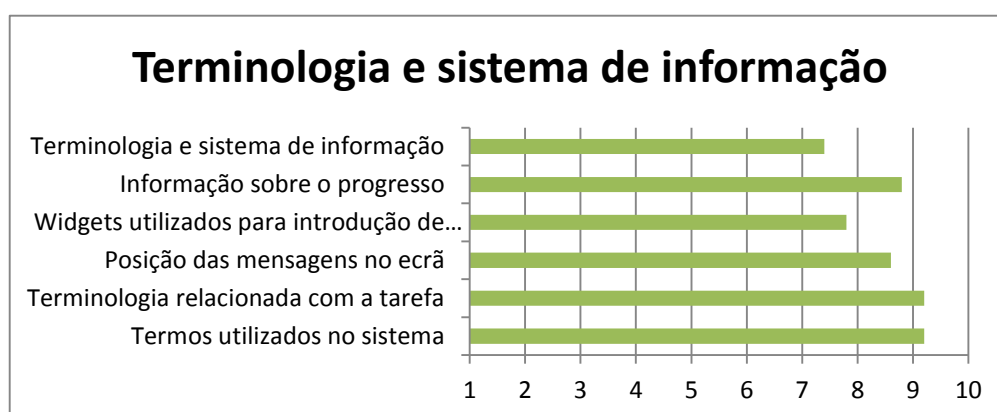


Gráfico 6.5 – Avaliação da terminologia utilizada

Em relação às capacidades do sistema, os resultados da avaliação foram um pouco mais baixos que em relação ao *layout* e à terminologia utilizados (como se pode

observar no Gráfico 6.6), mas ainda assim foram positivos, tendo-se registado como nota mais baixa a categoria de robustez da aplicação, com 7,5 valores. Como esta versão da aplicação não era a versão final compreende-se que os resultados sejam um pouco mais baixos, devido aos erros e falhas do sistema.

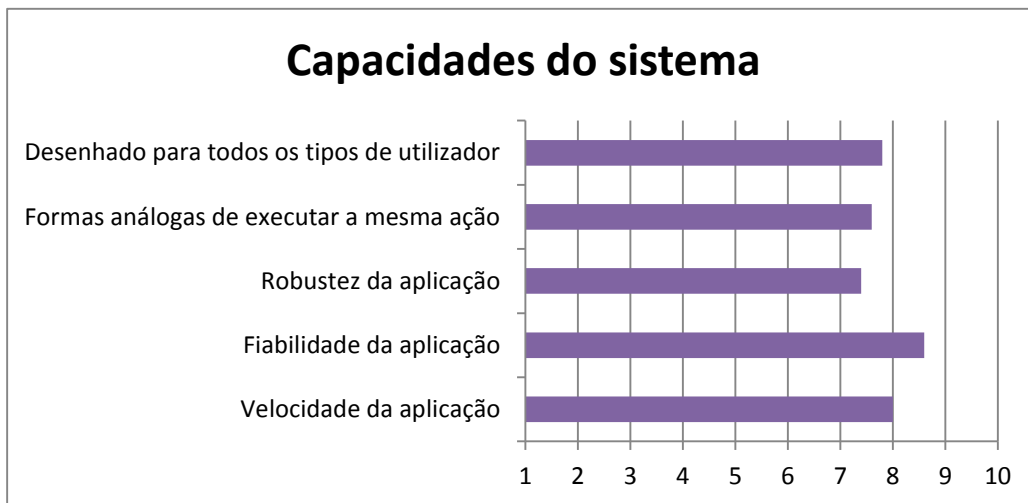


Gráfico 6.6 – Avaliação das capacidades do sistema

Quando se questionaram os participantes se tinham achado fácil a utilização da aplicação, se tinham percebido as mensagens e se tinham conseguido navegar facilmente entre as diferentes *views*, os utilizadores responderam muito positivamente, como se pode observar no Gráfico 6.7.

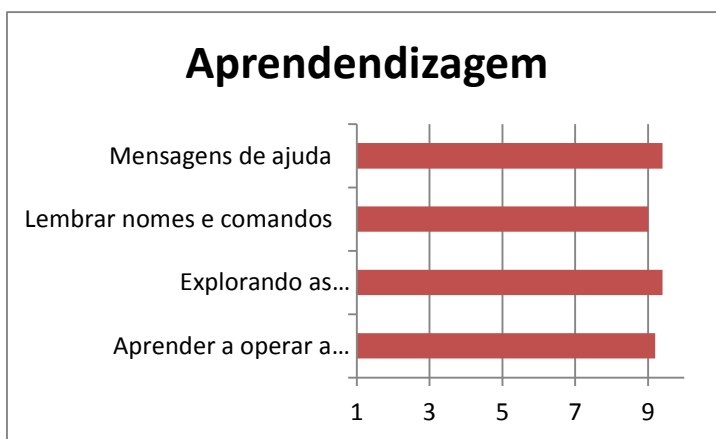


Gráfico 6.7 – Avaliação da aprendizagem do sistema

Posteriormente, realizaram-se questões de âmbito geral, de forma a tentar resumir a experiência de ter utilizado a aplicação durante esse curto período. No Gráfico 6.8 pode-se observar que os resultados foram satisfatórios na sua maioria.

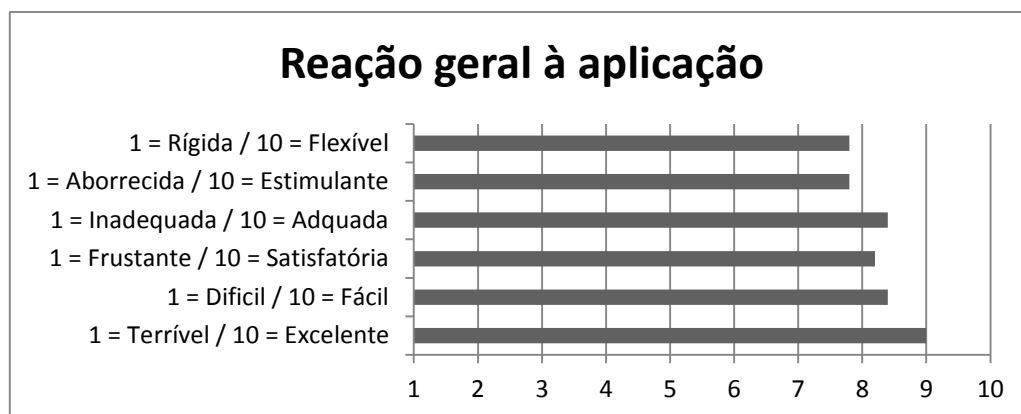


Gráfico 6.8 – Avaliação da reacção geral à aplicação

6.4. Discussão

Com os resultados obtidos durante este período de utilização e com os resultados dos testes de usabilidade pôde-se concluir alguns factos importantes para esta aplicação.

Ao analisarem-se as funcionalidades percebe-se que algumas são fundamentais, como por exemplo, mostrar os horários de uma forma dinâmica. Antes desta aplicação só era possível consultar os horários descarregando o ficheiro em PDF do *website* dos Horários do Funchal, e tinha-se de ter um conhecimento dos dias da semana e dos períodos utilizados pela empresa. Só por esta funcionalidade muitos utilizadores referiram que utilizariam a aplicação.

As funcionalidades relativas ao mapa e aos horários em tempo real foram das que mais despertavam interesse aos utilizadores quando ouviam falar que a aplicação continha este tipo de funcionalidades. Contudo não foram muito utilizadas quando comparadas com a visualização de horários. Provavelmente este facto deve-se a três factores: primeiro, um dos utilizadores não conseguiu utilizar os mapas devido ao tipo de dispositivo que tinha, segundo, porque os utilizadores conheciam o trajeto e as carreiras que queriam utilizar e por isso não exploraram muito através do mapa e terceiro, só dois dos utilizadores tinham internet móvel, daí que, quando não tinham ligação Wi-Fi não conseguiam aceder aos horários em tempo real.

As funcionalidades relativas ao agendamento, não foram muito utilizadas, mas visto que o tempo de testes foi curto e a amostra de utilizadores foi pequena decidiu-se

deixar esta funcionalidade para testar no futuro com uma amostra maior, por exemplo quando a primeira versão da aplicação fosse publicada.

Em relação aos textos e aos *layouts* escolhidos os níveis de satisfação foram muito bons, o que levou a concluir que não era preciso realizar muitas alterações nessa área.

7. Lançamento

“On the back end, software programming tools and Internet-based services make it easy to launch new global software-powered start-ups in many industries without the need to invest in new infrastructure and train new employees.”

(Marc Andreessen).

De modo a se ter uma ideia da opinião das pessoas decidiu-se realizar um lançamento da versão utilizada no último teste, e este lançamento teve por objetivo continuar o estudo anterior, mas com mais utilizadores e por um período mais extenso (quatro semanas).

Além disso, como foi feito em todos os estudos anteriores aproveitou-se este lançamento para testar o código em mais dispositivos, de modo a examinar a compatibilidade da aplicação. Para isso utilizou-se a plataforma oficial da Google, o *Google Play*.

7.1. Google Play

Ao utilizar-se a plataforma do Google Play tem-se acesso diversas vantagens que não se obtém em outros mercados. Por exemplo, ao utilizar o *Google Play* todos os utilizadores Android que tenham a versão oficial do sistema operativo, têm por defeito instalado a aplicação do *Google Play*.

Outra vantagem é o facto de através do Google Play os utilizadores poderem mandar um relatório com os erros que ocorrem durante o uso da aplicação, ou seja, quando a aplicação origina um erro durante a utilização, aparece uma janela *popup* a perguntar se o utilizador deseja notificar o programador sobre aquele erro, e dá a opção para escrever notas de algo que o utilizador queira comunicar. Posteriormente o programador recebe o relatório na consola de programador, como se pode observar na Figura 7.1, de modo a poder analisar e corrigir esses erros.

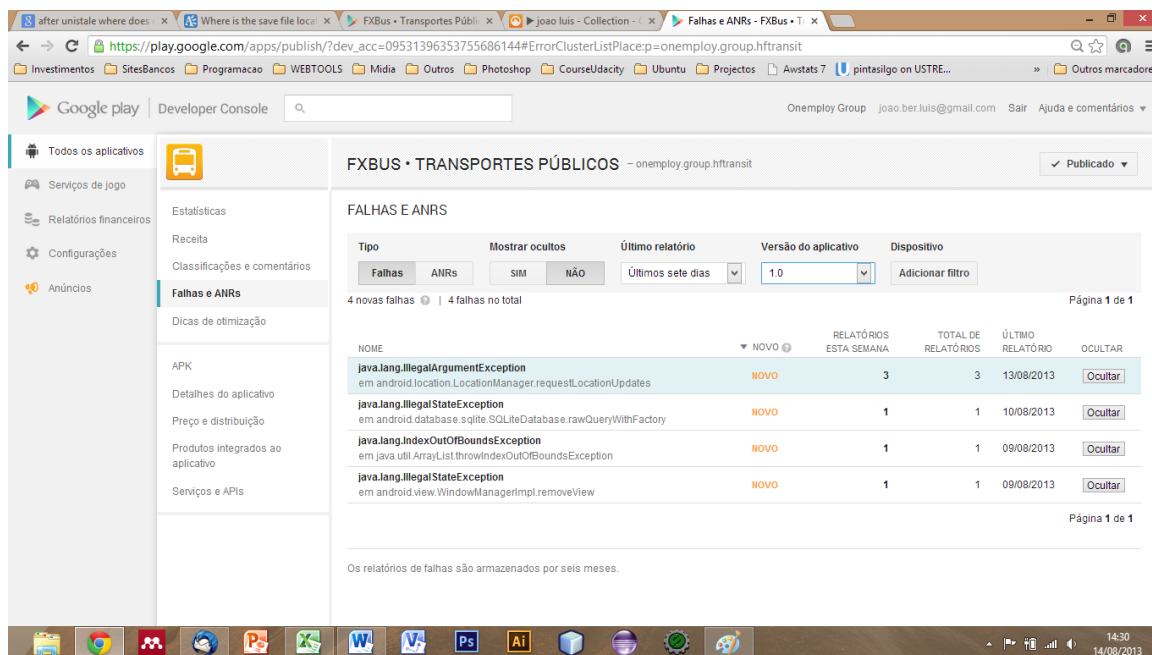


Figura 7.1 – Consola do programador, aba de relatórios de erro

É também possível que os utilizadores votem na aplicação e partilhem a sua opinião com outros utilizadores e com o programador. Desta forma pode-se registar a opinião geral dos utilizadores. Na consola de utilizador, o programador recebe as opiniões como se pode visualizar na Figura 7.2.

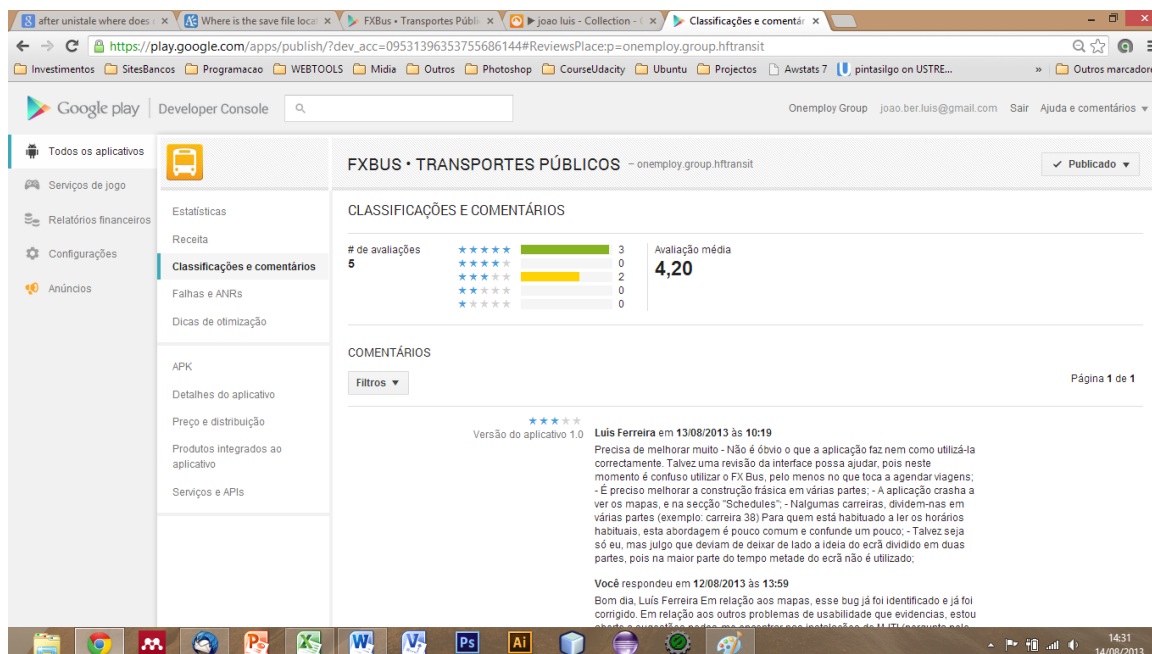


Figura 7.2 – Consola do programador, aba de classificações e comentários

Para os outros utilizadores, os comentários e as classificações ficam como demonstra a Figura 7.3.

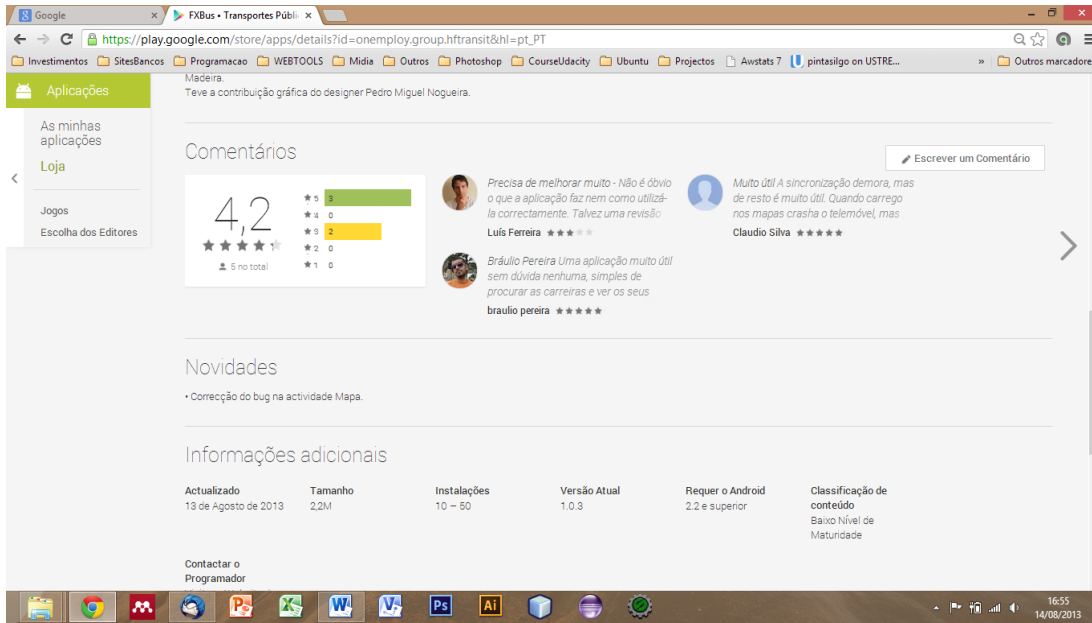


Figura 7.3 – Mercado Google Play, visualização da classificação e dos comentários

Através da consola do programador também se podem observar estatísticas de instalações e desinstalações da aplicação e outras informações, como os dispositivos que estão a utilizar a aplicação e qual o sistema operativo da Android que têm instalado. Pode-se observar o aspeto da consola na Figura 7.4.

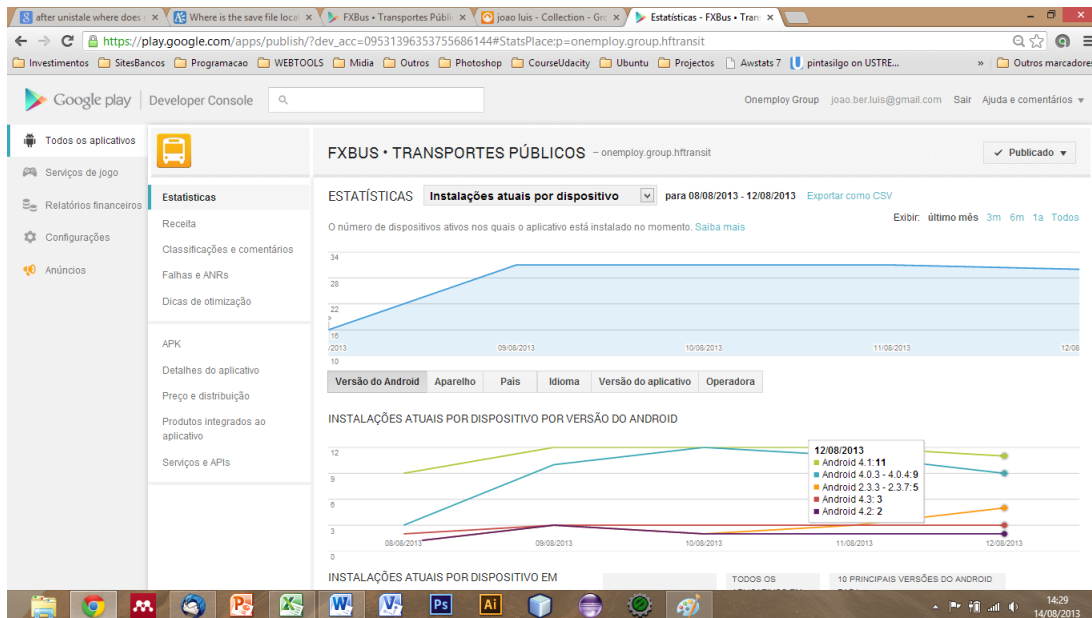


Figura 7.4 – Consola do programador, aba de estatísticas

7.1.1. Correção de Erros e Falhas

Nesta fase de lançamento, foram detetados alguns erros que não tinham sido detetados nas fases de testes. Muitos utilizadores confundiram a demora da primeira sincronização com um erro, ou seja, não acharam normal a aplicação demorar cerca de 7 minutos a sincronizar na primeira vez. Isto fez com que se pensasse noutra abordagem para sincronizar na primeira vez, em vez de adicionar os dados às tabelas linha a linha, utilizou-se o servidor para criar o ficheiro da base de dados com todos os dados e assim o dispositivo descarrega o ficheiro de uma só vez, demorando cerca de 3 segundos.

Também se descobriram alguns erros no mapa, pois o sistema de localização em alguns telemóveis sem GPS fazia o sistema falhar. Teve-se de fazer a verificação de *hardware* do dispositivo antes de fazer pedidos GPS.

7.1.2. Estudo de utilização

De modo a poder-se comparar os resultados do estudo de utilização anterior com a experiência dos utilizadores da versão de lançamento, utilizou-se um sistema de *logs* de utilização anónimos que descarregava para um servidor privado sempre que detetava uma ligação Wi-Fi.

Esta versão de testes esteve disponível desde o dia 9 de agosto até o dia 31 do mesmo mês. Durante este período o servidor armazenou cerca de 10000 registos, de um total de 72 utilizadores. Destes 72 utilizadores, 43 mantêm a aplicação ativa nos seus dispositivos, e destes conseguiu-se uma quantidade de dados significativa de 35 utilizadores. Os restantes 8 utilizadores, registaram-se menos de 30 inserções, provavelmente devido a terem instalado a aplicação numa fase tardia.

Analisando os dados recolhidos, verificou-se que as atividades mais utilizadas foram as atividades dos Próximos Autocarros, dos Horários e do Mapa. Para aceder a estas atividades utilizaram o menu na maior parte das vezes, visto que a utilização do Menu ocupa 26% e através da outra opção disponível, o Submenu, registou-se apenas 4%. As restantes actividades, Alarmes, Itinerários e Itinerários Mapa registaram 4%, 7% e 5% respetivamente.

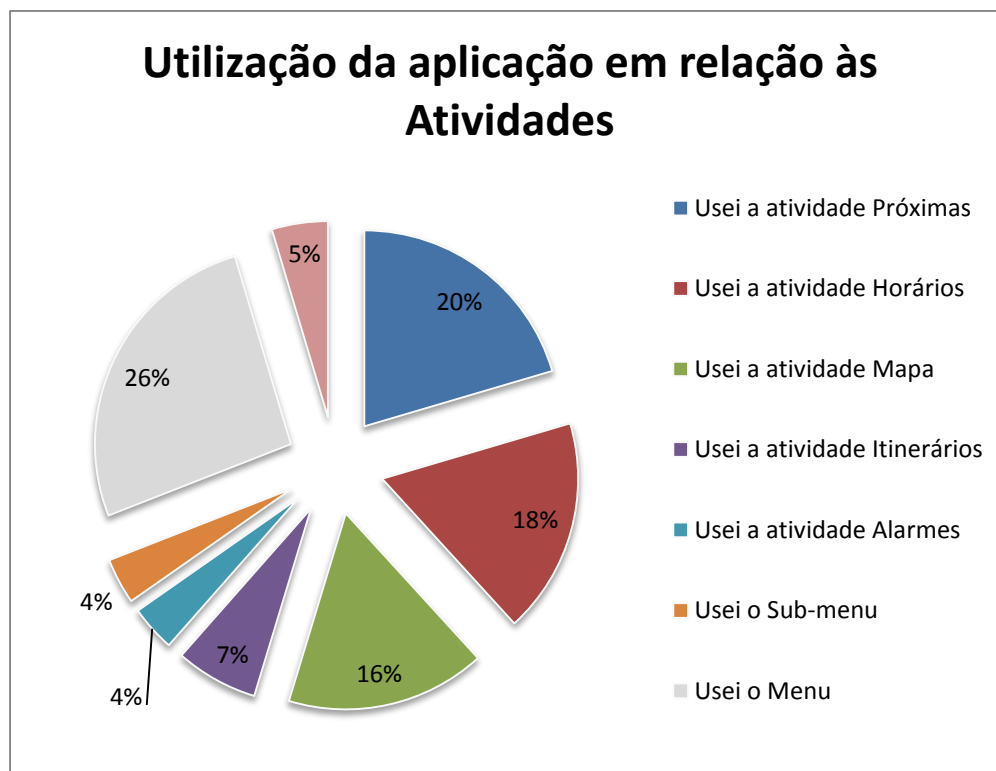


Figura 7.5 – Distribuição da utilização das atividades (segundo estudo)

Estes resultados dão suporte aos resultados do primeiro estudo, pois as atividades mais utilizadas foram as mesmas e as menos utilizadas também. Ao analisarem-se os resultados quanto às funcionalidades reparou-se que as duas funcionalidades com mais destaque foram a Procurar Carreira e a Ver Horário com, 25% e 31% respectivamente. Logo de seguida tem-se as funcionalidades de Ver Rota com 11% e Ver paragem X com 10%. Pesquisar por itinerários obteve 7% e as funcionalidades de Ver Todas as Paragens no mapa e ver os Horários em Tempo Real, obtiveram ambas 5%. As restantes funcionalidades obtiveram 2% ou menos.

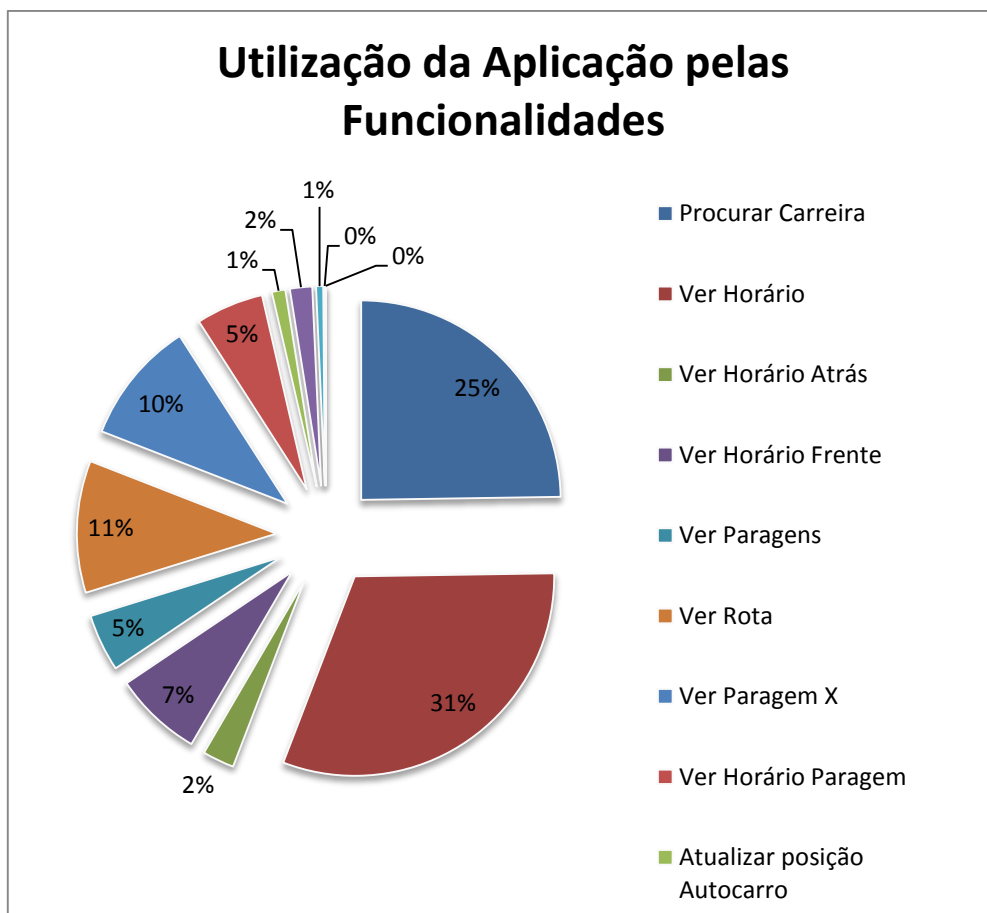


Figura 7.6 – Distribuição da utilização pelas funcionalidades (segundo estudo)

Mais uma vez os resultados não se diferenciaram muito dos do primeiro estudo, levando a concluir que para algumas funcionalidades pode ser necessário realizar melhorias, como por exemplo nos agendamentos em que não se verificou muita utilização. Outra possível conclusão que se pode tirar destas informações é que os utilizadores não têm por hábito planear as suas viagens de autocarro.

Em relação às outras funcionalidades, pode-se concluir que estas são essenciais e tiveram uma boa adesão durante este período de testes.

8. Conclusões

“There is a constant need for new systems and new software.”

(Marc Andreessen).

A realização deste projeto de dissertação representou um grande desafio que requereu vários conhecimentos adquiridos ao longo do curso e muitos outros novos que se teve de adquirir. A colaboração do pessoal dos Horários do Funchal e a orientação oferecida foram essenciais para garantir a conclusão deste projeto, e o *feedback* dos participantes na fase de testes e dos utilizadores no ensaio geral foram fundamentais para garantir a qualidade do *software*.

Ao desenvolver este projeto obteve-se um conhecimento maior sobre o sistema de gestão dos Horários do Funchal e sobre a sua arquitetura informática. No sistema de gestão pôde-se observar, por exemplo, como é realizado o planeamento dos horários de cada Carreira e em termos de arquitetura informática pôde-se observar como os Autocarros transmitem as informações para a central e como esta faz a gestão destas informações.

Durante o processo de *brainstorming*, pensou-se em várias ideias que pudessem melhorar a qualidade das viagens dos utentes dos Horários do Funchal. Destas ideias, a maioria foi implementada com sucesso, mas infelizmente algumas funcionalidades, como ter uma conta de utilizador associada aos cartões de viagem ou ao passe, de modo a que o utilizador soubesse o número de viagens restantes ou a data a que expirava o passe, não foram implementadas devido ao sistema atual dos Horários do Funchal não estar preparado para tal.

As fases de testes foram essenciais para descobrir e resolver erros na aplicação. Desde os primeiros testes realizados com os *mockups*, de modo a detetar erros na interface gráfica da aplicação, até aos testes realizados com os protótipos, onde se pôde detetar erros e falhas no código e conferir os resultados em relação à interface gráfica realizados nos *mockups*.

Durante a implementação pôde-se ainda obter um conhecimento mais profundo de como o *SDK* da Android funciona, e sobre o impacto que as tecnologias móveis têm na forma como se programa e como se utiliza uma aplicação. Principalmente, ao

analisarem-se os *layouts* utilizados, estes, diferenciam-se muito de um dispositivo *smartphone* para um *tablet* e ainda mais se se comparar com os *layouts* de uma aplicação para PC. Além do SDK da Android teve-se de aprofundar conhecimentos em outros SDK e API, de modo a obter um projeto mais rico, como por exemplo o SDK do Google *Maps* para Android que tornou possível a inserção de mapas e marcadores na aplicação e a API do Google *Places* ou do Google *Calendar* que trouxeram mais funcionalidades para a aplicação.

Comparando esta versão mobile com a versão *web* da aplicação móvel dos Horários do Funchal, pode-se constatar que apesar de as duas aplicações terem algumas funcionalidades em comum, estas funcionam de forma completamente diferente. A aplicação desta dissertação, como qualquer aplicação nativa, encontra-se otimizada para a plataforma Android. Por exemplo, se se comparar o acesso aos mapas, percebe-se que os movimentos são mais fluidos do que na API da Sapo utilizada na *web app* móvel dos Horários do Funchal. Esta e outras funcionalidades justificam a criação da aplicação nativa quando a quantidade de utilizadores da plataforma destino é significativa.

O lançamento da aplicação para o mercado global mostrou os resultados que se esperavam. A aplicação foi descarregada do mercado por diversos utilizadores com diferentes dispositivos Android e, além da taxa de desinstalações ser baixa, o *feedback* dos utilizadores ajudou a melhorar alguns aspetos da aplicação e a corrigir alguns erros que se desconheciam até o momento.

Esta aplicação pode ainda ser melhorada a nível de funcionalidades, e por isso mantem-se o diálogo com a empresa Horários do Funchal, para eventuais projetos futuros.

8.1. Trabalhos Futuros

Algumas das funcionalidades propostas para esta aplicação não foram implemenadas, podendo-se no futuro, com o interesse da empresa Horários do Funchal, completar estas funcionalidades.

A nível de funcionalidades pode-se referir que pelo menos a criação de uma conta de cliente, para o utilizador ter acesso a algumas informações do seu passe ou bilhete é

importante. Também a nível da funcionalidade de pesquisa por itinerários pode-se melhorar, se se introduzir a capacidade desta procurar itinerários por meio de mais do que uma Carreira.

Alguns utilizadores ainda sugeriram a criação de uma lista com as paragens, de modo a terem um acesso mais rápido, em vez de acederem sempre ao mapa, visto que este modo requer mais tempo.

Outra proposta que foi feita à empresa Horários do Funchal foi a de substituir os quiosques atuais, constituídos por PC e *touchscreen* caros, por equipamentos de baixo consumo e mais económicos, com o sistema Android. Se tal for aceite, pode-se utilizar uma versão desta aplicação, adaptada para vários utilizadores, onde estes possam pesquisar e obter várias informações dos Horários do Funchal.

Referências

- [1] A. T. Murray, R. Davis, R. J. Stimson, and L. Ferreira, "Public Transportation Access," *Transportation Research Part D: Transport and Environment*, vol. 3, no. 5, pp. 319–328, Sep. 1998.
- [2] T. Heimonen, "Information Needs and Practices of Active Mobile Internet Users." p. 8, 2012.
- [3] "Google Play - Tiramissu." [Online]. Available: <https://play.google.com/store/apps/details?id=edu.cmu.dv1.androidprod>. [Accessed: 25-Jun-2013].
- [4] "Google Play - OneBusAway." [Online]. Available: <https://play.google.com/store/apps/details?id=com.joulespersecond.seattlebusbot>. [Accessed: 25-Jun-2013].
- [5] "Agile Methodology." [Online]. Available: <http://agilemethodology.org/>. [Accessed: 27-Jun-2013].
- [6] J. Zimmerman, A. Tomasic, C. Garrod, D. Yoo, C. Hiruncharoenvate, R. Aziz, N. R. Thiruvengadam, Y. Huang, and A. Steinfeld, "Field Trial of Tiramisu : Crowd-Sourcing Bus Arrival Times to Spur Co-Design," in *Proceedings of the 2011 annual conference on Human factors in computing systems*, 2011, pp. 1677–1686.
- [7] B. Ferris, K. Watkins, and a. Borning, "Location-Aware Tools for Improving Public Transit Usability," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 13–19, Jan. 2010.
- [8] M. Bertolotto, G. O. Hare, R. Strahan, A. Brophy, A. Martin, and E. Mcloughlin, "Bus Catcher : a Context Sensitive Prototype System for Public Transportation Users." pp. 1–9, 2006.
- [9] M. Weiser, "Mark Weiser (1952–1999)," 1999.
- [10] "mobiThinking." [Online]. Available: <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#smartphoneos>. [Accessed: 16-Jul-2013].
- [11] "Android Developer." [Online]. Available: <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>. [Accessed: 17-Jul-2013].
- [12] "Android 1.5 Cupcake." [Online]. Available: <http://developer.android.com/about/versions/android-1.5.html>. [Accessed: 17-Jul-2013].

-
- [13] “Android Dashboards.” [Online]. Available: <http://developer.android.com/about/dashboards/index.html>. [Accessed: 17-Jul-2013].
- [14] A. Charland and B. LeRoux, “Mobile Application Development: Web vs. Native,” *Queue*, vol. 9, no. 4, p. 20, Apr. 2011.
- [15] “Download Eclipse and Android.” [Online]. Available: <http://developer.android.com/sdk/index.html>. [Accessed: 22-Jul-2013].
- [16] “Android SDK Manager.” [Online]. Available: <http://developer.android.com/sdk/installing/adding-packages.html>. [Accessed: 22-Jul-2013].
- [17] “Plugin ADT.” [Online]. Available: <http://developer.android.com/sdk/installing/bundle.html>. [Accessed: 22-Jul-2013].
- [18] “Google Calendar API.” [Online]. Available: <https://developers.google.com/google-apps/calendar/v3/reference/>. [Accessed: 22-Jul-2013].
- [19] “Google Maps Android SDK v2.” [Online]. Available: <https://developers.google.com/maps/documentation/android/start?hl=pt-PT>. [Accessed: 25-Jul-2013].
- [20] “Google Places API.” [Online]. Available: <https://developers.google.com/places/documentation/search>. [Accessed: 25-Jul-2013].
- [21] “Uses Case Guidelines.” [Online]. Available: <http://msdn.microsoft.com/en-us/library/vstudio/dd409432.aspx>. [Accessed: 26-Jul-2013].
- [22] “Requisitos.” [Online]. Available: <http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1718>. [Accessed: 26-Jul-2013].
- [23] L. Constantine, “Peer Reviews for Usability users interface,” vol. 18, no. 1, pp. 5–14.
- [24] “Android Principles.” [Online]. Available: <http://developer.android.com/design/get-started/principles.html>. [Accessed: 12-Jun-2013].
- [25] “Balsamiq Mockups.” [Online]. Available: <http://balsamiq.com/products/mockups/>. [Accessed: 26-Jul-2013].
- [26] G. Perlman’s, “Questionnaire for User Interface Satisfaction.” [Online]. Available: [http://hcibib.org/perlman/question.cgi?system=My HomePage](http://hcibib.org/perlman/question.cgi?system=My+HomePage)

-
- Maker&form=QUIS&email=marketing@myhomepage.com. [Accessed: 26-Jul-2013].
- [27] R. Hanmer, *Pattern-Oriented Software Architecture For Dummies*, 1d. ed. For Dummies 2013
- [28] B. C. Ed, M. Forss, I. Hagen, K. Hansson, J. Jonasson, M. Jonasson, F. Lott, and S. Olsson, *GoF Design Patterns - with examples using Java and UML2*. 2008.
- [29] "Android Activity." [Online]. Available: <http://developer.android.com/reference/android/app/Activity.html>. [Accessed: 30-Jul-2013].
- [30] "Android Multiscreen." [Online]. Available: <http://developer.android.com/training/multiscreen/screendensities.html>. [Accessed: 24-Set-2013].

Anexo A – Inquérito de avaliação do teste de usabilidade.

O inquérito que se segue foi utilizado para avaliar a opinião dos utilizadores ao testarem a aplicação. Foi criado com base no modelo *online* de inquérito para avaliação de testes de usabilidade de Gary Perlman [26].

HiQPdf Evaluation 07/26/2013

Inquérito de Usabilidade HF Transit - T2 (Protótipo Android App V1)

Este inquerito serve para registar e quantificar a satisfacao do utilizador quando usando a aplicacao HF Transit
* Required

Dados Pessoais

Questões pessoais de forma a traçar o seu perfil de utilizador

Qual é o seu nome? *

Qual é a sua idade? *

Qual ou quais as Carreiras em que se desloca? *
Escrever o número(s)!

Em média quantas viagens realiza semanalmente? *
(1 viagem = 1 deslocação num sentido)

Menos de 2 viagens

De 2 a 6 viagens

De 6 a 10 viagens

De 10 a 14 viagens

Mais de 14 viagens

Figura. A.1 – Inquérito de avaliação do teste de usabilidade (1/7)

Experiência de utilização

Questões sobre a experiência de ter utilizado a aplicação HF Transit

Opiniões sobre a usabilidade da Aplicação.
Erros, defeitos e outros problemas que tenhas encontrado!

Usas-te esta aplicação no teu dia-a-dia? *

Sim

Não

Qual a funcionalidade que achas que mais utilizas-te na aplicação HF Transit? *

Poder saber a hora do próximo autocarro de todas as carreiras

Poder agendar uma notificação para te lembrar de apanhar um autocarro

Ser notificado quando o autocarro sai da primeira paragem

Poder ver informações em Tempo Real como os horários, a tua localização e dos autocarros.

Poder pesquisar no mapa por paragens rotas e outras informações

Poder pesquisar por itinerários

Other:

Qual é a versão mais adequada para ti? *

Versão para SmartPhone

Versão para Tablet

O que recomendarias para adicionar a esta aplicação?

O que recomendarias para retirar desta aplicação?

Poder saber a hora do próximo autocarro de todas as carreiras

Poder agendar uma notificação para te lembrar de apanhar um autocarro

Ser notificado quando o autocarro sai da primeira paragem

Poder ver informações em Tempo Real como os horários, a tua localização e dos autocarros.

Poder pesquisar no mapa por paragens rotas e outras informações

Poder pesquisar por itinerários

Other:

Figura. A.2 – Inquérito de avaliação do teste de usabilidade (2/7)

Utilidade da aplicação											
	NA	1	2	3	4	5	6	7	8	9	10
Adicionar favoritos (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visualizar os Horários (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Próximas carreiras (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visualizar a rota no mapa (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visualizar o autocarro no mapa (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visualizar o horário na paragem (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pesquisar por lugares (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Adicionar alarme (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pesquisar por itinerários (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Serviço de avisar quando o autocarro sai da primeira paragem (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Guardar lugares (1 = Pouco 10 = Muito)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura. A.3 – Inquérito de avaliação do teste de usabilidade (3/7)

Achas que esta aplicação implementada num quiosque teria alguma utilidade? *

Sim
 Sim mas não para mim
 Não sei
 Não

Satisfação de Interface de Utilizador

Questões sobre a aplicação e a sua interface. (Baseado em <http://hcibib.org/perlman/question.cgi>)

Reação geral à aplicação

	NA	1	2	3	4	5	6	7	8	9	10
1 = Terrível 10 = Excelente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 = Difícil 10 = Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 = Frustrante 10 = Satisfatória	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 = Inadequada 10 = Adquada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 = Aborrecida 10 = Estimulante	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1 = Rígida 10 = Flexível	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura. A.4 – Inquérito de avaliação do teste de usabilidade (4/7)

Layout de visualização

	NA	1	2	3	4	5	6	7	8	9	10
Leitura dos caracteres (1 = Difícil 10 = Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Animações nas tarefas (1 = Pioram 10 = Melhoram)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Negritos e cores nos Layouts (1 = Pioram 10 = Melhoram)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organização da Informação (1 = Confusa 10 = Clara e Objectiva)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Disposição dos Layouts (1 = Confusa 10 = Clara)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura. A.5 – Inquérito de avaliação do teste de usabilidade (5/7)

Terminologia e sistema de informação											
	NA	1	2	3	4	5	6	7	8	9	10
Termos utilizados no sistema (1 = Inconsistente 10 = Consistente)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Terminologia relacionada com a tarefa (1 = Nunca 10 = Sempre)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Posição das mensagens no ecrã (1 = Inconsistente 10 = Consistente)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Widgets utilizados para introdução de dados (1 = Confusa 10 = Clara e Objectiva)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Informação sobre o progresso (1 = Nunca 10 = Sempre)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mensagens de Erro (1 = Pioram 10 = Ajudam)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura. A.6 – Inquérito de avaliação do teste de usabilidade (6/7)

Aprendizagem											
	NA	1	2	3	4	5	6	7	8	9	10
Aprender a operar a aplicação (1 = Difícil 10 = Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Explorando as funcionalidades (1 = Difícil 10 = Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lembrar nomes e comandos (1 = Difícil 10 = Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mensagens de ajuda (1 = Pioram 10 = Ajudam)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Capacidades do sistema											
	NA	1	2	3	4	5	6	7	8	9	10
Velocidade da aplicação (1 = Lenta 10 = Rápida)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fiabilidade da aplicação (1 = Péssima 10 = Fiável)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Robustez da aplicação (1 = Péssima 10 = Robusta)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Formas análogas de executar a mesma ação (1 = Poucas 10 = Muitas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Desenhado para todos os tipos de utilizador (1= Nunca 10 = Sempre)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura. A.7 – Inquérito de avaliação do teste de usabilidade (7/7)

Anexo B – Primeiro cenário de usabilidade

Abaixo pode-se observar o *template* do cenário de usabilidade usado na primeira fase de testes com o protótipo de baixo nível (*mockups*).

Teste de usabilidade app +Transit

Nome: _____

Cenário 1: Imagina que és o Tiago Neves e estás no Funchal, numa sexta-feira e queres ir para a tua casa que se localiza no Trapiche. Neste momento marca no teu relógio 10H52.

1. Queres agendar um alarme para apanhar o próximo autocarro N11.
2. Agora que marcaste a tua viagem, queres saber se tens viagens suficientes para subir.
3. Agora reparaste que não tens o teu cartão de viagem contigo e foste comprar um ao quiosque e queres adicionar ao sistema.
4. Agora reparaste que o teu número de telemóvel que está no sistema é o antigo e queres actualizar.

Cenário 2: Enquanto esperas pelo autocarro decides ver se existe alternativas e procuras pelo roteiro da carreira N11.

1. Decides ver o percurso da carreira N11 no mapa.
2. Constatas que a carreira N10 passa também, mas não passa na paragem onde estás. Decides então adicionar a carreira N11 aos Favoritos.
3. Decides agora ver onde está a localização do autocarro, mas no esquemático.
4. Reparas que ainda falta um pouco e decides ver os horários para a carreira N11, mas para amanhã.
5. Como amanhã também vais descer para o funchal, decides marcar um alarme para descer às 11H35 e outro para subires às 16H35.

Cenário 3: A app avisa-te que o autocarro está a sair e tu decides que está na hora de ficar atento. Finalmente, tu apanhas o autocarro e chegas a casa são e salvo e bem satisfeito, porque tudo correu como planeado. Agora reparas que na app tens uma secção chamada lugares e sentes interesse ao ponto de ires investigar.

1. Ao abrires vêes que a categoria “Meus lugares” está vazia e decides adicionar a localização da tua casa.

2. Depois adicionas mais uns quantos e agora queres ver que outros lugares existem na categoria “Populares”.
3. Reparas que existe o “Jardim Botânico” e decides ver mais.
4. Como já tinhas ido de carro ao jardim botânico, agora vês os comentários e deixas o teu.
5. Agora também queres deixar a tua classificação sobre este lugar.
6. Ficas curioso para saber como chegar lá de autocarro e procuras como o fazer.

Cenário 4: Como tu gostas bastante do Jardim botânico e queres lá voltar, decides que desta vez vais poupar uns trocos na gasolina e vais apanhar o autocarro. Então agendas uma viagem para o dia seguinte. Depois de voltar da visita, queres ir ao shopping e não sabes como lá chegar de transportes públicos.

1. Procuras como chegar ao Shopping.
2. Depois de saber, queres ver o que tens de fazer e queres visualizar no Mapa.

Cenário 5: Enquanto esperas pelo autocarro, queres saber informações sobre o tempo e outras notícias relevantes.

1. Procuras por informações sobre o tempo.
2. Procuras por notícias.

Anexo C – Segundo cenário de usabilidade

Abaixo pode-se observar o *template* do cenário de usabilidade usado na segunda fase de testes com o protótipo de alto nível.

Teste de usabilidade app FXBus

Nome: _____

Cenário 1: Imagina que estás no Funchal e queres ir para a tua casa que se localiza no Trapiche.

1. Queres agendar um alarme para apanhar o próximo autocarro número 11 - Graça.
2. Agora vês que te enganaste e queres editar o último agendamento para o dia de amanhã.
3. Depois vês que afinal é melhor apanhar boleia e pretendes apagar o último agendamento.

Cenário 2: Enquanto esperas pelo autocarro, decides ver se existem alternativas e procuras pelo roteiro da carreira número 11 - Graça.

1. Decides ver o percurso da carreira número 11 - Graça no mapa.
2. Procuras no mapa alternativas à carreira número 11 - Graça e verificas que na primeira paragem desta carreira, podes apanhar a carreira número 10 que também passa na tua casa.
3. Decides agora procurar no mapa carreiras que passem nas tuas imediações.
4. Agora atualizas a posição do autocarro, a ver se mudou de paragem.
5. Também queres verificar quais os autocarros que passam no Madeira Shopping e fazes uma procura no mapa.
6. Agora queres ver o horário em tempo real da primeira paragem da carreira número 11 - Graça.

Cenário 3: Pretendes consultar os horários para hoje e para amanhã de várias carreiras.

1. Procuras pelo horário da carreira número 12 - Jamboto (hospital) para o dia de hoje.
2. Agora vês o horário da mesma carreira para o dia de amanhã.

-
3. Agora procuras pelo horário da carreira número 12 - Centro (via viveiros) para o dia de hoje.
 4. Agora agendas um alarme para a próxima hora, desta mesma carreira.

Cenário 4: Agora decides pesquisar os itinerários disponíveis para te deslocares do ponto A até ao ponto B.

1. Decides que o teu ponto A será a baixa do funchal ao perto das galarias São Lourenço.
2. Decides que o teu ponto B será a Universidade.
3. Decides guardar o teu ponto B, visto que pretendes utiliza-lo mais tarde.
4. Pesquisas por Itinerários disponíveis entre o teu ponto A e B.
5. Conferes o resultado no Mapa.

Anexo D – Implementação na versão para *Tablet*

A versão para *Tablet*, como foi referido anteriormente, contém as mesmas funcionalidades que a versão para *smartphone*, simplesmente são apresentadas ao utilizador vistas mais completas em cada atividade.

Pode-se observar na Figura. D.1 o *Layout* apresentado ao utilizador para a actividade Próximas Carreiras.



Figura. D.1 – *Layout* da atividade Próximas Carreiras na versão para *Tablet*

Na Figura. D.2 pode-se observar o *Layout* apresentado ao utilizador para a atividade Alarmes.

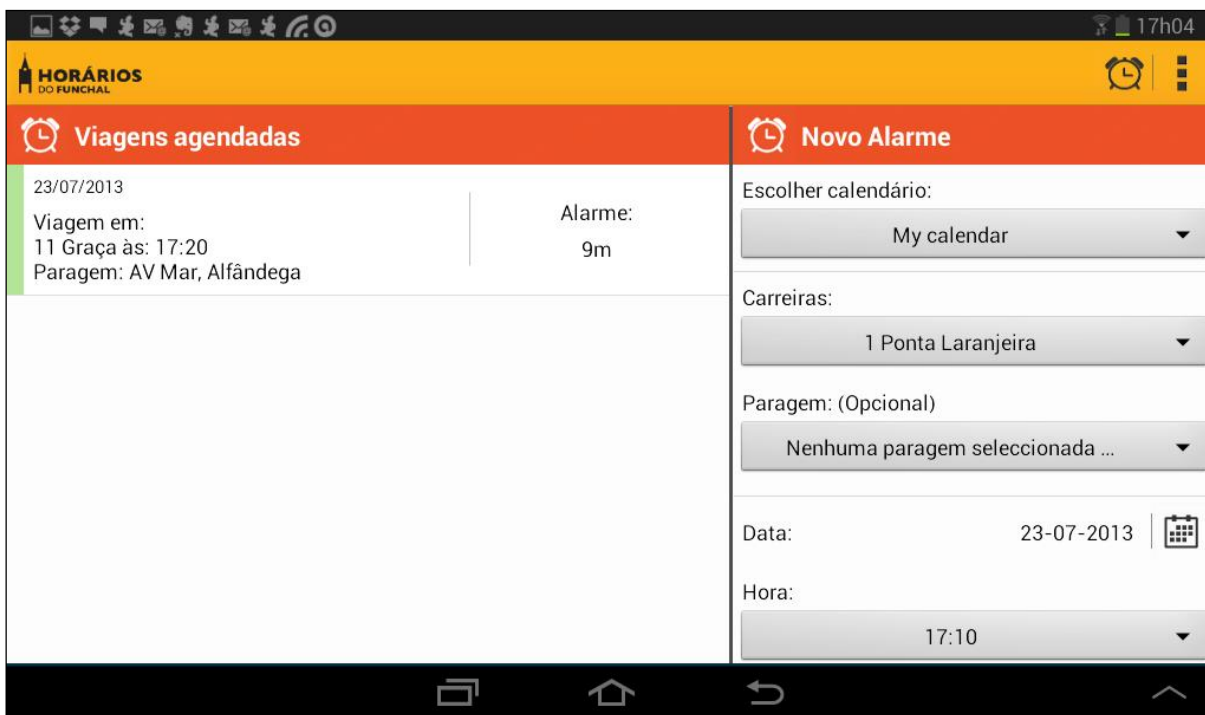


Figura. D.2 – Layout da atividade Alarmes na versão para Tablet

Pode-se observar na Figura. D.3 o Layout apresentado ao utilizador para a atividade Mapa.

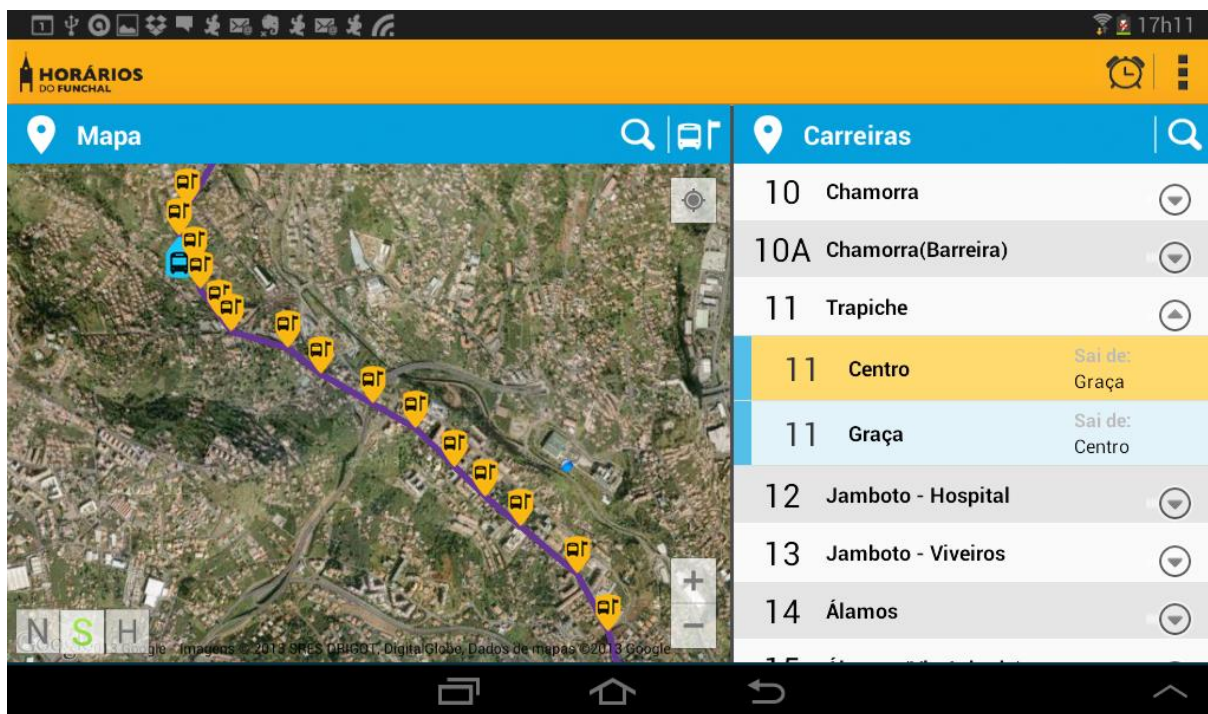


Figura. D.3 – Layout da atividade Mapa na versão para Tablet

Na Figura. D.4 pode-se observar o *Layout* apresentado ao utilizador para a atividade Horários.



Figura. D.4 – *Layout* da atividade Horários na versão para *Tablet*

Na Figura. D.5 pode-se observar o *Layout* apresentado ao utilizador para a atividade Itinerários (pesquisa).



Figura. D.5 – *Layout* da atividade Itinerários (pesquisa) na versão para *Tablet*

Na Figura. D.6 pode-se observar o *Layout* apresentado ao utilizador para a atividade Itinerários (resultados).

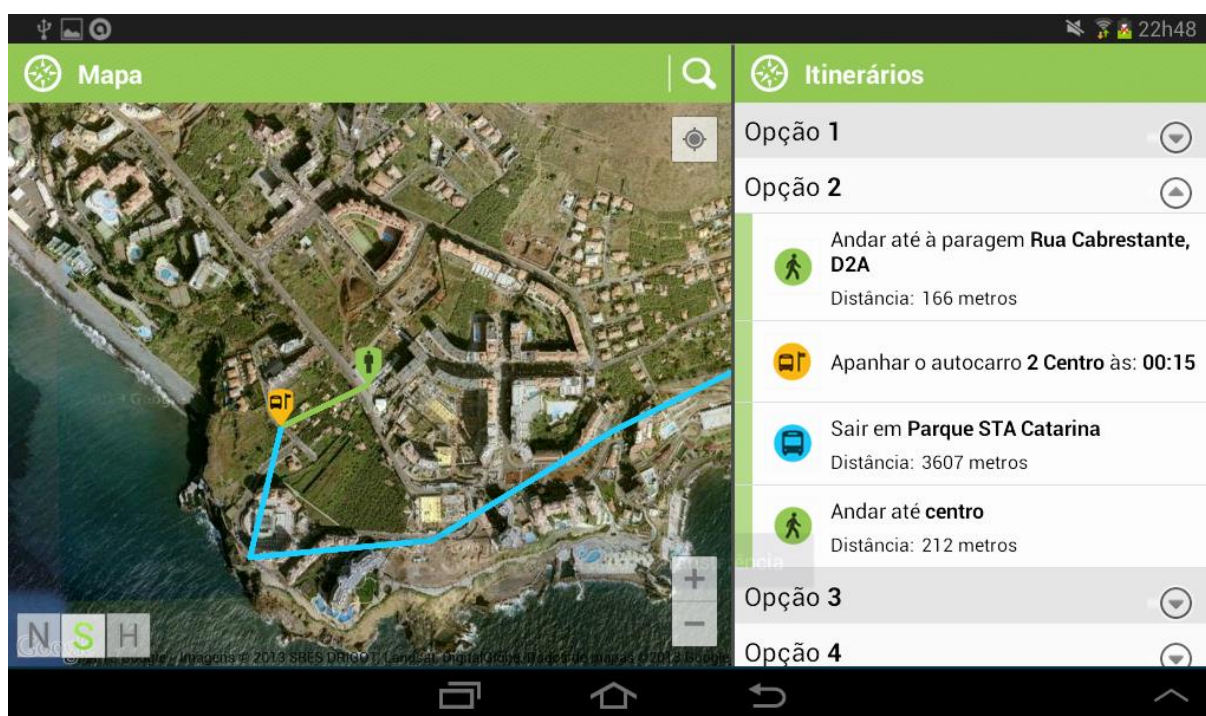


Figura. D.6 – *Layout* da atividade Itinerários (resultados) na versão para *Tablet*