

Final

Aplicação Móvel para Percurso Turístico

PROJETO DE MESTRADO

André Bruno Ferreira Freitas

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

setembro | 2016

E Apl
D-R(2ex.)

T/H
004
FRE Ahl
+ PD-R(2ex.)

Aplicação Móvel para Percorso Turístico

PROJETO DE MESTRADO

André Bruno Ferreira Freitas

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADORA
Karolina Baras

CO-ORIENTADORA
Raquel Brazão



APLICAÇÃO MÓVEL PARA PERCURSO TURÍSTICO

André Bruno Ferreira Freitas
Mestrado em Engenharia Informática

ORIENTADORA: KAROLINA BARAS
COORIENTADORA: RAQUEL BRAZÃO

FUNCHAL, NOVEMBRO 2016



Aplicação Móvel para Percurso Turístico

André Bruno Ferreira Freitas

Constituição do júri de provas públicas:

Eduardo Miguel Dias Marques, Prof. Auxiliar da Universidade da Madeira, Presidente

Diogo Nuno Crespo Ribeiro Cabral, Prof. Auxiliar Convidado da Universidade da Madeira, Vogal

Karolina Baras, Prof.^a Auxiliar da Universidade da Madeira, Vogal

Novembro 2016

Funchal – Portugal

Resumo

O planejamento de roteiros a múltiplos Pontos de Interesse (PDI) com restrições associadas pode ser algo complexo, existindo diversas variáveis com influência sobre a qualidade dos mesmos. A qualidade de um roteiro depende, objetivamente, de variáveis como a proximidade, tempo, custo financeiro e acessibilidade (no caso de pessoas portadoras de deficiência). Também depende, mais subjetivamente, das preferências pessoais de cada pessoa.

A solução a ser apresentada tem como principal objetivo a criação de roteiros que apresentem elevado nível de qualidade, sendo que a qualidade irá depender das variáveis anteriormente descritas. O desenvolvimento do restante documento irá fazer o enquadramento deste problema, dentro de estudos existentes na temática da modelação do comportamento do turista. Estes estudos têm como principal objetivo a identificação de guidelines de design para sistemas que sirvam como guia a roteiros turísticos. A temática de trabalho também irá ser contextualizada dentro de outros problemas de orientação já conhecidos. Outro aspeto a ser descrito será um comparativo entre diversas soluções existentes. Também irá descrita a arquitetura do sistema desenvolvido, que inclui os requisitos funcionais, cenários de qualidade, diversas vistas possíveis para a plataforma e análise das soluções tecnológicas de desenvolvimento existentes. Dado que a plataforma foi desenvolvida em iterações, ir-se-á apresentar o desenvolvimento e funcionalidades por cada iteração produzida. Por fim, irão ser identificados alguns dos testes efetuados sobre a plataforma e resultados associados, sendo descrito algumas conclusões obtidas a partir dos mesmos.

Palavras-chave: Problemas de orientação, Roteiros turísticos, aplicações móveis.

Abstract

Planning a tour to multiple Points of interest (POI) with multiple restrictions can be somehow complex, and there are several variables with influence on the tour quality. Objectively, the quality of a tour depends on variables such as proximity, time, financial cost, accessibility (for disabled people). Subjectively, it also depends on personal preferences.

The solution being presented aims to create tours that have high quality level, and the quality will depend on the variables described above. The development of the remaining document will make the framework of this problem, within existing studies on the topic of modeling tourist behavior. These studies have as main objective the identification of design guidelines for systems that serve as guide to tourist itineraries. The main topic of this work will be also be contextualized within other already known orientation problems. Another aspect to be described will be a comparison between several existing solutions. It will also describe the developed system architecture, which includes functional requirements, quality scenarios, different views for the platform, analyzing the existing technological development solutions. Since the platform was developed in iterations, the development and features produced by each iteration will be presented. Finally, some conclusions will be presented based on tests performed over the platform and associated results.

Keywords: Orientation problem, Sightseeing tours, Mobile applications.

Agradecimentos

Gostaria de agradecer à orientadora Professora Karolina Baras pela orientação e disponibilidade apresentadas durante o desenvolvimento de todo o projeto.

Quero também ao pessoal da Câmara Municipal do Funchal pela disponibilidade que apresentaram durante as reuniões relacionadas com este trabalho.

Por fim, queria agradecer à minha família pela disponibilidade para testar a aplicação e por todo o apoio e incentivo que me têm transmitido, não só durante o desenvolvimento do projeto, mas também durante toda a minha formação académica.

Índice

1. Introdução	1
1.1 Motivação.....	1
2. Estado da arte	5
2.1 Modelação do comportamento do Turista e aspetos de design associados	5
2.2 Estudo do problema de planeamento de roteiros turísticos e possíveis abordagens	6
2.3 Estudo das soluções disponíveis como guia turístico.....	7
3. Arquitetura de Software	17
3.1 Requisitos funcionais	17
3.2 Attribute-driven design	18
3.3 Estudo das soluções tecnológicas disponíveis.....	28
4. Desenvolvimento da aplicação.....	33
4.1 Primeira Iteração	34
4.2 Segunda Iteração	46
4.3 Terceira Iteração.....	64
5. Testes e Resultados	83
5.1 Questionários.....	83
5.1.1 Discussão dos resultados dos questionários	84
5.2 Comparação do desempenho entre as bases de dados Neo4j e MySQL	86
5.2.1 Discussão dos resultados dos testes às bases de dados.....	88
5.3 Limitações da Aplicação	88
6. Conclusões	91
6.1 Objetivos atingidos.....	92
6.2 Trabalho futuro.....	93
Referências	95
Anexos.....	99

Lista de Tabelas

Tabela 1 - Comparação entre aplicações.....	15
Tabela 2 - Cenários de qualidade ordenados por prioridade	19
Tabela 3 - Cenário de qualidade para o atributo disponibilidade	21
Tabela 4 - Cenário de qualidade para o atributo desempenho.....	22
Tabela 5 - Cenário de qualidade para o atributo desempenho.....	23
Tabela 6 - Cenário de qualidade para o atributo modificabilidade	24
Tabela 7 - Cenário de qualidade para o atributo Interoperabilidade	25
Tabela 8 - Vantagens e Desvantagens da arquitetura Rich client hybrid	27
Tabela 9 - Comparação entre elementos das BDs MySQL e Neo4j	65
Tabela 10 - Tempos de execução para a BD MySQL (segundos)	87
Tabela 11 - Tempos de execução para a BD Neo4j (segundos).....	87

Lista de Figuras

Figura 1 - Aplicação Sygic Travel	8
Figura 2 - Aplicação Tourias – App&Web Travel Guide	9
Figura 3 - Madeira Map and Guide	10
Figura 4 - Aplicação Madeira.Best Activities Tours.....	11
Figura 5 - Aplicação Foursquare.....	12
Figura 6 - Aplicação TripAdvisor	13
Figura 7 - Aplicação Fx Guide.....	14
Figura 8 - Diagrama de casos de utilização;.....	17
Figura 9 - Vista componente-conector do sistema	26
Figura 10 - Menu principal.....	38
Figura 11 – Vista de PDIs	39
Figura 12 - Vista PDI detalhes	39
Figura 13 - Vista PDI rota.....	40
Figura 14 - Vista roteiros personalizados.....	41
Figura 15 - Vista roteiros detalhes	42
Figura 16 - Vista roteiro rota.....	43
Figura 17 - Vista roteiro criar/editar	43
Figura 18 - Vista roteiros adicionar PDIs.....	44
Figura 19 - Vista roteiros adicionar roteiros como template.....	45
Figura 20 - Esquema da base de dados Neo4j simplificado.....	51
Figura 21 - Vista Principal	54
Figura 22 - Vista PDIs (Separador Todos).....	55
Figura 23 - Vista PDI detalhes	56
Figura 24 - Vista PDI rota.....	57
Figura 25 - Vista de roteiros.....	59
Figura 26 - Vista roteiros detalhes	61
Figura 27 - Vista roteiro rota.....	62
Figura 28 - Vista das definições.....	62
Figura 29 - Esquema do modelo relacional.....	65
Figura 30 - Diagrama de classes do servidor	66
Figura 31 - Vista de Registo.....	68
Figura 32 - Vista Principal	69
Figura 33 - Vista PDIs.....	70
Figura 34 - Separador Histórico dos PDIs.....	71
Figura 35 - Filtros para os PDIs	71

Figura 36 - Vista de detalhes do PDI	72
Figura 37 - Vista PDI rota	73
Figura 38 - Vista Roteiros	74
Figura 39 - Vista de detalhes do roteiro	75
Figura 40 - Vista de rota para os PDIs do roteiro.....	75
Figura 41 - Vista de criação de roteiro	76
Figura 42 - Vistas adicionar PDIs e roteiros	77
Figura 43 - Vista de criação automática de roteiros	78
Figura 44 - Vista das definições da aplicação	79
Figura 45 - Vista da agenda.....	80
Figura 46 - Notificações de evento agendado (sem internet e com internet)	80
Figura 47 - Notificação de evento de proximidade de PDI (sem internet e com internet)	81
Figura 48 - Protótipo de alta fidelidade para 1ª iteração	92
Figura 49 - Protótipo de alta fidelidade para 2ª iteração	92
Figura 50 - Protótipo de alta fidelidade para 3ª iteração	93
Figura 51 - Protótipo de baixa fidelidade 1ª iteração secção PDIs.....	99
Figura 52 - Protótipo de baixa fidelidade 1ª iteração secção Recommended Tours	100
Figura 53 - Protótipo de baixa fidelidade 1ª iteração secção Custom Tours.....	101
Figura 54 - Protótipo de baixa fidelidade 2ª iteração secção Registo/Login.....	102
Figura 55 - Protótipo de baixa fidelidade 2ª iteração secção PDIs.....	103
Figura 56 - Protótipo de baixa fidelidade 2ª iteração secção Roteiros	104
Figura 57 - Protótipo de baixa fidelidade 2ª iteração secção Agenda	105
Figura 58 - Protótipo de baixa fidelidade 2ª iteração secção Definições e Dados do Utilizador	106

Lista de Acrónimos

AES	Advanced Encryption Standard
API	Application programming interface
App	Aplicação
ATAM	Architecture Tradeoff Analysis Method
ATM	Automated teller machine
BD	Base de dados
CLI	Command-line interface
CMF	Câmara Municipal do Funchal
GIS	Geographic information system
GPS	Global Positioning System
GUI	Graphical user interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
MCTOPTW	Multi-constrained team orienteering problem with time windows
MVC	Model–view–controller
MySQL	My Structured Query Language
OP	Orienteering problem
OS	Operating system
PCTSP	Prize collecting TSP
PDI/POI	Ponto de interesse/Point of interest
PHP	PHP: Hypertext Preprocessor
PTP	Profitable tour problem
Rest	Representational state transfer
SDK	Software development kit
SSL	Secure Sockets Layer
TSP	Travelling Salesman Problem
TTDP	Tourist trip design problem
UI	User Interface
URL	Uniform Resource Locator
XAMPP	Cross-Platform, Apache, MariaDB, PHP and Perl
XML	Extensible Markup Language

1. Introdução

Tradicionalmente, a divulgação de pontos de interesse (PDIs) existentes baseava-se na distribuição de panfletos pelas agências de viagem ou hotéis, sendo este um método pouco eficiente a dar a conhecer todos os locais/roteiros de potencial interesse para os turistas, sendo que por vezes muitos locais ficariam por visitar dada a falta de divulgação de alguns PDIs e das restrições (tempo, custo, etc.) que os turistas apresentam.

Nos últimos anos, com a massificação dos *smartphones* e com as potencialidades associadas aos mesmos, o número de aplicações desenvolvidas têm tido um enorme crescimento. No caso do turismo, dada a falta de eficácia de divulgação de PDIs através dos métodos tradicionais, das limitações que os turistas apresentam e sendo este um setor que em muitos casos tem um grande impacto económico, a procura e oferta de aplicações associadas a esta categoria tem crescido imenso. Os principais objetivos destas aplicações consistem, na maioria das situações, em divulgar informação acerca dos PDIs e otimizar os recursos do turista de forma a que este possa usufruir do maior número de PDIs que possível.

1.1 Motivação

A Região Autónoma da Madeira, nomeadamente o concelho do Funchal é uma área com um elevado fluxo de turismo, sendo este o setor que gera a maior fonte de receitas para a economia regional [1]. Muitos destes turistas apresentam agendas limitadas temporalmente, que em alguns casos não têm duração superior a um dia, no caso dos cruzeiristas. Tendo em conta os aspetos anteriormente referidos, justifica-se a criação de um guia à medida das restrições apresentadas pelos utilizadores, sendo que esta dissertação irá apresentar uma solução plausível para as necessidades dos mesmos.

Atualmente, uma das soluções mais viáveis para a criação de guias turísticos à medida, passa pelo desenvolvimento de aplicações móveis para *smartphone*, dado que muitos destes dispositivos apresentam capacidade de processamento e memória relativamente elevados (para executar aplicações relativamente complexas), apresentam conectividade à internet (permitindo fazer uso de serviços externos) e integram recetores de GPS (facilitando a extração de informação acerca do contexto geográfico do

utilizador). Para além das vantagens anteriormente referidas este tipo de dispositivos apresentam-se amplamente massificados e são dispositivos pessoais que quase sempre acompanham os utilizadores, permitindo obter dados com elevada qualidade acerca do ambiente e contexto de um grande número de utilizadores.

1.2 Objetivos

Os principais objetivos do projeto consistem no desenvolvimento de uma aplicação para Android que permita visualizar informação acerca de PDIs existentes no concelho do Funchal e obter recomendações sobre os mesmos, informar acerca das rotas existentes para os PDIs, criação e sugestão de roteiros personalizados à medida do utilizador e agendamento de visitas aos PDIs e aos roteiros personalizados.

Outro dos objetivos pretendidos é o desenvolvimento de uma plataforma de serviços Web que forneçam suporte à aplicação, de modo a facilitar a atualização da informação e execução de processos e leitura de dados que exijam elevados recursos de *hardware* dos dispositivos móveis dos quais poderiam sobrecarregar os mesmos, em termos de processamento e memória.

1.3 Estrutura do documento

Este capítulo pretende apresentar o contexto do problema em que o projeto se insere, sendo apresentadas algumas justificações para a solução a ser apresentada. Também apresenta os principais objetivos pretendidos para o projeto.

O segundo capítulo exhibe um resumo acerca das conclusões obtidas a partir de alguns estudos que pretendem efetuar a modelação do comportamento do turista, de modo a obter um conjunto de guidelines a utilizar em sistemas que sirvam como guia a roteiros turísticos. Ainda neste capítulo irá ser apresentada uma breve síntese acerca dos tipos de problema de planeamento de roteiros turísticos possíveis e qual destes em que este projeto se insere. Também expõe um comparativo entre algumas soluções existentes que apresentam alguns objetivos semelhantes a esta solução.

O terceiro capítulo, esquematiza os diversos aspetos que compõem a arquitetura completa do sistema, entre os quais se destacam os requisitos funcionais do sistema, a utilização do método “*Attribute-driven design*” que identifica diversos atributos de qualidade, cenários de qualidade e a vista componente-conector que representa o sistema. Também irão ser apresentadas diversas opções de arquiteturas móveis disponíveis sobre as quais o sistema pode ser implementado. Por fim, irá ser feita uma análise das tecnologias disponíveis em cada uma das camadas da plataforma.

O quarto capítulo descreve o desenvolvimento das várias iterações de desenvolvimento da aplicação, descrevendo os vários protótipos desenvolvidos, os aspetos do *design* da UI introduzidos na aplicação, as ferramentas utilizadas, a explicação dos algoritmos utilizados, a arquitetura do sistema por cada iteração, testes efetuados e no final uma breve conclusão acerca de aspetos que foi necessário modificar na presente iteração.

O quinto capítulo descreve um conjunto de testes efetuados sobre a aplicação desenvolvida (questionários e testes às bases de dados), sendo descrito em detalhe os resultados obtidos e que conclusões se pode constatar a partir desses resultados. Ainda neste capítulo foram identificadas algumas das limitações presentes no sistema desenvolvido.

Por fim, no sexto capítulo irão ser apresentadas as conclusões obtidas do resultado final do projeto, bem como irão ser identificados aspetos que poderão ser desenvolvidos em futuras iterações.

2. Estado da arte

Neste capítulo inicialmente irão ser descritos alguns estudos cujo principal objetivo consiste no estudo do comportamento do turista para que sejam identificados quais as implicações em termos de design que devem ser tidos em conta durante o desenvolvimento de novas tecnologias orientadas para o turismo.

Posteriormente, irá também ser descrito o problema de planeamento de roteiros turísticos e possíveis abordagens ao mesmo, sendo também identificada qual a abordagem que melhor se adequa ao contexto deste projeto.

Por fim, irão ser identificadas as possíveis soluções existentes no mercado apresentando as vantagens e desvantagens de cada uma destas.

2.1 Modelação do comportamento do Turista e aspetos de design associados

Barry Brown e Matthew Chalmers [43] efetuaram um estudo etnográfico de forma analisar o comportamento dos turistas e conseguiram concluir diversos aspetos que devem ser tidos em conta no design de aplicações direcionadas para o turismo. O primeiro aspeto identificado é que deve existir um meio de partilha de informação acerca de um percurso quer seja dentro do mesmo grupo de turistas (comunicando a localização de cada elemento do grupo ou apenas apresentando a mesma rota para todos os elementos deste grupo), ou com outros turistas que não se conheçam (Partilha da experiência do roteiro através da criação de comentários associados ao mesmo). Outro aspeto assinalado é funcionalidade de guia eletrónico, ou seja, apresente dados acerca de um PDI que contenha uma descrição, localização e se possível um modo de comparar estes dados (embora seja difícil de implementar um comparador para ecrãs de pequenas dimensões). Também deveria existir suporte para um modo que permita o turista vaguear por uma área (indicar apenas se o turista se dirige na direção correta do PDI em vez gerar uma rota fixa para esse mesmo local), de modo a garantir a flexibilidade do roteiro. Outro aspeto identificado é a existência de roteiros flexíveis que apresentam um conjunto de PDIs que permitem ao utilizador ter um conhecimento de informações destes antes de efetuar a rota. Por fim, se possível devem apresentar um meio de comunicar a experiência obtida, de forma que possam posteriormente partilhar a sua experiência de visita.

Outro estudo efetuado por Tuomas Vaitinen e David McGookin [44] dentro do mesmo âmbito embora usando métodos diferentes, obteve algumas conclusões

semelhantes ao nível da partilha da partilha da informação. Outro aspeto de design identificado por estes foi que as rotas devem ser mantidas flexíveis para que o utilizador a qualquer momento possa ajustá-las mesmo durante o percurso, pois frequentemente os turistas mudam de ideias em relação aos locais que pretendem ir. Também foi apontado que deveria de existir a possibilidade de modificação de vistas que apresentam os PDIs consoante as necessidades atuais do turista. Ainda um último aspeto assinalado é a identificação à distância do destino pretendido mesmo que a visibilidade para o mesmo esteja bloqueada, de forma a tornar navegação mais interessante.

É de notar que aspetos como a flexibilidade das rotas e modificação das vistas foram aspetos tidos em conta durante o desenvolvimento da aplicação móvel. Já outros aspetos relacionados com a partilha de informação acerca dos PDIs visitados foram definidos como trabalho futuro a desenvolver na aplicação, devido às restrições temporais associadas ao mesmo.

2.2 Estudo do problema de planeamento de roteiros turísticos e possíveis abordagens

O estudo do problema de planeamento de roteiros turísticos (TTDP - *Tourist trip design problem*), tem como principal objetivo satisfazer de forma ótima os interesses do turista, tendo em conta diversos parâmetros e restrições, sendo que existem 3 formas principais de abordar este problema através do uso de diversos modelos de pesquisa, algoritmos e metodologias [2]:

1. PTP (*Profitable tour problem*) - Procura de um percurso que maximize os ganhos (satisfação do utilizador) subtraindo o custo de viagem. Esta abordagem é uma extensão do TSP (*Travelling Salesman Problem*);

2. PCTSP (*Prize collecting TSP*) - Procura encontrar um percurso que minimize o custo de viagem, tentando manter os ganhos não inferiores a um determinado valor (o ganho é uma restrição/imposição). Esta abordagem é uma extensão do TSP;

3. OP (*Orienteering problem*) - Procura de um percurso que maximize o total dos ganhos obtidos enquanto mantém o custo dentro de um determinado valor (o custo é uma restrição/imposição). Esta abordagem pretende responder mais ao problema TTDP do que ao problema TSPP (*TSP with profits*);

Das diversas abordagens anteriormente referidas, a que foi seguida durante o desenvolvimento do projeto foi a OP. Neste caso, o custo corresponde às restrições impostas pelo utilizador, sendo estas nomeadamente: a disponibilidade temporal, financeira, a distância do ponto base, a existência de acessibilidades para pessoas portadoras de deficiência. Das diversas variantes de OPs que existem a que mais se aproxima ao contexto deste projeto é a MCTOPTW (*Multi-constrained team orienteering problem with time windows*), onde cada nó (PDI) apresenta um conjunto de atributos (neste caso são o custo, distância, tempo de visita, etc.) cuja soma entre diversos nós (PDIs) não deve exceder um valor máximo, que é definido pelas restrições do utilizador.

Nesta situação, é apenas apresentada uma única janela temporal sendo esta o espaço de tempo que o turista delimita para a sua permanência na localidade do Funchal. A principal desvantagem desta abordagem é que não permite gerar rotas para períodos de tempo que incluam múltiplos dias. Existem outras abordagens que incluem múltiplas janelas temporais, mas como um dos principais focos da aplicação são os turistas com agendas bastante curtas (ex. cruzeiristas) e dada a elevada complexidade destas abordagens, optou-se por focar numa única janela temporal.

Os atributos selecionados associados ao ganho e custo de um roteiro que foram anteriormente identificados para serem utilizados na aplicação móvel a ser desenvolvida foram definidos em parte, durante as reuniões na CMF e também tendo em conta alguns parâmetros identificados nas referências bibliográficas [2].

2.3 Estudo das soluções disponíveis como guia turístico

Existem inúmeras aplicações que servem como guia turístico, sendo selecionados alguns exemplos que demonstram quais as funcionalidades tipicamente encontradas neste tipo de aplicações e que apresentam cobertura para locais/serviços existentes na Região Autónoma da Madeira.

2.3.1 Sygic Travel

O Sygic Travel [37] desenvolvido pela Sygic, é uma aplicação disponível para Android e iOS. Esta aplicação permite definir roteiros para múltiplos dias. A forma de criação dos roteiros consiste na escolha manual cada dos PDIs sendo posteriormente requisitado qual o dia a que pretende adicionar esse PDI sendo automaticamente criado um roteiro para esse dia. A navegação até aos PDIs é feita externamente através de outra

aplicação de GPS que exista instalada no dispositivo. A utilização de mapas *offline* só pode ser efetuada ao adquirir uma aplicação de GPS desenvolvida pela mesma empresa. Esta aplicação apresenta uma vertente mais comercial que faz pesquisas a sites externos à aplicação relacionados com a pesquisa de hotéis e transportes (Por exemplo: Booking.com, Sygic e Tripomatic), sendo o utilizador redirecionado para o website através do browser do dispositivo, após ter escolhido o serviço que pretende adquirir. Por fim, esta aplicação apresenta dados acerca das condições meteorológicas para os dias seguintes.

Existe outra aplicação, o Tripomatic, que foi comprada pela Sygic e cujas funcionalidades e serviços pagos são idênticos, sendo que esta apresenta-se fragmentada em inúmeras aplicações idênticas, cuja única diferença é o local a que se destinam.

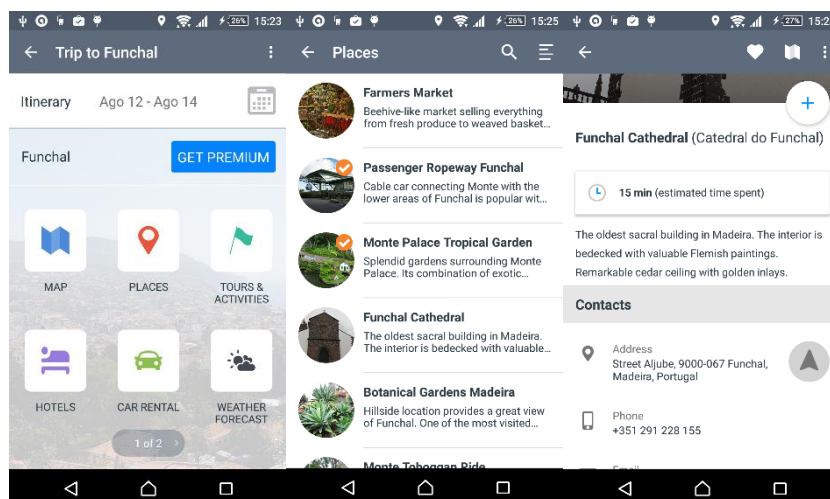


Figura 1 - Aplicação Sygic Travel

2.3.2 Tourias – App&Web Travel Guide

A Tourias – App&Web Travel Guide [38] desenvolvido pela Tourias, é uma aplicação que serve como guia turístico com informação detalhada acerca dos PDIs. Esta aplicação apresenta funcionalidade de indicação dos PDIs próximos do utilizador. Também apresenta informação acerca de roteiros guiados, que depois de selecionados, é efetuado o acesso a um serviço externo de modo a adquirir o respetivo roteiro guiado. Esta aplicação apresenta uma funcionalidade de realidade aumentada que permite usar a câmara do *smartphone* combinada com alguns sensores, como o de GPS e o giroscópio, para detetar a posição de um local que seja identificado pela aplicação. Também permite

fazer o uso dos mapas *offline* através do OpenStreetMap, embora a obtenção de rotas até aos referidos locais, seja feita por uma aplicação de GPS externa. Por fim, esta aplicação apresenta dados acerca das condições meteorológicas para os dias seguintes. Um aspeto negativo a apontar sobre esta aplicação é que apresenta publicidade na sua versão gratuita.

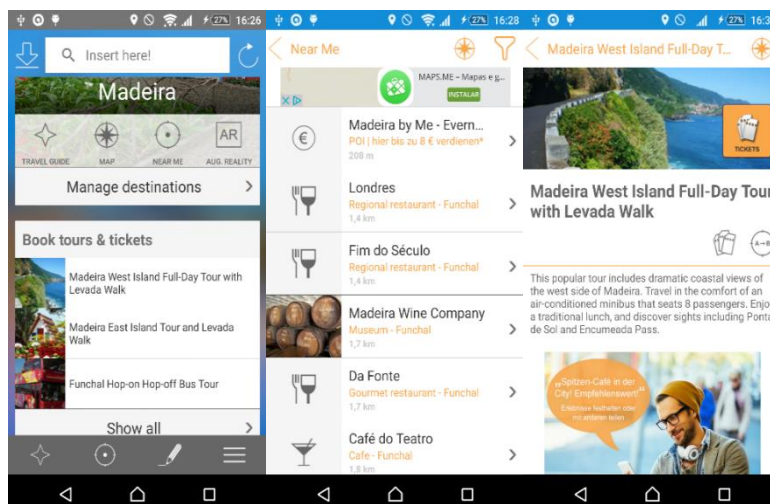


Figura 2 - Aplicação Tourias – App&Web Travel Guide

2.3.3 Madeira Map & Guide

O Madeira Map & Guide [39] é desenvolvido pela Free Travel & Tourist Guides. Esta aplicação apresenta dados acerca de excursões guiadas, eventos e passeios, localização de ATMs, farmácias, etc. Também apresenta um conversor de moeda, um pequeno dicionário e informações meteorológicas. Alguns aspetos negativos a apontar sobre esta aplicação é que apresenta uma excessiva vertente comercial (quase todos os conteúdos vão coincidir com compra de serviços), apresenta publicidade na sua versão gratuita, apresenta-se amplamente incompleta e muitas das opções seleccionadas acabam por redirecionar a uma ligação externa.

Apesar de apresentar mapas sem requerer ligação à internet (OpenStreetView), não apresenta um modo de navegação interno da aplicação, sendo o utilizador redirecionado para o site do Google Maps.

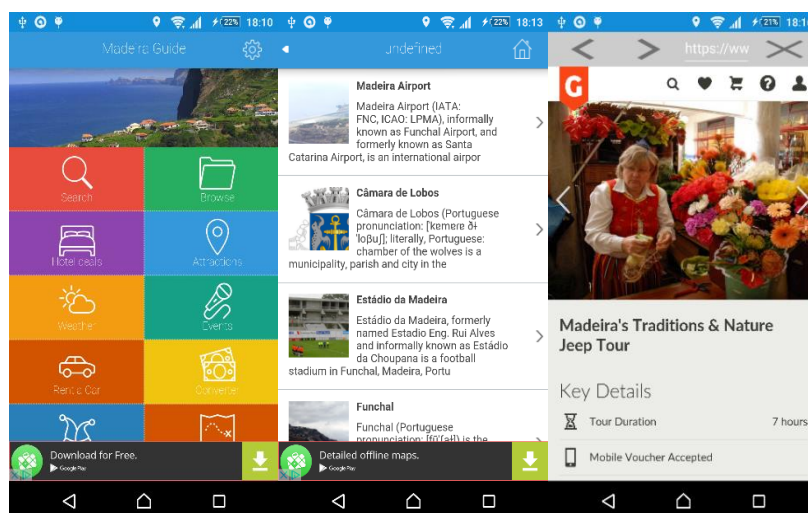


Figura 3 - Madeira Map and Guide

2.3.4 Madeira.Best Activities Tours

O Madeira.Best Activities Tours [40] é desenvolvido pela Wow!Systems Lda. Esta aplicação foca-se na divulgação e reserva de atividades de lazer, tais como: excursões guiadas, eventos, passeios e atividades desportivas e culturais. Alguns aspetos a apontar sobre esta aplicação é que apresenta uma vertente somente comercial (permite efetuar reservas), apresenta informação bastante completa e isenta de publicidade. O principal aspeto negativo é que apresenta uma navegação bastante confusa para a obtenção de rotas para as atividades/eventos. Ao visualizar os detalhes das atividades/eventos é apresentada informação escrita acerca da localização, mas esta, em muitos casos apresenta-se bastante vaga (Indica apenas o nome da área do evento em vez de uma morada ou coordenadas geográficas exatas). Nesta aplicação, também existe o redireccionamento para ligações externas, mas apenas no momento da reserva da atividade/evento.

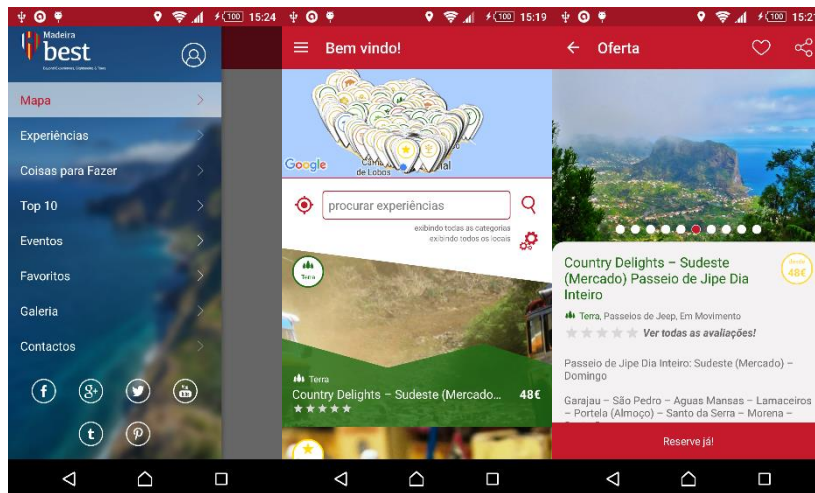


Figura 4 - Aplicação Madeira.Best Activities Tours

2.3.5 Foursquare

O Foursquare [41] é desenvolvido pela empresa com o mesmo nome. Esta aplicação apresenta-se como um guia/jogo/plataforma social para diversos locais, sendo que esta se apresenta disponível para a Madeira. Os seus conteúdos são gerados e confirmados pelos utilizadores tendo em conta o contexto geográfico e interesses dos mesmos. Esta plataforma apresenta uma enorme quantidade de PDIs e informação sobre os mesmos. A navegação até aos PDIs é feita externamente através de outra aplicação de GPS que exista instalada no dispositivo. Outra funcionalidade a identificar é relacionada com os horários sendo identificados horários populares que representam, na maior parte das situações, as melhores horas para usufruir dos PDIs. Esta aplicação apresenta-se isenta de publicidade. A principal desvantagem da aplicação está relacionada com a subjetividade da informação gerada pelos utilizadores (Ex. não existe um preço identificado, sendo apenas identificado se o local é caro ou barato).

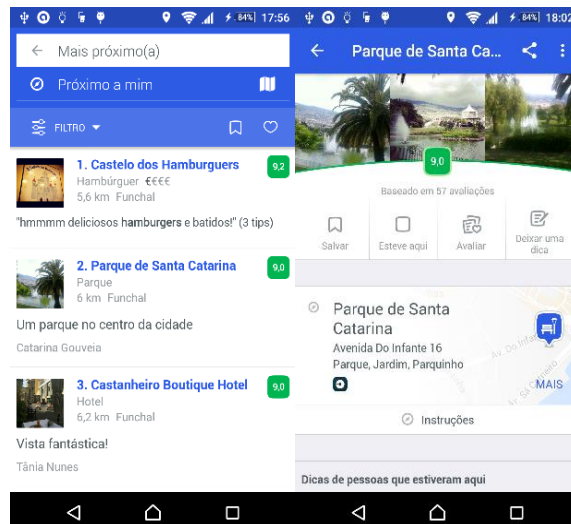


Figura 5 - Aplicação Foursquare

2.3.6 TripAdvisor

O TripAdvisor [41] é desenvolvido pela empresa com o mesmo nome. Esta apresenta-se no formato de website e aplicação. Esta plataforma foca-se na divulgação de PDIs direcionados para o turismo e divulgação das opiniões dos utilizadores em relação a estes mesmos locais. Esta apresenta uma forte vertente comercial na medida em que a maioria dos locais exibidos são estabelecimentos comerciais, alojamento, atividades ao ar livre, excursões, etc. Também é possível visualizar informação acerca de locais públicos de acesso livre (ex.: Jardins públicos, igrejas, etc.), embora o nível de detalhe da informação apresentada para estes seja bastante mais reduzido. Nesta aplicação, também existe o redirecionamento para ligações externas em algumas secções (ex.: Fórum, Reserva de serviços, etc.). Tal como outras aplicações deste género o TripAdvisor, também depende de aplicações externas para a criação rotas que permitam guiar o utilizador até ao local pretendido por GPS. Por fim, o TripAdvisor apresenta-se isento de publicidade.

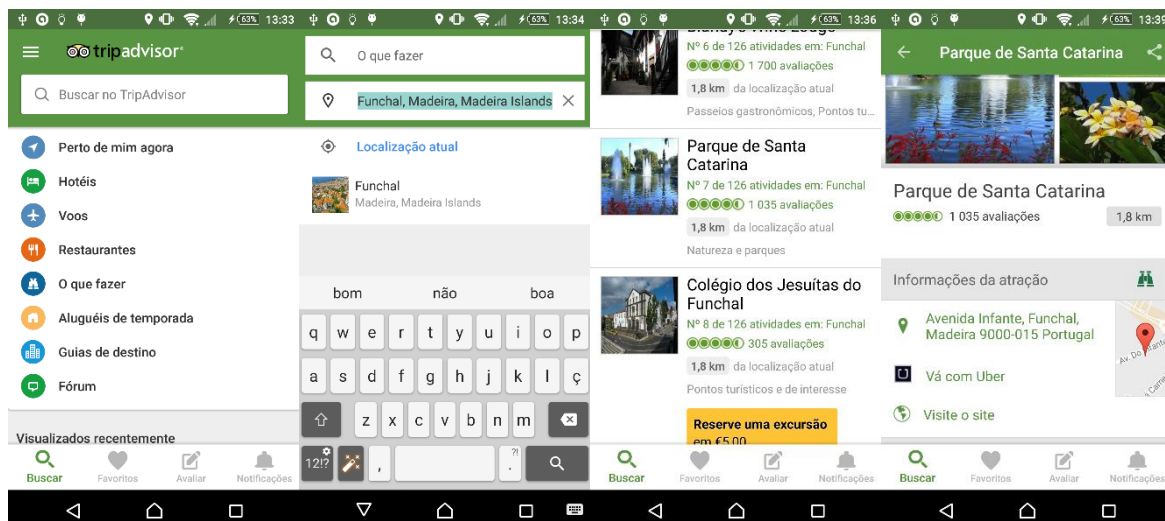


Figura 6 - Aplicação TripAdvisor

2.3.7 Fx Guide

A aplicação Fx Guide é a aplicação a ser desenvolvida no âmbito deste projeto. Esta aplicação tem como principal objetivo a criação simplificada de roteiros turísticos à medida das preferências e recursos dos turistas que visitem o concelho do Funchal, sendo para isto tido em conta fatores como: Tempo; Localização; Custo financeiro; Interesses pessoais; Acessibilidade para pessoas portadoras de deficiência. Esta apresenta um conjunto de PDIs de interesse turístico associados apenas ao conselho do Funchal. Outra funcionalidade a identificar são as recomendações tanto de PDIs como roteiros, sendo que estas têm em consideração aspetos como os interesses do turista, distância e custo de entrada. Também apresenta funcionalidades relacionadas com o agendamento de visitas, deteção de proximidade em relação aos PDIs, histórico de visitas e favoritos. Um aspeto relevante desta aplicação é a possibilidade de utilização sem ligação à internet. Por fim, a criação de rotas até aos locais pretendidos pode ser feita, ou através da respetiva aplicação, ou através de uma aplicação externa.

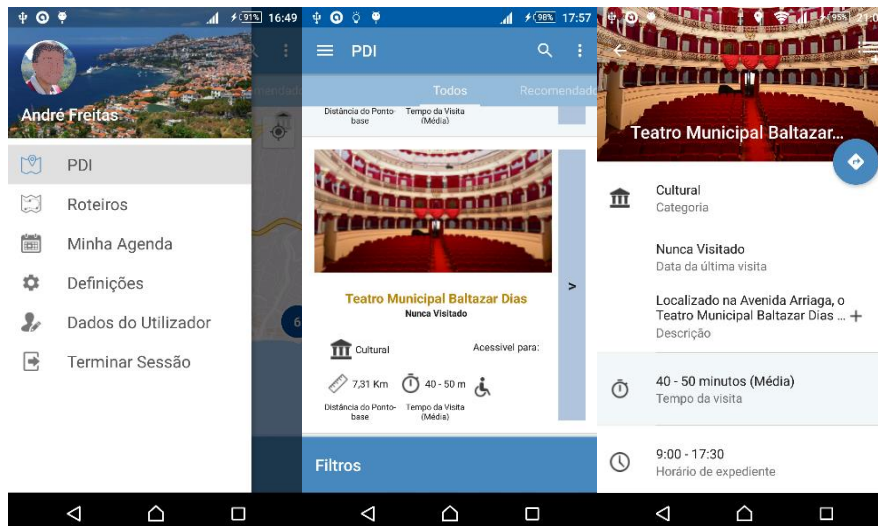


Figura 7 - Aplicação Fx Guide

2.3.8 Conclusão

O principal objetivo deste tipo de aplicações geralmente consiste em disponibilizar informação acerca de PDIs, eventos, atividades, etc. A maioria das aplicações do tipo guia turístico apresentam conteúdos com uma vertente comercial, que permitem efetuar reserva de serviços. Normalmente, a navegação até aos PDIs é feita externamente através de outra aplicação de GPS que exista instalada no dispositivo. Um aspeto negativo que algumas aplicações deste género partilham é o excesso de publicidade. Muitas aplicações apresentam roteiros estáticos, guiados e pagos. Em alguns casos, estas aplicações servem como suporte para empresas que vendem serviços, sendo apenas uma plataforma de publicidade dos respetivos serviços. Existem algumas aplicações que dependem, em parte, de conteúdo gerado pelos utilizadores, sendo este um método eficiente de obter informação quando apresentam um grande número de utilizadores.

Por fim, o que distingue principalmente a aplicação apresentada neste trabalho (Fx Guide) das restantes é a capacidade de gerar roteiros personalizados à medida dos interesses do turista.

Na seguinte tabela (*Tabela 1*) é efetuada a comparação entre as várias soluções mencionadas anteriormente.

	Sem publicidade	Utilização Offline	Vertente comercial	Informação gerada pelos utilizadores	Utilidades (ex. meteorologia, agenda, etc.)	Vertente social	Navegação através de aplicação externa
Sygyt Travel	✓	✗	✓	✗	✓	✗	✓
Tourias – App&Web Travel Guide	✗	✓	✓	✗	✓	✗	✓
Madeira Map & Guide	✗	✗	✓	✗	✓	✗	✗
Madeira.Best Activities Tours	✓	✗	✓	✗	✗	✗	✗
Foursquare	✓	✗	✗	✓	✗	✓	✓
TripAdvisor	✓	✗	✓	✗	✗	✗	✓
Ex Guide	✓	✓	✗	✗	✓	✗	✓

Tabela 1 - Comparação entre aplicações

3. Arquitetura de Software

Neste capítulo, irá ser definida a arquitetura do sistema que incorpora os requisitos funcionais, atributos e cenários de qualidade, as diferentes vistas possíveis para o sistema, a análise das arquiteturas de *software* possíveis para sistemas móveis e por fim, o estudo das soluções tecnológicas disponíveis para as diferentes camadas do sistema.

3.1 Requisitos funcionais

Durante esta secção irá ser representado um diagrama de casos de utilização (Figura 8) acerca das funcionalidades requeridas para o sistema. Posteriormente, irá ser justificada a origem das funcionalidades apresentadas.

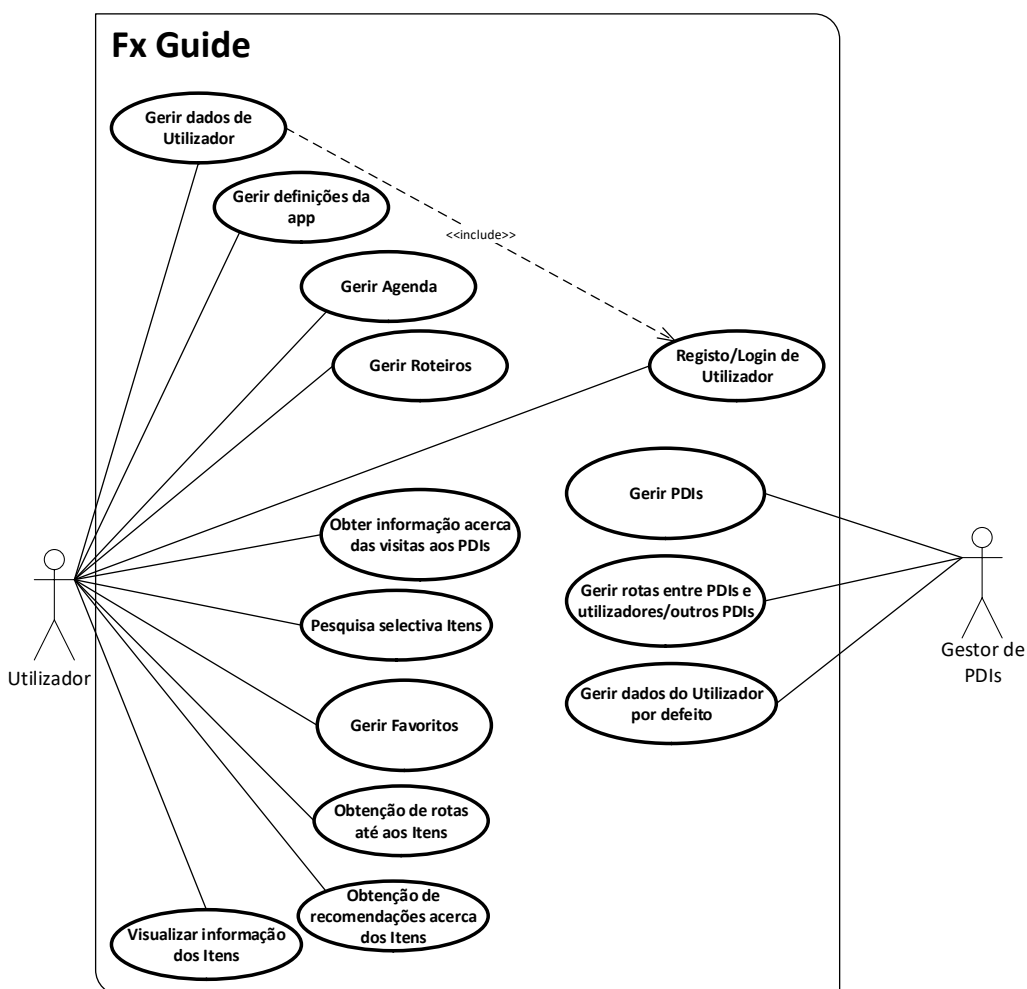


Figura 8 - Diagrama de casos de utilização;

Os casos de utilização identificados para o utilizador correspondem às funcionalidades desempenhadas pela aplicação móvel desenvolvida. Já os casos de utilização identificados para o gestor de PDIs correspondem a um programa desenvolvido cujo principal objetivo é preencher as bases de dados com informação acerca dos PDIs a partir de ficheiros KML e respetivas rotas entre PDIs. Este programa foi desenvolvido apenas com o intuito de ser provisório, enquanto não fossem cedidas outras fontes de informação detalhadas acerca dos PDIs (Bases de dados da CMF), sendo que esta informação acabou por nunca ser cedida, até à presente data.

Algumas funcionalidades apresentadas tiveram origem na informação obtida durante as reuniões na CMF. Outras funcionalidades foram extraídas da literatura associada a este tema [43, 44, 2]. Por fim, outras funcionalidades foram identificadas tendo em conta algumas das funcionalidades existentes nas aplicações apresentadas na *secção 2.3*.

3.2 Attribute-driven design

O *attribute-driven design* é um método para definir a arquitetura de software em que o processo de design baseia-se na identificação de atributos de qualidade sobre os quais o software deve ser implementado. O *attribute-driven design* usa um processo de decomposição recursiva dos elementos de um sistema, sendo que durante esta decomposição são definidas táticas e padrões arquiteturais de modo a garantir os cenários associados aos atributos de qualidade identificados [45].

Durante esta secção irá ser descrito o método *attribute-driven design* que permitiu definir a arquitetura de software do sistema desenvolvido, sendo para isto apresentados os cenários de qualidade obtidos, tendo em conta os atributos de qualidade definidos pelo grau de importância. Posteriormente irão ser representados os vários elementos que compõem o sistema e as relações entre eles. Um dos métodos que deveria ter sido utilizado para avaliar a arquitetura de *software* obtida é o *Architecture Tradeoff Analysis Method* (ATAM), mas dadas restrições temporais e de recursos humanos não se pode proceder à realização do mesmo. O ATAM tem como principal objetivo identificar que implicações as decisões arquiteturais definidas durante o *attribute-driven design* apresentam os sobre objetivos de negócio definidos pelos *stakeholders*, permitindo assim um método de negociação dos objetivos pretendidos para o projeto [46].

Idealmente, o ATAM não deve ser realizado pela mesma equipa que desenvolveu o sistema, pois pretende-se que a análise do sistema seja o mais imparcial que possível.

3.2.1 Atributos e cenários de qualidade

Os atributos de qualidade definidos como mais importantes são: a disponibilidade, o desempenho, a segurança, a modificabilidade e a interoperabilidade. A *Tabela 2* apresenta os atributos de qualidade ordenados por relevância com os respetivos cenários de qualidade.

Nível de prioridade	Atributo de qualidade	Cenário de qualidade
1	Disponibilidade	A aplicação deverá apresentar funcionalidades, independentemente do estado de ligação à internet.
2	Desempenho	A aplicação deverá apresentar-se disponível num período máximo de 30 segundos, em caso de falha na conectividade com o servidor.
3	Segurança	A aplicação deverá garantir confidencialidade na transmissão de dados.
4	Modificabilidade	A aplicação deverá permitir a alteração do tipo de base de dados sem necessitar de modificar lógica da aplicação.
5	Interoperabilidade	A aplicação deverá poder trocar mensagens com serviços externos (servidor e serviço de direções do Google Maps) com uma taxa de sucesso maior ou igual a 85%.

Tabela 2 - Cenários de qualidade ordenados por prioridade

A identificação dos atributos de qualidade com maior nível de prioridade (Disponibilidade, Desempenho), teve por base em algumas falhas identificadas no primeiro protótipo de alta fidelidade desenvolvido, durante a avaliação do mesmo.

O atributo modificabilidade teve origem em consequência de algumas das táticas utilizadas para tentar garantir a existência do atributo desempenho, nomeadamente identificar qual a base de dados que garante maior desempenho por parte do sistema.

O atributo de segurança foi introduzido dado este ser um aspeto fundamental, principalmente em situações em que se lida com dados sensíveis como a localização física do utilizador.

Por fim, o atributo de interoperabilidade, foi introduzido dada a necessidade utilização de serviços externos como por exemplo o serviço de direções do Google Maps.

Seguidamente, procede-se à especificação alargada dos cenários desenvolvidos para cada atributo de qualidade, sendo identificadas para cada um dos cenários possíveis táticas que garantam o respetivo atributo de qualidade.

3.2.1.1 Disponibilidade

Atributo de Qualidade	Disponibilidade
Cenário # 1	A aplicação deverá apresentar funcionalidades, independentemente do estado de ligação à internet.
Motivação	Garantir um meio de utilização da aplicação em ambientes que não permitam acesso à internet
Estimulo	Falha na conectividade com o servidor
Fonte do estímulo	Ocorrência de erro no servidor ou outros fatores externos que influenciem a qualidade de ligação.
Ambiente	Ambiente normal de funcionamento onde não há ligação com o servidor
Artefacto	Sistema do cliente (<i>smartphone</i>)
Resposta	Execução das funcionalidades (Leitura de dados em cache e criação de pedidos numa lista temporária que irá ser executada logo que a conectividade com o servidor seja restabelecida.)

Tabela 3 - Cenário de qualidade para o atributo disponibilidade

Táticas de disponibilidade

Redundância de dados - Manter um repositório local de dados em XML, sendo armazenados todos os dados acerca dos pontos de interesse e respetivas rotas até ao ponto base e vice-versa, sendo este repositório acedido caso não seja detetada qualquer ligação ao servidor. Posteriormente este repositório deve ser atualizado cada vez que o utilizador efetuar acesso aos dados disponibilizados pelo servidor externo.

Criação de listas de eventos - Manter localmente numa lista ordenada os eventos que o utilizador tenha ativado, mas que devido à falta de ligação à internet não seja possível ressincronizá-los com o servidor. Posteriormente quando a ligação ao servidor for reestabelecida, essa mesma lista de eventos deve ser executada de forma sequencial.

3.2.1.2 Desempenho

Atributo de Qualidade	Desempenho
Cenário #2	A aplicação deverá apresentar-se disponível num período máximo de 30 segundos, em caso de falha na conectividade com o servidor.
Motivação	Garantir a rapidez de resposta da aplicação mesmo em situações que não exista ligação à internet
Estimulo	Falha na conectividade com o servidor
Fonte do estímulo	Ocorrência de erro no servidor ou outros fatores externos que influenciem a qualidade de ligação.
Ambiente	Ambiente normal de funcionamento onde não há ligação com o servidor
Artefacto	Sistema do cliente (<i>smartphone</i>)
Resposta	Apresentar um meio de interação com o utilizador (Leitura de dados em cache, criação de pedidos numa lista temporária que irá ser executada logo que a conectividade com o servidor seja reestabelecida ou apresentação de mensagem de erro no caso das funcionalidades que não se apresentem disponíveis <i>offline</i> .)
Medida da resposta	No máximo 30 segundos

Tabela 4 - Cenário de qualidade para o atributo desempenho

Táticas de desempenho

Delimitar tempos de execução - A resposta aos pedidos deve apresentar um limite temporal, sendo neste caso de 30 segundos. Após este tempo deve ser considerado que houve uma falha na ligação.

Manter múltiplas cópias dos dados - Para além dos dados disponíveis no servidor manter dados localmente (cache) e manter os dados em memória em situações que tenha de ser efetuado o acesso aos mesmos em subsecções que venham posteriormente a ser seleccionadas.

Táticas de usabilidade

Heurísticas de Nielsen:

- “Utilizador controla e exerce livre-arbítrio” - Permitir a escolha de alternativas enquanto o sistema espera pela resposta do servidor e após a ocorrência de erro.
- “Ajudar utilizador a reconhecer, diagnosticar e recuperar dos erros” - Introdução de mensagens de erro adequadas e possíveis soluções.

3.2.1.3 Segurança

Atributo de Qualidade	Segurança
Cenário # 3	A aplicação deverá garantir confidencialidade na transmissão de dados.
Motivação	Garantir a privacidade dos dados de utilizador.
Estimulo	Pedido ao servidor.
Fonte do estímulo	Utilizador.
Ambiente	Ambiente normal de funcionamento.
Artefacto	Ligação entre cliente e servidor.
Resposta	Uso de mecanismos que permitam ocultar (encriptação/desencriptação) os dados transmitidos.
Medida da resposta	Ocultação dos dados transmitidos pela utilização de analisadores de pacotes.

Tabela 5 - Cenário de qualidade para o atributo desempenho

Táticas de segurança

Encriptação de dados - Os dados dos pedidos ao servidor devem ser encriptados, se possível, usando SSL através do protocolo HTTPS, protegendo de ataques do tipo *Man-in-the-middle*.

3.2.1.4 Modificabilidade

Atributo de Qualidade	Modificabilidade
Cenário # 4	A aplicação deverá permitir a alteração do tipo de base de dados sem necessitar de modificar lógica da aplicação.
Motivação	Ambiguidade acerca da fonte de dados definitiva a utilizar na aplicação
Estímulo	Necessidade de modificação da fonte de dados sobre a qual a plataforma faz utilização
Fonte do estímulo	Desenvolvedores
Ambiente	Ambiente normal de funcionamento.
Artefacto	Servidor
Resposta	Uso de mecanismos que permitam o acesso genérico a um conjunto de dados
Medida da resposta	Introdução uma nova base de dados sem alterações na lógica do servidor.

Tabela 6 - Cenário de qualidade para o atributo modificabilidade

Táticas de modificabilidade

Redução do Coupling - Abstrair módulos que façam acesso ao servidor através da introdução de um serviço comum que abstraia os métodos particulares de obtenção da base de dados da lógica de aplicação. Utilização do padrão *Bridge*.

3.2.1.5 Interoperabilidade

Atributo de Qualidade	Interoperabilidade
Cenário # 5	A aplicação deverá poder trocar mensagens com serviços externos (servidor e serviço de direções do Google Maps) com uma taxa de sucesso maior ou igual a 85%.
Motivação	Garantir elevada qualidade de utilização da aplicação.
Estimulo	Pedido de troca de dados
Fonte do estímulo	Execução de funcionalidade que tenha dependência de serviço externo
Ambiente	Ambiente normal de funcionamento onde há ligação com os servidores externos
Artefacto	Ligação entre cliente e servidor.
Resposta	Obtenção de respostas no formato apropriado que são utilizados pela aplicação
Medida da resposta	A percentagem de respostas bem-sucedidas maior ou igual a 85%.

Tabela 7 - Cenário de qualidade para o atributo Interoperabilidade

Táticas de desempenho

Redução do overhead - Reduzir a sobrecarga introduzida por cada pedido individual, substituindo-os por múltiplos pedidos de menor complexidade. No caso da criação das rotas, introdução de algoritmos computacionalmente mais eficientes, como por exemplo Breadth-first search.

Manter múltiplas cópias dos dados - Em situações em que são efetuadas múltiplas *queries* da mesma informação, manter em memória parte dos dados que são pesquisados múltiplas vezes de modo a reduzir o número de *queries* efetuadas à base de dados.

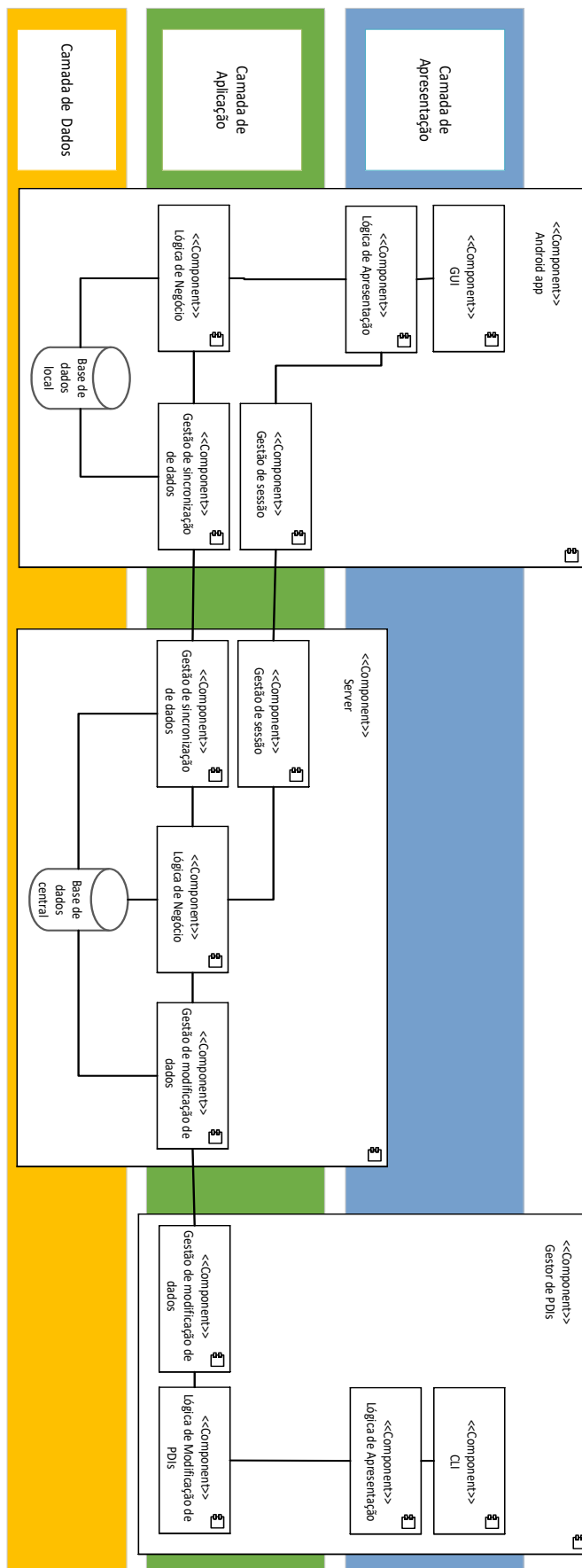


Figura 9 - Vista componente-conector do sistema

Descrição das camadas

- Apresentação:
 - Responsabilidades: Executar a GUI/CLI do *software* do lado do cliente que depende da lógica de apresentação;
 - Localização: Terminal do Cliente;
- Aplicação:
 - Responsabilidades: Gere a sessão do utilizador, mecanismos de sincronização de conteúdos e algoritmos da plataforma entre o cliente e o servidor;
 - Localização: Na sua maioria no cliente, sendo alguns módulos existentes também no servidor, apenas para funcionalidades de maior complexidade (ex.: criação de roteiros);
- Dados:
 - Responsabilidades: Armazenar em base de dados separadas os dados da plataforma. Também deve fornecer respostas aos pedidos da camada de aplicação. A base de dados local apresenta uma cópia parcial dos dados da base de dados central, permitindo assim a utilização da aplicação *offline*;
 - Localização: A localização BD central é no servidor e a BD local é no lado do cliente;

A abordagem para vista anteriormente representada tem por base a arquitetura *Rich client hybrid* [3]. Esta arquitetura é direccionada para plataformas móveis, sendo necessário efetuar algumas alterações para se adaptar ao contexto deste trabalho, bem como integrar o componente de Gestão de PDIs. Esta abordagem comparativamente a outras existentes apresenta o seguinte conjunto de vantagens e desvantagens:

Vantagens	Desvantagens
GUI numa aplicação nativa apresenta normalmente melhor usabilidade que uma aplicação Web.	Podem ocorrer quebras de desempenho devido à largura de banda da rede ser baixa;
A plataforma pode ser utilizada com ou sem ligação à internet;	Depender de um servidor com requisitos bastante exigentes em termos de desempenho (Elevado número de clientes ligados em simultâneo);
Bom desempenho, pois a largura de banda da rede utilizada é mais reduzida, dado que a GUI é carregada diretamente no cliente.	Elevada complexidade, pois faz uso de mecanismos complexos de atualização, sincronismo e verificação da consistência da secção entre Cliente/Servidor;
	Se existirem múltiplas plataformas a atualização torna-se bastante complexa;

Tabela 8 - Vantagens e Desvantagens da arquitetura *Rich client hybrid*

Tendo em conta os aspetos associados a esta arquitetura, considero-a a mais adequada dado o contexto do projeto pois, permite ao utilizador fazer uso da plataforma enquanto ligado à rede, mas também pode ser utilizada parcialmente sem acesso à internet.

3.3 Estudo das soluções tecnológicas disponíveis

Nesta secção irão ser especificadas as tecnologias por cada camada (aplicação, comunicação entre cliente-servidor e de armazenamento de dados) que melhor se adequam à presente situação, sendo identificadas as suas vantagens e desvantagens.

3.3.1 Camada de apresentação e aplicação

Uma possibilidade para o desenvolvimento da camada de aplicação são os **ambientes de desenvolvimento para aplicações nativas**. A definição de aplicação nativa utilizada nesta secção, consiste em uma aplicação que é escrita e desenvolvida para um sistema operativo particular [47]. Esta apresenta-se como uma alternativa vantajosa pois as aplicações desenvolvidas através deste meio exibem elevado nível de otimização ao serem executadas numa plataforma específica. Este método de desenvolvimento possibilita o uso de aplicações *offline*. Também permite a utilização de funções muito específicas de um terminal/OS muito particular. Por fim, permite elevado nível de integração entre *software* e *hardware*. A principal desvantagem é que cada aplicação desenvolvida apenas corre num único OS, sendo necessário criar múltiplas aplicações diferentes para cada OS, dificultando também a tarefa de atualização das mesmas.

Outra possibilidade de desenvolvimento é através do uso de **Frameworks de desenvolvimento para aplicações multiplataforma**. A definição de aplicação multiplataforma utilizada nesta secção, consiste em uma aplicação que é escrita e desenvolvida para ser executada em múltiplos sistemas operativos [48]. Estas permitem a reutilização do trabalho desenvolvido entre diferentes plataformas, exigindo um menor conhecimento técnico acerca das ferramentas de desenvolvimento e facilitam a tarefa de atualização da aplicação. Também permitem a criação de aplicações *offline*. Como desvantagens as aplicações produzidas usando este tipo de ferramentas, geralmente apresentam uma usabilidade e um desempenho inferiores às aplicações nativas. Outra desvantagem é que apesar de ser uma única tecnologia, é necessário ter conhecimento das múltiplas variações entre cada plataforma, aumentando a dificuldade e o tempo de desenvolvimento e atualização,

comparativamente a aplicações web. Por fim, outra desvantagem é que este tipo de tecnologia, por vezes demora algum tempo até suportar novas funcionalidades, sem qualquer garantia que no futuro continue a ter suporte para novas atualizações.

Um dos requisitos mais importantes da aplicação a desenvolver neste projeto é a possibilidade de utilização sem conectividade à internet tornando ambos os métodos aceitáveis para a presente situação. No entanto, como o atributo de qualidade **desempenho** apresenta maior nível de prioridade que o atributo de qualidade **interoperabilidade**, considerou-se o desenvolvimento de aplicações nativas mais adequado que o uso de uma *framework* de desenvolvimento de aplicações multiplataforma. Outro aspeto a ter em conta nesta escolha foi uma aplicação que se encontra a ser desenvolvida pela CMF onde, possivelmente, este projeto poderá vir a ser integrado, sendo que o desenvolvimento dessa aplicação fazia uso do SDK para Android desenvolvido pela Google, daí ter sido escolhido a metodologia de desenvolvimento de aplicações nativas.

3.3.2 Comunicação entre cliente e servidor

Uma das possibilidades para a comunicação entre cliente-servidor é através do uso de **tecnologias Push de forma combinada com o uso de Restful Web Services** cuja comunicação é através do protocolo *Hypertext Transfer Protocol* (HTTP). Este método permite comunicação bidirecional entre o cliente e o servidor. Outra vantagem é que estes apresentam elevado suporte para o desenvolvimento de aplicações nativas, embora também existam soluções *push* multiplataforma. Este tipo de comunicação requer um baixo consumo energético por parte dos terminais móveis. Por fim, este não se apresenta sujeito a quebras de ligação quando ocorre alteração do estado da aplicação, algo que ocorre frequentemente quando, por exemplo, o terminal entra em modo de repouso. A única desvantagem é a dependência de tecnologias proprietárias com um custo associado.

Uma alternativa é a através da utilização de *web services* cuja comunicação é feita através de **websockets**. Este tipo de comunicação também permite comunicação bidirecional com baixo consumo energético. Outra vantagem é que os *websockets* dependem de um grande conjunto de bibliotecas *open source* multiplataforma, sendo assim de utilização gratuita. As principais desvantagens são que este meio não apresenta suporte nativo e sofre quebras de ligação com alteração do estado da aplicação.

Optou-se pelo uso de *Restful Web Services* pois, nesta situação, não foi definida nenhuma funcionalidade que necessitasse de comunicação bidirecional, podendo ser introduzido, posteriormente o uso de tecnologias *push* caso haja necessidade de comunicação bidirecional, sendo que o elevado suporte nativo facilita a introdução deste tipo de mecanismos. Em relação à tecnologia escolhida, optou-se pela criação de um servidor Apache cuja linguagem de desenvolvimento é PHP principalmente devido à familiarização pessoal com este tipo de linguagem e por esta ser *open source*.

3.3.3 Armazenamento de dados

Nesta situação, as duas alternativas consideradas são as bases de dados relacionais e não relacionais.

Uma base de dados relacional é um tipo de base de dados que está organizada pelo modelo relacional de dados proposto por E. F. Codd [49], sendo que cada grupo de itens é organizado por tabelas que apresentam colunas e linhas. Cada item da tabela é inserido numa linha e é designado por registo ou tuplo. Cada coluna apresenta uma lista de valores dos atributos associados a cada tuplo inserido na tabela. O uso de bases de dados relacionais favorece uma maior robustez e consistência dos dados e desfavorece ao apresentar uma inferior flexibilidade de modificação e disponibilidade entre diversos servidores.

Uma base de dados não relacional consiste num grupo alargado de bases de dados que não apresentam uma estrutura de tabelas usada nas bases de dados relacionais [50]. Estas bases de dados apresentam, normalmente, um conjunto de vantagens e desvantagens comuns que são contrastantes às bases de dados relacionais. O uso de bases de dados não relacionais favorece a flexibilidade de modificação estrutural e disponibilidade entre diversos servidores e desfavorece ao apresentar uma inferior robustez e consistência dos dados.

A escolha final do tipo de base de dados a utilizar tem sempre em conta o tipo de requisitos pretendidos para cada projeto, sendo que ambas podem ser soluções vantajosas para diferentes situações. Sendo assim ambos os tipos de base de dados irão ser utilizados, sendo posteriormente feitos alguns testes ao desempenho das mesmas, com o objetivo de analisar qual destas apresenta melhor desempenho, pois foi definido na *secção 3.2.1* que o atributo de qualidade **desempenho** é relevante para este projeto.

O sistema de gestão de base de dados relacional considerada foi MySQL devido à familiarização pessoal com o mesmo e por este ser *open source*.

O sistema de gestão de base de dados não relacional considerado é o Neo4j, sendo este orientado a grafos. A escolha da utilização de uma base de dados orientada a grafos deve-se à natureza do tipo de informação a armazenar e de *queries* a efetuar. Neste caso, pretende-se guardar um conjunto de dados associados a diversos PDIs e um conjunto de rotas possíveis estes mesmos PDIs, sendo que o tipo de pesquisas a efetuar implica a utilização de alguns algoritmos de pesquisa de modo a encontrar o percurso mais curto entre os PDIs, sendo que o Neo4j permite executar estes algoritmos de pesquisa diretamente sobre a linguagem de consulta utilizada (Cypher).

4. Desenvolvimento da aplicação

Neste capítulo irá ser descrito o processo de desenvolvimento do sistema. É de referir que a metodologia utilizada foi incremental, na medida em que as funcionalidades da aplicação foram sendo desenvolvidas em várias iterações, sendo que cada uma destas iterações iniciou-se com a criação de protótipos de baixa fidelidade, que depois de validados progrediram para protótipos de alta fidelidade.

Foram produzidas três iterações. A primeira iteração teve como objetivo a análise do domínio do problema de modo a identificar quais as funcionalidades que deveriam conter o sistema final, sendo para isto produzido um protótipo de baixa fidelidade, que depois evoluiu para um protótipo de alta fidelidade e por fim, este protótipo foi avaliado por pessoal da CMF, sendo avaliadas as possíveis alterações a efetuar e novas funcionalidades a introduzir.

A segunda iteração consistiu na criação de um protótipo de baixa fidelidade integrando novas funcionalidades identificadas na primeira iteração. Foi efetuada a pesquisa de dados acerca de PDIs que existam no Funchal, fez-se a implementação do servidor Apache com funcionalidades que permitam ler e escrever PDIs e rotas entre estes. Fez-se a implementação da BD neo4j, implementação do sistema de gestão de PDIs, foram modificados os aspetos menos positivos do *design* da GUI produzida na primeira iteração, foi feita a integração da aplicação com os *webservices* produzidos pelo servidor e por fim foram efetuados alguns testes de utilizador (questionários) de modo a avaliar o protótipo de alta fidelidade produzido.

Na terceira iteração, procedeu-se novamente à modificação da GUI da aplicação para introduzir alguns padrões de *design* definidos pela Google para Android. Desenvolveu-se funcionalidades relacionadas com a criação e recomendação de roteiros, recomendação de PDIs, agendamento de itens, gestão dos dados do utilizador, favoritos e suporte para múltiplas línguas. Implementou-se também uma nova base de dados MySQL, sendo necessário implementar o padrão de desenho *Bridge* para permitir a abstração entre as classes que fazem acesso às diferentes bases de dados.

4.1 Primeira Iteração

Como já fora referido, a primeira iteração teve o objetivo de estudar o problema de modo a identificar quais as funcionalidades que deveria conter o sistema final, sendo para isto produzido um protótipo de baixa fidelidade, que depois evoluiu para um protótipo de alta fidelidade e por fim, este protótipo foi avaliado por pessoal da CMF, sendo avaliadas as possíveis alterações a efetuar e novas funcionalidades a introduzir.

4.1.1 Protótipo de baixa fidelidade

Os protótipos de baixa fidelidade foram desenvolvidos com recurso ao *software* Pencil [4], sendo que a principal vantagem deste é ser *open source*. O principal objetivo deste *software* é a prototipagem de GUIs para diferentes plataformas. Dado este não apresentar suporte atualizado para prototipagem de aplicações que apresentem elementos de design associados às versões mais recentes do OS Android, foi necessário a introdução de um *plugin* externo para o efeito [5].

O protótipo de baixa fidelidade desenvolvido nesta iteração apresenta alguns esquemas associados às vistas que permitem a utilização das funcionalidades relacionadas com os PDIs, nomeadamente:

- visualização de PDIs (em modo mapa e lista);
- apresentação dos filtros;
- detalhes dos PDIs;
- criação de rotas até aos PDIs;

Este protótipo de baixa fidelidade também exhibe alguns esboços relacionados com as vistas que possibilitam as funcionalidades associadas aos roteiros: os roteiros personalizados e os roteiros recomendados. Ambas as secções dos roteiros são idênticas sendo que a única diferença reside no facto de os roteiros personalizados permitirem a criação e edição de roteiros. Inicialmente os roteiros recomendados iriam ser roteiros pré-existentes definidos pela CMF, sendo que o suporte a este tipo de dados nunca chegou a ser fornecido, sendo posteriormente abandonada a ideia de uso de roteiros pré-existentes. Os protótipos de baixa fidelidade apresentam alguns esquemas associados às funcionalidades relacionadas com os roteiros, nomeadamente:

- visualização de roteiros (em modo mapa e lista);
- apresentação de filtros;
- detalhes dos roteiros;
- criação de rotas até aos PDIs dos roteiros;

As imagens dos protótipos podem ser visualizadas no anexo A.

4.1.2 Protótipo de alta fidelidade

Seguidamente ir-se-á apresentar os aspetos tidos em conta para o desenvolvimento do protótipo de alta fidelidade em termos de design e ferramentas utilizadas para o desenvolvimento do mesmo. Por fim, irão ser apresentadas as diferentes vistas do protótipo de alta fidelidade e suas funcionalidades.

4.1.2.1 Design e iconografia utilizados

Toda a iconografia utilizada nos protótipos de alta fidelidade proveio do site Google Design [9] e do site icon8 [10]. Ambos os sites apresentam um grande conjunto de ícones gratuitos que seguem as guidelines indicadas pela Google para o *design* de ícones para Android [11].

4.1.2.2 Ferramentas utilizadas

Nesta subsecção irão ser especificadas quais ferramentas externas que foram integradas com o ambiente de desenvolvimento de aplicações nativas Android (Android Studio) e como estas foram utilizadas durante o desenvolvimento do protótipo de alta fidelidade.

4.1.2.2.1 Android-maps-utils - Extração de dados de ficheiros KML

Inicialmente, foi requisitado para o protótipo de alta-fidelidade que para além de adicionar um marcador ao local do PDI, fossem criados polígonos representando os edifícios que compõem cada PDI, tendo sido para isto fornecidos diversos ficheiros KMZ gerados através de *software* do tipo GIS (*Geographic information system*). Estes ficheiros na realidade são um único ficheiro KML repartido e comprimido, que depois da descompressão através da utilização de softwares, como o Winrar, permite a obtenção de dados no formato KML passíveis de ser utilizados diretamente pelo Google Maps. Este tipo de ficheiro é uma variante do XML, sendo mais direcionado para conter informações geográficas. O esquema completo XML para o KML está bem documentado pela Google, facilitando a extração de dados através de um *parser* [6]. A API nativa do Google Maps para Android não permite efetuar a extração de dados deste tipo de ficheiros, sendo assim foi necessário utilizar uma biblioteca externa (android-maps-utils) [7] para a extração dos dados do KML. Após extraída a informação conseguiu-se obter os dados acerca do nome de cada PDI e geometria dos edifícios que compõem o mesmo (através de múltiplas coordenadas geográficas). Por fim, os edifícios foram desenhados através de polígonos que

apresentavam as coordenadas obtidas. As coordenadas dos marcadores que representavam cada PDI foram calculadas através da obtenção do centro geométrico (*Centroide*) de p polígonos diferentes (fórmula (3)) [8].

$$C = \frac{\sum_{i=1}^l x_i}{l} \quad (1)$$

$$C_{latitude} = \frac{\sum_{i=1}^p C_{i_{latitude}} A_i}{\sum_{i=1}^p A_i}, C_{longitude} = \frac{\sum_{i=1}^p C_{i_{longitude}} A_i}{\sum_{i=1}^p A_i} \quad (2)$$

$$Centroide = (C_{latitude}, C_{longitude}) \quad (3)$$

O calculo do centro geométrico para múltiplos polígonos exige o cálculo separado do centro geométrico quer para a latitude ($C_{latitude}$) quer para a longitude ($C_{longitude}$), sendo a fórmula (2) utilizada para determinar ambos os valores. Nesta fórmula (2) o valor de p representa o número total de polígonos sobre os quais se pretende calcular o centro geométrico, A_i o valor da área associada ao polígono i , $C_{i_{latitude}}$ e $C_{i_{longitude}}$ correspondem aos valores do centro geométrico para o polígono i .

Os valores de $C_{i_{latitude}}$ e $C_{i_{longitude}}$ podem ser determinados através da utilização da fórmula (1). Nesta fórmula (1) a variável C corresponde ao valor do centro geométrico de 1 polígono para apenas uma dimensão, l representa o número total de pontos que dão forma ao polígono, x_i corresponde ao valor das coordenadas para o ponto i do polígono.

Os dados extraídos não correspondem a pontos de interesse turísticos, uma vez que apenas foi fornecido um ficheiro temporário para testes cujo conteúdo apresentava informação acerca de instituições de ensino.

4.1.2.2.2 AndroidSlidingUpPanel – Painel deslizante

A biblioteca `AndroidSlidingUpPanel` [12], disponibiliza um novo elemento para a UI que consiste num painel deslizante, que pode ser ativado ao arrastar ou ao selecionar o cabeçalho do mesmo. Para a utilização deste elemento é necessário criar um `FrameLayout` (Elemento que vai ser sobreposto pelo painel deslizante) e outro *layout* do tipo `RelativeLayout` ou `LinearLayout`, sendo este último a vista deslizante. Também é necessário definir um cabeçalho para a vista deslizante, sendo considerado o primeiro *layout* filho do painel deslizante. A vantagem da utilização deste elemento é que permite a introdução de mais elementos visuais sem sobrecarregar a atual vista, pois os elementos podem ser mostrados e escondidos a qualquer momento.

4.1.2.2.3 Google Maps Directions API – Dados acerca das rotas entre PDIs

A API de direções do Google Maps é um *webservice* disponibilizado pela Google [13]. Este serviço cria um objeto JSON contendo informação detalhada, acerca da distância e tempo de viagem, coordenadas dos múltiplos locais onde ocorre mudança de direção (Ex. virar para uma nova rua). Também são fornecidos dados no formato de texto semelhantes aos disponibilizados por uma aplicação de GPS (ex. Vire à esquerda na rua X). Este serviço é gratuito até ao limite de 1000 pedidos por dia sem a criação de uma chave que identifique a aplicação e 2500 pedidos diários com a criação de uma chave identificadora. Quaisquer outros pedidos que superem estes valores são rejeitados, sendo que a partir daqui requerem uma licença de utilização paga. A utilização desta API é feita pela manipulação de um endereço URL permitindo especificar os diferentes pontos da rota por coordenadas ou pelo nome da rua. Outro parâmetro que pode ser modificado é o meio de locomoção podendo ser de 4 tipos: *driving*, *walking*, *transit* e *cycling*. Por fim, este serviço permite a otimização da rota até no máximo 8 pontos diferentes. Existem outros parâmetros disponíveis, mas só se fez uso dos anteriormente referidos.

4.1.2.3 Vistas do protótipo de alta fidelidade

Esta secção tem como objetivo apresentar as vistas desenvolvidas no protótipo de alta fidelidade e consequentemente apresentar as funcionalidades associadas às mesmas. Os dados apresentados nas vistas do protótipo de alta fidelidade são genéricos cujo objetivo é apenas apresentar o aspeto da GUI para cada vista.

4.1.2.3.1 Menu principal

O menu principal (Figura 10) apresenta três opções possíveis: PDI, Roteiros recomendados e Roteiros personalizados.

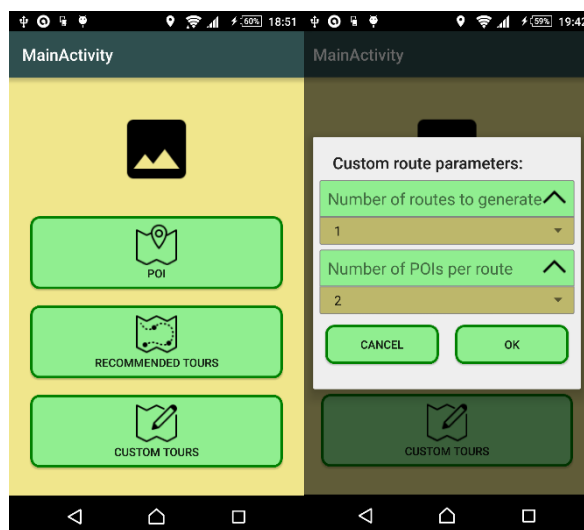


Figura 10 - Menu principal

Das três opções apresentadas, apenas duas se apresentam funcionais (PDI e Roteiros personalizados). Ao selecionar a opção PDI a aplicação inicia a extração dos ficheiros KML, levando alguns segundos no processo, sendo o utilizador notificado deste processo através de um objeto Toast (Notificação de texto de curta duração).

No caso de selecionar a opção de roteiros personalizados, é apresentado um diálogo de modo a gerar um conjunto de roteiros que apresentam um conjunto de PDIs selecionados de forma aleatória, sendo que os únicos parâmetros personalizáveis são o número de roteiros e o número de PDIs por roteiro. Os roteiros gerados apresentam uma rota otimizada, sendo esta obtida através da API de direções do Google Maps.

4.1.2.3.2 Vista de PDIs

Ao selecionar a opção PDI no menu principal são apresentados dois separadores que permitem alternar entre modo mapa e modo lista (Figura 11).

O separador mapa apresenta um conjunto de marcadores e de polígonos que representam cada PDI. Estes marcadores ao serem selecionados permitem o acesso a outro menu que apresenta os detalhes do respetivo PDI.

A vista em lista apresenta um conjunto de itens que correspondem a cada PDI. Neste caso para aceder aos detalhes do PDI é necessário selecionar o respetivo *layout* do item.

O menu PDIs também apresenta um conjunto de filtros, sendo que estes podem ser visualizados ao selecionar a barra (*Filters*) no canto inferior da vista de PDIs. Os filtros apresentam um conjunto de campos relacionados com as informações apresentadas nos PDIs, sendo que neste protótipo os filtros não se apresentam funcionais. Em algumas situações foi necessário modificar o *design* dos tipos de *input* existentes nos filtros. Por exemplo, um dos *inputs* modificados foi o Spinner/Dropdown da categoria, sendo que este tipo de *input* por defeito não permite a existência de imagens por cada item. Para contornar esta situação foi necessário criar um ArrayAdapter e um *layout* que irá preencher cada item.

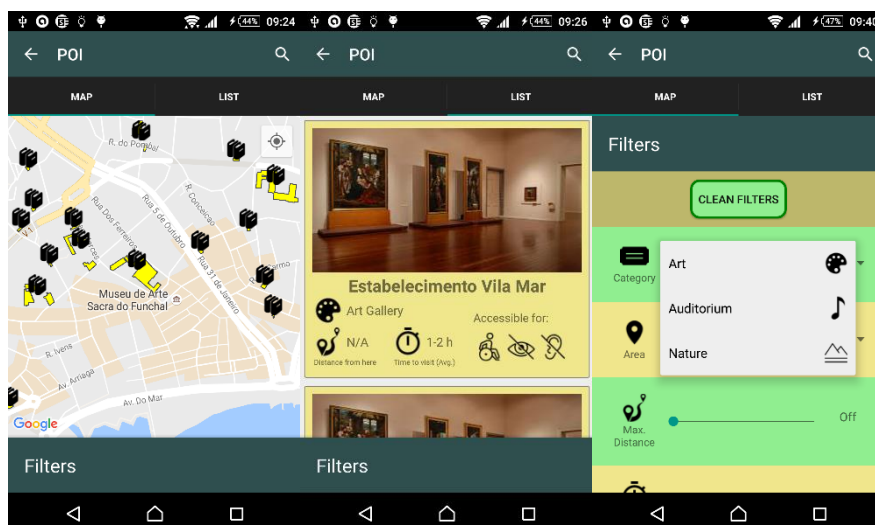


Figura 11 – Vista de PDIs

4.1.2.3.3 Vista PDI detalhes

Ao selecionar um PDI é apresentada a vista de detalhes do mesmo (Figura 12).

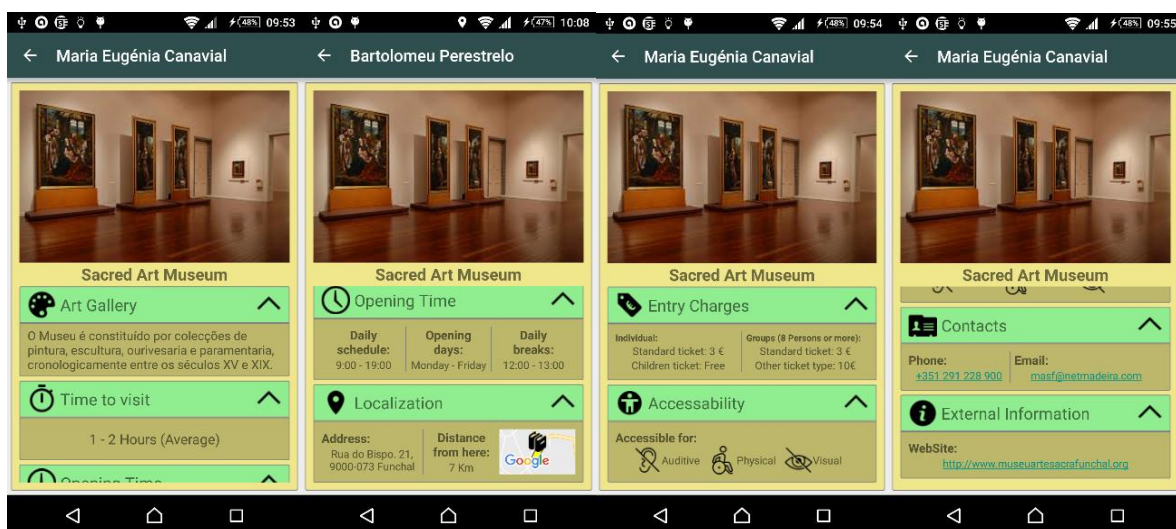


Figura 12 - Vista PDI detalhes

Esta vista apresenta um conjunto de dados informativos associados ao PDI. Cada secção deste menu apresenta a opção de mostrar/ocultar o campo (Símbolos \wedge e \vee no lado direito do cabeçalho do campo). Ao selecionar a opção de ocultar (\wedge) o campo é iniciada uma animação que gera movimento do *layout* informativo para debaixo do cabeçalho sendo que após 1 segundo de movimento o *layout* informativo é ocultado.

4.1.2.3.4 Vista PDI rota

Nos detalhes da vista ao selecionar o campo localização, a aplicação inicia a vista de rota para o PDI (Figura 13).

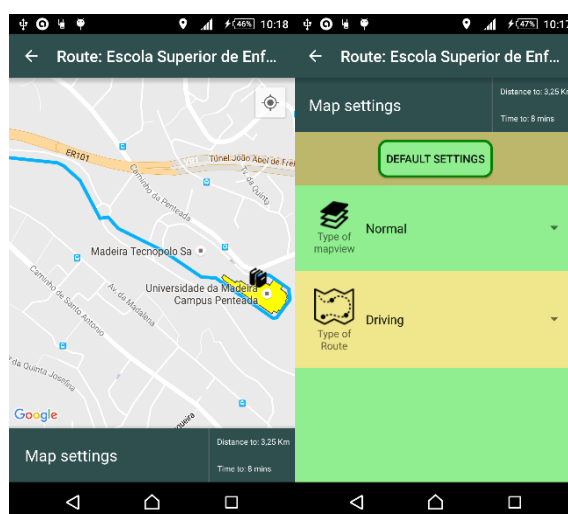


Figura 13 - Vista PDI rota

A criação da rota apresentada pela *polyline* azul sobre o mapa e a obtenção dos dados da distância e do tempo é feita através do serviço de direções do Google Maps. Para criar a *polyline* é necessário decodificar um campo encriptado do objeto JSON enviado pelo serviço [14]. Já a extração da distância e do tempo é mais simplificada, dado que estas informações se apresentam de forma legível no corpo do objeto JSON (routes => legs => distance/duration => value). A atualização da *polyline* é feita em cada 10 metros e para este protótipo apenas quando o *smartphone* apresenta ligação à internet.

Nesta vista, fez-se novamente uso do painel deslizante, sendo que este apresenta *inputs* que permitem modificar as definições do tipo de mapa e do tipo de rota pretendidos.

4.1.2.3.5 Menu Roteiros personalizados

No menu principal ao selecionar a opção de roteiros personalizados, são gerados um conjunto de roteiros com um grupo de PDIs selecionados aleatoriamente, como já fora referido. Após a seleção do mesmo é apresentada a vista presente na Figura 14.

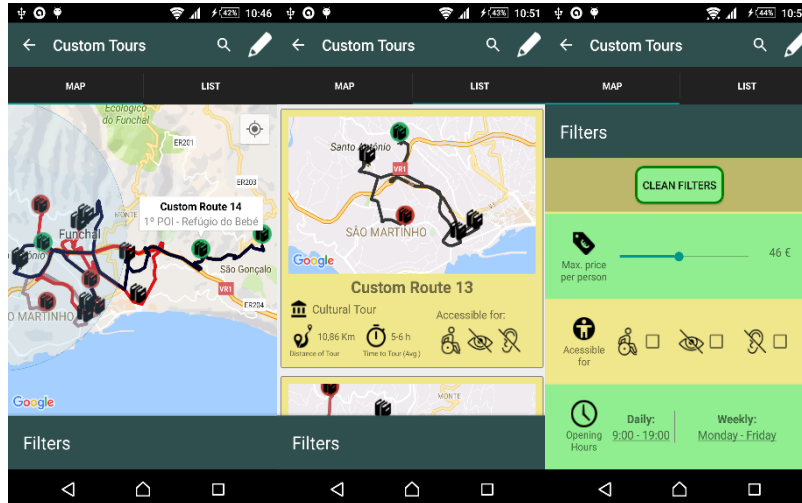


Figura 14 - Vista roteiros personalizados

A apresentação da GUI nesta vista apresenta muitas semelhanças com a vista de PDIs. As *polylines* aqui apresentadas foram obtidas do mesmo modo que foi identificado na secção anterior, embora apresentem algumas diferenças para tornar estas interativas. Por defeito, as *polylines* são apenas múltiplas linhas retas criadas entre diversos pontos situados em locais onde há mudança de direção (tais como cruzamentos, curvas, etc.), não havendo forma de obter as suas coordenadas intermédias, apenas dos pontos nas suas extremidades. Nesta iteração fez-se apenas a deteção do local que foi selecionado do mapa, sendo este comparado com as coordenadas das extremidades das múltiplas linhas da *polyline*. Caso estas se encontrem num raio de 100 metros do local selecionado é ativado o evento que permite visualizar os detalhes da *polyline*. Esta metodologia não funciona para situações em que existem dois pontos intermédios da *polyline* com mais de 100 metros de distância, pois permite a existência de locais sem sensibilidade ao toque. Este aspeto foi corrigido na iteração seguinte através de um método que irá ser posteriormente explicado.

Nesta situação, os marcadores com o fundo verde e vermelho representam, respetivamente, o PDI de início e de fim do roteiro. Os filtros também se apresentam de forma idêntica ao anteriormente exibido. Uma opção aqui presente é o botão com a forma de lápis que permite iniciar a vista para criar um novo roteiro. Ao selecionar os marcadores,

nesta situação é apresentada uma InfoWindow que indica qual o roteiro a que este pertence e qual a sua posição no roteiro.

4.1.2.3.6 Vista roteiro detalhes

A vista de detalhes de um roteiro é apresentada na Figura 15.

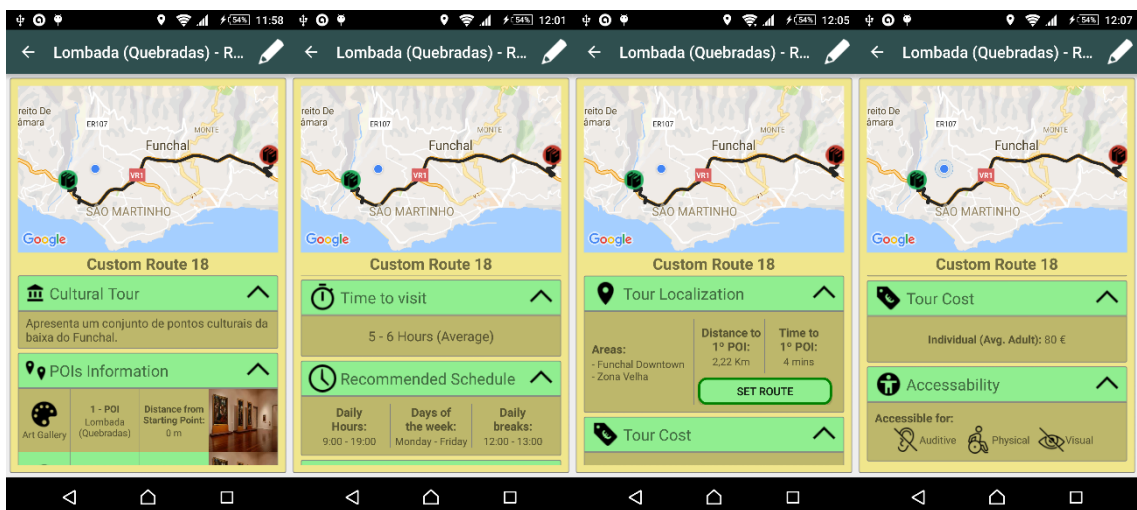


Figura 15 - Vista roteiros detalhes

A vista de detalhes dos roteiros em termos de apresentação da GUI é um pouco semelhante à de detalhes do PDI. Nesta situação é apresentada também uma lista de PDIs que compõem o roteiro. Cada um destes itens quando selecionado, permitem a inicialização da vista de detalhes do PDI. Nesta situação, o ícone do lápis permite editar o presente roteiro.

4.1.2.3.7 Vista roteiro rota

Ao seleccionar a opção *SET ROUTE*, é aberta a vista de rota para cada PDI do roteiro (Figura 16).

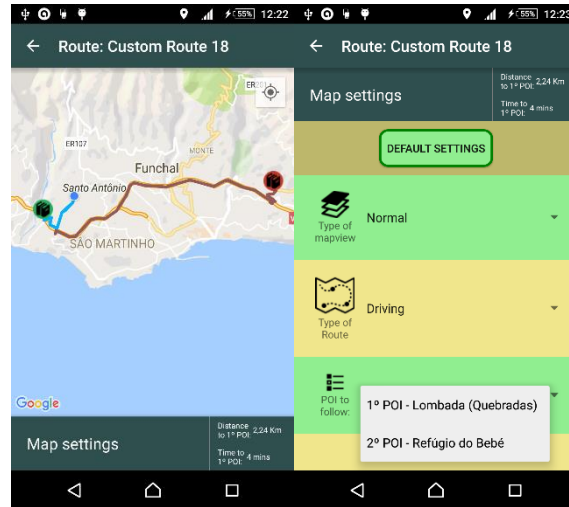


Figura 16 - Vista roteiro rota

A vista de rota para os PDIs do roteiro em termos de apresentação da GUI é um pouco semelhante à de rota para PDI simples. Nesta situação são criadas duas *polylines*, uma para o PDI para onde o utilizador se dirige e outra para definir o percurso completo do roteiro. O utilizador ao alcançar um PDI do roteiro é redireccionado para o seguinte, sendo removida a *polyline* anterior e criada uma nova para o efeito.

4.1.2.3.8 Vista roteiro criar/editar

Ao seleccionar a opção de criar ou editar roteiro é apresentada a vista presente na Figura 17.

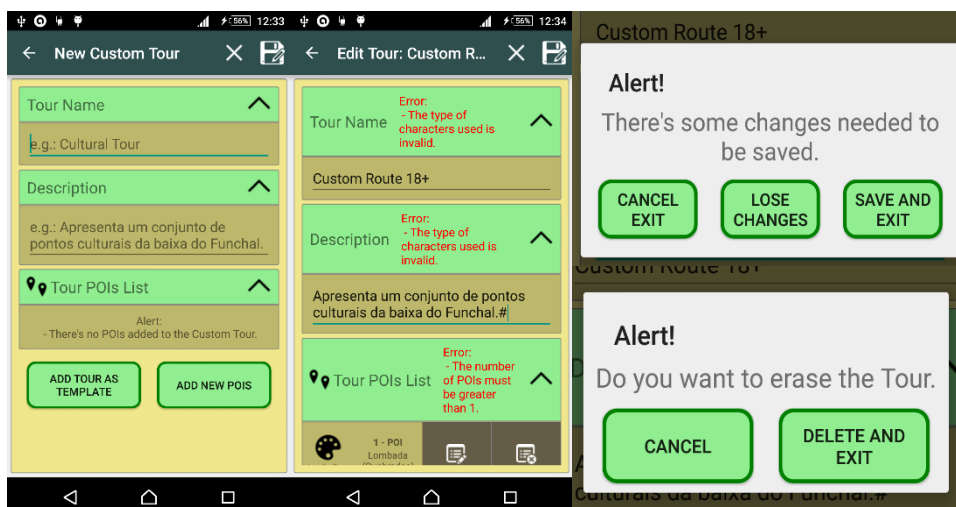


Figura 17 - Vista roteiro criar/editar

Tanto o campo de nome do roteiro como a descrição apresentam validação de tipo e número de caracteres introduzidos, sendo apenas permitidos caracteres alfanuméricos. No caso da lista de PDIs existe a restrição que não podem ser adicionados menos de 2 e mais de 10 PDIs. Os ícones no canto superior direito servem para, respetivamente, apagar e guardar o roteiro criado/editado. Quando é seleccionada a opção de apagar aparece a mensagem de alerta presente no canto inferior direito da Figura 17. Quando o utilizador efetuou qualquer tipo de modificações no roteiro e retrocede sem guardar é mostrado o alerta exibido no canto superior direito da Figura 17.

4.1.2.3.9 Vista roteiro adicionar PDIs

Na vista de criar/editar roteiro ao seleccionar a opção de adicionar novos PDIs é iniciada a vista presente na Figura 18.

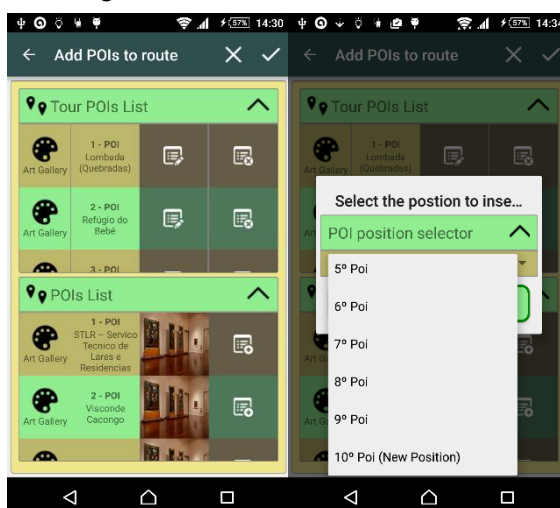


Figura 18 - Vista roteiros adicionar PDIs

Nesta situação é apresentada a lista de PDIs que estão atualmente associados ao roteiro na parte superior do ecrã e na parte inferior do ecrã a lista de todos os PDIs existentes no sistema. No canto superior direito estão presentes dois botões que servem, respetivamente, para rejeitar ou aceitar os PDIs adicionados. Os dois botões presentes em cada item do roteiro, servem respetivamente para editar a posição do PDI no roteiro e remover o PDI do roteiro. O botão presente em cada item da lista geral de PDIs serve para adicionar o PDI ao roteiro. Ao seleccionar a opção editar de um PDI é apresentado um novo diálogo, tal como pode ser observado no lado direito da Figura 18.

4.1.2.3.10 Vista roteiro adicionar roteiros como *template*

Na vista de criar/editar roteiro ao seleccionar a opção de adicionar roteiros como *template* é iniciada a vista presente na Figura 19.

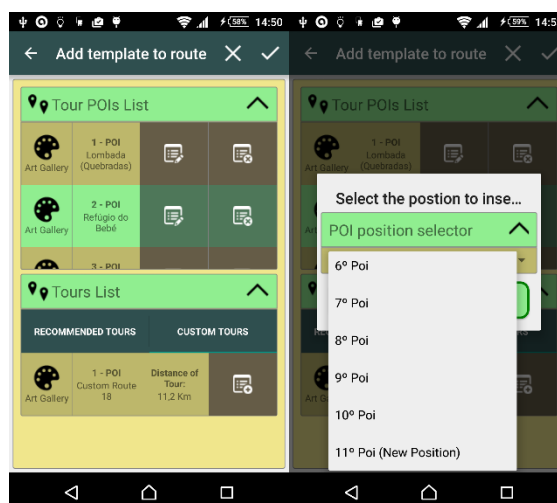


Figura 19 - Vista roteiros adicionar roteiros como *template*

Esta vista é bastante semelhante à vista que permite adicionar novos PDIs, sendo que a única variação é no canto inferior da vista, mais propriamente na secção lista de roteiros que apresenta todos os roteiros existentes no sistema. Neste protótipo apenas estão disponíveis os roteiros personalizados.

4.1.3 Conclusão

Após a conclusão do protótipo de alta-fidelidade foram discutidas novas funcionalidades a integrar na aplicação.

Um aspeto discutido foi a modificação da informação utilizada no protótipo, visto que apenas se tratavam de dados de teste.

Outro aspeto a melhorar foi o desempenho da aplicação, sendo que aspetos como o elevado número de edifícios desenhados e mostrados em simultâneo tornavam a aplicação um pouco lenta. A extração dos dados diretamente de um ficheiro KML para centenas de locais é um processo relativamente demorado, comparativamente a outros formatos como por exemplo JSON. O armazenamento dos dados em memória após a extração dos mesmos também contribuiu para um fraco desempenho do protótipo.

Por fim, outro aspeto a melhorar foi a GUI, de modo a que permita a integração de *design* com outra aplicação a ser desenvolvida pela CMF.

4.2 Segunda Iteração

Na 2ª iteração, o objetivo principal foi definir uma arquitetura, sendo que esta já fora referenciada na secção 3. Também pretende-se implementar parte desta arquitetura, sendo que esta deverá ser integrada com parte dos elementos do protótipo produzido na 1ª iteração, de modo a que estes possam ser reutilizados. Por fim, pretende-se efetuar alguns testes ao novo protótipo de alta-fidelidade produzido.

4.2.1 Modificações efetuadas

Para esta iteração, os dados fornecidos foram um ficheiro KML com dados acerca de PDIs direcionados para o turismo. Este ficheiro, contrariamente ao referido na primeira iteração, não apresenta as coordenadas dos edifícios, sendo este aspeto ignorado devido aos problemas de desempenho gerados pelos polígonos sobre o mapa. O novo ficheiro KML só apresenta o nome do local e as coordenadas que representam o centro geométrico de cada PDI. Outros dados necessários para preencher a informação acerca dos PDIs tiveram de ser pesquisados e preenchidos manualmente no ficheiro KML.

Outro aspeto a modificar foi a fonte dos dados das rotas criadas, dado que o limite de pedidos gratuitos para o *webservice* de direções do Google Maps é relativamente curto, sendo frequentemente bloqueado devido ao excesso de pedidos efetuados na 1ª iteração. Nesta iteração, pretende-se armazenar todas as combinações de rotas, distâncias e tempos de viagem possíveis entre PDIs, numa base de dados localizada num servidor, sendo que a aplicação móvel deve efetuar pedidos diretamente a este servidor. O serviço do Google Maps só deverá ser utilizado quando o utilizador esteja a seguir uma rota, apresente ligação à internet e percorra uma distância significativa (20 metros) até a aplicação efetuar um novo pedido. Para os restantes pedidos acerca da distância, tempo e criação de roteiros deverão ser utilizados os dados presentes na base de dados existente no servidor implementado.

A melhoria do desempenho da aplicação deve ser feita também ao substituir o armazenamento de dados localmente num ficheiro KML por outros ficheiros no formato JSON, sendo este um formato muito mais rápido para extrair dados e gera ficheiros que ocupam menor espaço em memória que o formato KML [15].

Um dos principais aspetos que tornavam a aplicação lenta era a necessidade de fazer imensos pedidos ao *webservice* do Google Maps e obrigar a aplicação a reinterpretar

os dados obtidos de modo a extrair dados de todas as rotas obtidas cada vez que o utilizador muda de posição. A tática que permite melhorar o desempenho deste aspeto é o estabelecimento de um ponto-base (ex. Hotel onde o utilizador irá ficar alojado), sendo todas as rotas, distâncias e tempos calculados uma única vez durante a definição deste local. Posteriormente, irá ser necessário armazenar na base de dados todas as rotas, distâncias e tempos possíveis entre o ponto-base definido pelo utilizador e os PDIs definidos na plataforma.

Por fim, outro aspeto a ser modificado é a GUI, sendo feitas alterações no esquema de cores de alguns *layouts*.

4.2.2 Protótipos de baixa fidelidade

Os protótipos de baixa fidelidade apresentam esquemas que representam as vistas associadas a um conjunto alargado de funcionalidades sendo que algumas destas nunca chegaram a ser implementadas no protótipo de alta fidelidade devido a restrições temporais. A seguinte lista de funcionalidades indica o conjunto total de funcionalidades associadas às vistas presentes nos esboços representados no protótipo de baixa fidelidade.

- Visualização dos PDIs em diversos tipos de vista (mapa e lista);
- Visualização dos detalhes dos pontos de interesse com respetiva informação sobre os mesmos;
- Criação de diversos tipos de rota até ao ponto de interesse;
- Visualização de rotas em diversos tipos de vista (mapa e lista);
- Visualização dos detalhes das rotas com respetiva informação sobre as mesmas;
- Criação de diversos tipos de rota até cada um dos pontos de interesse das referidas rotas;
- Criação de rotas personalizadas;
- Agendamento das visitas aos pontos de interesse e rotas;
- Lista de favoritos dos pontos de interesse;
- Lista de favoritos das rotas;
- *Login* dos utilizadores;
- Registo dos utilizadores;
- Preferências do utilizador, em relação às categorias dos PDIs/Rotas que pretende visitar;
- Avaliação dos PDIs;
- Visualização da opinião de outros utilizadores acerca de um determinado PDI;
- Visualização dos PDIs mais visitados;
- Recomendação de PDIs de interesse do utilizador;
- Recomendação rotas de interesse do utilizador;
- Visualização do histórico de visitas aos PDIs individuais;

- Visualização do histórico de visitas aos PDIs das rotas;
- Associação de fotos ao PDI;
- Associação de curtos clipes (vídeos) ao PDI;
- Associação de fotos aos PDIs da Rota;
- Associação de curtos clipes (vídeos) aos PDIs da Rota;
- Pesquisa seletiva de PDIs;
- Pesquisa seletiva de rotas;

As imagens do protótipo podem ser visualizadas no anexo B.

4.2.3 Protótipo de alta fidelidade

Seguidamente ir-se-á apresentar os aspetos tidos em conta para o desenvolvimento do protótipo de alta fidelidade, sendo descrito como os dados para PDIs foram obtidos e quais as ferramentas utilizadas durante a fase de desenvolvimento do protótipo de alta fidelidade (incluindo a implementação do servidor, da base de dados e do sistema de preenchimento da base de dados, entre outros). Por fim, irão ser apresentadas as diferentes vistas do protótipo de alta fidelidade e suas funcionalidades.

4.2.3.1 Obtenção de dados para os PDIs

Uma das primeiras tarefas da 2ª iteração foi a obtenção de dados acerca dos PDIs. A obtenção de dados foi feita manualmente, sendo posteriormente guardados num ficheiro KML. Os dados foram obtidos de diversos sites com informação detalhada acerca de inúmeros PDIs existentes na Região Autónoma da Madeira. Os principais *websites* de onde a informação fora extraída, são o site da CMF [16] e o site da direção regional de turismo da madeira [17]. Após a extração dos dados, estes foram colocados numa secção do ficheiro KML delimitada pela *tag description* que suporta HTML. Foi construída uma tabela em HTML contendo pares *key-value* sendo aqui inseridos os dados obtidos.

4.2.3.2 Ferramentas utilizadas

Nesta subsecção irão ser especificadas quais as ferramentas externas que foram integradas com o ambiente de desenvolvimento de aplicações nativas Android (Android Studio), com o ambiente de desenvolvimento Java (IntelliJ idea) e com o ambiente de desenvolvimento PHP (IntelliJ idea com plugin PHP 162.1121.34). Também será especificado como estas foram utilizadas.

4.2.3.2.1 XAMPP - Implementação do servidor

A implementação do servidor foi feita através do *software* XAMPP [18]. Este *software* foi escolhido por ser *open source* e por apresentar um funcionamento que me é familiar. Este permite a criação de um servidor HTTP Apache que permite o desenvolvimento de scripts em PHP.

Apesar da não utilização de uma *framework* PHP, utilizou-se o padrão arquitetural MVC, de modo a que exista independência entre as camadas da vista (aplicação android), controlador (classes onde a aplicação Android efetua pedidos HTTP) e modelo (classes responsáveis pelos dados e pela lógica no servidor), sendo que a estrutura das mensagens trocadas é o único aspeto que deve ser mantido.

A implementação do servidor foi feita de tal modo a que possam ser efetuadas modificações na lógica do lado do servidor, sem ter de ser modificado qualquer aspeto no lado do cliente, desde que o formato dos parâmetros dos dados enviados como resposta (objetos JSON) seja mantido.

Também é possível modificar o tipo de base de dados sem que haja diferença nos dados obtidos do lado do cliente, sendo este um aspeto que irá ser abordado em maior detalhe na 3ª iteração do projeto.

Do lado do cliente, existe também a utilização do padrão arquitetural MVC, sendo as vistas criadas em XML, o controlador são as classes denominadas de atividades (fazem a escolha do *layout* XML a ser apresentado e envia os parâmetros dos modelos para a vista) e o modelo são as classes designadas de Helper que contêm a lógica de negócio e fazem a extração e conversão dos dados locais ou dos *webservices* em objetos que possam ser utilizados para preencher as vistas.

4.2.3.2.2 Neo4j community edition e neo4jphp - criação da interface de comunicação e implementação da base de dados neo4j

Para a criação de uma base de dados neo4j podem ser seguidas duas abordagens. A primeira consiste, na utilização de um serviço de *hosting* que suporte neo4j (ex.: Graphenedb) [19]. Outra abordagem consiste na utilização de um computador pessoal em que seja instalado o *software* desenvolvido pela equipa de desenvolvimento do neo4j [20].

Inicialmente, utilizou-se ambas as abordagens de modo a comparar o desempenho, sendo que o Graphenedb apresentava claramente um tempo de escrita bastante mais elevado, quando comparado com a utilização de um computador pessoal utilizando o Neo4j community edition (Versão gratuita).

O teste efetuado teve como objetivo verificar qual o tempo de escrita de todos os dados relacionados com os PDIs e rotas entre estes, sendo efetuadas três tentativas. Usando o serviço Graphenedb a conclusão do referido teste demorou entre 45 a 50 minutos. Já no caso do uso de um computador pessoal com o *software* Neo4j community edition instalado, o tempo foi entre 7 a 10 minutos.

Um dos problemas do neo4j é que este apresenta uma API REST cujos parâmetros utilizados podem ser bastante complexos, dado que utiliza objetos JSON bastante extensos e com campos fixos, obrigando a memorizar a grande maioria destes campos. Uma das soluções disponíveis para este tipo de problemas é através da utilização de *wrappers* PHP (dado que o servidor utiliza PHP). Um dos *wrappers* PHP *open source* mais conhecidos que pode ser utilizado com neo4j é o neo4jphp [21]. Este simplifica os pedidos a efetuar, na medida em que abstrai a componente dos pedidos HTTP, transformando-os em métodos que são facilmente compreensíveis. Outro aspeto que torna o neo4jphp fácil de utilizar é que este se apresenta muito bem documentado [22].

A forma como é feito o acesso do *webservice* Apache até à base de dados neo4j é através de classes cujos métodos evocam os métodos do neo4jphp de forma a obter os dados pretendidos.

Seguidamente, irá ser apresentado o esquema da base de dados Neo4j implementada, apenas para 2 PDIs e 1 utilizador a título de exemplo (Figura 20).

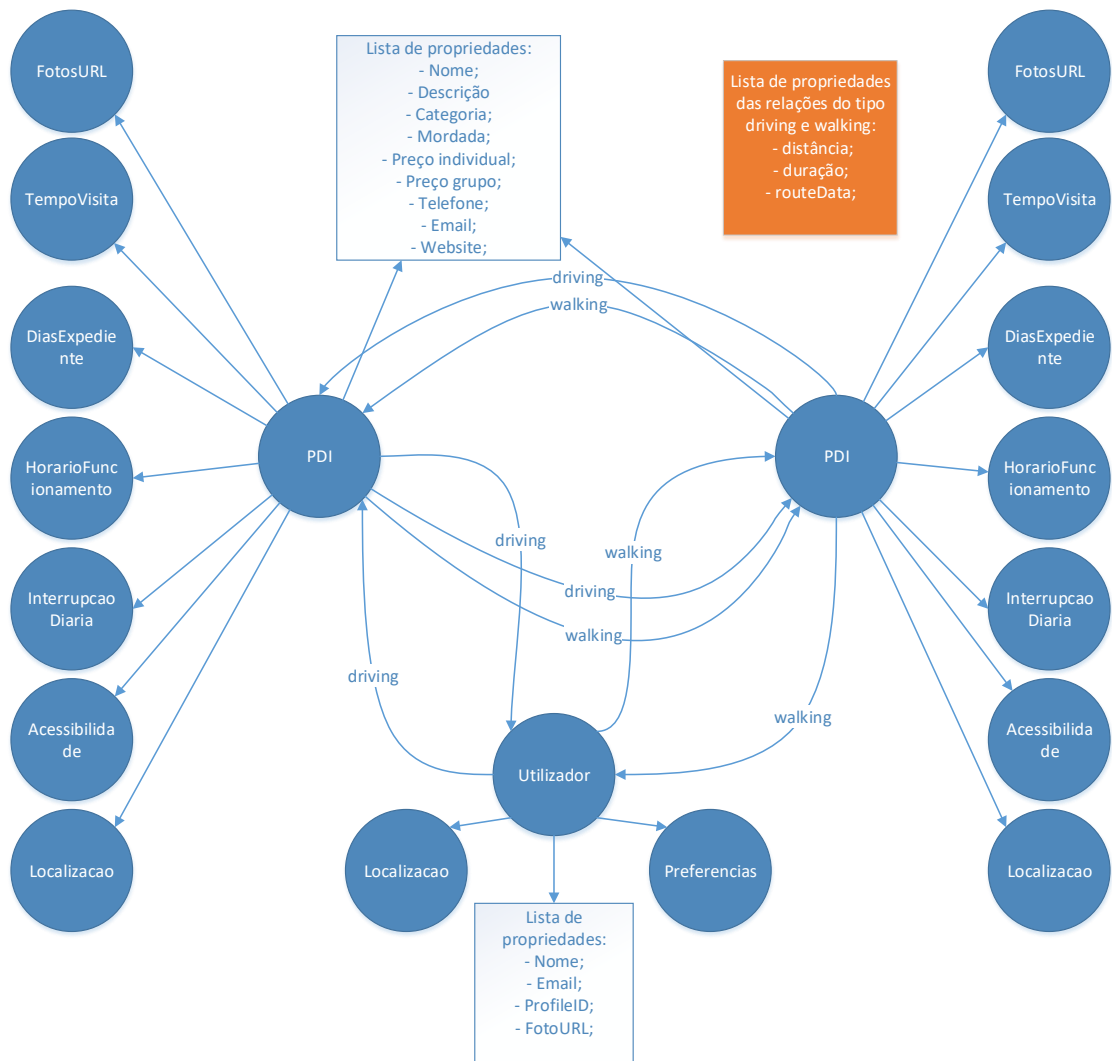


Figura 20 - Esquema da base de dados Neo4j simplificado

Apenas foram representadas as propriedades dos nós e das relações mais relevantes, sendo que as restantes propriedades foram omitidas, de modo a tornar o esquema mais legível.

4.2.3.2.3 Sistema de preenchimento da base de dados

Na primeira iteração, os dados utilizados dos ficheiros KML foram extraídos diretamente para o *smartphone*. A principal desvantagem deste método é que acaba por prejudicar o desempenho da aplicação. Nesta iteração, criou-se um programa em Java que funciona sobre a linha de comandos e cujo objetivo é extrair os dados do ficheiro KML e popular as bases de dados. Este programa também serve para obter as rotas entre PDIs e popular a base de dados com estes dados. Por fim, outra responsabilidade deste programa é a da gestão do utilizador por defeito (Utilizador que é criado quando não é feito registo dos dados pessoais ao iniciar a aplicação).

O desenvolvimento deste programa foi feito em Java de modo a permitir a reutilização do parser KML referido na primeira iteração. Foi necessário utilizar a biblioteca externa Jsoup de modo a simplificar a extração dados HTML contidos no KML [23].

Todos os objetos trocados entre o cliente e o servidor foram convertidos para o formato JSON, sendo usada a biblioteca Gson [24] para o efeito. Esta permite a conversão direta de objetos Java em JSON e vice-versa sem necessidade de ter qualquer conhecimento acerca da estrutura interna da classe desse objeto. Esta biblioteca foi também frequentemente utilizada na aplicação móvel.

Por fim, a troca de pedidos fez uso da biblioteca Apache HttpComponents [25] que simplifica a operação de criação de pedidos HTTP. Apesar de ser uma biblioteca Java apresenta-se incompatível com desenvolvimento de aplicações nativas usando o SDK Android.

4.2.3.2.4 Aplicação móvel – pedidos HTTP.

Como fora referido anteriormente, o uso da biblioteca Apache HttpComponents é incompatível com desenvolvimento de aplicações nativas Android, apenas nas suas versões mais recentes. Atualmente, os métodos recomendados passam pela utilização, ou da Classe AsyncTask [26], ou pela utilização da biblioteca android-volley [27]. Optou-se pela biblioteca android-volley devido à sua simplicidade comparativamente à utilização da Classe AsyncTask.

4.2.3.2.5 Android-maps-utils – Cluster de marcadores.

Nesta iteração, voltou-se a utilizar a biblioteca Android-maps-utils. Neste caso, utilizou-se outro módulo que permite a criação agrupamentos de marcadores (Clusters). A principal vantagem deste módulo, é na melhoria da apresentação de um grande conjunto de marcadores agrupados num espaço relativamente pequeno. Outra vantagem desta funcionalidade, é na melhoria do desempenho da aplicação, na medida em que estes marcadores são somente carregados à medida que o utilizador se vai aproximando destes.

4.2.3.3 Vistas do protótipo de alta fidelidade

Esta secção tem como objetivo apresentar as vistas desenvolvidas no protótipo de alta fidelidade e consequentemente apresentar as funcionalidades associadas às mesmas.

O protótipo de alta fidelidade produzido, como fora referido anteriormente, sofreu modificações de modo a integrar algumas das funcionalidades definidas no início da 2ª iteração. Em termos de *design*, foram feitas algumas alterações de modo a melhorar a integração com a aplicação a ser desenvolvida pela CMF.

Outro aspeto integrado nesta versão é o suporte para múltiplas línguas (Português e Inglês), sendo que a língua da aplicação varia consoante o idioma definido no equipamento do utilizador.

O protótipo produzido não apresenta a funcionalidade de criação de rotas manualmente devido às restrições temporais, sendo esta funcionalidade introduzida na 3ª iteração.

Ao iniciar a aplicação pela primeira vez é requisitada a ligação à internet para permitir a obtenção de dados do Google Maps acerca do concelho do Funchal. Inicialmente são carregados para a memória do dispositivo os dados dos PDIs que estão presentes nos recursos locais da aplicação. Posteriormente, é também carregado um conjunto de rotas por defeito entre a posição do utilizador por defeito e a localização dos PDIs. A posição do utilizador por defeito é situada no cais do Funchal, dado que um dos potenciais utilizadores são os cruzeiristas e por este ser um local que pode ser facilmente identificado. Nesta iteração apenas é possível a utilização do utilizador por defeito, sendo que este não permite a modificação das suas características.

4.2.3.3.1 Vista Principal

A vista principal (Figura 21) é uma atividade que apresenta as subvistas das opções selecionadas, sendo estas modificadas através da modificação do Fragmento visível (um fragmento é um elemento visual que permite apresentar uma subvista sem a necessidade de eliminar a vista completa para apresentar uma nova vista. Existem múltiplos fragmentos na vista principal que são mostrados/ocultados quando são escolhidas as diferentes opções disponíveis). Para aceder ao menu principal da vista inicial deve-se selecionar o botão do lado esquerdo da barra de cabeçalho localizada no canto superior do ecrã ou ao fazer o movimento *swipe* a partir do lado esquerdo do ecrã. Após isto é apresentado o menu principal, sendo este presente numa *NavigationView* (Painel deslizante lateral que apresenta uma lista de itens).

O menu principal apresenta 6 opções possíveis: PDI, Roteiros, Agenda, Definições, Dados do Utilização e Terminar Sessão. Das opções apresentadas apenas, PDI, Roteiros e Definições funcionam nesta iteração.

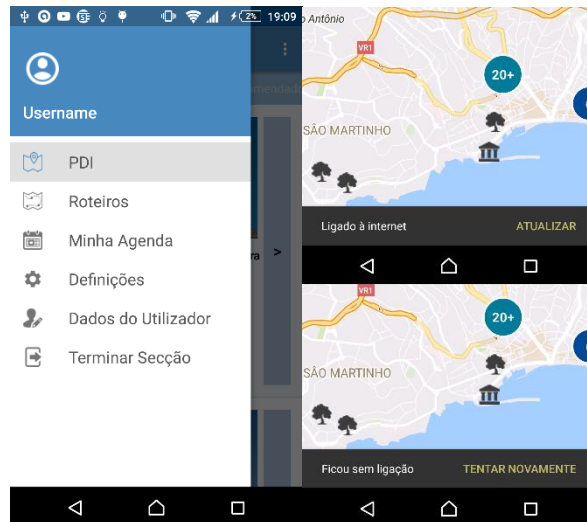


Figura 21 - Vista Principal

A vista principal, bem como todas as restantes vistas, tem a capacidade de detetar a alteração do estado de ligação à internet, sendo este apresentado através de um objeto do tipo Snackbar, permitindo também a opção de restabelecer a ligação e atualizar a vista quando visível. Dado esta funcionalidade ser partilhada por todas as vistas e de modo a otimizar o código implementado, considerou-se a criação de uma classe abstrata comum que apresenta a responsabilidade de coordenar os eventos relacionados com a modificação do estado da ligação à internet. Posteriormente, os métodos abstratos irão ser implementados na definição das respetivas atividades (subclasses).

Outro aspeto a referir na barra do canto superior da vista principal, é que esta apresenta alterações que coincidem com os tipos de fragmentos que são apresentados.

4.2.3.3.2 Vista PDIs

A vista PDIs é um fragmento carregado sobre a vista principal (Figura 22). Este fragmento apenas contém um objeto do tipo PagerTabStrip, que irá conter todos os separadores associados às diferentes opções possíveis para os PDIs.

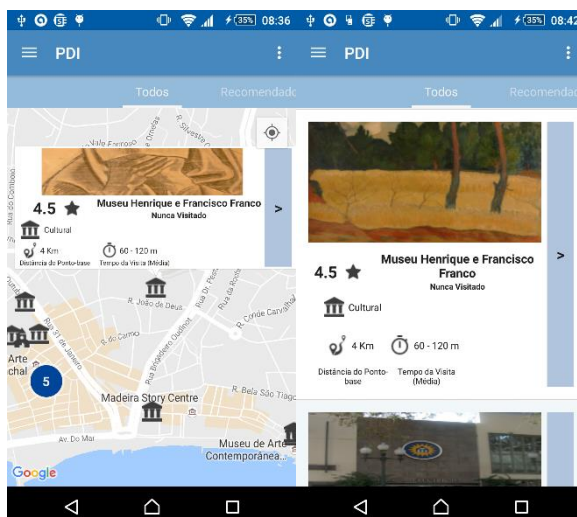


Figura 22 - Vista PDIs
(Separador Todos)

A principal vantagem deste tipo de objeto é que apresenta uma navegação simples (pode ser modificado o separador visível ao selecionar o cabeçalho para o seguinte/anterior e também pode ser mudado através do movimento *swipe*, semelhante às páginas de um livro) e permite adicionar novas funcionalidades dinamicamente, dado que só é necessário criar um novo separador.

Todos separadores do PagerTabStrip, implementam diferentes fragmentos cuja única diferença é na implementação de dois métodos que correspondem à leitura dos PDIs (pois cada separador deve ler um conjunto de PDIs diferentes, consoante a sua funcionalidade) e inicialização de outros processos dedicados a uma funcionalidade concreta de cada separador (funciona como uma *callback* que é executada durante a criação da vista do fragmento). Os restantes métodos responsáveis pela lógica da vista são partilhados entre fragmentos.

A tática utilizada para a criação dos fragmentos por cada separador (Classes) que permitem a partilha da lógica da GUI passa pela criação de um fragmento comum designado por Poi_MainFragment. Este fragmento é uma classe abstrata e contém toda a lógica relacionada com carregamento dos dados dos PDIs na GUI. Nesta classe são invocados os métodos abstratos que contêm a lógica de negócio relacionada com a obtenção dos

PDIIs pretendidos. A implementação dos métodos abstratos é efetuada durante a implementação das respetivas subclasses de Poi_MainFragment que vão popular individualmente cada separador do PagerTabStrip.

Ao selecionar a opção PDI a aplicação efetua um pedido ao servidor de modo a obter informação acerca dos PDIIs pretendidos e dos dados das rotas entre estes e o ponto-base. Caso o pedido seja bem-sucedido, os novos dados obtidos devem substituir os dados armazenados na memória do dispositivo, de modo a atualizá-los. Caso ocorra alguma falha na ligação, é mostrada uma mensagem a indicar que não existem PDIIs disponíveis ou, caso existam PDIIs e rotas que tenham sido carregados anteriormente em memória, esta informação é apresentada, permitindo assim a utilização da aplicação *offline*, sendo esta tática utilizada para as restantes funcionalidades poderem ser utilizadas *offline*.

Inicialmente, a aplicação apresenta por defeito a vista em modo mapa, podendo esta ser alternada para o modo lista nas definições. Ao selecionar um marcador no modo mapa, a informação (InfoWindow) exibida é idêntica ao modo lista. Para este protótipo apenas se implementou o separador “*Todos*” que apresenta a informação acerca de todos os PDIIs existentes no sistema sem diferenciação.

4.2.3.3.3 Vista PDI detalhes

Ao selecionar um PDI é apresentada a vista de detalhes do mesmo (Figura 23).

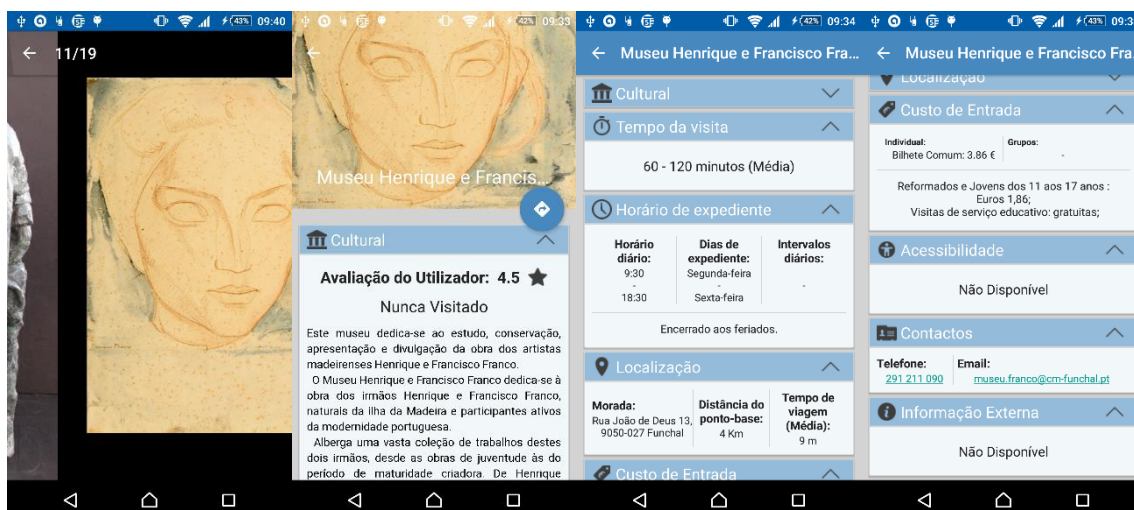


Figura 23 - Vista PDI detalhes

Na vista de detalhes do PDI, o tipo de *layout* principal foi modificado, sendo neste caso utilizado uma NestedScrollView. A vantagem deste *layout* é que permite o ajuste do conteúdo do ecrã à medida que o utilizador faz *scroll* sobre a vista.

A seleção das imagens do PDI também foi modificada. Neste caso, é possível fazer o movimento *swipe* diretamente sobre a imagem de modo a mudar para a imagem seguinte/anterior. Ao selecionar a imagem, é iniciada outra vista que apresenta a imagem em ecrã inteiro, sendo também possível navegar pelas imagens através do movimento de *swipe*.

Em termos dos dados apresentados, a estrutura do *layout* apresenta-se bastante semelhante à primeira iteração, sendo que as únicas mudanças efetuadas foram em termos de *design*.

4.2.3.3.4 Vista PDI rota

Nos detalhes do PDI ao selecionar o botão redondo no canto superior direito, a aplicação inicia a vista de rota para o PDI (Figura 24).

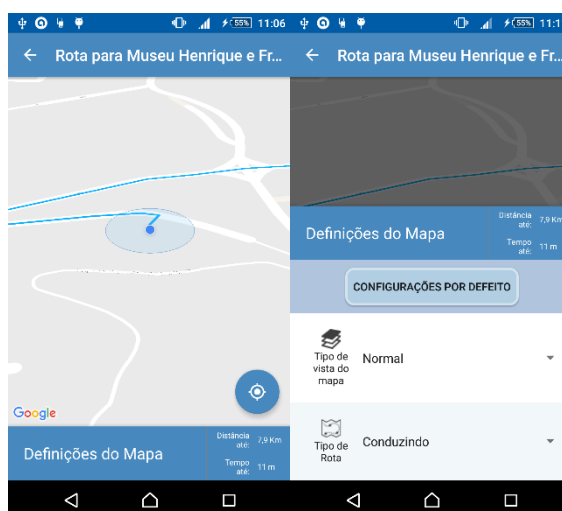


Figura 24 - Vista PDI rota

As diferenças existentes, comparativamente à 1ª iteração, é que na 1ª iteração as rotas deveriam ser obtidas exclusivamente online, sendo necessária ligação permanente à internet para existir a atualização da *polyline*. Nesta iteração, pelo facto de existirem dados das rotas na memória do dispositivo permite a utilização da funcionalidade, mesmo sem conectividade à internet.

Quando existe ligação à internet a atualização dos dados da rota é efetuada através de um novo pedido ao *webservice* do Google Maps.

Quando não há ligação à internet a rota deve ser atualizada exclusivamente ao adicionar e remover pontos à *polyline*. Adicionar pontos tem o propósito de permitir ao utilizador conhecer qual foi o caminho que este seguiu, caso se desvie da rota sugerida,

para que este possa fazer a rota no sentido inverso (embora esta rota não seja possível nos casos em que o utilizador esteja a conduzir devido às regras de trânsito) até começar a rota sugerida.

A adição de pontos é feita tendo por base o cálculo do ponto médio entre a extremidade da *polyline* e a posição atual do utilizador, sendo posteriormente criada uma extensão da *polyline* até ao ponto médio (*PtoMedio*) (8). As fórmulas utilizadas foram as seguintes [28]:

$$B_x = \cos(lat_2) * \cos(lon_1 - lon_2) \quad (4)$$

$$B_y = \cos(lat_2) * \sin(lon_1 - lon_2) \quad (5)$$

$$lat_3 = \text{atan} \left(\sin(lat_1) + \sin(lat_2), \sqrt{(\cos(lat_1) + B_x)^2 + B_y^2} \right) \quad (6)$$

$$lon_3 = lon_1 + \text{atan}(B_y, \cos(lat_1) + B_x) \quad (7)$$

$$PtoMedio = (lat_3, lon_3) \quad (8)$$

As variáveis lat_1 , lon_1 e lat_2 , lon_2 são respetivamente as coordenadas do ponto inicial e final sobre os quais se pretende calcular o ponto médio. As coordenadas do ponto médio são as variáveis lat_3 , lon_3 , sendo estes valores determinados, respetivamente, pelas fórmulas (6) e (7).

Para remoção de pontos foi calculada a distância existente entre a extremidade da *polyline* e a localização atual do utilizador, sendo que todos os pontos que apresentarem uma distância inferior a 20 metros serão removidos.

Os dados acerca da distância e do tempo de deslocação só podem ser atualizados quando existe ligação à internet.

Outro aspeto modificado foi a posição de visibilidade sobre o mapa, sendo que nesta iteração a vista segue o ponto que indica a localização atual do utilizador. Neste caso, a vista é atualizada cada vez que são recebidas novas coordenadas GPS. A direção e ângulo de inclinação da vista sobre o mapa depende da direção para onde se encontra a *polyline* relativamente à posição atual do utilizador, sendo sempre apontada nesse sentido. O botão redondo no canto inferior direito, serve para modificar a vista do mapa para os

parâmetros anteriormente identificados, permitindo assim focar na posição atual, sendo útil no caso de o utilizador navegar no mapa para um local distante do local atual.

4.2.3.3.5 Vista Roteiros

No menu principal ao selecionar a opção de roteiros personalizados, são gerados um conjunto de roteiros com PDIs selecionados aleatoriamente. Caso não haja ligação à internet, é apresentado um aviso, identificando a causa de erro com um botão que permite atualizar a vista, sendo que nesta iteração esta funcionalidade não pode utilizada *offline*.

Após a seleção do mesmo é apresentada a vista presente na Figura 25.

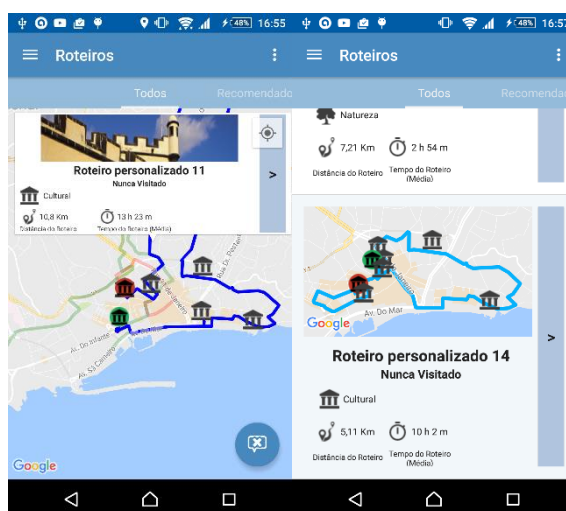


Figura 25 - Vista de roteiros

A apresentação da GUI nesta vista exibe muitas semelhanças com a vista de PDIs, sendo aqui também utilizada a PagerTabStrip com múltiplas vistas divididas por separadores. Também nesta vista o único separador disponível para esta iteração é o separador “Todos”.

Como já foi referido anteriormente, as *polylines* não apresentam qualquer tipo de interação, sendo compostas por múltiplos pontos principais unidos por uma linha. Para tornar as *polylines* interativas foi necessário fragmentá-las em múltiplos pontos situados entre cada par de pontos principais, sendo que quando o utilizador seleciona o mapa é iniciado um ciclo cujo objetivo é verificar se nas proximidades do local selecionado existe algum dos referidos subpontos, com uma distância/raio máximo de 50 metros. Seguidamente apresenta-se as fórmulas utilizadas para calcular os múltiplos pontos intermédios entre cada par de pontos principais da *polyline* [28].

$$brng = atan2(\sin(lon_2 - lon_1) * \cos(lat_2), (\cos(lat_1) * \sin(lat_2)) - (\sin(lat_1) * \cos(lat_2) * \cos(lon_2 - lon_1))) \quad (9)$$

$$lat_3 = asin((\sin(lat_1) * \cos(\frac{distPto_1}{R})) + (\cos(lat_1) * \sin(\frac{distPto_1}{R}) * \cos(brng))) \quad (10)$$

$$lon_3 = lon_1 + atan2\left(\sin(brng) * \sin(\frac{distPto_1}{R}) * \cos(lat_1), \cos(\frac{distPto_1}{R}) - (\sin(lat_1) * \cos(lat_2))\right) \quad (11)$$

$$PtoIntermedio = (lat_3, lon_3) \quad (12)$$

As variáveis lat_1, lon_1 e lat_2, lon_2 são respetivamente as coordenadas do ponto inicial e final sobre os quais se pretende calcular um ponto intermédio. As coordenadas do ponto intermédio (12) são as variáveis lat_3, lon_3 , sendo estes valores determinados, respetivamente, pelas fórmulas (10) e (11). A variável $brng$ (9) indica o valor do Bearing inicial, ou seja, o ângulo associado à direção do movimento de um corpo entre 2 pontos relativamente ao Polo norte geográfico, seguindo a trajetória mais curta possível e tendo em conta a curvatura da terra, no momento em que o corpo se encontra no ponto inicial (lat_1, lon_1). A variável $distPto_1$ indica a distância do ponto pretendido em relação ao ponto inicial e a variável R indica o raio da terra.

Quando nenhuma das *polylines* está selecionada, é apenas apresentado o PDI inicial (fundo verde) e final (vermelho) de cada um dos roteiros, sendo que as cores das *polylines* apresentam um igual nível de transparência. Quando uma *polyline* é selecionada, todos os marcadores das restantes *polylines* (roteiros) são ocultados, sendo que estas modificam o nível de transparência (mais transparência), de modo a facilitar a identificação da rota selecionada. Na *polyline* do roteiro selecionado são apresentados todos os marcadores dos PDIs desse mesmo roteiro.

Quando uma *polyline* é selecionada, é executado o evento que apresenta a janela informativa do roteiro associado à *polyline*. Por fim, o botão redondo no canto inferior

direito que pode ser visualizado na Figura 25, serve para ocultar/mostrar a janela informativa do roteiro/PDI, de modo a facilitar a visualização do mapa, pois a janela informativa ocupa um espaço significativo do ecrã principalmente em *smartphones* onde o tamanho do ecrã é reduzido.

4.2.3.3.6 Vista roteiro detalhes

Ao iniciar a vista de detalhes de um roteiro é apresentada a vista presente na Figura 26.



Figura 26 - Vista roteiros detalhes

A vista de detalhes dos roteiros em termos de apresentação da GUI é um pouco semelhante à de detalhes do PDI, sendo que o tipo de modificações que esta sofreu da 1ª para a 2ª iteração foram modificações idênticas às referidas na vista de detalhes dos PDIs. Tal como a vista de detalhes do PDI, é possível mudar a fotografia visível ao efetuar o movimento de *swipe* sobre a foto na parte superior da vista, sendo que ao selecionar a foto, esta apresenta outra vista em modo de ecrã completo. Outro aspeto a referenciar é que os PDIs do roteiro podem ser acedidos ao selecionar um item da lista de PDIs.

4.2.3.3.7 Vista roteiro rota

Nos detalhes do roteiro ao selecionar o botão redondo no canto superior direito, a aplicação inicia a vista de rota para os PDIs do roteiro (Figura 27).

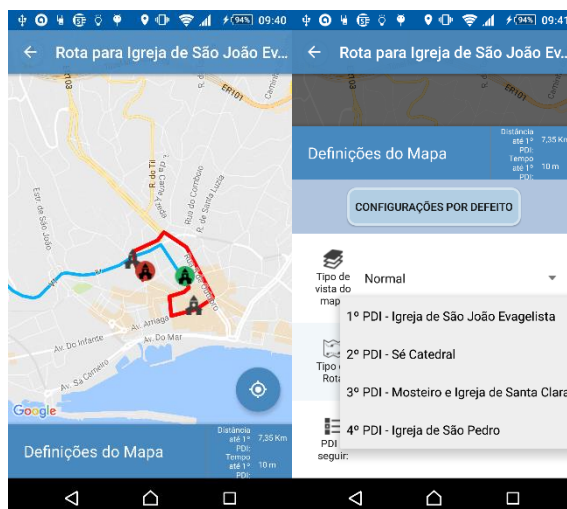


Figura 27 - Vista roteiro rota

A vista de rota para os PDIs do roteiro em termos de apresentação da GUI é um pouco semelhante à vista da rota para PDI simples, sendo que todas as alterações efetuadas acerca da atualização da *polyline* quando não há ligação à internet e acerca da vista sobre o mapa também aqui foram aplicadas. Tal como foi referido anteriormente, nesta iteração não é possível a utilização das funcionalidades dos roteiros *offline*.

4.2.3.3.8 Vista definições

No menu principal, é possível selecionar as definições (Figura 28).

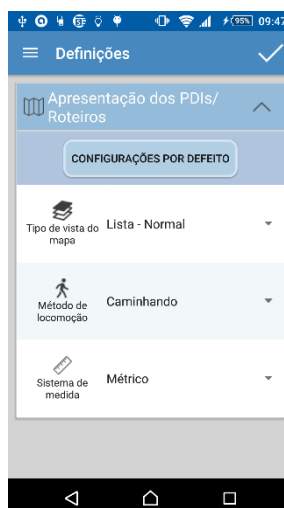


Figura 28 - Vista das definições

Esta vista, permite a modificação de um conjunto de aspetos gerais da aplicação. Um dos aspetos que as definições permitem mudar é o tipo de vista pretendido, podendo ser alternada entre o modo mapa e lista, que foram anteriormente apresentados, sendo

também possível escolher o tipo de mapas visíveis nessas vistas (Normal, Satélite, Terreno). Outro aspeto que pode ser modificado é o tipo de locomoção pretendido (caminhando ou conduzindo), sendo que isto tem influência nas distâncias e tempos de viagem apresentados e consequentemente nas rotas sugeridas. Por fim, outro aspeto personalizável é o sistema de medida utilizado, de modo a modificar o tipo de unidades de distância apresentadas. Em relação, às definições relacionadas com a língua a apresentar não se achou conveniente colocar nesta secção, pois a escolha da língua é feita de forma automática pela aplicação consoante o idioma do OS Android utilizado.

4.2.4 Conclusão

No final da 2ª iteração, apresentou-se a aplicação ao encarregado pelo *design* da aplicação a ser desenvolvida pela CMF, sendo que este forneceu quais as guidelines que deveriam ser seguidas, de modo a ter uma aplicação com dimensões, tipos de letra, cor e estilo de apresentação semelhantes. Também foi aconselhado seguir as guidelines sugeridas pela Google para o *design* de aplicações para Android [30]. Por fim, foi fornecido um *Launcher icon* que é apresentado no menu das aplicações do OS Android.

No final desta iteração também se produziu um questionário para permitir efetuar testes de utilizador em relação ao protótipo desenvolvido, sendo que este tema foi abordado em maior detalhe na *secção 5.1*.

4.3 Terceira Iteração

A 3ª iteração foi a última iteração implementada. Esta iteração teve como objetivo a continuação e melhoria de diversos aspetos identificados na 2ª iteração. Também como objetivo pretende-se no final desta iteração, efetuar diversos testes à aplicação e às bases de dados utilizadas de modo a analisar o desempenho do sistema final obtido.

4.3.1 Modificações efetuadas

Uma das alterações efetuadas foi em termos de *design* sendo que, para esta iteração, foram seguidas algumas das guidelines sugeridas pela Google para o *design* de aplicações para Android [30]. Estas recomendações incluem aspetos relacionados com as dimensões/proporções que cada elemento deve ocupar no ecrã, dimensão do espaçamento entre itens, etc.

Outro aspeto seguido, em termos de *design*, foram um conjunto de guidelines definidas pela CMF, sendo que estas integram elementos relacionados com a cor, formato do texto, tipo de *layout*, etc. Este último conjunto de guidelines apresenta o problema de ser demasiado específico, pois são focadas na aplicação a ser desenvolvida pela CMF, sendo que para diversos tipos de *layout* usados na aplicação desenvolvida não apresentava qualquer tipo de informação.

Por fim, introduziu-se algumas modificações na aplicação de modo a que esta se torne compatível com todas as versões do Android até à versão 4.0.

4.3.2 Protótipo de alta fidelidade

Seguidamente ir-se-á apresentar os aspetos tidos em conta para o desenvolvimento do protótipo de alta fidelidade, sendo descrito como foi implementada a base de dados MySQL e o padrão de desenho “*Bridge*”. Também pretende-se descrever como foi encriptada a comunicação cliente-servidor. Por fim, irão ser apresentadas as diferentes vistas do protótipo de alta fidelidade e suas funcionalidades.

4.3.2.1 Implementação da base de dados MySQL e Implementação do padrão de desenho “*Bridge*”

Um dos aspetos requeridos para esta iteração foi a implementação de uma base de dados MySQL que apresente um conjunto de estruturas equivalentes à base de dados Neo4j implementada. Apesar da base de dados MySQL ser bastante diferente da base de dados Neo4j, em termos de estruturas apresentadas, existe uma equivalência entre ambas (Tabela 9).

Mysql	Neo4j
Tabelas Principais	Etiquetas atribuídas aos nós, Relações entre nós com propriedades
Tuplo	Nó
Tabelas intermédias	Relações entre nós sem propriedades
Nome das Colunas	Propriedades

Tabela 9 - Comparação entre elementos das BDs MySQL e Neo4j

Para a implementação desta BD foi utilizado o *software* MySQL Workbench [31].

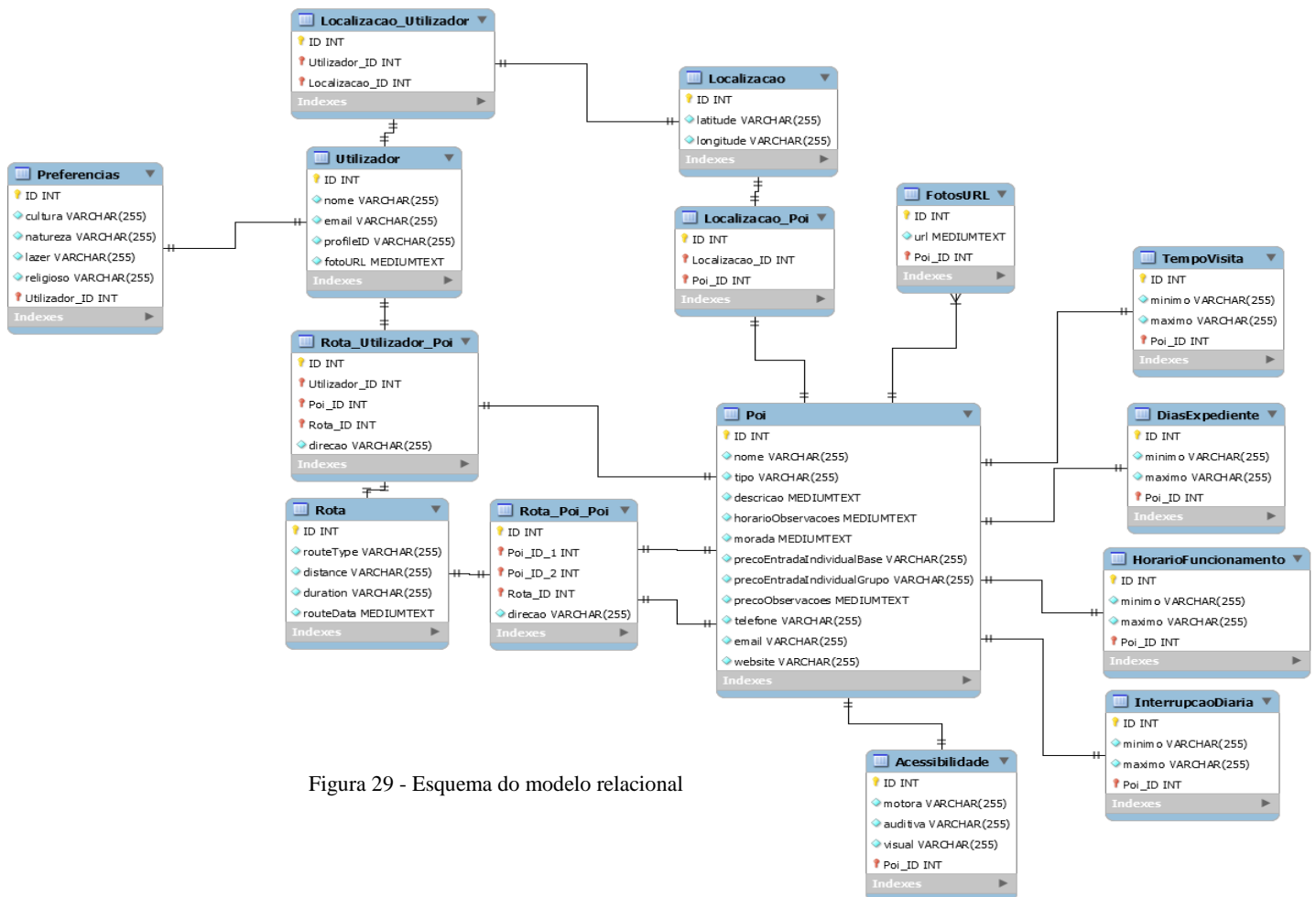


Figura 29 - Esquema do modelo relacional

Após a implementação da base de dados foi necessário alterar a lógica do servidor de modo a permitir a modificação da lógica de negócio independentemente da base de dados utilizada. Para isto foi necessário utilizar o padrão de desenho de *software* *Bridge*. Por definição o padrão *Bridge* indica que “*Decouple an abstraction from its implementation so that the two can vary independently.*” [32]. Neste caso, a abstração são as interfaces que fazem acesso aos objetos que permitem o acesso concreto a cada base de dados. O diagrama de classes presente na *Figura 30* representa a estrutura de classes implementadas no servidor. Dada a extensão do diagrama optou-se apenas por representar as classes do sistema, ocultando os detalhes acerca dos atributos e métodos.

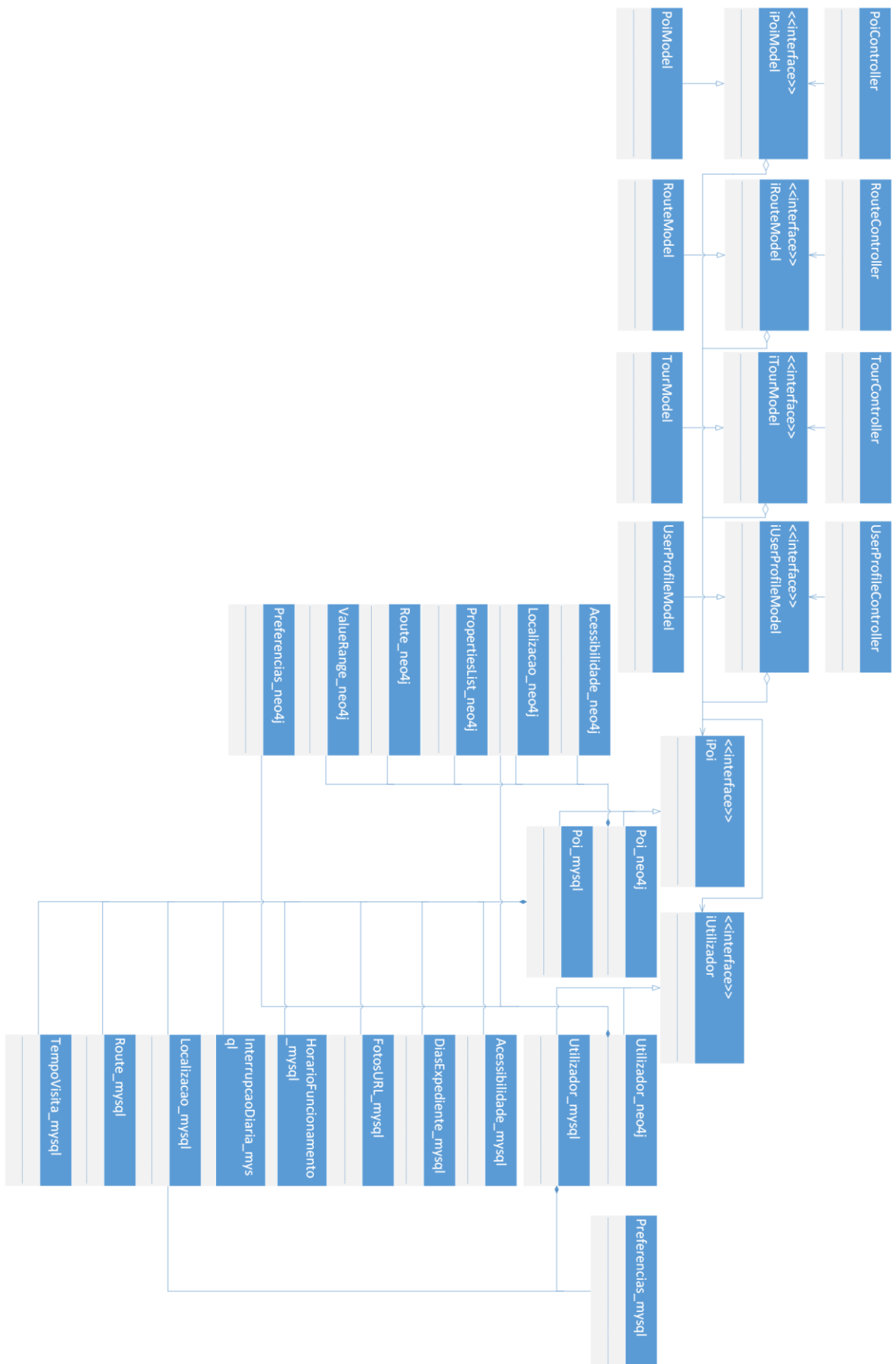


Figura 30 - Diagrama de classes do servidor

4.3.2.2 Encriptação dos dados

Outro aspeto introduzido nesta iteração foi a encriptação de dados. Durante as definições dos requisitos foi estabelecido que idealmente deveria ser usado o protocolo HTTPS, permitindo assim uso de criptografia durante a troca de mensagens com o servidor, sendo este um protocolo criptográfico recomendado para aplicações em Android que façam uso de *webservices* [33].

Dado os custos associados à aquisição de um certificado, optou-se por outra solução recomendada como alternativa ao uso do protocolo SSL. Essa solução consiste na utilização do *Advanced Encryption Standard* (AES) de modo a encriptar as mensagens em blocos de 128 bits. Nesta situação, é utilizada uma chave simétrica aleatória, que é encriptada e enviada juntamente com todas as mensagens trocadas, de modo a diminuir a probabilidade de gerar um padrão que seja detetável por um ataque do tipo *man-in-the-middle* durante a troca de mensagens. Apesar deste método apresentar um bom nível de segurança, ainda assim apresenta algum tipo de vulnerabilidade, pois caso o intermediário descubra qual a técnica de criptografia utilizada, qual a secção da mensagem que apresenta a chave e como descriptá-la, poderá ter acesso a toda a mensagem. A maior desvantagem da utilização deste tipo de mecanismos de segurança é no desempenho da aplicação, principalmente em situações em que exista elevada carga de pedidos.

4.3.2.3 Vistas do protótipo de alta fidelidade

Esta secção tem como objetivo apresentar as vistas desenvolvidas no protótipo de alta fidelidade e conseqüentemente apresentar as funcionalidades associadas às mesmas.

O protótipo de alta fidelidade, sofreu diversas alterações de modo a integrar as guidelines definidas pela Google para *design* em Android [30].

Em termos de funcionalidades, este protótipo apresenta todas as funcionalidades, indicadas na secção dos requisitos funcionais. Algumas das funcionalidades definidas nos protótipos anteriores foram removidas nesta fase, devido aos limites temporais apresentados e também porque estas apresentavam um baixo nível de prioridade, comparativamente aos objetivos inicialmente designados.

4.3.2.3.1 Vista do registo

Ao iniciar a aplicação pela primeira vez são disponibilizadas 2 opções (Figura 31). A primeira opção serve para efetuar um registo através do Google Plus. A segunda opção serve para ignorar o registo, sendo ativado o utilizador por defeito.

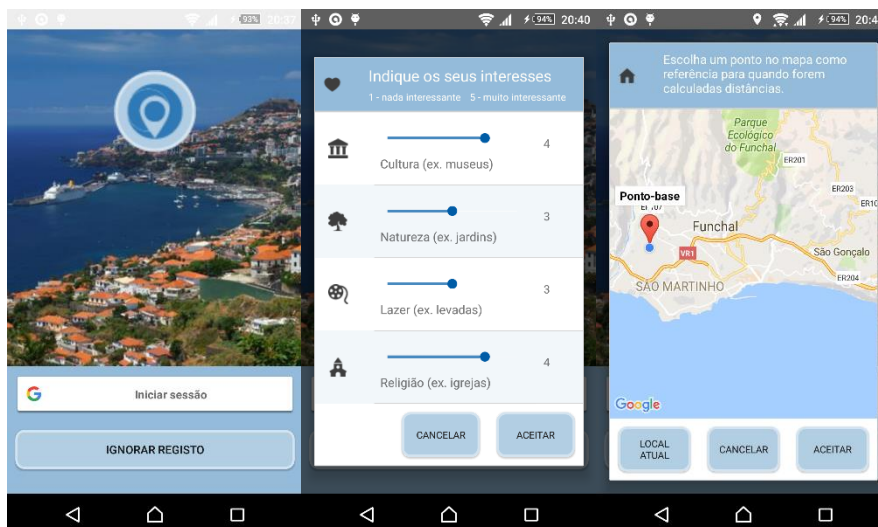


Figura 31 - Vista de Registo

Optou-se por apenas disponibilizar o registo do Google Plus, dado que este tipo de conta é um requisito obrigatório para a utilização do OS Android, daí ser um tipo de conta que todos os utilizadores possuem. Este tipo de registo torna o processo de inicialização bastante mais rápido em relação a um formulário de registo convencional, na medida em que, em vez de requisitar um conjunto de alargado de dados, faz uso direto dos dados do Google Plus. Após a seleção desta opção é pedida a introdução dos interesses do utilizador e do ponto-base. O formulário dos interesses do utilizador é composto por um curto questionário que usa a escala de Likert (1 a 5) em todas as questões. A seleção do ponto-base é feita sobre o Google Maps sendo restringida a área da ilha da Madeira. Após o preenchimento dos dados iniciais, é iniciada a obtenção de todas as rotas possíveis entre o ponto-base e os PDIs em ambos os sentidos (ida e volta). Com a conclusão deste processo é inicializada a vista principal da aplicação.

Caso seja selecionada a opção ignorar registo, é inicializado o processo de carregamento dos roteiros existentes nos recursos da aplicação para a memória do dispositivo. Terminado este processo, é inicializada a vista principal.

As principais diferenças entre o registo do utilizador e ignorar registo, é no tipo de preferências do utilizador e no tipo de dados acerca da distância do ponto-base. Na opção ignorar registo, é selecionado o utilizador por defeito, cujo ponto-base é o cais do

Funchal (Pontinha). Nesta situação, os interesses do utilizador apresentam o nível máximo de interesse (5 – muito interessante), sendo que o tipo de recomendações obtidas tem apenas em conta a distância e o tempo de deslocação até ao PDI.

4.3.2.3.2 Vista Principal

A vista principal (Figura 32) apresenta um *design* bastante semelhante ao protótipo de apresentado na 2ª iteração, sendo que nesta situação todas as opções do menu principal se apresentam funcionais. Neste menu é apresentada também a opção de iniciar a sessão com o Google Plus, quando o utilizador não iniciou previamente a sessão sendo esta opção apresentada no canto superior do menu principal.

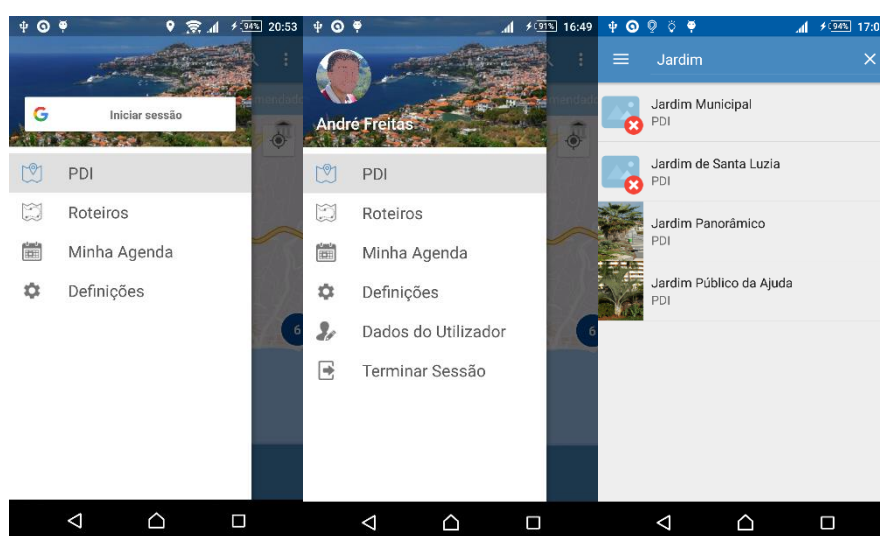


Figura 32 - Vista Principal

Para além das funcionalidades identificadas na 2ª iteração, esta vista também apresenta a funcionalidade de pesquisa, sendo nesta situação criado um *layout* que é modificado consoante o tipo de texto introduzido após ser selecionado o ícone da lupa. Esta funcionalidade apresenta uma lista de itens que contém os PDIs e os roteiros e é disponibilizada independentemente da opção selecionada no menu principal.

4.3.2.3.3 Vista PDIs

A vista de PDIs (Figura 33) sofreu alterações menores em termos da iconografia, tipografia e disposição do *layout*. Neste caso, todos os separadores apresentam-se funcionais.

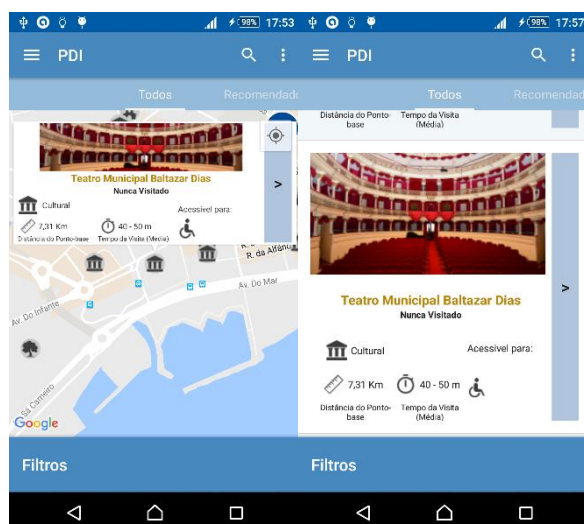


Figura 33 - Vista PDIs

As vistas entre separadores são idênticas, sendo que o histórico é a única vista que apresenta alguma diferenciação. Em termos de funcionalidades, os separadores apresentam as seguintes funcionalidades:

- Todos – Apresenta um conjunto de PDIs de forma indiferenciada;
- Recomendados – Apresenta um conjunto de PDIs com uma compatibilidade com o utilizador de 75 %, no mínimo. O valor da percentagem da compatibilidade com o utilizador é obtido tendo em conta:
 - Interesses do utilizador – Valor máximo 40 % na compatibilidade;
 - Distância do ponto-base – Valor máximo 30 % na compatibilidade;
 - Custo da entrada – Valor máximo 30 % na compatibilidade;
- Favoritos – Apresenta o conjunto de PDIs identificados como favoritos, sendo que estes também podem ser identificados em outras vistas pelo nome do PDI com uma coloração dourada;
- Histórico – Apresenta o conjunto de PDIs visitados anteriormente (Figura 34), sendo que nesta secção os PDIs podem ser filtrados por intervalo de datas. Esta opção pode ser modificada ao seleccionar o botão circular azul no canto inferior direito;



Figura 34 - Separador Histórico dos PDIs

Outro aspeto introduzido nesta iteração são os filtros, sendo que estes permitem “filtrar” o conteúdo apresentado em todos os separadores da vista de PDIs, consoante o *input* do utilizador. O formulário dos filtros pode ser apresentado ao seleccionar a barra na parte inferior da vista de PDIs, identificada com o texto Filtros (Figura 35).

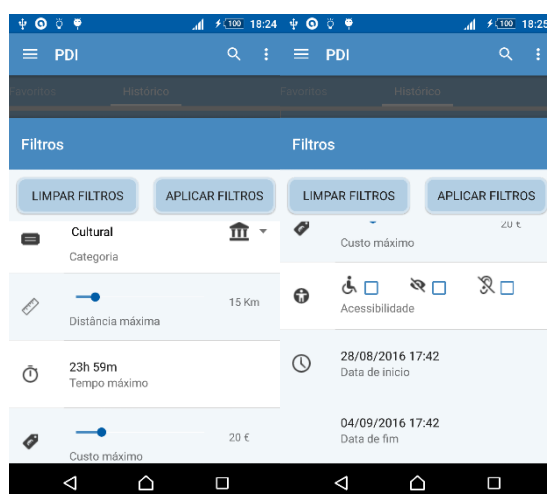


Figura 35 - Filtros para os PDIs

O tipo de filtros apresentados está associado aos parâmetros dos PDIs. A opção limpar filtros repõe os valores dos filtros para valores que permitem visualizar todos os PDIs disponíveis. A opção de aplicar filtros atualiza a vista seleccionando apenas os PDIs que apresentam dados que coincidem com os filtros seleccionados. O processo de filtragem dos PDIs tem em conta os dados apresentados na secção de detalhes dos PDIs, sendo aplicadas um conjunto de condições de modo a garantir que os PDIs seleccionados se apresentam dentro dos parâmetros inseridos pelo utilizador.

4.3.2.3.4 Vista PDI detalhes

A vista de detalhes de um PDI (Figura 36) nesta iteração, apesar de manter o mesmo tipo de dados relativamente às iterações anteriores, foi modificada em termos de *design* de modo a coincidir com tipo de *design* sugerido pela Google para uma vista de detalhes [30]. Neste caso, manteve-se a NestedScrollView, mas alterou-se a forma de apresentação de cada campo do PDI.

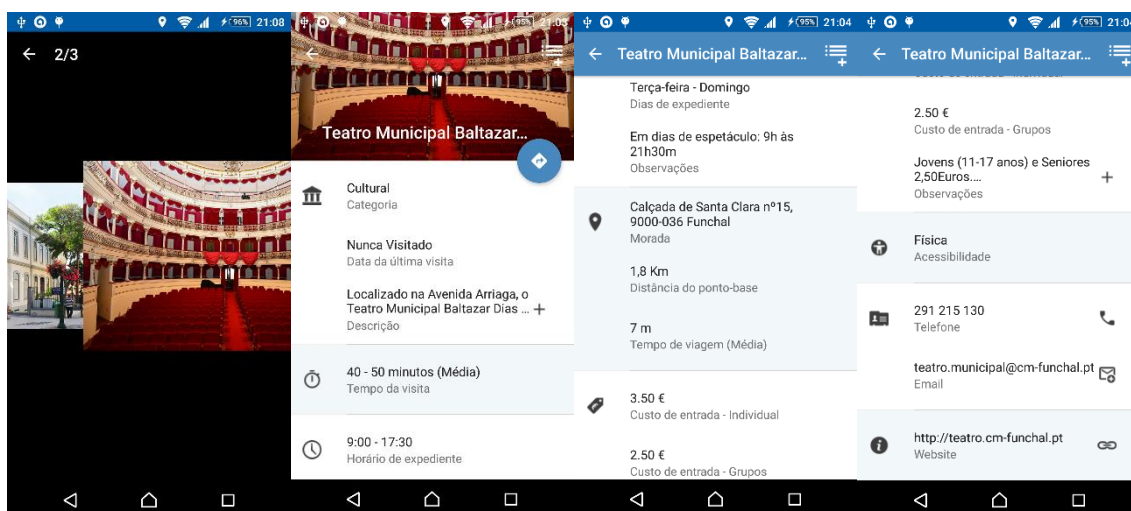


Figura 36 - Vista de detalhes do PDI

Nesta vista, cada campo de texto que ocupe um espaço superior a 2 linhas pode ser expandido através de uma caixa de diálogo que é mostrada ao selecionar o respetivo campo.

Ao selecionar o ícone no canto superior direito é apresentado um conjunto de opções em relação a este PDI:

- Adicionar/Remover dos favoritos – Ao selecionar esta opção o PDI é adicionado aos favoritos, sendo que o nome do PDI muda para uma cor dourada e é exibida uma mensagem a notificar deste evento;
- Adicionar à agenda – Ao selecionar esta opção é exibido um seletor de data e hora para identificar qual a data e hora pretendidas para iniciar a visita ao PDI;
- Adicionar aos roteiros – É exibida uma lista de roteiros sobre os quais o PDI pode ser adicionado, sendo que após a seleção do roteiro é iniciada a vista de criação/edição de roteiros;

Outro aspeto a realçar acerca desta vista é a possibilidade de poder interagir diretamente com os contactos (telefone e email) e informação externa (website) ao selecionar o respetivo *layout*.

4.3.2.3.5 Vista PDI rota

A vista da rota para o PDI (Figura 37) apresenta algumas mudanças comparativamente à 2ª iteração, sendo a maioria destas em termos de *design*.

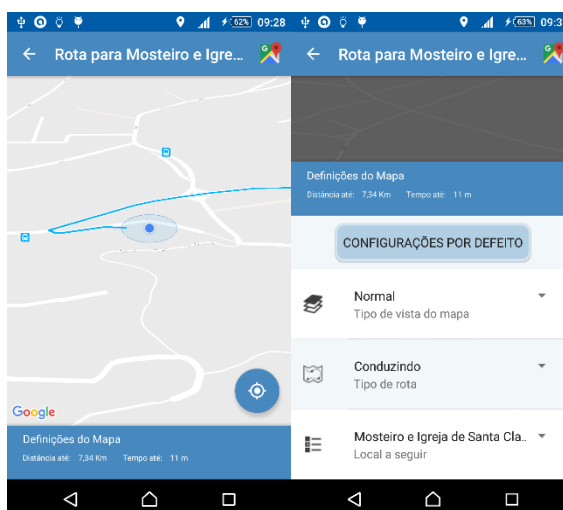


Figura 37 - Vista PDI rota

Uma das funcionalidades introduzidas é a opção da utilização da aplicação do Google Maps como alternativa a esta aplicação. O Google Maps pode ser iniciado ao selecionar o ícone no canto superior direito.

Outro aspeto introduzido é o seletor do local a seguir, sendo possível criar uma rota desde o ponto-base até ao PDI e vice-versa. Anteriormente apenas era possível obter a rota de ida para o PDI.

4.3.2.3.6 Vista Roteiros

A vista de roteiros (Figura 38) sofreu alterações menores em termos da iconografia, tipografia e disposição do *layout*. Neste caso, todos os separadores apresentam-se funcionais.

É possível a utilização da maioria das funcionalidades associadas aos roteiros sem depender de ligação à internet. Apenas as recomendações requerem obrigatoriamente ligação à internet para serem geradas novas recomendações. Caso não haja ligação, são carregadas as últimas recomendações apresentadas quando existia ligação à internet.

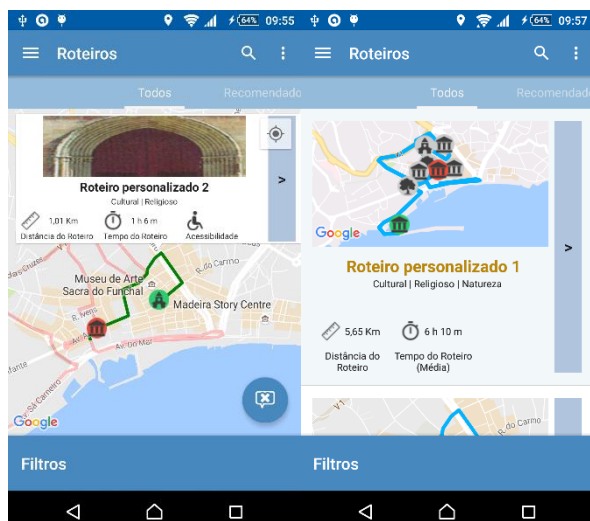


Figura 38 - Vista Roteiros

Tal como na vista de PDIs, as vistas entre separadores são idênticas. Em termos de funcionalidades, os separadores apresentam as seguintes funcionalidades:

- Todos – Apresenta um conjunto de roteiros de forma indiferenciada;
- Recomendados – São criados 3 roteiros de forma automática, sendo estes adequados para as próximas 24 horas. O primeiro roteiro apresentado é um roteiro de curta duração, que deve durar, no máximo, 2 horas e apresentar, no mínimo, 2 PDIs. O segundo roteiro é um roteiro de média duração, que deve durar, no máximo, 4 horas e apresentar, no mínimo, 4 PDIs. Por fim, o terceiro roteiro é um roteiro de longa duração que apresenta uma duração máxima de 8 horas e apresenta, no mínimo, 8 PDIs. A forma como são criados os roteiros automaticamente irá ser descrita na vista de criação de roteiros.
- Favoritos – Apresenta o conjunto de roteiros identificados como favoritos, sendo que estes também podem ser identificados em outras vistas pelo nome do roteiro com uma coloração dourada (igual aos PDIs);

Outro aspeto introduzido nesta iteração são os filtros, sendo que estes apresentam um *design* e funcionalidade idêntica aos filtros dos PDIs, mas com condições de filtragem adaptadas ao tipo de dados apresentados pelos detalhes dos roteiros.

4.3.2.3.7 Vista roteiro detalhes

Na vista de detalhes dos roteiros (Figura 39) efetuou-se modificações de *design* idênticas às referidas na vista de detalhes do PDI.

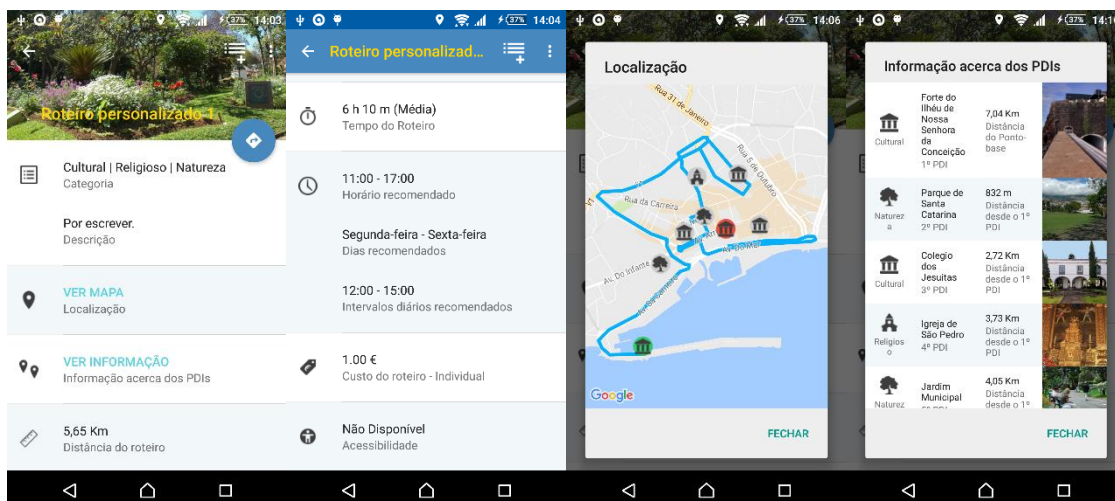


Figura 39 - Vista de detalhes do roteiro

Em termos de funcionalidades, a vista de detalhes do roteiro é bastante semelhante à vista de detalhes do PDI, na medida em que apresenta o ícone no canto superior direito do ecrã que permite adicionar/remover dos favoritos e adicionar à agenda. Outras opções apresentadas nesta vista são editar e apagar o roteiro. Editar o roteiro inicia a vista de criação de roteiro preenchida com os dados do atual roteiro. Apagar o roteiro remove todas as referências dos favoritos e da agenda ao atual roteiro sendo por fim fechada a atual vista de detalhes do roteiro.

4.3.2.3.8 Vista roteiro rota

Na vista de rota até aos PDIs dos roteiros (Figura 40), todas as modificações em termos de *design* e funcionalidades foram idênticas às apresentadas na vista de rota até a um PDI. Neste caso, optou-se por dar maior transparência à rota do roteiro de modo a realçar a atual rota a ser percorrida pelo utilizador (*polyline* azul).

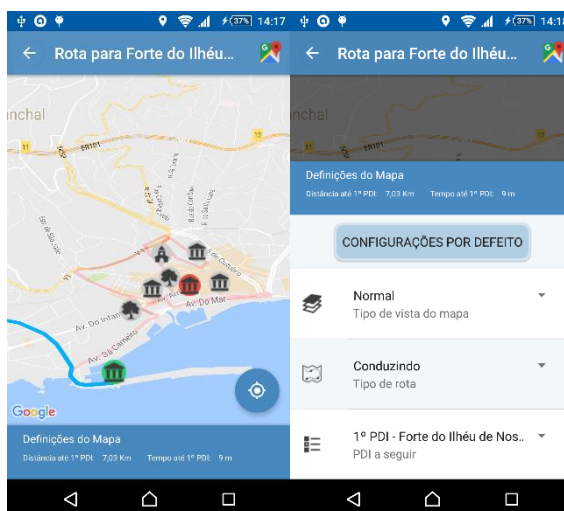


Figura 40 - Vista de rota para os PDIs do roteiro

4.3.2.3.9 Vista de criação de roteiros

Na vista geral dos roteiros (Figura 38), ao seleccionar o ícone no canto superior direito, representado por 3 pontos, é possível seleccionar a funcionalidade de criação de um novo roteiro (Figura 41).

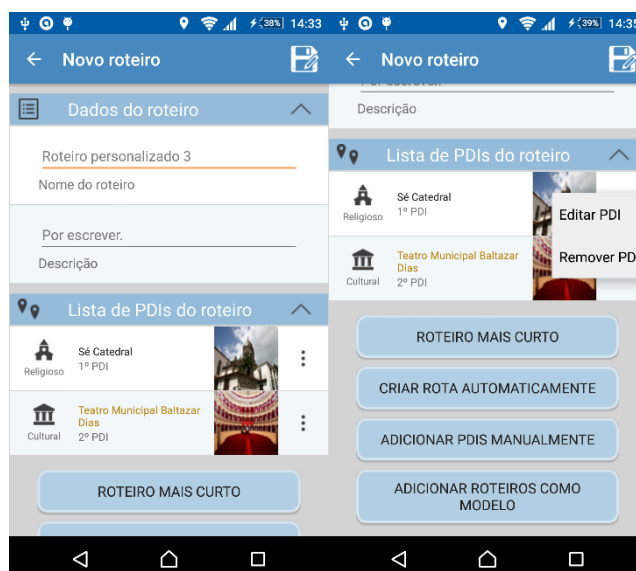


Figura 41 - Vista de criação de roteiro

A mesma vista é também utilizada na edição do roteiro, sendo que as funcionalidades são idênticas para ambas as secções. A vista de criação de roteiros apresenta dois *inputs* de texto, o nome e descrição do roteiro. Contrariamente à 1ª iteração, em vez de se apresentar avisos acerca do limite e tipo de caracteres, optou-se simplesmente por restringir o *input* de texto, quando é realizada uma operação inválida. Nesta vista, é possível alterar a ordem dos PDIs no roteiro (opção editar) e remover os PDIs do roteiro.

Para adicionar novos PDIs ao roteiro deve ser seleccionada a opção adicionar PDIs manualmente, sendo exibida outra vista que apresenta a lista de PDIs do presente roteiro e de todos os PDIs existentes no sistema de forma indiferenciada (Figura 42). Ao seleccionar a opção adicionar PDI, é apresentado um diálogo de modo a indicar a posição para inserir o novo PDI, sendo os restantes PDIs movidos de posição de modo a permitir a inserção deste PDI.

Outra opção possível é adicionar roteiros como modelo (Figura 42). Esta opção permite copiar todos os PDIs de um roteiro existente para o atual roteiro. O funcionamento é bastante semelhante ao de adicionar PDIs manualmente, sendo que nesta situação são disponibilizadas duas listas de roteiros (roteiros personalizados e recomendados) divididas por separadores. Após ser seleccionada a opção de adicionar o

roteiro todos os PDIs são colocados de forma consecutiva a partir de uma posição identificada entre os PDIs adicionados anteriormente, sendo que estes irão ser deslocados de modo a permitir a inserção de PDIs.



Figura 42 - Vistas adicionar PDIs e roteiros

Na vista de criação de roteiros, existe outra funcionalidade que é otimização do percurso gerado, cujo objetivo é criar o caminho mais curto possível entre PDIs, sendo estes reorganizados para o efeito. Para a realização deste tipo de operação é requerida a ligação à internet, sendo que esta função não se apresenta disponível sem ligação ao servidor, dada a sua complexidade e quantidade de dados requeridos para o cálculo deste percurso. Esta função requer criação de um array com todas as permutações de percursos possíveis entre PDIs, sem calcular os valores da distância entre estes. Após isto são calculados para cada par de PDIs de cada percurso possível as respetivas distâncias. Caso esta distância ultrapasse o valor mínimo conhecido, o cálculo da distância dos próximos PDIs é cancelado e é iniciada a iteração seguinte para o próximo percurso possível. Um aspeto importante durante este processo é manter os dados obtidos das *queries* anteriores em memória e pesquisar inicialmente se o valor da distância pretendida foi calculado anteriormente e está presente em memória, antes de efetuar uma nova pesquisa à base de dados, sendo que isto aumenta o desempenho desta funcionalidade. Existem restrições acerca do número de PDIs (no mínimo 2 e no máximo 8 PDIs) que podem ser otimizados, devido aos elevados tempos de conclusão da operação para um elevado número de PDIs.

Outra funcionalidade disponibilizada na criação de um roteiro é a criação automática de um roteiro (Figura 43). Esta funcionalidade é semelhante à recomendação de roteiros, só que nesta situação o utilizador pode modificar um conjunto significativo

de parâmetros de modo a obter um roteiro mais adequado às suas necessidades. A utilização desta funcionalidade, remove quaisquer PDIs que tenham sido adicionados anteriormente, sendo o utilizador alertado para esta situação caso seleccione esta opção e existam PDIs adicionados ao roteiro.

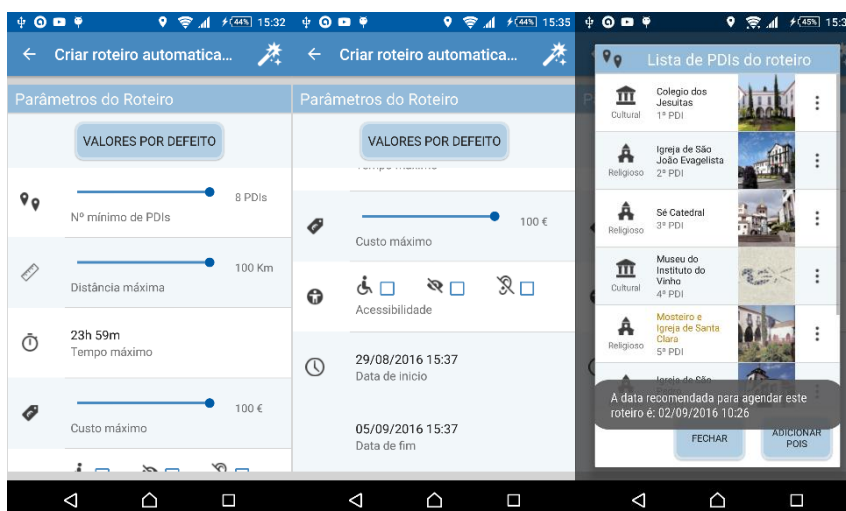


Figura 43 - Vista de criação automática de roteiros

Apesar dos parâmetros serem semelhantes aos filtros, o funcionamento é bastante diferente destes. Nesta situação, inicialmente são seleccionados os PDIs que correspondem aos parâmetros introduzidos. Posteriormente os PDIs são organizados por disponibilidade por semanas que compõem o intervalo de tempo seleccionado, sendo identificados também o dia e hora de disponibilidade diária. De seguida, são removidos os dias que não cumpram as restrições acerca do número mínimo de PDIs. Posteriormente, são calculadas as percentagens de compatibilidade para cada PDI ainda disponível no conjunto de dias seleccionado. Seguidamente, é seleccionado o dia que apresenta o ganho médio dos PDIs com a compatibilidade mais elevada. Após isto são criados, por iterações, múltiplos roteiros otimizados, sendo que em cada iteração é adicionado um novo PDI ao roteiro (é seleccionado o PDI que apresenta a maior percentagem de compatibilidade com o utilizador para o dia seleccionado). Os PDIs vão sendo removidos do dia seleccionado à medida que são adicionados no roteiro ou permitam a criação de um roteiro que não satisfaça os parâmetros introduzidos pelo utilizador. Por fim, a criação do roteiro é terminada quando já não existam mais PDIs por adicionar. O cálculo da data e hora recomendadas é feito de modo a garantir a existência do maior número possível de PDIs abertos ao público. Esta data e hora recomendadas são posteriormente apresentadas numa opção que permite o agendamento do novo roteiro antes deste ser guardado.

Por fim, na vista de criação de roteiros é possível guardar e agendar o roteiro criado/modificado ao selecionar a disquete no canto superior direito.

Das operações referidas anteriormente, as de otimização do roteiro, criação automática de roteiros e guardar roteiro requerem ligação à internet, não existindo a possibilidade de utilização *offline*. O motivo desta decisão é que as referidas funcionalidades requerem um conjunto bastante alargado de dados que foram armazenados do lado do servidor, de modo a não sobrecarregar a memória do dispositivo.

Em todas as listas de itens apresentadas (roteiros e PDIs) ao serem selecionadas, através do toque, permitem a inicialização das respetivas vistas de detalhes.

4.3.2.3.10 Vista definições

A vista das definições (Figura 44) mantém o mesmo conjunto de funcionalidades referidas na 2ª iteração, sendo que as únicas alterações ocorridas foram em termos de *design*.

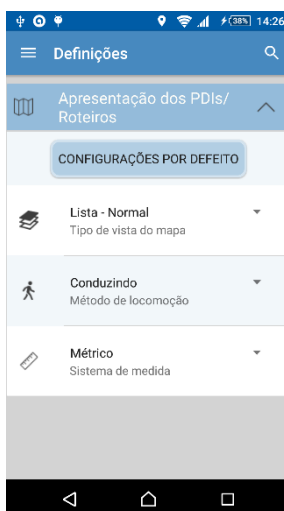


Figura 44 - Vista das definições da aplicação

4.3.2.3.11 Vista agenda

No menu principal da vista principal, é possível selecionar a opção agenda (Figura 45).

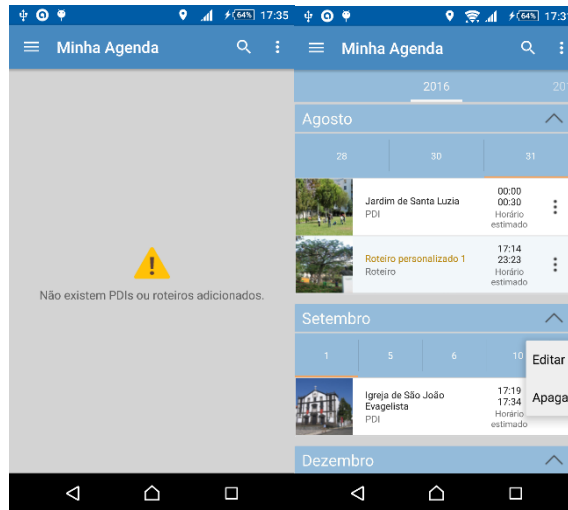


Figura 45 - Vista da agenda

Esta vista é composta por um PagerTabStrip que contém diversos separadores idênticos, sendo um por cada ano de itens agendados. Dentro de cada separador são criadas múltiplas subsecções divididas por mês, sendo que cada uma destas contém outra lista de separadores que dividem as diversas listas de itens por dia. Nesta vista é possível editar (através de um seletor de data e hora) e apagar os horários da agenda.

Ao ser agendado um item são feitas validações de modo a que não exista sobreposição de horários. No momento da criação de um item na agenda é criado um BroadcastReceiver que guarda a referência ao evento agendado, sendo que quando a data e hora do sistema coincide com a data e hora dos itens agendados é criada uma notificação (Figura 46). Este BroadcastReceiver funciona de forma independente à aplicação, ou seja, não é necessário ter a aplicação aberta para que estes estejam ativos, sendo que este tipo de processo foi definido para iniciar com a inicialização do OS Android.

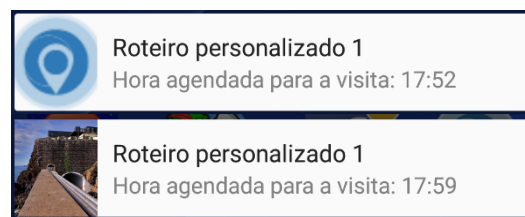


Figura 46 - Notificações de evento agendado (sem internet e com internet)

A notificação ao ser selecionada inicializa, de forma imediata, a vista de detalhes do PDI/Roteiro.

4.3.2.3.12 Dados do utilizador

No menu principal da vista principal, é possível selecionar a opção dados do utilizador, apenas se a sessão tiver sido iniciada. O tipo de vista apresentado é idêntico ao registo, sendo que neste caso os formulários já se apresentam preenchidos com os dados anteriormente introduzidos. Esta secção pode ser acedida sem ligação à internet, embora para confirmar as modificações efetuadas é requerida a ligação à internet.

4.3.2.3.13 Terminar sessão

No menu principal da vista principal, é possível selecionar a opção terminar sessão. Nesta situação os dados associados ao utilizador armazenados localmente são apagados (Neste caso, esta informação é apenas mantida no servidor), dando lugar ao utilizador por defeito, sendo necessário carregar as rotas por defeito entre o ponto-base e os PDIs para este tipo de utilizador.

4.3.2.3.14 Notificação por proximidade

O *smartphone* ao se aproximar fisicamente das coordenadas indicadas para cada PDI é ativado um evento que notifica o utilizador que chegou ao local de um PDI. Este evento funciona de forma bastante semelhante à agenda, sendo que única diferença é que enquanto na agenda a entidade a ser observada é o relógio do sistema, neste caso, é o recetor de GPS. No instante que este evento é ativado, para além da notificação recebida (Figura 47), é armazenado na memória do dispositivo, mais propriamente na secção do histórico, a data e hora do evento. A notificação é emitida apenas uma vez em cada 24 horas para cada PDI, de modo a que o utilizador não esteja constantemente a ser notificado enquanto permanece fisicamente no PDI.

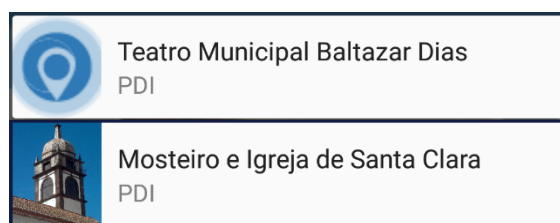


Figura 47 - Notificação de evento de proximidade de PDI (sem internet e com internet)

4.3.2.3.15 Outros aspetos

Todas as secções da aplicação que requerem algum tempo para obter a informação (informação obtida através da internet) apresentam um *layout* de carregamento, normalmente apresentado no formato de uma Progressbar. Na vista principal, a disposição do *layout* foi criada de tal forma a que não seja necessário esperar que um processo termine para mudar de vista. Existem algumas situações em que o ecrã é bloqueado durante o carregamento de dados, sendo este aspeto intencional dado que se o utilizador escolher outra opção pode interromper um processo a meio e tornar parte dos dados corrompidos (ex. Registo do utilizador e finalização da criação de roteiros).

Outro aspeto tido em conta durante o desenvolvimento da GUI é o ajustamento da mesma para ecrãs de diferentes tamanhos, sendo utilizadas medidas relativas em termos de percentagem de espaço ocupado no ecrã do dispositivo. Esta medida é razoável, embora idealmente deveriam ter sido criados múltiplos *layouts* para diferentes tamanhos de ecrãs, sendo que estes não foram criados devido às restrições temporais do projeto.

Nesta iteração foi criada a documentação completa do código desenvolvido.

4.3.3 Conclusão

Após o final da 3ª iteração efetuou-se alguns testes de utilização da aplicação com diversos dispositivos físicos, sendo verificado que a aplicação apresenta algum grau de incompatibilidade ao nível do *design* da GUI com as versões do Android anteriores à versão 5.0 (Lollipop). A incompatibilidade existente é observável na apresentação de alguns ícones, tipos de texto, sombras e transparências que não se apresentam de forma idêntica das versões 4.x para as versões 5 e 6 do Android. Apesar disto, a aplicação apresenta-se funcional para qualquer versão do Android que seja igual ou superior à versão 4.0, permitindo abranger cerca de 98,2 % dos dispositivos Android existentes [36].

Também no final desta iteração efetuou-se um conjunto de testes a ambas as bases de dados utilizadas (MySQL e Neo4j), sendo que os resultados dos testes demonstraram que a base de dados MySQL apresenta maior desempenho, comparativamente à base de dados Neo4j dentro do contexto deste projeto. A descrição dos testes e discussão dos resultados está apresentada na *secção 5.2*.

5. Testes e Resultados

Este capítulo vem introduzir alguns dos testes efetuados durante o desenvolvimento da aplicação, sendo que após a descrição de cada um dos testes irá ser feita uma breve discussão acerca das conclusões obtidas e que impacto estas tiveram no restante desenvolvimento da aplicação. No final deste capítulo irão ser assinaladas algumas das limitações identificadas na aplicação.

5.1 Questionários

No final da 2ª iteração desenvolveu-se um questionário destinado a potenciais utilizadores da aplicação, sendo que para o preenchimento do mesmo foi requisitado que os utilizadores efetuassem testes a algumas funcionalidades presentes na aplicação que podia ser descarregada diretamente do *link* presente na descrição do questionário. O motivo dos testes prévios à aplicação deve-se a que algumas das funcionalidades são difíceis de ser testadas durante o preenchimento do mesmo (ex. Rotas para os PDIs e roteiros).

O objetivo do questionário foi compreender quais as prioridades em termos de funcionalidades para aplicação, tentar detetar a existência de erros em outros tipos de dispositivos Android, obter algum *feedback* em relação ao *design* e por fim tentar compreender se o tipo de informação apresentada é adequada para o fim pretendido.

O questionário é composto por 78 perguntas, sendo a maioria destas de resposta fechada seguindo a escala de Likert de 4 e 5 pontos. As principais razões da escolha da escala de Likert é porque as perguntas criadas usando esta escala são facilmente interpretáveis e consequentemente rapidamente respondidas por ser um método bastante conhecido. Outro motivo é por esta permitir a extração simplificada de dados quantitativos permitindo obter rapidamente conclusões e gráficos associados às respostas.

As perguntas que apresentam apenas 4 pontos têm como objetivo obter uma opinião concreta acerca de um tópico, pois a existência de uma opção intermédia não permite obter qualquer conclusão em alguns tópicos. Já nas perguntas de 5 pontos a opção intermédia é relevante para a obtenção de conclusões, mesmo que estas sejam neutras.

A criação dos questionários foi efetuada com auxílio da ferramenta Google Forms [29]. O questionário apresenta-se disponível no Anexo C e os resultados dos mesmos apresentam-se disponíveis no Anexo D.

Inicialmente, os testes com utilizadores deveriam ter sido efetuados com utilizadores da CMF, sendo isto definido durante a 1ª iteração. Infelizmente, devido a alguns atrasos, estes acabaram por não ser realizados com os utilizadores inicialmente previstos.

Por fim, os testes foram efetuados com algumas pessoas conhecidas que apresentaram disponibilidade para os mesmos (9 pessoas), sendo que os resultados foram apenas obtidos perto do fim da 3ª iteração. Daí que os resultados obtidos possam não ter tido grande impacto sobre o desenvolvimento da aplicação na 3ª iteração.

Os resultados apresentam-se disponíveis no Anexo D.

5.1.1 Discussão dos resultados dos questionários

Ao analisar os resultados dos questionários conclui-se que, segundo os inquiridos, na generalidade o *design* da aplicação está adequado, sendo que a informação apresenta um bom nível de detalhe e é de fácil interpretação.

Já na relevância dos dados apresentados em termos da vista de detalhes de PDIs, a grande maioria dos dados foram designados de muito relevantes ou imprescindíveis (avaliação dos utilizadores, horário de expediente, observações, morada, distância do ponto-base, tempo de viagem, custo das entradas, acessibilidade, contactos). Ainda na mesma vista alguns resultados apresentaram divergências acerca da relevância dos mesmos, sendo que um número significativo de utilizadores indicou que estes campos eram pouco relevantes (as fontes de informação externas e a data da última visita ao PDI).

Na secção do questionário que diz respeito aos dados da vista de detalhes dos roteiros, informações como o mapa de localização dos pontos de interesse, informação acerca da distância dos pontos de interesse, tempo médio do roteiro, horário recomendado, custo do roteiro e acessibilidade, foram considerados muito relevantes ou imprescindíveis. Já dados como a data do último percurso do roteiro e a descrição do roteiro criada pelo utilizador foram indicados, em algumas situações, como dados pouco relevantes e irrelevantes.

Outro aspeto averiguado com o questionário foi o tempo de carregamento das vistas, sendo indicado por alguns inquiridos que o arranque inicial da aplicação e o carregamento de itens na vista principal apresentavam um tempo de carregamento excessivo.

A funcionalidade de criação de rotas até aos locais também foi avaliada pelos inqueridos, sendo que estes indicaram que esta informação era bastante relevante ou imprescindível e que o tipo de dados apresentados nesta vista é adequado para o fim pretendido.

Outras funcionalidades que poderiam ser implementadas no futuro (até ao momento da criação do questionário), como a criação de roteiros personalizados, recomendações de PDIs e roteiros, favoritos, filtros, agenda, avaliação de PDIs e publicação de fotos foram consideradas muito relevantes, sendo que algumas destas foram implementadas na 3ª iteração. Já outras funcionalidades indicadas a implementar no futuro, como o caso do sistema de comentários e histórico de vistas foram consideradas pouco relevantes por alguns utilizadores.

Também é de notar que a aplicação apresentou alguns erros durante a utilização da mesma, sendo que estes erros estão diretamente relacionados com a versão do Android utilizada, dado que apenas ocorreram erros nas versões do Android anteriores à versão 5.0 (Lollipop), sendo esta a versão do OS do dispositivo que foi utilizado para testes durante o desenvolvimento da aplicação.

5.2 Comparação do desempenho entre as bases de dados Neo4j e MySQL

No final da 3ª iteração foi efetuada uma comparação entre o desempenho dos pedidos efetuados ao servidor utilizando ambas as bases de dados (Neo4j e MySQL). Para isto fez-se um conjunto de 10 testes em 7 funcionalidades diferentes, sendo por fim calculado o tempo médio de execução de cada funcionalidade. Os valores apresentados estão em segundos. É de notar que os parâmetros utilizados em todos os pedidos efetuados para cada funcionalidade foram sempre os mesmos para ambas as bases de dados. As funcionalidades escolhidas para testes são as funcionalidades usadas com maior frequência pela aplicação, sendo as seguintes:

- **Ler todos os PDIs** – É efetuada uma pesquisa acerca de todos os PDIs existentes na base de dados;
- **Ler rota entre ponto-base e PDI** – Foi escolhido um PDI aleatório, sendo lidos os dados existentes acerca da rota entre este PDI e um utilizador também aleatório;
- **Ler utilizador** – Foi efetuada uma pesquisa de um utilizador concreto;
- **Ler dados do roteiro** – São lidas todas as sub-rotas entre PDIs de um roteiro e o utilizador em ambos os sentidos (ida e volta);
- **Gerar recomendações de roteiros** – São criados 3 roteiros recomendados para um utilizador concreto;
- **Otimizar rota do roteiro** – É enviado um roteiro, sendo que a resposta é o mesmo roteiro apresentando a sequência de PDIs otimizada por distância;
- **Criar rota automática** – É criada uma rota automática que apresenta, no mínimo, 8 PDIs;

Funcionalidade	1ª Teste	2ª Teste	3ª Teste	4ª Teste	5ª Teste	6ª Teste	7ª Teste	8ª Teste	9ª Teste	10ª Teste	Média (segundos)
Ler todos os PDIs	0,268	0,259	0,258	0,263	0,256	0,251	0,259	0,252	0,258	0,260	0,258
Ler rota entre ponto-base e PDI	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004	0,004
Ler utilizador	0,004	0,003	0,003	0,004	0,003	0,003	0,004	0,003	0,004	0,004	0,004
Ler dados do roteiro	0,077	0,071	0,072	0,077	0,074	0,067	0,075	0,068	0,076	0,072	0,073
Gerar recomendações de roteiros	9,994	9,881	9,815	9,937	9,822	9,832	9,871	9,918	9,813	9,784	9,867
Otimizar rota do roteiro	3,524	3,477	3,502	3,519	3,488	3,520	3,452	3,563	3,496	3,462	3,500
Criar rota automática	6,618	6,669	6,564	6,611	6,616	6,564	6,584	6,652	6,562	6,651	6,609

Tabela 10 - Tempos de execução para a BD MySQL (segundos)

Funcionalidade	1ª Teste	2ª Teste	3ª Teste	4ª Teste	5ª Teste	6ª Teste	7ª Teste	8ª Teste	9ª Teste	10ª Teste	Média (segundos)
Ler todos os PDIs	0,995	1,164	1,383	0,962	1,087	1,184	1,097	1,178	1,243	1,039	1,133
Ler rota entre ponto-base e PDI	0,015	0,015	0,015	0,014	0,015	0,015	0,014	0,016	0,015	0,015	0,015
Ler utilizador	0,015	0,018	0,014	0,015	0,016	0,018	0,013	0,018	0,013	0,015	0,016
Ler dados do roteiro	0,087	0,130	0,090	0,084	0,086	0,089	0,083	0,089	0,083	0,089	0,091
Gerar recomendações de rotas	18,611	18,659	18,314	18,960	18,567	18,477	18,691	18,254	18,841	19,516	18,689
Otimizar rota do roteiro	3,737	3,746	3,716	3,723	3,684	3,711	3,707	3,686	3,681	3,658	3,705
Criar rota automática	10,999	11,446	11,268	10,690	11,166	11,565	11,305	11,053	10,984	11,632	11,211

Tabela 11 - Tempos de execução para a BD Neo4j (segundos)

5.2.1 Discussão dos resultados dos testes às bases de dados

Ao analisar os resultados obtidos, conclui-se que a base de dados MySQL é mais rápida do que a base de dados Neo4j, sendo este um resultado algo inesperado. O motivo deste resultado é que o Neo4j apresenta-se como uma solução mais adequada quando se pretende efetuar pesquisas que necessitem de percorrer um grafo com uma grande profundidade (grande número de nós, na ordem dos milhares de nós) [35], sendo que neste caso apenas se percorreu 1 nível de profundidade.

5.3 Limitações da Aplicação

A aplicação desenvolvida apresenta um conjunto de limitações associadas à forma de como esta foi implementada, sendo que algumas destas foram citadas anteriormente.

Uma das primeiras limitações identificadas está relacionado com a funcionalidade de criação de rotas para os PDIs e os roteiros, dado que caso não exista ligação à internet torna-se impossível a atualização da rota, sendo apenas feita a adição e remoção de pontos da *polyline*, nas situações em que o utilizador, respetivamente, siga a rota apresentada ou se desvie do percurso indicado. Neste caso, não existe a possibilidade de atualização dos dados acerca da distância e do tempo do percurso.

Outra limitação presente no sistema desenvolvido está associada à impossibilidade de criação e recomendação de roteiros sem a existência de ligação à internet. Esta limitação deve-se ao facto que os dados acerca de todas as rotas possíveis não estarem alojados no terminal do cliente (*Smartphone*). Seria possível contornar esta situação ao alojar todas estas rotas no terminal móvel embora como contrapartida a aplicação iria passar a ocupar uma porção do espaço de memória significativa, principalmente em alguns dispositivos móveis mais antigos que apresentam um espaço de memória interna bastante reduzido.

Outra limitação identificada é referente ao número máximo de 8 PDIs possíveis que um roteiro pode apresentar para o seu trajeto poder ser otimizado. Esta limitação deve-se ao facto que o tempo de execução desta funcionalidade para um conjunto de PDIs superior a 8 elementos ser bastante elevado, havendo o risco de exceder o limite de *timeout* (30 segundos).

Outra limitação é o valor do tempo médio de visita apresentado, que em muitas situações pode não corresponder ao valor médio real. Nesta situação o valor médio utilizado é um valor representativo para cada categoria de PDIs (apenas para os PDIs que não foi possível obter informação durante a fase de pesquisa de dados). Um meio de contornar esta limitação passaria por manter os valores representativos por cada categoria, mas também contabilizar o valor do tempo médio real que cada visitante leva para visitar cada PDI. Quando o número de visitas total a um PDI fosse significativo, o valor do tempo médio apresentado iria passar a ser o valor do tempo médio real, substituindo assim o valor do tempo médio representativo por categoria.

Por fim, uma outra limitação da aplicação é o tempo de carregamento dos PDIs e roteiros crescer de forma constante à medida que são adicionados novos PDIs e roteiros. O problema desta situação é que para um grande número de PDIs e roteiros estes tempos podem se tornar excessivamente longos, até ao ponto de se tornar intoleráveis. Uma possível tática que poderia ser implementada para evitar esta limitação passaria pela utilização do padrão de desenho de software *Lazy Loading* para os PDIs e roteiros, sendo que estes seriam carregados à medida que fosse necessário a apresentação dos dados associados aos mesmos. Uma das desvantagens da utilização deste padrão é que, apesar tornar o tempo de carregamento inicial mais rápido, poderia tornar a navegação pela aplicação mais lenta, pois os tempos de carregamento dos itens, à medida que fossem apresentados, iriam ser significativamente mais longos. Neste momento, a tática de *Lazy Loading* é apenas utilizada no carregamento dos dados acerca da distância e tempo de viagem cujo pedido ao servidor é efetuado apenas quando um item está prestes a ficar visível (No caso da vista mapa, o instante em que o utilizador seleciona um marcador. Já no caso da vista em lista à medida que o utilizador efetua *scroll* sobre a lista de itens). O motivo da não utilização deste padrão diretamente sobre todos os PDIs e roteiros está associada à vista em mapa dos itens que exige o carregamento imediato de todos os itens no momento em que a vista é instanciada, pelo menos para a região que está inicialmente visível no mapa. Este aspeto poderia ser corrigido ao efetuar um pedido para servidor de modo a que este apenas envie inicialmente os dados associados à localização dos PDIs e os IDs associados aos mesmos (reduzindo significativamente a quantidade de dados que *smartphone* iria ter de interpretar no momento da instanciação da vista), sendo que os restantes dados iriam ser posteriormente obtidos apenas no momento em que o utilizador seleciona-se o marcador associado a um dos PDIs.

6. Conclusões

O projeto desenvolvido inicialmente começou com um objetivo, a criação de uma aplicação que complemente a experiência de alguns percursos turísticos preexistentes com elementos de realidade aumentada, sendo que ao longo do desenvolvimento do projeto foi necessário modificar alguns dos objetivos do projeto, devido à ausência de alguns dados que permitam apresentar informação acerca destes percursos.

Em alternativa, optou-se por criar uma aplicação que não dependesse de um conjunto de dados/rotas fixas, sendo para isto criado um sistema de gestão de PDIs, que apresenta responsabilidade de gerir os dados dos PDIs e das rotas entre estes e os utilizadores, de modo a permitir a modificação destes dados a qualquer momento.

Outro aspeto que teve de ser modificado ao longo do projeto foi o modo de como os roteiros são apresentados, sendo que na versão final da aplicação estes são definidos manualmente pelo utilizador ou de forma automática consoante os “ganhos” calculados para um conjunto de PDIs, dentro de um conjunto de restrições e de uma janela temporal.

Os protótipos desenvolvidos foram modificados ao longo do tempo de modo a acomodar melhoramentos e novas funcionalidades em relação às iterações anteriores. Em todas as iterações efetuaram-se sempre melhorias em termos de *design* e usabilidade. A implementação dos *webservices* foi gradual, consoante a introdução, de novas funcionalidades na aplicação.

Toda a lógica do lado do servidor apresenta-se independente do lado do cliente através da utilização do padrão arquitetural MVC. A lógica do servidor também se apresenta independente da base de dados utilizada através da utilização do padrão de desenho de *software Bridge*.

Após alguns testes, optou-se por utilizar a base de dados MySQL dado que esta apresenta um tempo de leitura e escrita superior para um pequeno conjunto de nós, que é o caso deste projeto. No futuro, caso o número de nós aumente, na ordem dos milhares de nós, torna-se viável a utilização da base de dados Neo4j, dado que o tempo de escrita e leitura para esta situação aumenta significativamente para a base de dados MySQL comparativamente ao Neo4j [35].

Por fim, a aplicação desenvolvida, consiste num sistema que serve como guia turístico à medida das necessidades do utilizador, sendo que este tem em conta aspetos

como a proximidade, o custo e os interesses pessoais do utilizador em relação a um conjunto de PDIs existentes no concelho do Funchal.

6.1 Objetivos atingidos

Em termos de objetivos atingidos, na 1ª iteração desenvolveu-se um protótipo de modo a obter *feedback* detalhado acerca do tipo de aplicação pretendida, dado que inicialmente não existiam dados concretos acerca de funcionalidades detalhadas pretendidas para a aplicação. Por fim, a análise deste protótipo, serviu para definir quais os conceitos a abordar na aplicação final.

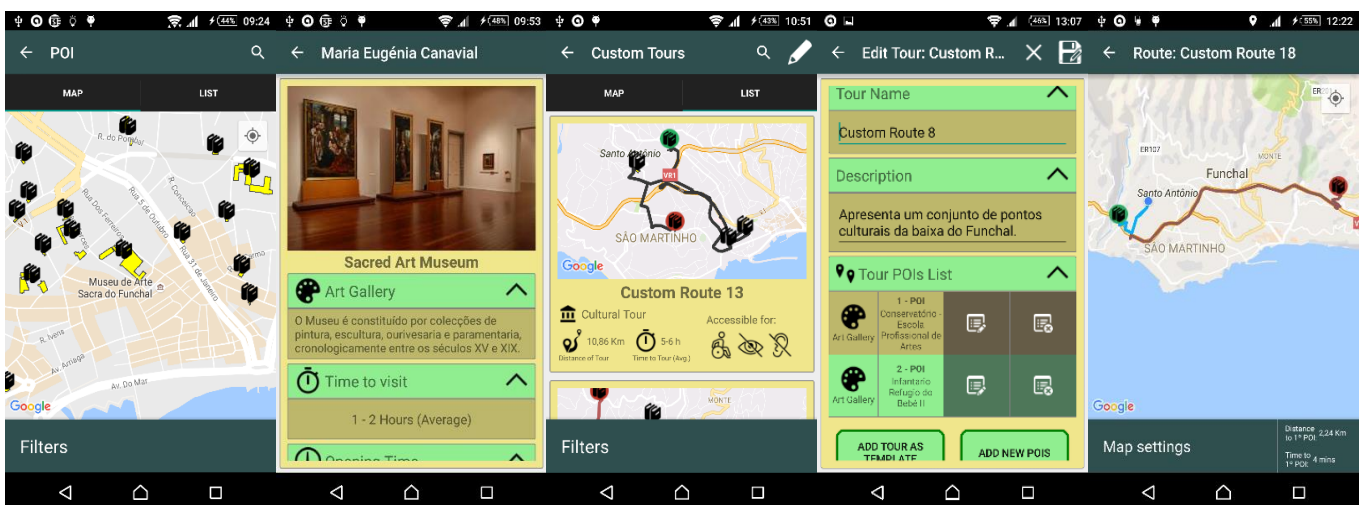


Figura 48 - Protótipo de alta fidelidade para 1ª iteração

Os objetivos atingidos na 2ª iteração foram o estabelecimento de uma arquitetura para o sistema, a pesquisa de dados, a criação de um sistema de gestão dos PDIs/dados e por fim a integração dos dados dentro de um protótipo de alta fidelidade, que foi posteriormente testado.

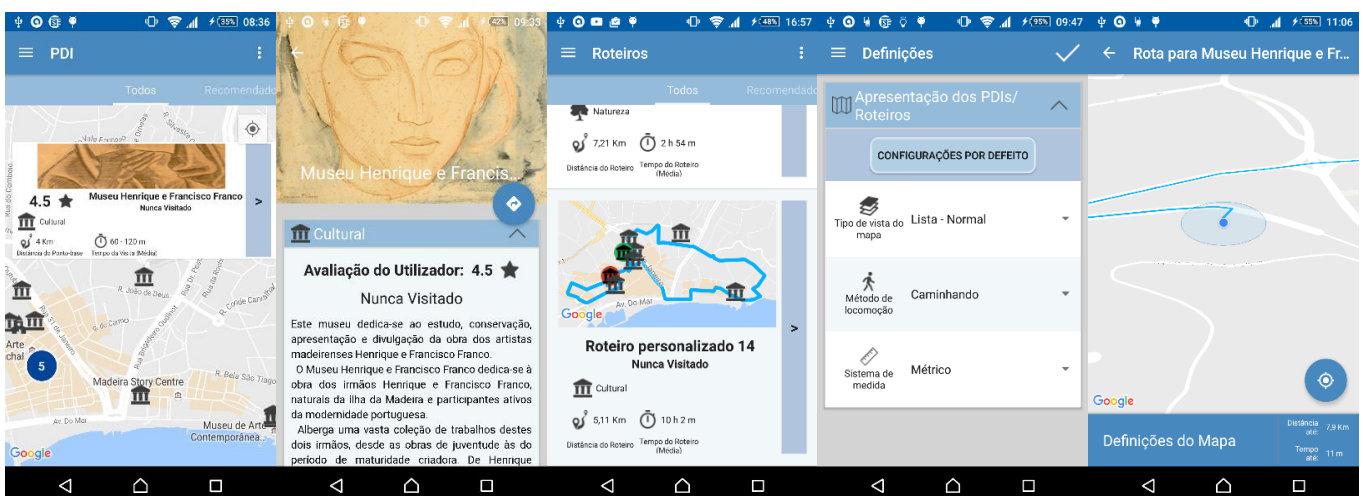


Figura 49 - Protótipo de alta fidelidade para 2ª iteração

Na 3ª iteração, os objetivos atingidos foram a implementação das funcionalidades relacionadas com recomendação de PDIs e roteiros, criação de roteiros, agendamento de PDIs e roteiros. E por fim implementou-se a base de dados MySQL sendo feita a comparação entre o desempenho a base de dados Neo4j com a base de dados MySQL.

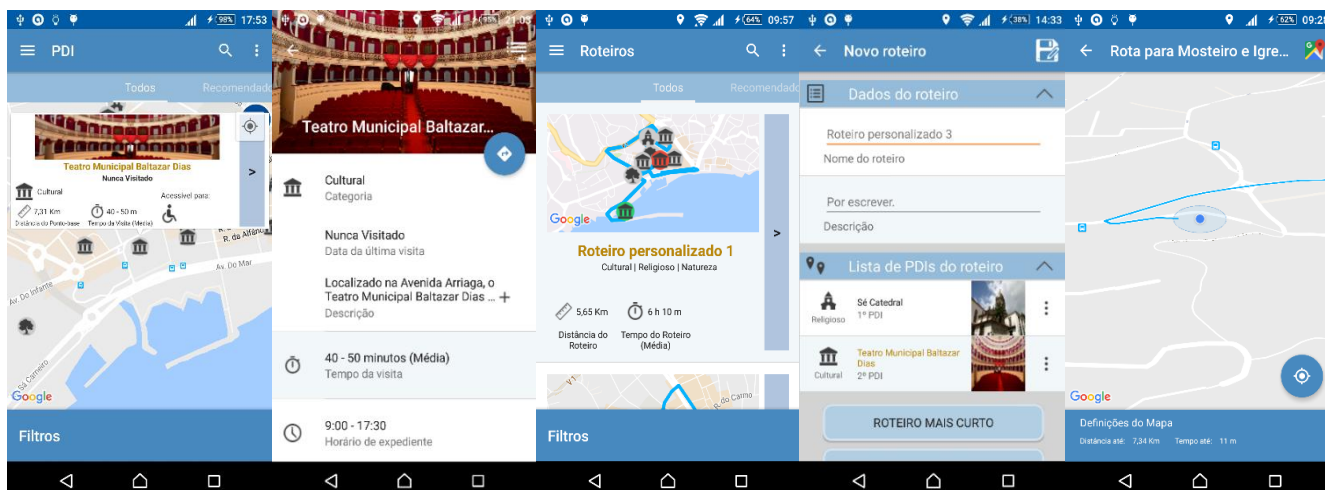


Figura 50 - Protótipo de alta fidelidade para 3ª iteração

Em termos de funcionalidades, desenvolveu-se a maioria das funcionalidades pretendidas.

A aplicação final apresenta o seguinte conjunto de funcionalidades relacionadas com os PDIs: Apresentação de dados detalhados; Criação de rotas; Recomendação de PDIs; Lista de favoritos; Histórico de visitas; Deteção de proximidade de PDIs; Agendamento; Utilização *offline*.

A aplicação também apresenta o seguinte conjunto de funcionalidades relacionadas com os Roteiros: Criação de roteiros personalizados; Apresentação de dados detalhados; Criação de rotas; Recomendação de Roteiros; Lista de favoritos; Agendamento; Utilização *offline*.

6.2 Trabalho futuro

Como trabalho futuro, pretende-se incluir uma vertente mais social na aplicação.

Esta vertente social deveria integrar um sistema de opiniões (comentários e avaliação de 1 a 5) acerca de cada PDI, sendo que cada um destes apresentaria uma avaliação média. Outra funcionalidade dentro desta vertente social seria a introdução dos locais mais visitados. Os algoritmos utilizados nas recomendações deveriam ser modificados de modo a integrar as opiniões gerais dos utilizadores. Também poder-se-ia

desenvolver um modo de partilha fotos diretamente a partir da aplicação. As visitas aos locais poderiam ser partilhadas através das redes sociais.

Outro aspeto que possivelmente seria introduzido no futuro é a reestruturação do servidor de modo a que este suportasse a utilização de *websockets*, principalmente devido ao elevado desempenho que estes apresentam em termos de tempo de comunicação [34].

Futuramente, a arquitetura do sistema poderia ser reestruturada, caso a aplicação apresente uma quantidade significativa de utilizadores. Esta reestruturação deveria apresentar um conjunto mais alargado de servidores redundantes, para permitir que a plataforma suporte uma elevada carga de pedidos em simultâneo, sendo que os pedidos deveriam encaminhados através de um intermediário seguindo o padrão arquitetural *Broker*. Uma das principais vantagens deste padrão é o aumento do desempenho, por aliviar a carga a que cada servidor se apresenta exposto ao reencaminhar os pedidos e respostas para diferentes servidores.

Apesar da aplicação apresentar suporte para ecrãs de diferentes tamanhos através da utilização de tamanhos relativos em termos da percentagem do ecrã ocupado, idealmente poderiam ser criados *layouts* dedicados para tablets onde, por exemplo, apresentar-se-ia a lista de PDIs e de detalhes de cada PDI simultaneamente numa única vista, contribuindo assim para a melhoria da usabilidade em sistemas que apresentem ecrãs de grandes dimensões.

Outro aspeto que poderia ser implementado é o suporte para outras línguas, para além de Português e Inglês, sendo isto realizado através da criação de um ficheiro XML contendo todas as *strings* utilizadas na aplicação.

Por fim, outro aspeto a ser repensado no futuro seria o tipo de ambiente de desenvolvimento da aplicação sendo que, ou implementar-se-ia uma nova versão da aplicação dedicada a outros sistemas como o Windows Phone ou o iOS, dado que estes apresentam uma quantidade significativa de utilizadores [42], ou implementar-se-ia uma nova versão da aplicação num ambiente de desenvolvimento de aplicações multiplataforma, embora as vistas iriam ter que ser desenvolvidas separadamente para cada sistema operativo. Como já se referiu anteriormente, optou-se por um ambiente de desenvolvimento de aplicações nativas, dado que a aplicação a ser desenvolvida pela CMF, onde possivelmente esta aplicação poderá a vir ser integrada como um módulo, é desenvolvida utilizando o SDK para Android desenvolvido pela Google.

Referências

- [1] - Gov-madeira.pt. (2016). Governo Regional da Madeira. [online] Available at: <http://www.gov-madeira.pt/madeira/conteudo/displayconteudo.do2?numero=263> [Accessed 11 Aug. 2016].<http://www.gov-madeira.pt/madeira/conteudo/displayconteudo.do2?numero=263>.
- [2] - Gavalas, D., Konstantopoulos, C., Mastakas, K. and Pantziou, G. (2014). A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3), pp.291-328.
- [3] - Gruhn, V. and Köhler, A. (2006). Aligning Software Architectures of Mobile Applications on Business Requirements. In: 18th international conference on Advanced Information Systems Engineering. Leipzig, Germany, pp.1113-1124.
- [4] - Pencil.evolus.vn. (2016). Home - Pencil Project. [online] Available at: <http://pencil.evolus.vn/> [Accessed 22 Apr. 2016].
- [5] - GitHub. (2016). nathanielw/Android-Lollipop-Pencil-Stencils. [online] Available at: <https://github.com/nathanielw/Android-Lollipop-Pencil-Stencils> [Accessed 28 Jun. 2016].
- [6] - Google Developers. (2016). KML Reference | Keyhole Markup Language | Google Developers. [online] Available at: <https://developers.google.com/kml/documentation/kmlreference> [Accessed 31 Aug. 2016].
- [7] - GitHub. (2016). googlemaps/android-maps-utils. [online] Available at: <https://github.com/googlemaps/android-maps-utils> [Accessed 31 Aug. 2016].
- [8] - Mathworld.wolfram.com. (2016). Geometric Centroid -- from Wolfram MathWorld. [online] Available at: <http://mathworld.wolfram.com/GeometricCentroid.html> [Accessed 31 Aug. 2016].
- [9] - Google Design. (2016). Google Design. [online] Available at: <https://design.google.com/> [Accessed 31 Aug. 2016].
- [10] - Icons8. (2016). 33,200 Free Icons - The Largest Icon Pack Ever. [online] Available at: <https://icons8.com/> [Accessed 31 Aug. 2016].
- [11] - Developer.android.com. (2016). Icon Design Guidelines | Android Developers. [online] Available at: https://developer.android.com/guide/practices/ui_guidelines/icon_design.html [Accessed 31 Aug. 2016].
- [12] - GitHub. (2016). umano/AndroidSlidingUpPanel. [online] Available at: <https://github.com/umano/AndroidSlidingUpPanel> [Accessed 31 Aug. 2016].
- [13] - Google Developers. (2016). Google Maps Directions API | Google Developers. [online] Available at: <https://developers.google.com/maps/documentation/directions/> [Accessed 31 Aug. 2016].

- [14] - Wptrafficanalyzer.in. (2016). Route between two locations with waypoints in Google Map Android API V2 | Knowledge by Experience. [online] Available at: <http://wptrafficanalyzer.in/blog/route-between-two-locations-with-waypoints-in-google-map-android-api-v2/> [Accessed 31 Aug. 2016].
- [15] - Strassner, T. (2015). XML vs JSON. Medford, Massachusetts.
- [16] - Cm-funchal.pt. (2016). Município do Funchal. [online] Available at: <http://www.cm-funchal.pt/> [Accessed 31 Aug. 2016].
- [17] - Visitmadeira.pt. (2016). Site oficial do Turismo da Madeira. [online] Available at: <http://www.visitmadeira.pt/> [Accessed 31 Aug. 2016].
- [18] - Apachefriends.org. (2016). XAMPP Installers and Downloads for Apache Friends. [online] Available at: https://www.apachefriends.org/pt_br/index.html [Accessed 31 Aug. 2016].
- [19] - Graphenedb.com. (2016). Neo4j Cloud Hosting, Neo4j Hosting | GrapheneDB. [online] Available at: <http://www.graphenedb.com/> [Accessed 31 Aug. 2016].
- [20] - Neo4j Graph Database. (2016). Neo4j: The World's Leading Graph Database. [online] Available at: <https://neo4j.com/> [Accessed 31 Aug. 2016].
- [21] - GitHub. (2016). jadell/neo4jphp. [online] Available at: <https://github.com/jadell/neo4jphp> [Accessed 31 Aug. 2016].
- [22] - GitHub. (2016). jadell/neo4jphp. [online] Available at: <https://github.com/jadell/neo4jphp/wiki> [Accessed 31 Aug. 2016].
- [23] - Hedley, J. (2016). jsoup Java HTML Parser, with best of DOM, CSS, and jquery. [online] Jsoup.org. Available at: <https://jsoup.org/> [Accessed 31 Aug. 2016].
- [24] - GitHub. (2016). google/gson. [online] Available at: <https://github.com/google/gson> [Accessed 31 Aug. 2016].
- [25] - Hc.apache.org. (2016). Apache HttpComponents - Apache HttpComponents. [online] Available at: <https://hc.apache.org/> [Accessed 31 Aug. 2016].
- [26] - Developer.android.com. (2016). AsyncTask | Android Developers. [online] Available at: <https://developer.android.com/reference/android/os/AsyncTask.html> [Accessed 31 Aug. 2016].
- [27] - GitHub. (2016). mcxiaoke/android-volley. [online] Available at: <https://github.com/mcxiaoke/android-volley> [Accessed 31 Aug. 2016].
- [28] - Chris Veness, w. (2016). Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript. [online] Movable-type.co.uk. Available at: <http://www.movable-type.co.uk/scripts/latlong.html> [Accessed 31 Aug. 2016].
- [29] - Google.com. (2016). Google Forms - create and analyze surveys, for free.. [online] Available at: <https://www.google.com/forms/about/> [Accessed 31 Aug. 2016].

- [30] - Material design guidelines. (2016). Metrics & keylines - Layout - Material design guidelines. [online] Available at: <https://material.google.com/layout/metrics-keylines.html> [Accessed 31 Aug. 2016].
- [31] - Mysql.com. (2016). MySQL :: MySQL Workbench. [online] Available at: <https://www.mysql.com/products/workbench/> [Accessed 31 Aug. 2016].
- [32] – Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (2015). Design patterns. Boston, Mass.: Addison-Wesley.
- [33] - Developer.android.com. (2016). Security Tips | Android Developers. [online] Available at: <https://developer.android.com/training/articles/security-tips.html> [Accessed 31 Aug. 2016].
- [34] - Liu, Q. and Sun, X. (2012). Research of Web Real-Time Communication Based on Web Socket. International Journal of Communications, Network and System Sciences, pp.797-801.
- [35] - Joishi, J. and Sureka, A. (2014). Vishleshan. Proceedings of the 19th International Database Engineering & Applications Symposium on - IDEAS '15.
- [36] - Developer.android.com. (2016). Dashboards | Android Developers. [online] Available at: <https://developer.android.com/about/dashboards/index.html> [Accessed 6 Sep. 2016].
- [37] - Sygic Travel. (2016). Sygic Travel. [online] Available at: <http://sygictravel.com/> [Accessed 7 Sep. 2016].
- [38] - Tourias-mobile.com. (2016). Reiseinhalte und mobile Reisebegleiter für die Tourismusbranche - Reiseinhalte und mobile Reisebegleiter für die Tourismusbranche. [online] Available at: <http://www.tourias-mobile.com/> [Accessed 7 Sep. 2016].
- [39] - Hikers Bay - Best time to travel. (2016). HikersBay - Check when is the best time to visit Madeira. [online] Available at: <http://hikersbay.com/europe/portugal/madeira?lang=en> [Accessed 7 Sep. 2016].
- [40] - Madeira.Best. (2016). Madeira.Best - Beyond Experiences Sightseeing and Tours in Madeira and Porto Santo islands. [online] Available at: <https://madeira.best/> [Accessed 7 Sep. 2016].
- [41] - Pt.foursquare.com. (2016). San Antonio | Gastronomía, vida nocturna, entretenimiento. [online] Available at: <https://pt.foursquare.com/> [Accessed 7 Sep. 2016].
- [42] - Statista. (2016). Mobile OS market share 2016 | Statistic. [online] Available at: <http://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/> [Accessed 7 Sep. 2016].
- [43] - Brown, B. and Chalmers, M. (2003). Tourism and mobile technology. ECSCW 2003, pp.335-354.
- [44] - Vaittinen, T. and McGookin, D. (2016). Phases of Urban Tourists' Exploratory Navigation.

- [45] – Wood, W. (2007). A Practical Example of Applying Attribute-Driven Design (ADD), Version 2.0. 1st ed. Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute.
- [46] - Nord, R. (2003). Integrating the architecture tradeoff analysis method (ATAM) with the cost benefit analysis method (CBAM). 1st ed. Pittsburgh, Pa.: Carnegie Mellon University, Software Engineering Institute.
- [47] - Jobe, W. (2013). Native Apps Vs. Mobile Web Apps. International Journal of Interactive Mobile Technologies (IJIM), 7(4), p.27.
- [48] - RapidValue Solutions, (2012). How to Choose the Right Architecture For Your Mobile Application.
- [49] - Codd, E. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), pp.377-387.
- [50] - Edlich, P. (2016). NOSQL Databases. [online] Nosql-database.org. Available at: <http://nosql-database.org/> [Accessed 12 Nov. 2016].

Anexos

Anexo A - Protótipo de Baixa Fidelidade 1ª Iteração

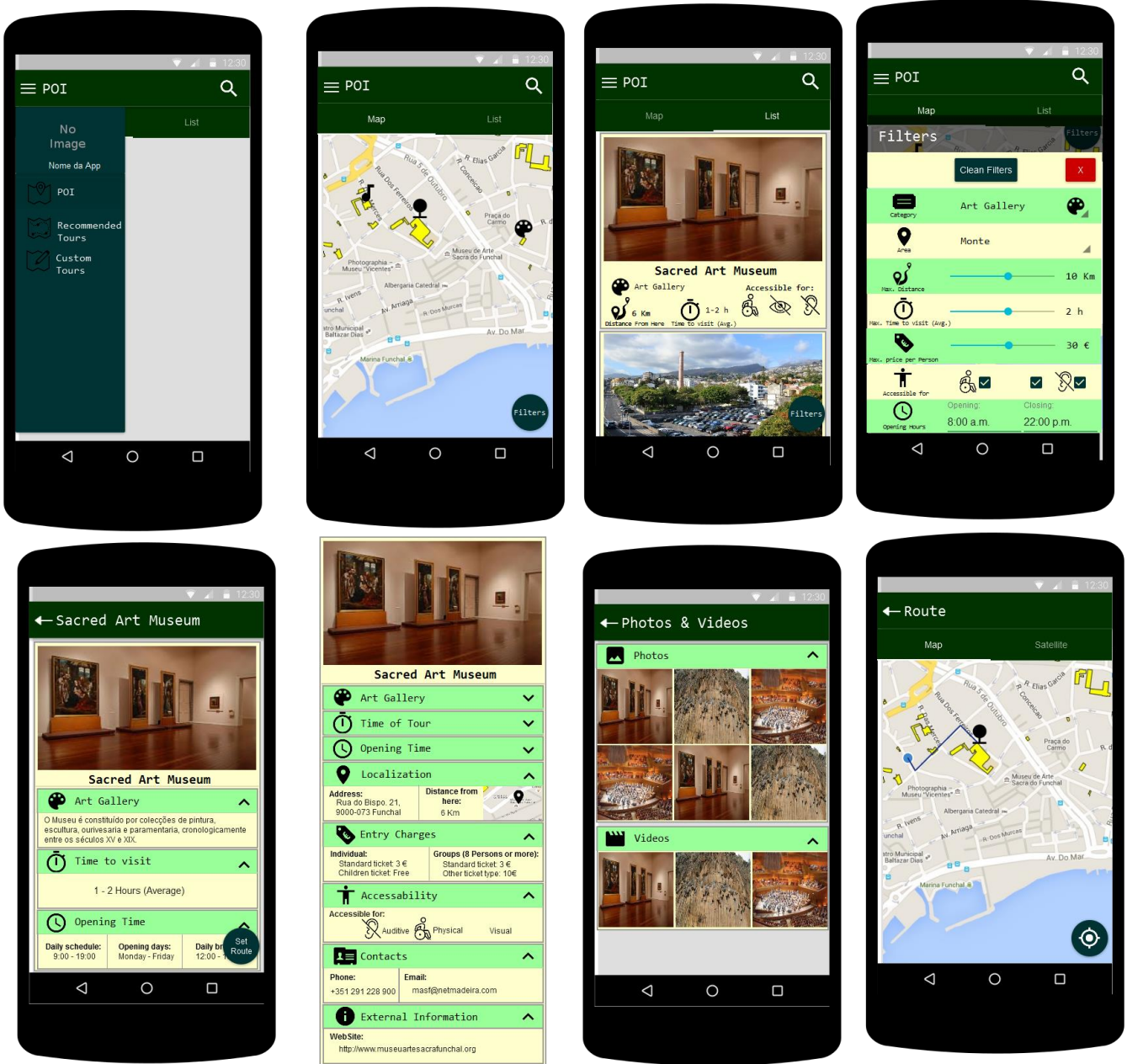


Figura 51 - Protótipo de baixa fidelidade 1ª iteração secção PDIs

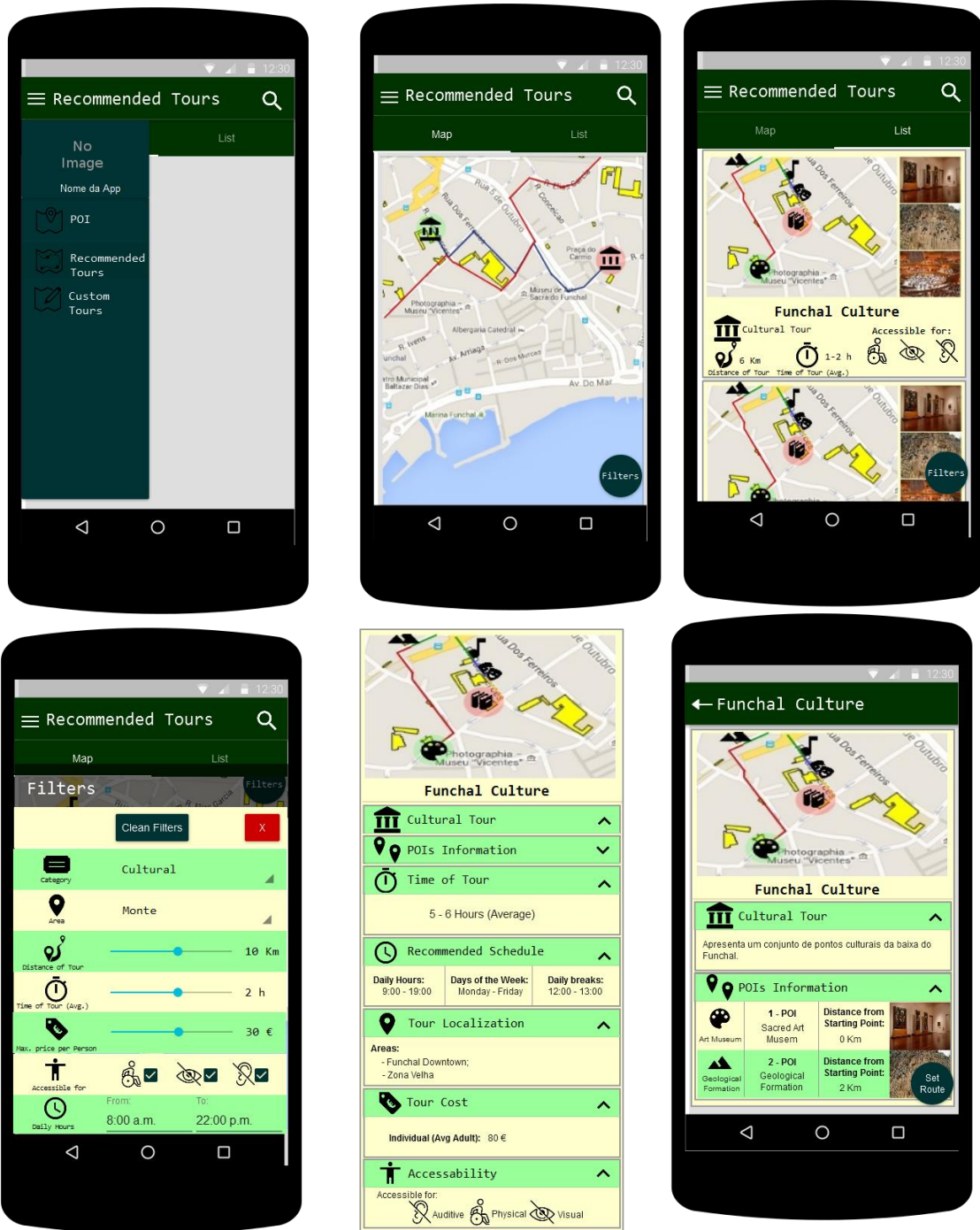


Figura 52 - Protótipo de baixa fidelidade 1ª iteração secção Recommended Tours

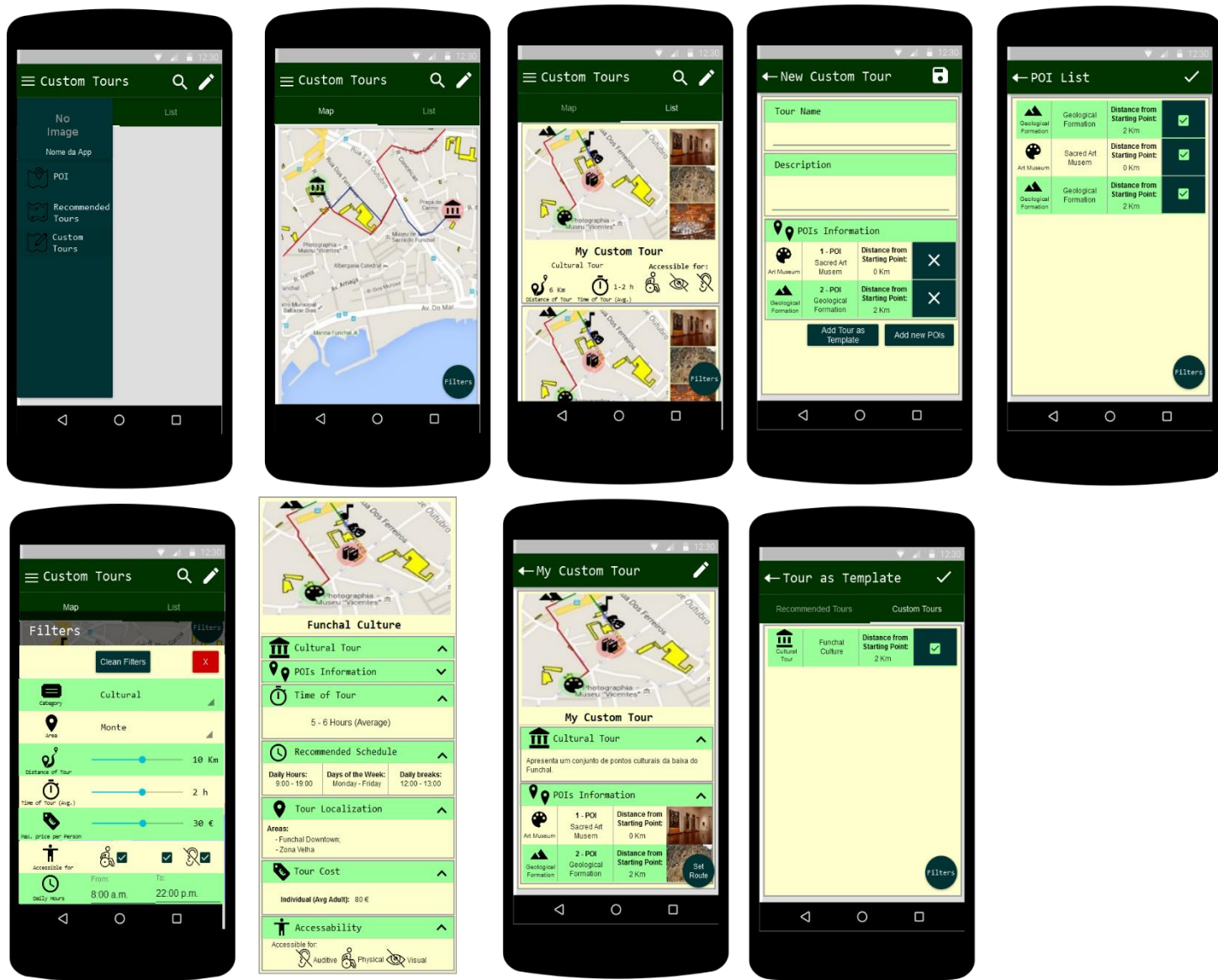


Figura 53 - Protótipo de baixa fidelidade 1ª iteração secção Custom Tours

Anexo B - Protótipo de Baixa Fidelidade 2ª Iteração

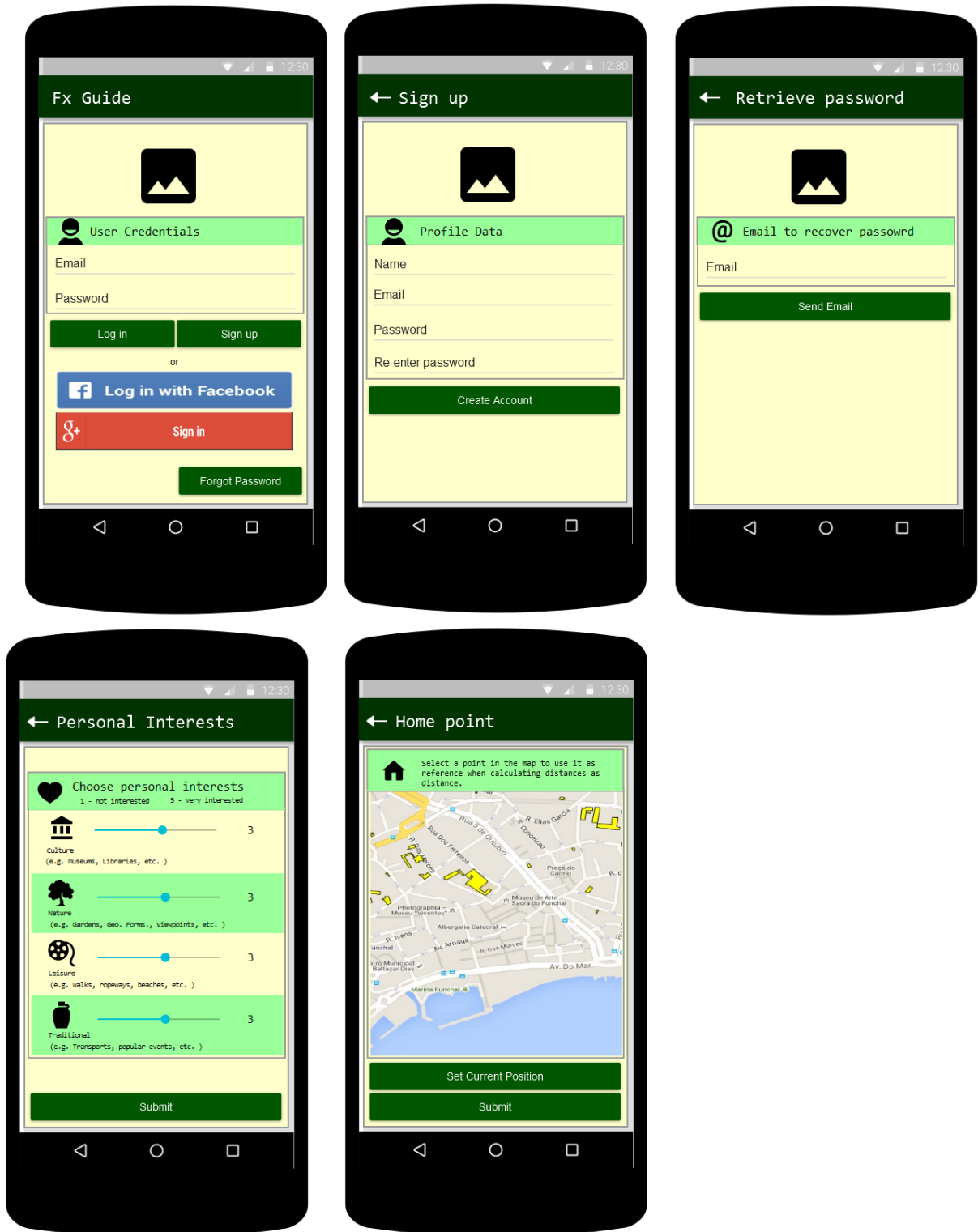


Figura 54 - Protótipo de baixa fidelidade 2ª iteração secção Registo/Login

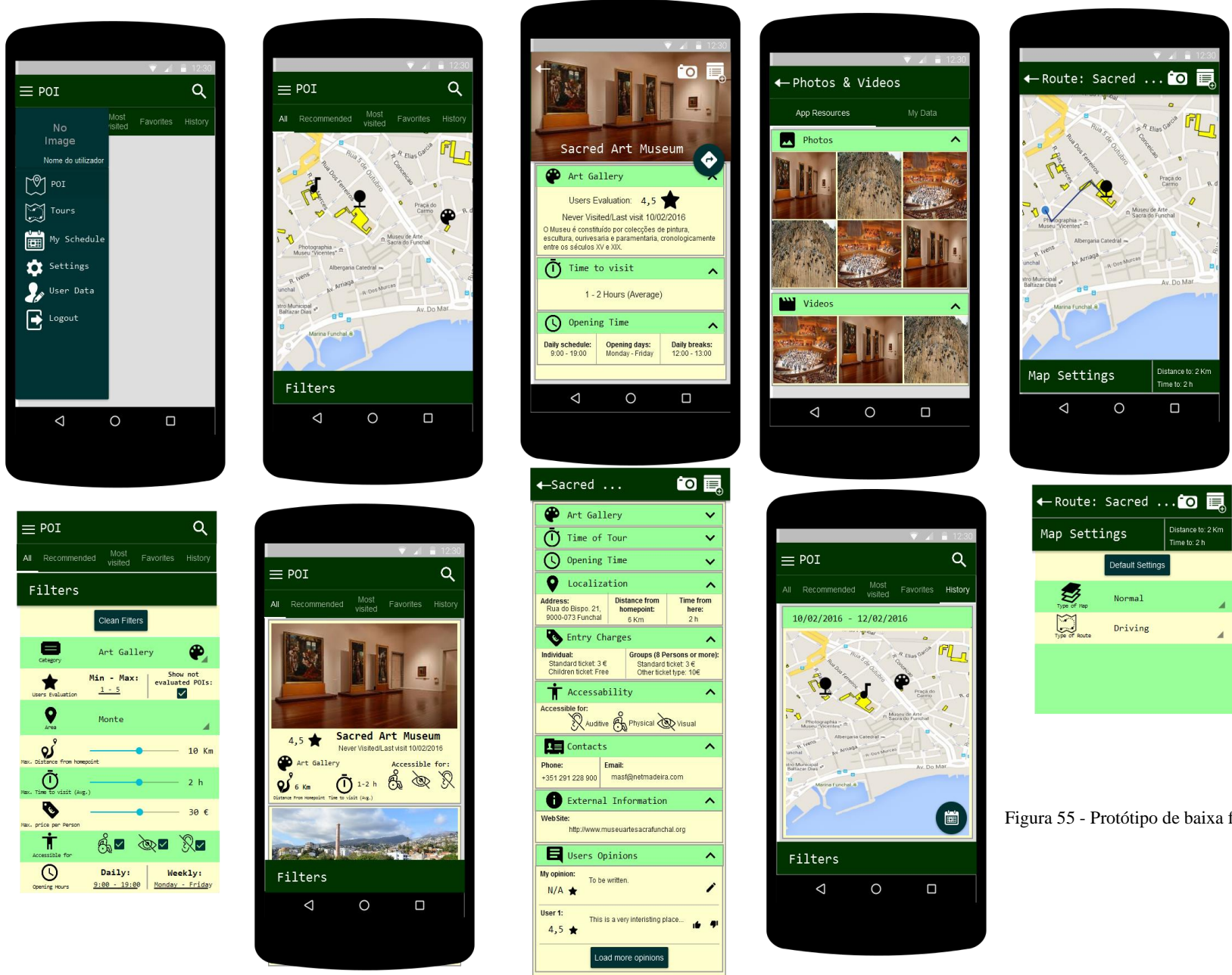


Figura 55 - Protótipo de baixa fidelidade 2ª iteração secção PDIs

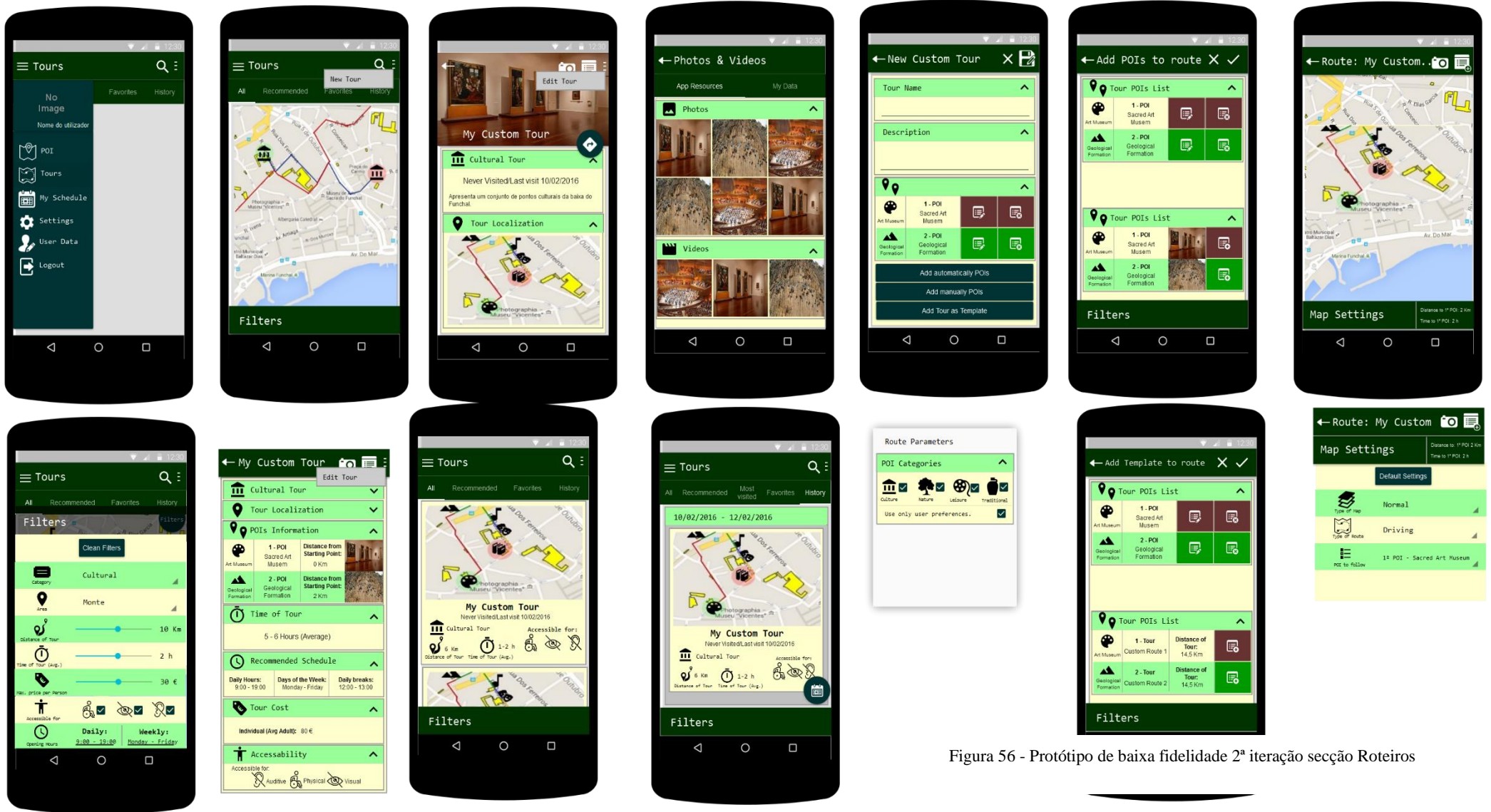


Figura 56 - Protótipo de baixa fidelidade 2ª iteração secção Roteiros

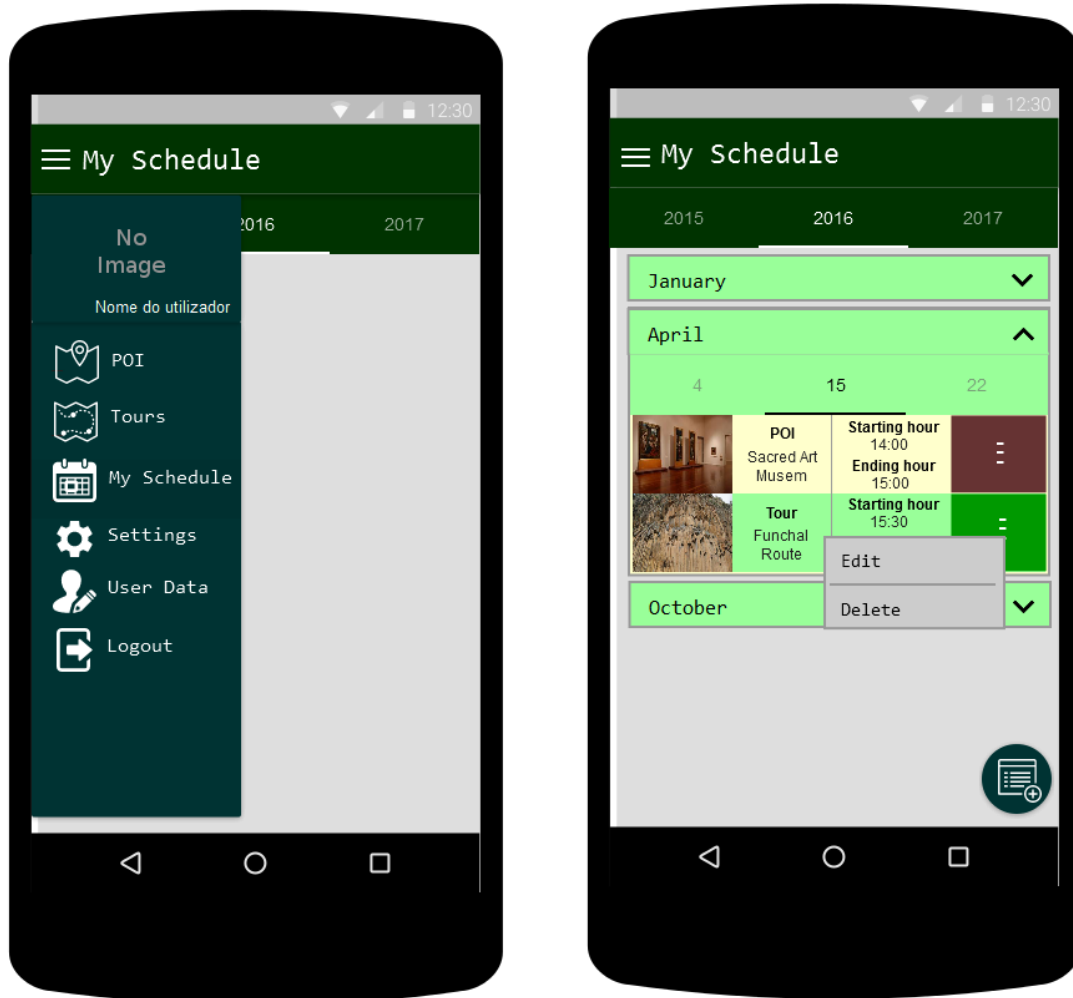


Figura 57 - Protótipo de baixa fidelidade 2ª iteração secção Agenda

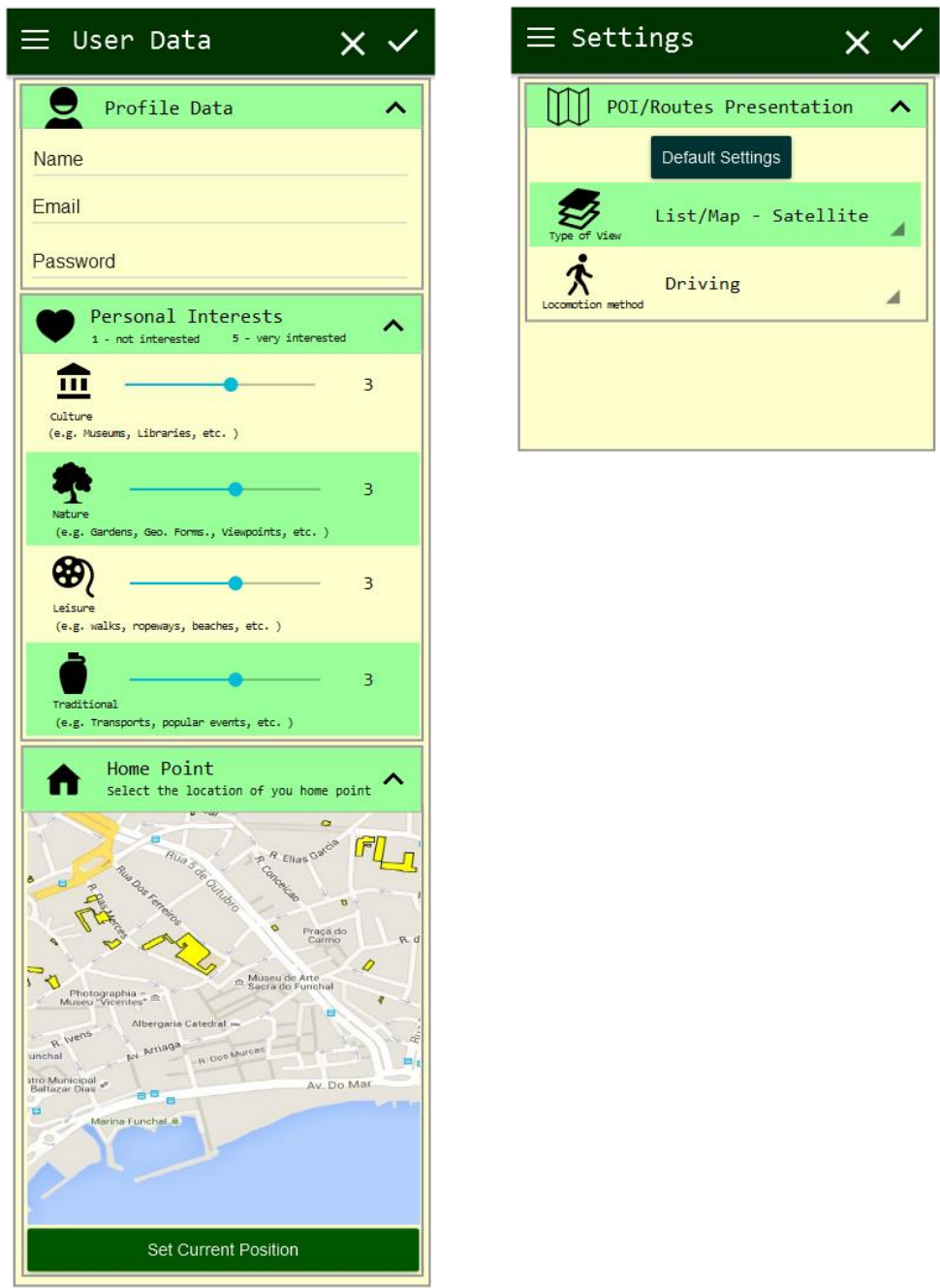


Figura 58 - Protótipo de baixa fidelidade 2ª iteração secção Definições e Dados do Utilizador

Anexo C – Formulário dos questionários

Fx Guide - Testes

Este questionário tem o intuito de avaliar o atual e futuro desenvolvimento de uma aplicação para Android cujo principal objetivo é a criação e sugestão de roteiros turísticos personalizados, sendo que os locais a visitar são pontos de interesse localizados dentro do conselho do Funchal.

Para a realização deste questionário é necessário que efetue previamente alguns testes à aplicação no seu smartphone.

Algumas perguntas da secção 5 e 9 necessitam que faça uso das funcionalidades de criação de rotas para um ponto de interesse ou para um roteiro (Não necessita de ir aos pontos de interesse, basta que teste um pouco as rotas criadas para concluir qual o grau de dificuldade de orientação).

Aplicação apresenta-se disponível na seguinte ligação:

<https://drive.google.com/file/d/0B8QxFC1-5eWXNnhkbURmMUJxZ0k/view?usp=sharing>

Nota:

Apenas seguintes funcionalidades se encontram disponíveis:

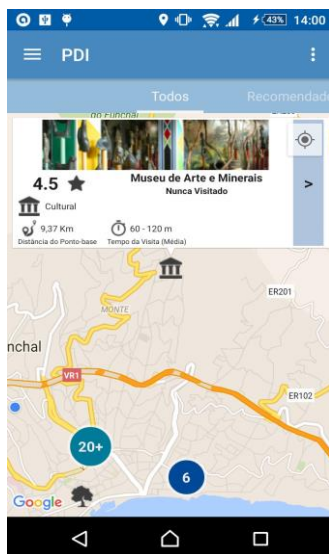
- Secção PDI -> Separador Todos;
- Secção Roteiros-> Separador Todos;
- Secção Definições;

***Obrigatório**

Apresentação dos pontos de interesse (PDI) - Mapa

Ao iniciar a aplicação é possível seleccionar um menu que apresenta informação acerca dos pontos de interesse (PDI) existentes sobre um mapa.

Menu dos pontos de interesse sobre o mapa



1. Como avalia a apresentação dos pontos de interesse sobre o mapa? * Marcar apenas uma oval.

1 2 3 4 5

Muito Fraco Muito Bom

2. Qual o grau de dificuldade de interpretação dos ícones utilizados nos pontos de interesse? * Marcar apenas uma oval.

1 2 3 4 5

Difícil interpretação Fácil interpretação

3. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos pontos de interesse no mapa?

.....

.....

.....

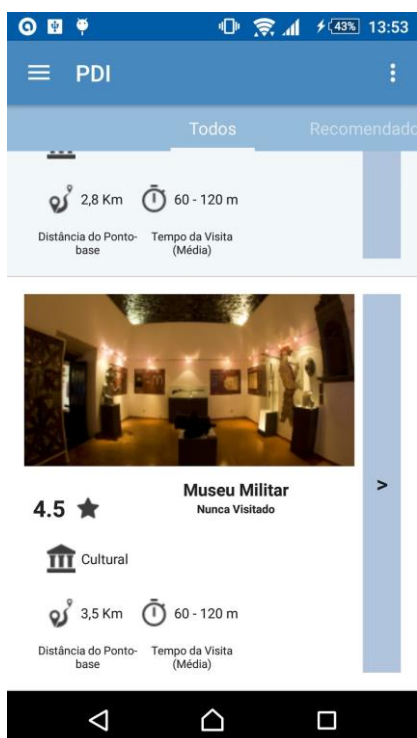
.....

.....

Apresentação dos pontos de interesse (PDI) - Lista

Ao seleccionar as Definições tem a opção de alternar entre diversos tipos de vista, sendo possível escolher uma vista designada por lista. Ao seleccionar esta opção os pontos de interesse irão passar a ser apresentados como uma lista de itens.

Menu dos pontos de interesse em lista



4. Como avalia a apresentação dos pontos de interesse em lista? * Marcar apenas uma oval.

1 2 3 4 5

Muito Fraco Muito Bom

5. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos pontos de interesse em lista?

.....

.....

.....

.....

.....

Apresentação dos detalhes dos pontos de interesse

Ao seleccionar cada um dos pontos de interesse ser-lhe-á apresentado um conjunto de dados detalhados acerca desse mesmo local.

Menu dos detalhes de um ponto de interesse - Parte 1



Menu dos detalhes de um ponto de interesse - Parte 2



Menu dos detalhes de um ponto de interesse - Parte 3



6. Como avalia a apresentação destas informações? * Marcar apenas uma oval.

	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

7. Como avalia o nível de detalhe da informação apresentada? * Marcar apenas uma oval.

	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

8. Qual o grau de dificuldade de interpretação da informação apresentada? * Marcar apenas uma oval.

	1	2	3	4	5	
Difícil interpretação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fácil interpretação

9. Qual o grau de importância de cada uma das informações apresentadas anteriormente? * Marcar apenas uma oval por linha.

	Irrelevante	Pouco relevante	Muito relevante	Imprescindível
Avaliação dos utilizadores	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Data da última visita	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Descrição do ponto de interesse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tempo médio de visita	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Horário diário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dias de expediente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Intervalos diários	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Observações acerca do horário de expediente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Morada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Distância do ponto-base	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tempo de viagem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Custo das entradas Individuais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Custo das entradas para grupos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Observações acerca dos custos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dados acerca da acessibilidade para pessoas portadoras de deficiência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Contactos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fontes de informação externas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação e qualidade destes dados?

.....

.....

.....

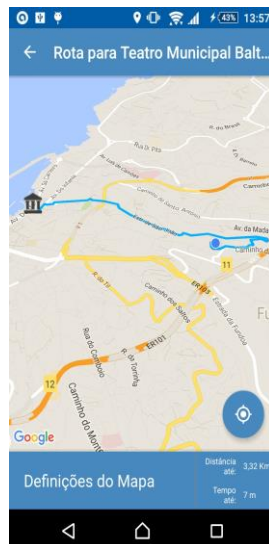
.....

.....

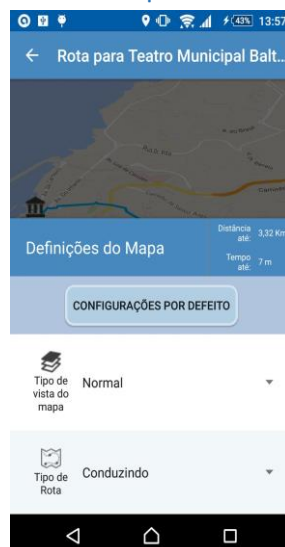
Apresentação de rotas até ao ponto de interesse.

Dentro dos detalhes de um ponto de interesse é possível seleccionar uma opção que permite criar uma rota para o referido ponto de interesse. (Botão circular no canto superior esquerdo).

Apresentação da rota até ao ponto de interesse - Parte 1



Apresentação da rota até ao ponto de interesse - Parte 2



11. Como avalia a apresentação destas informações? * Marcar apenas uma oval.

1 2 3 4 5

Muito Fraco Muito Bom

12. Como avalia a eficácia da rota obtida para chegar ao local pretendido? Marcar apenas uma oval.

1 2 3 4 5

Difícil orientação Fácil orientação

13. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos dados da rota?

.....

.....

.....

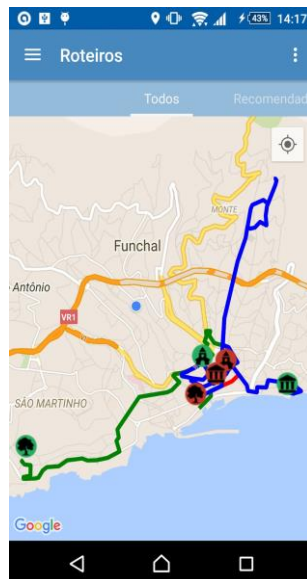
.....

.....

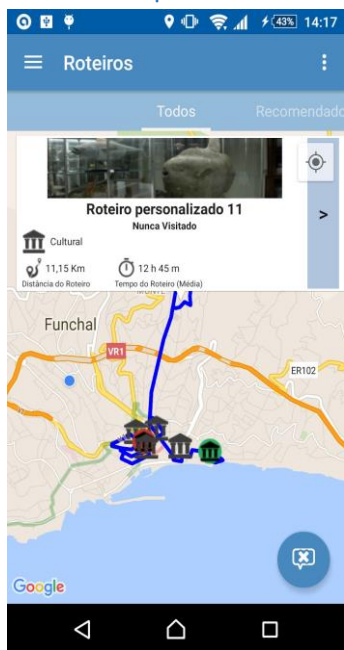
Apresentação dos Roteiros - Mapa

Ao voltar ao menu inicial dos Pontos de interesse é possível seleccionar, no menu lateral esquerdo, a opção de Roteiros. Para obter a vista em mapa é necessário novamente voltar às definições da aplicação e modificar o tipo de vista seleccionado.

Menu dos roteiros sobre o mapa - Parte 1



Menu dos roteiros sobre o mapa - Parte 2



14. Como avalia a apresentação dos roteiros sobre o mapa? * Marcar apenas uma oval.

1 2 3 4 5

Muito Fraco Muito Bom

15. Qual o grau de dificuldade de interpretação dos ícones utilizados nos roteiros? * Marcar apenas uma oval.

1 2 3 4 5

Difícil interpretação Fácil interpretação

16. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos roteiros sobre o mapa?

.....

.....

.....

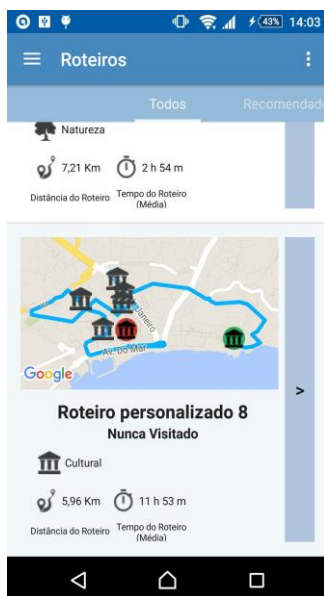
.....

.....

Apresentação dos Roteiros - Lista

Ao selecionar as Definições têm a opção de alternar entre diversos tipos de vista, sendo possível escolher uma vista designada por lista. Ao selecionar esta opção os roteiros irão passar a ser apresentados como uma lista de itens.

Menu dos roteiros em lista



17. Como avalia a apresentação dos roteiros em lista? * Marcar apenas uma oval.

1 2 3 4 5

Muito Fraco Muito Bom

18. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos roteiros em lista?

.....

.....

.....

.....

.....

Apresentação dos detalhes dos roteiros

Ao selecionar cada um dos roteiros ser-lhe-á apresentado um conjunto de dados detalhados acerca desse mesmo roteiro.

Menu dos detalhes de um roteiro - Parte 1



Menu dos detalhes de um roteiro - Parte 2



Menu dos detalhes de um roteiro - Parte 3



19. Como avalia a apresentação destas informações? * Marcar apenas uma oval.

	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

20. Como avalia o nível de detalhe da informação apresentada? * Marcar apenas uma oval.

	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

21. Qual o grau de dificuldade de interpretação da informação apresentada? * Marcar apenas uma oval.

	1	2	3	4	5	
Difícil interpretação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fácil interpretação

22. Qual o grau de importância de cada uma das informações apresentadas anteriormente? * Marcar apenas uma oval por linha.

	Irrelevante	Pouco relevante	Muito relevante	Imprescindível
Data do ultimo percurso pelo roteiro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Descrição do utilizador acerca do roteiro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mapa das localizações dos pontos de interesse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Informação acerca dos pontos de interesse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tempo médio do percurso do roteiro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Horário diário	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dias de expediente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Intervalos diários	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Custo do roteiro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acessibilidade para pessoas portadoras de deficiência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

23. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação e qualidade destes dados?

.....

.....

.....

.....

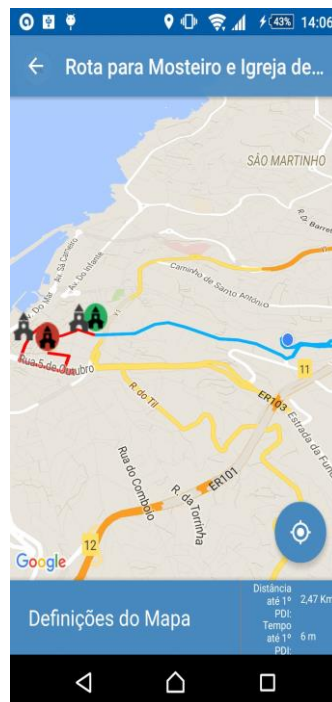
.....

Apresentação de rotas até aos pontos de interesse de um roteiro.

Dentro dos detalhes de um roteiro é possível selecionar uma opção que permite criar rotas, sequencialmente, para cada um dos pontos de interesse associados com o roteiro (Botão circular no canto superior esquerdo).

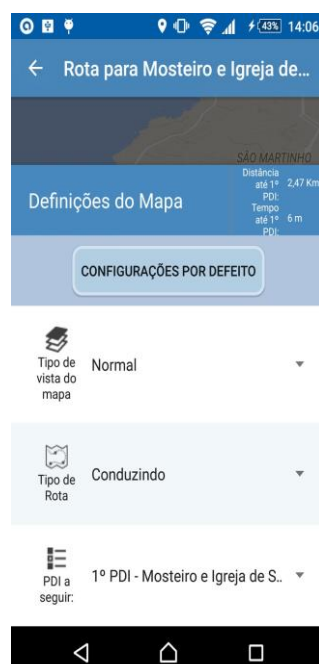
Apresentação da rota até aos pontos de interesse do roteiro

Parte 1



Apresentação da rota até aos pontos de interesse do roteiro

Parte 2



24. Como avalia a apresentação destas informações? * Marcar apenas uma oval.

	1	2	3	4	5	
Muito Fraco	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muito Bom

25. Como avalia a eficácia das rotas obtidas para percorrer o roteiro pretendido? Marcar apenas uma oval.

	1	2	3	4	5	
Difícil orientação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fácil orientação

26. Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos dados das rotas?

.....

.....

.....

.....

.....

Perguntas acerca do desempenho da aplicação

27. Como classifica os tempos de carregamento das diferentes secções da aplicação? * Marcar apenas uma oval.

	1	2	3	4	5	
Bastante lento	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante rápido

28. Existem algumas secções que considere serem especialmente lentas? Se sim, identifique quais as secções.

Marcar tudo o que for aplicável.

- Processo de carregamento inicial da
- aplicação
-
- Menu principal PDI em mapa.
- Menu principal PDI em lista
- Menu de detalhes de cada PDI
- Menu de rotas de cada PDI
- Menu principal Roteiros em mapa.
- Menu principal Roteiros em lista
- Menu de detalhes de cada roteiro
- Menu de rotas de cada roteiro
- Definições

Outra:

29. Identifique o número de erros ocorridos durante o uso da aplicação.

* Marcar apenas uma oval.

- Não ocorreram erros
- 1 - 2 erros
- 2 - 5 erros
- + 5 erros
- Não consegui usar a aplicação devido à ocorrência de erros ao iniciar a aplicação

30. Existem algumas secções que durante a sua utilização apresentaram erros de forma persistente? Se sim, identifique quais as secções. Marcar tudo o que for aplicável.

- Processo de carregamento inicial da
- aplicação
-
- Menu principal PDI em mapa.
- Menu principal PDI em lista
- Menu de detalhes de cada PDI
- Menu de rotas de cada PDI
-
- Menu principal Roteiros em mapa.
- Menu principal Roteiros em lista
- Menu de detalhes de cada roteiro
- Menu de rotas de cada roteiro

Definições

Outra:

31. Existem outros aspetos que considere que prejudicam o desempenho da aplicação? Se sim, identifique quais?

.....

.....

.....

.....

.....

32. Que aspetos considera que devam ser melhorados quando a aplicação é utilizada sem ligação à Internet?

.....

.....

.....

.....

.....

Perguntas acerca de outras funcionalidades a definir.

Futuramente pretendesse apresentar um conjunto mais alargado de funcionalidades, sendo que o seguinte conjunto de perguntas tem o intuito de definir as prioridades para as atuais e futuras funcionalidades a definir.

33. Defina o nível de importância de cada uma das funcionalidades apresentadas anteriormente. * Marcar apenas uma oval por linha.

	Irrelevante	Pouco relevante	Muito relevante	Imprescindível
Apresentação dos pontos de interesse em diversos tipos de vista (mapa e lista)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentação dos detalhes dos pontos de interesse com respetiva informação sobre os mesmos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Criação de diversos tipos de rota até ao ponto de interesse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentação dos roteiros em diversos tipos de vista (mapa e lista)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentação dos detalhes dos roteiros com respetiva informação sobre os mesmos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Criação de diversos tipos de rota até cada um dos pontos de interesse dos referidos roteiros	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

34 - Defina o nível de importância de cada uma das funcionalidades que deverão ser desenvolvidas futuramente. *

Marcar apenas uma oval por linha.

	Irrelevante	Pouco relevante	Muito relevante	Imprescindível
Criação de roteiros personalizados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Agendamento das visitas aos pontos de interesse e roteiros	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Seleção dos pontos de interesse favoritos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Seleção dos roteiros favoritos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Avaliação dos pontos de interesse (1 a 5)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Criação e visualização de opiniões/comentários acerca dos pontos de interesse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Filtragem dos pontos de interesse pelos mais visitados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Recomendação de pontos de interesse tendo por base os interesses do utilizador	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Recomendação de roteiros tendo por base os interesses do utilizador	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Histórico dos pontos de interesse visitados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Histórico dos roteiros concluídos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Associação de fotografias/vídeos pessoais aos pontos de interesse visitados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Associação de fotografias/vídeos pessoais aos roteiros efetuados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pesquisa seletiva dos pontos de interesse tendo em conta: custos, horários, distância, etc.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pesquisa seletiva de roteiros tendo em conta: custos, horários, distância, etc.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

35. Identifique outras funcionalidades que considere relevantes que não foram mencionadas anteriormente.

.....

.....

.....

.....

.....

Anexo D – Respostas aos questionários

Pergunta #1

Como avalia a apresentação dos pontos de interesse sobre o mapa?

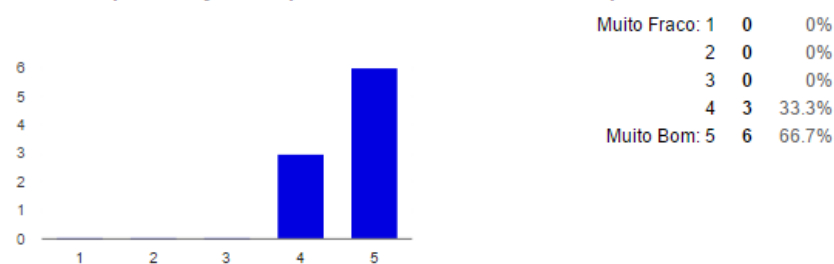


Figura 59 - Gráfico dos resultados da pergunta #1 do inquérito acerca da aplicação

Pergunta #2

Qual o grau de dificuldade de interpretação dos ícones utilizados nos pontos de interesse?

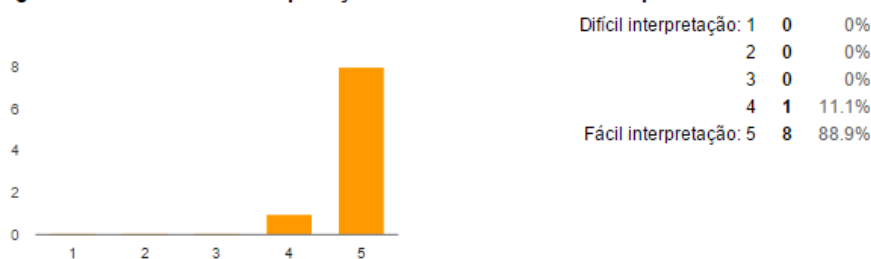


Figura 60 - Gráfico dos resultados da pergunta #2 do inquérito acerca da aplicação

Pergunta #3

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos pontos de interesse no mapa?

Resumo das respostas obtidas:

- Não.

Pergunta #4

Como avalia a apresentação dos pontos de interesse em lista?

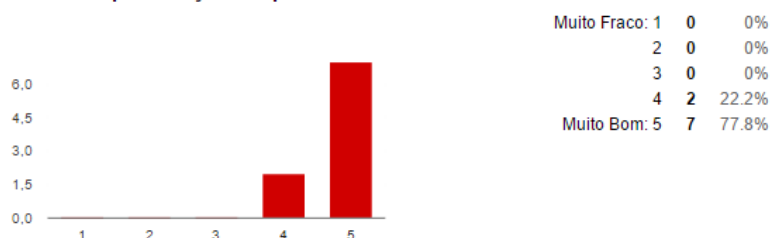


Figura 61 - Gráfico dos resultados da pergunta #4 do inquérito acerca da aplicação

Pergunta #5

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos pontos de interesse em lista?

Resumo das respostas obtidas:

- Não.
- Sim, deveria ser mais fácil a mudança entre vistas.

Pergunta #6

Como avalia a apresentação destas informações?

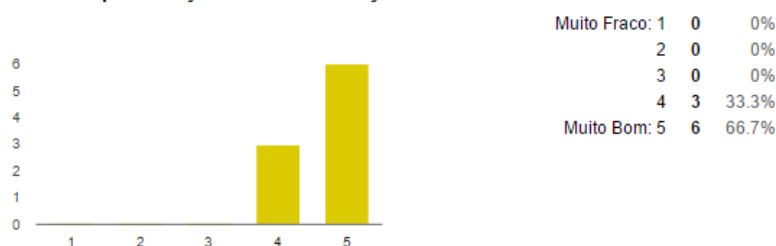


Figura 62 - Gráfico dos resultados da pergunta #6 do inquérito acerca da aplicação

Pergunta #7

Como avalia o nível de detalhe da informação apresentada?

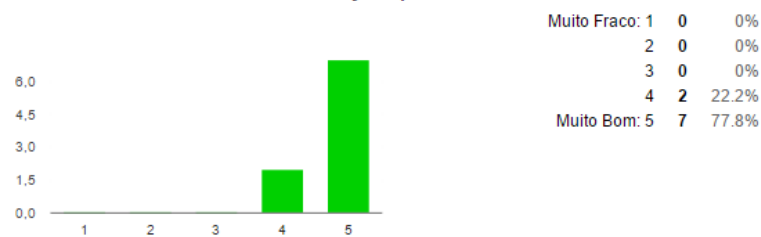


Figura 63 - Gráfico dos resultados da pergunta #7 do inquérito acerca da aplicação

Pergunta #8

Qual o grau de dificuldade de interpretação da informação apresentada

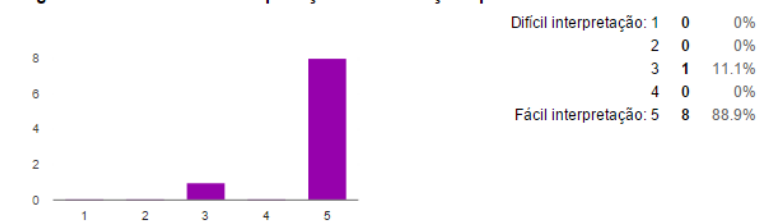


Figura 64 - Gráfico dos resultados da pergunta #8 do inquérito acerca da aplicação

Pergunta #9

Avaliação dos utilizadores [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]



Figura 65 - Gráfico dos resultados da pergunta #9 do inquérito acerca da aplicação

Pergunta #10

Data da última visita [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]

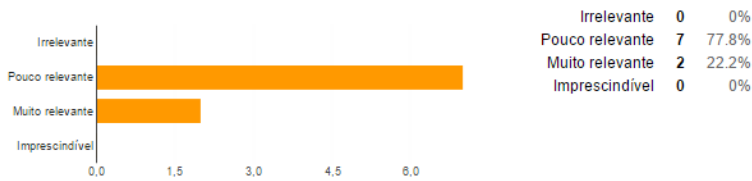


Figura 66 - Gráfico dos resultados da pergunta #10 do inquérito acerca da aplicação

Pergunta #11

Horário diário [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]

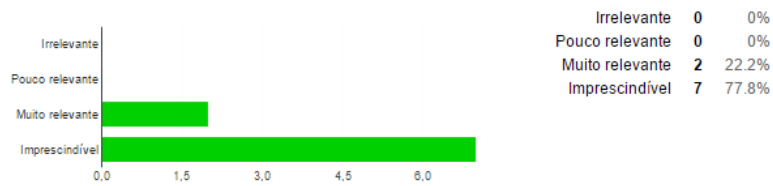


Figura 67 - Gráfico dos resultados da pergunta #11 do inquérito acerca da aplicação

Pergunta #12

Dias de expediente [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]

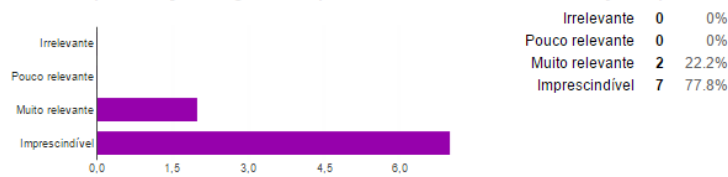


Figura 68 - Gráfico dos resultados da pergunta #12 do inquérito acerca da aplicação

Pergunta #13

Intervalos diários [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]



Figura 69 - Gráfico dos resultados da pergunta #13 do inquérito acerca da aplicação

Pergunta #14

Observações acerca do horário de expediente [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]



Figura 70 - Gráfico dos resultados da pergunta #14 do inquérito acerca da aplicação

Pergunta #15



Figura 71 - Gráfico dos resultados da pergunta #15 do inquérito acerca da aplicação

Pergunta #16

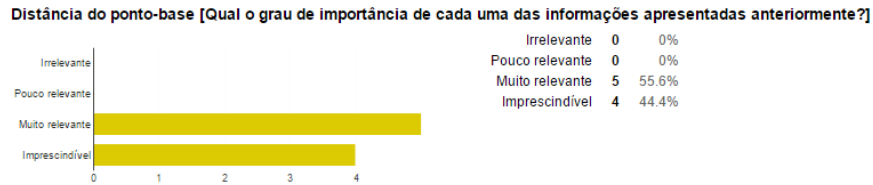


Figura 72 - Gráfico dos resultados da pergunta #16 do inquérito acerca da aplicação

Pergunta #17

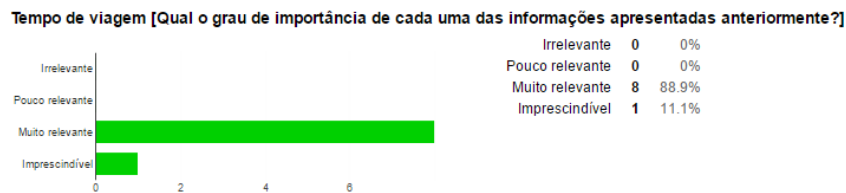


Figura 73 - Gráfico dos resultados da pergunta #17 do inquérito acerca da aplicação

Pergunta #18

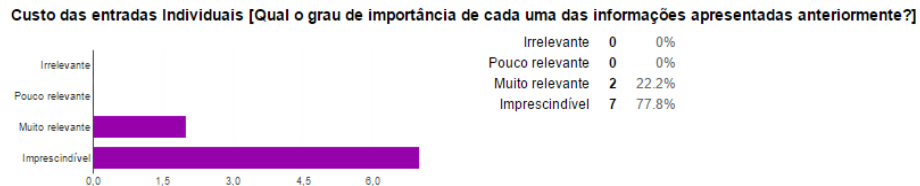


Figura 74 - Gráfico dos resultados da pergunta #18 do inquérito acerca da aplicação

Pergunta #19

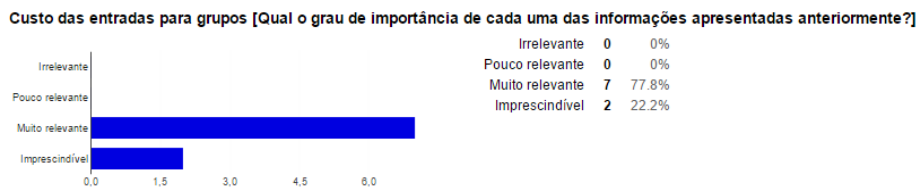


Figura 75 - Gráfico dos resultados da pergunta #19 do inquérito acerca da aplicação

Pergunta #20

Observações acerca dos custos [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]

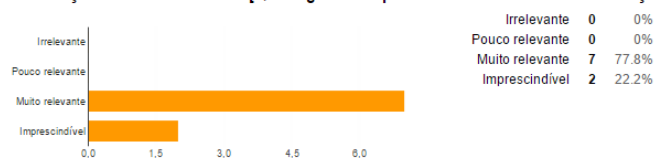


Figura 76 - Gráfico dos resultados da pergunta #20 do inquérito acerca da aplicação

Pergunta #21

Dados acerca da acessibilidade para pessoas portadoras de deficiência [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]



Figura 77 - Gráfico dos resultados da pergunta #21 do inquérito acerca da aplicação

Pergunta #22

Contactos [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]

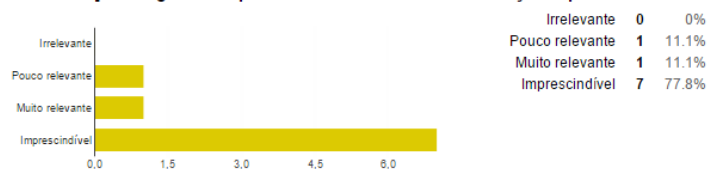


Figura 78 - Gráfico dos resultados da pergunta #22 do inquérito acerca da aplicação

Pergunta #23

Fontes de informação externas [Qual o grau de importância de cada uma das informações apresentadas anteriormente?]

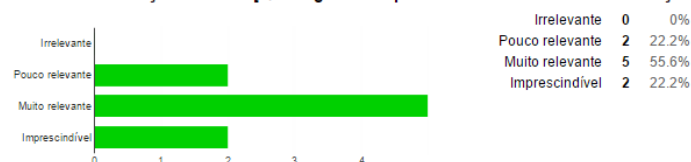


Figura 79 - Gráfico dos resultados da pergunta #23 do inquérito acerca da aplicação

Pergunta #24

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação e qualidade destes dados?

Resumo das respostas obtidas:

- Não.

Pergunta #25

Como avalia a apresentação destas informações?

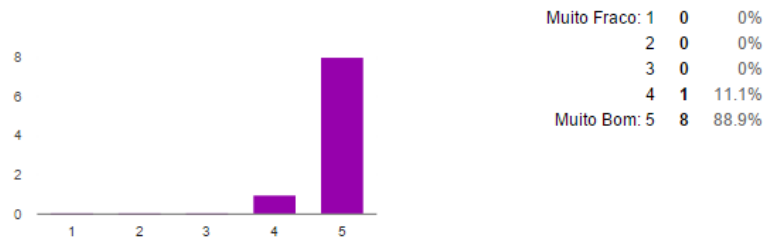


Figura 80 - Gráfico dos resultados da pergunta #25 do inquérito acerca da aplicação

Pergunta #26

Como avalia a eficácia da rota obtida para chegar ao local pretendido?

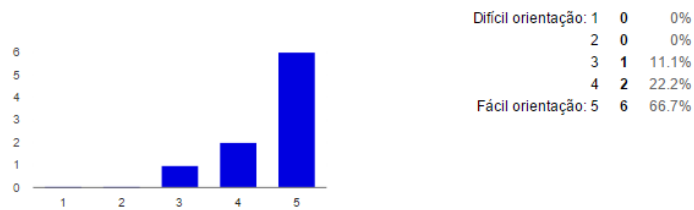


Figura 81 - Gráfico dos resultados da pergunta #26 do inquérito acerca da aplicação

Pergunta #27

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos dados da rota?

Resumo das respostas obtidas:

- Não.
- As informações da distância e tempo devem ser um pouco maiores.

Pergunta #28

Como avalia a apresentação dos roteiros sobre o mapa?

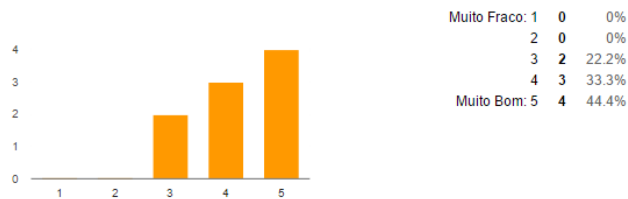


Figura 82 - Gráfico dos resultados da pergunta #28 do inquérito acerca da aplicação

Pergunta #29

Qual o grau de dificuldade de interpretação dos ícones utilizados nos roteiros?

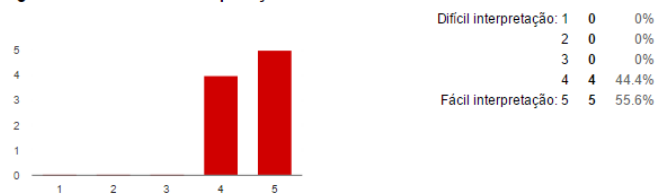


Figura 83 - Gráfico dos resultados da pergunta #29 do inquérito acerca da aplicação

Pergunta #30

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos roteiros sobre o mapa?

Resumo das respostas obtidas:

- Não.

Pergunta #31

Como avalia a apresentação dos roteiros em lista?



Figura 84 - Gráfico dos resultados da pergunta #31 do inquérito acerca da aplicação

Pergunta #32

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos roteiros em lista?

Resumo das respostas obtidas:

- Não.

Pergunta #33

Como avalia a apresentação destas informações?

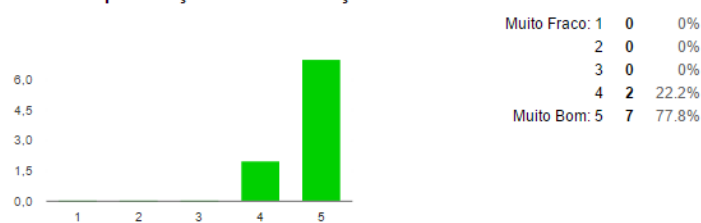


Figura 85 - Gráfico dos resultados da pergunta #33 do inquérito acerca da aplicação

Pergunta #34

Como avalia o nível de detalhe da informação apresentada?



Figura 86 - Gráfico dos resultados da pergunta #34 do inquérito acerca da aplicação

Pergunta #35



Figura 87 - Gráfico dos resultados da pergunta #35 do inquérito acerca da aplicação

Pergunta #36

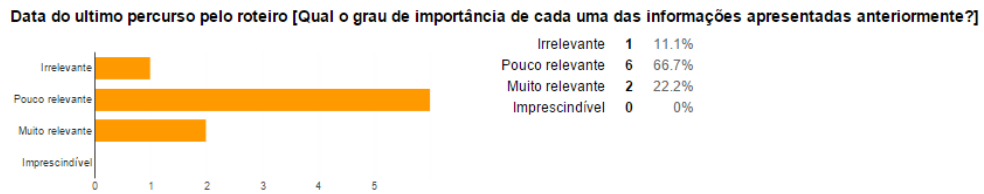


Figura 88 - Gráfico dos resultados da pergunta #36 do inquérito acerca da aplicação

Pergunta #37

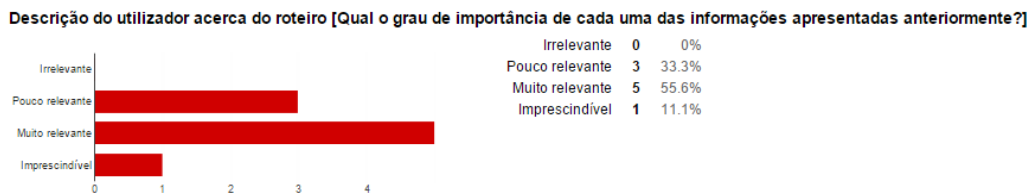


Figura 89 - Gráfico dos resultados da pergunta #37 do inquérito acerca da aplicação

Pergunta #38

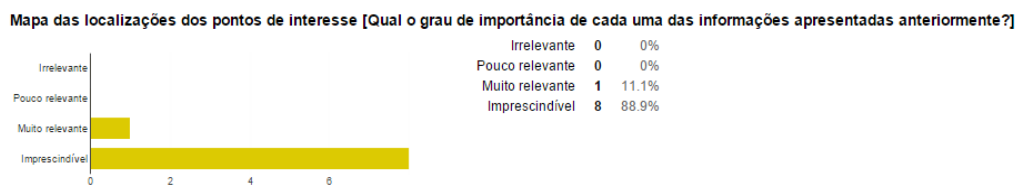


Figura 90 - Gráfico dos resultados da pergunta #38 do inquérito acerca da aplicação

Pergunta #39

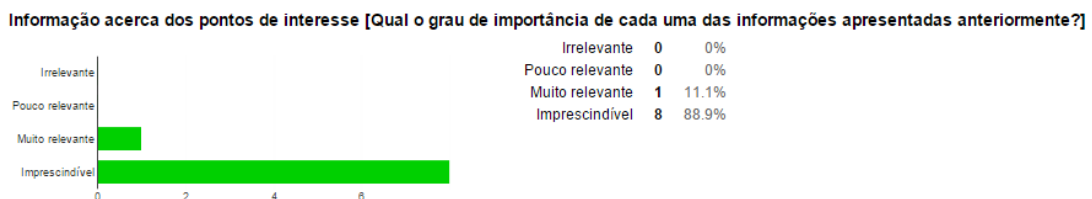


Figura 91 - Gráfico dos resultados da pergunta #39 do inquérito acerca da aplicação

Pergunta #40

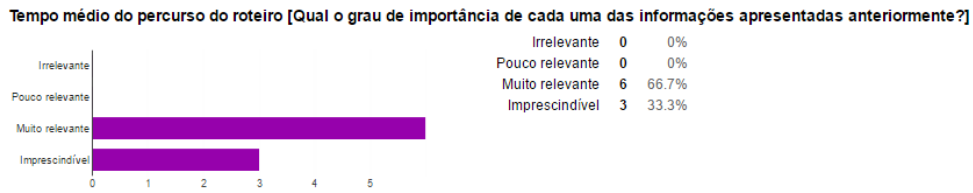


Figura 92 - Gráfico dos resultados da pergunta #40 do inquérito acerca da aplicação

Pergunta #41

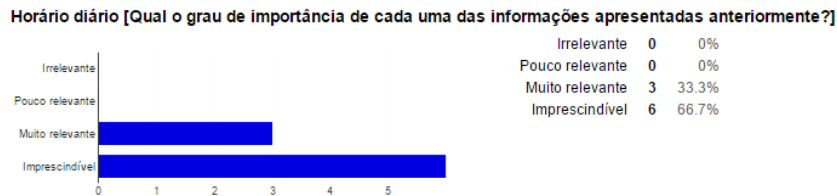


Figura 93 - Gráfico dos resultados da pergunta #41 do inquérito acerca da aplicação

Pergunta #42

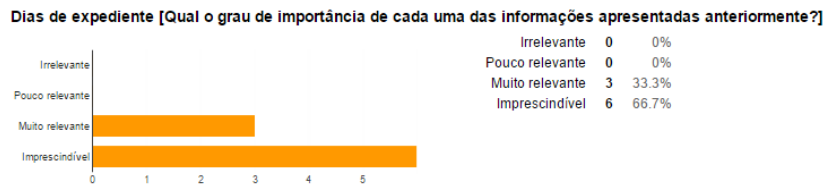


Figura 94 - Gráfico dos resultados da pergunta #42 do inquérito acerca da aplicação

Pergunta #43

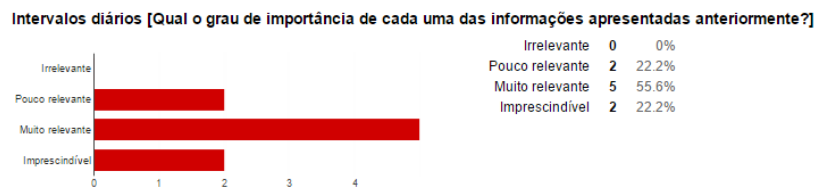


Figura 95 - Gráfico dos resultados da pergunta #43 do inquérito acerca da aplicação

Pergunta #44

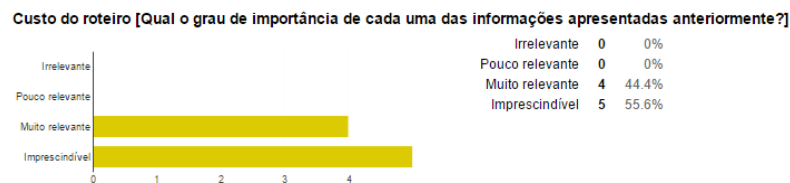


Figura 96 - Gráfico dos resultados da pergunta #44 do inquérito acerca da aplicação

Pergunta #45

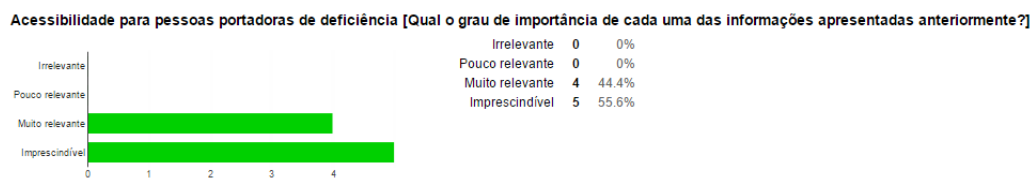


Figura 97 - Gráfico dos resultados da pergunta #45 do inquérito acerca da aplicação

Pergunta #46

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação e qualidade destes dados?

Resumo das respostas obtidas:

- Não.

Pergunta #47



Figura 98 - Gráfico dos resultados da pergunta #47 do inquérito acerca da aplicação

Pergunta #48

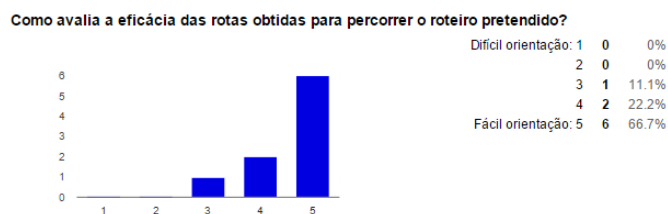


Figura 99 - Gráfico dos resultados da pergunta #48 do inquérito acerca da aplicação

Pergunta #49

Existe algum aspeto particular que deva ser alterado, de modo a melhorar a apresentação dos dados das rotas?

Resumo das respostas obtidas:

- Não.

Pergunta #50

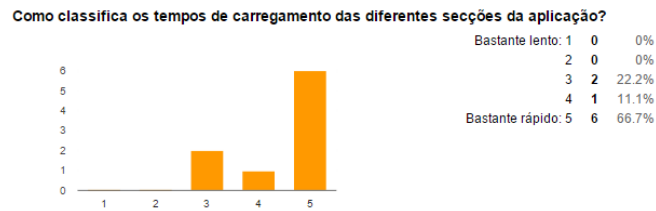


Figura 100 - Gráfico dos resultados da pergunta #50 do inquérito acerca da aplicação

Pergunta #51

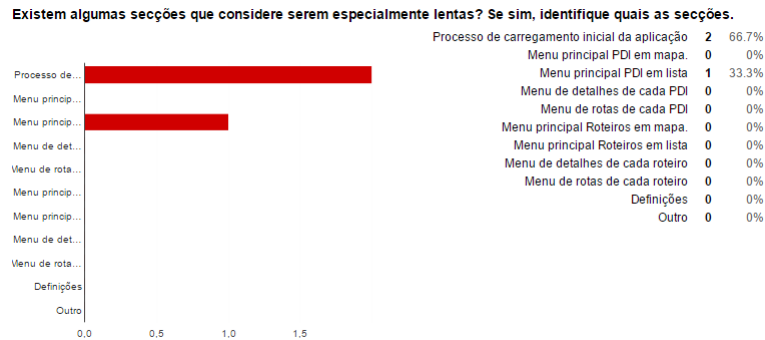


Figura 101 - Gráfico dos resultados da pergunta #51 do inquérito acerca da aplicação

Pergunta #52

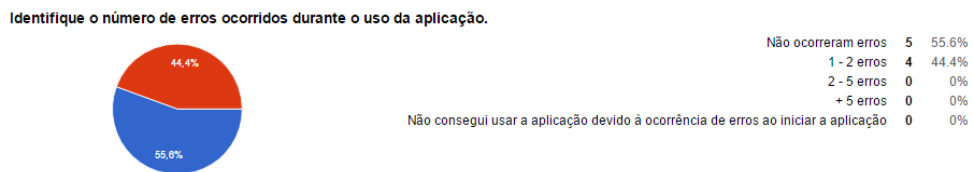


Figura 102 - Gráfico dos resultados da pergunta #52 do inquérito acerca da aplicação

Pergunta #53

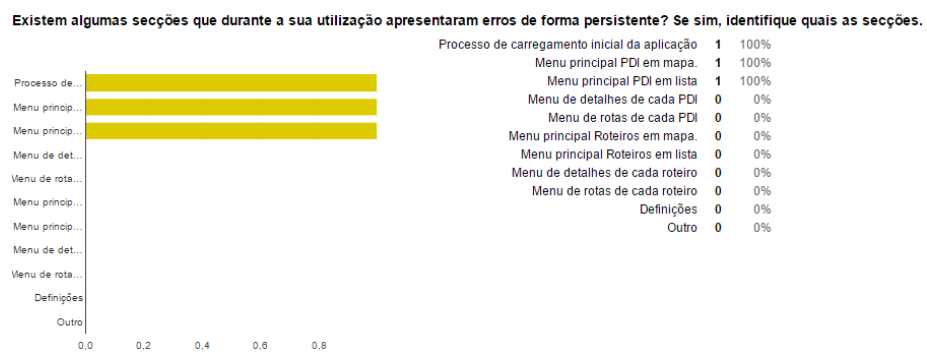


Figura 103 - Gráfico dos resultados da pergunta #53 do inquérito acerca da aplicação

Pergunta #54

Existem outros aspetos que considere que prejudicam o desempenho da aplicação?
Se sim, identifique quais?

Resumo das respostas obtidas:

- Não.

Pergunta #55

Que aspetos considera que devam ser melhorados quando a aplicação é utilizada sem ligação à Internet?

Resumo das respostas obtidas:

- Algumas fotos não carregam.

Pergunta #56

Apresentação dos pontos de interesse em diversos tipos de vista (mapa e lista) [Defina o nível de importância de cada uma das funcionalidades apresentadas anteriormente.]

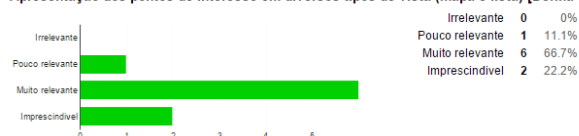


Figura 104 - Gráfico dos resultados da pergunta #56 do inquérito acerca da aplicação

Pergunta #57

Apresentação dos detalhes dos pontos de interesse com respetiva informação sobre os mesmos [Defina o nível de importância de cada uma das funcionalidades apresentadas anteriormente.]



Figura 105 - Gráfico dos resultados da pergunta #57 do inquérito acerca da aplicação

Pergunta #58

Criação de diversos tipos de rota até ao ponto de interesse [Defina o nível de importância de cada uma das funcionalidades apresentadas anteriormente.]

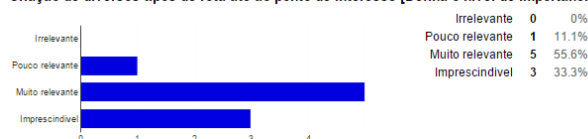


Figura 106 - Gráfico dos resultados da pergunta #58 do inquérito acerca da aplicação

Pergunta #59

Apresentação dos roteiros em diversos tipos de vista (mapa e lista) [Defina o nível de importância de cada uma das funcionalidades apresentadas anteriormente.]



Figura 107 - Gráfico dos resultados da pergunta #59 do inquérito acerca da aplicação

Pergunta #60

Apresentação dos detalhes dos roteiros com respetiva informação sobre os mesmos [Defina o nível de importância de cada uma das funcionalidades apresentadas anteriormente.]



Figura 108 - Gráfico dos resultados da pergunta #60 do inquérito acerca da aplicação

Pergunta #61



Figura 109 - Gráfico dos resultados da pergunta #61 do inquérito acerca da aplicação

Pergunta #62

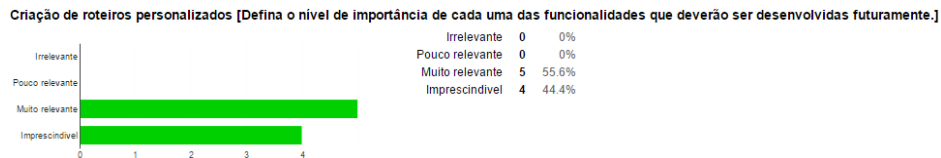


Figura 110 - Gráfico dos resultados da pergunta #62 do inquérito acerca da aplicação

Pergunta #63

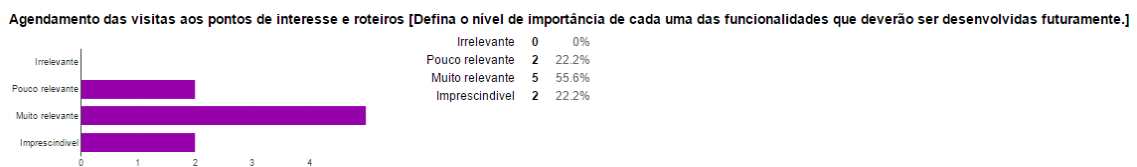


Figura 111 - Gráfico dos resultados da pergunta #63 do inquérito acerca da aplicação

Pergunta #64

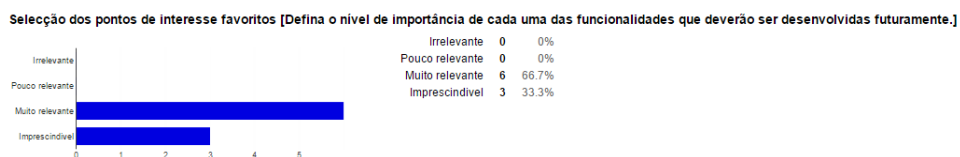


Figura 112 - Gráfico dos resultados da pergunta #64 do inquérito acerca da aplicação

Pergunta #65

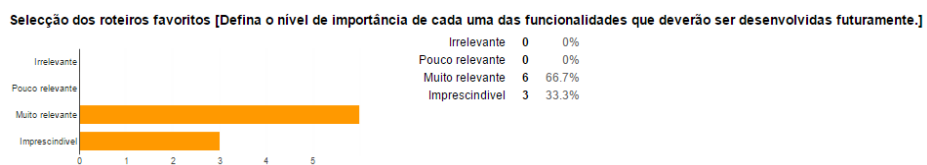


Figura 113 - Gráfico dos resultados da pergunta #65 do inquérito acerca da aplicação

Pergunta #66

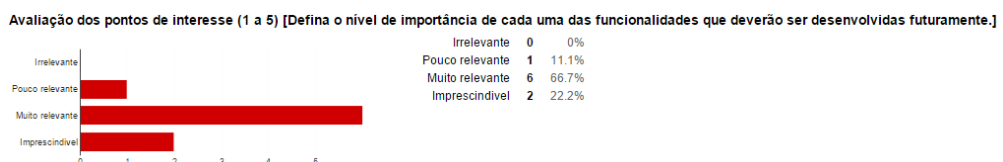


Figura 114 - Gráfico dos resultados da pergunta #66 do inquérito acerca da aplicação

Pergunta #67

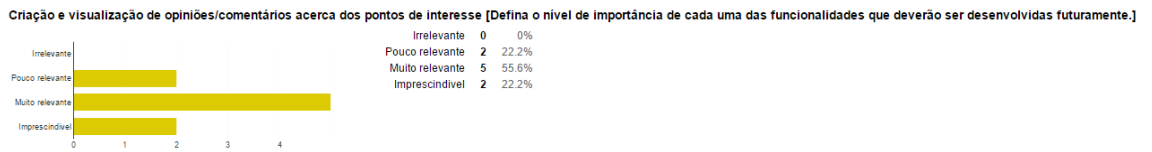


Figura 115 - Gráfico dos resultados da pergunta #67 do inquérito acerca da aplicação

Pergunta #68



Figura 116 - Gráfico dos resultados da pergunta #68 do inquérito acerca da aplicação

Pergunta #69

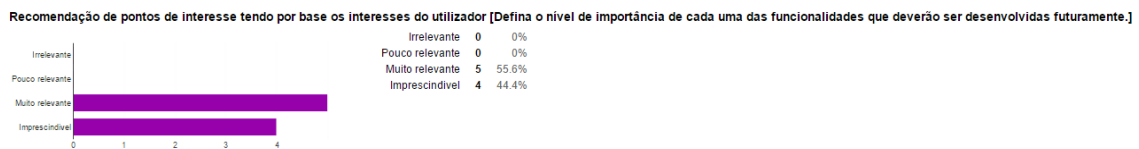


Figura 117 - Gráfico dos resultados da pergunta #69 do inquérito acerca da aplicação

Pergunta #70

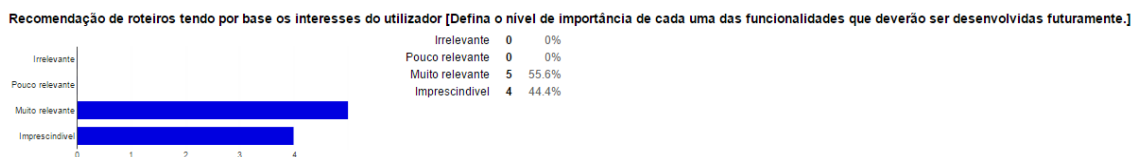


Figura 118 - Gráfico dos resultados da pergunta #70 do inquérito acerca da aplicação

Pergunta #71

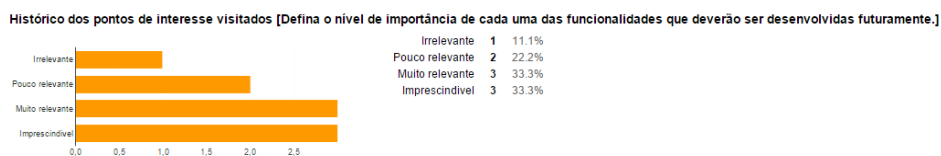


Figura 119 - Gráfico dos resultados da pergunta #71 do inquérito acerca da aplicação

Pergunta #72

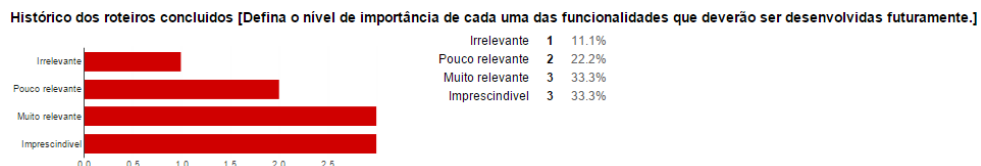


Figura 120 - Gráfico dos resultados da pergunta #72 do inquérito acerca da aplicação

Pergunta #73

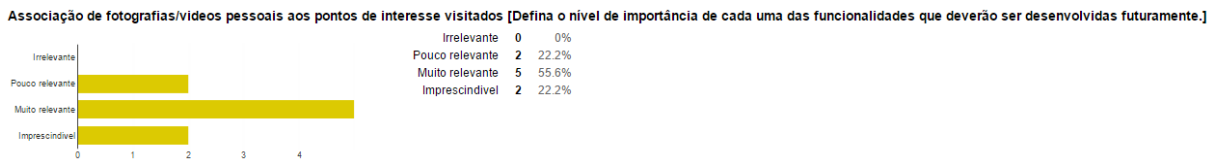


Figura 121 - Gráfico dos resultados da pergunta #73 do inquérito acerca da aplicação

Pergunta #74

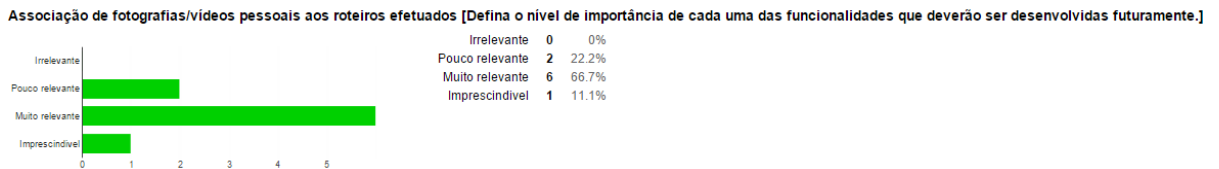


Figura 122 - Gráfico dos resultados da pergunta #74 do inquérito acerca da aplicação

Pergunta #75

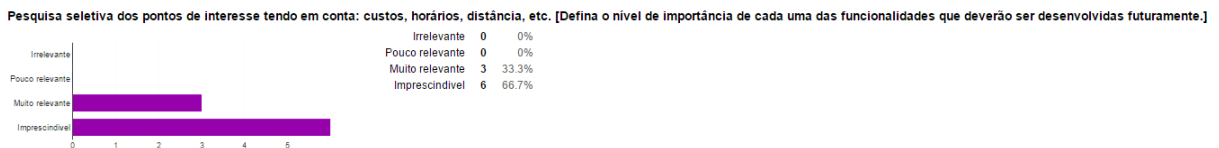


Figura 123 - Gráfico dos resultados da pergunta #75 do inquérito acerca da aplicação

Pergunta #76

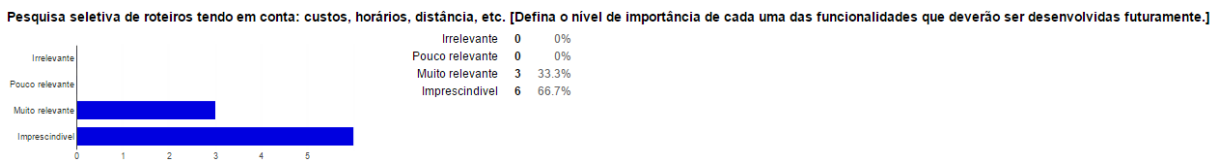


Figura 124 - Gráfico dos resultados da pergunta #76 do inquérito acerca da aplicação