



GeoTáxi ? Desenvolvimento de plataforma de recepção e atribuição de serviços de Táxis em pequenas localidades

TIAGO MANUEL DA CUNHA SILVA

novembro de 2017

GEO TÁXI – DESENVOLVIMENTO DE PLATAFORMA DE RECEÇÃO E ATRIBUIÇÃO DE SERVIÇOS DE TÁXIS EM PEQUENAS LOCALIDADES

Tiago Manuel Cunha Silva



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2017

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores no ramo de Telecomunicações

Candidato: Tiago Manuel Cunha Silva, N° 1120762, 1120762@isep.ipp.pt

Orientação científica: Carlos Ribeiro Campos, crc@isep.ipp.pt

Empresa: *Gisgeo – Information System*

Orientação na Empresa: Ricardo Baptista, ricardo.baptista@gisgeo.pt



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

31 de outubro de 2017

Agradecimentos

Tenho que agradecer ao *mui nobre* Instituto Superior de Engenharia do Porto, por estes últimos anos, pela formação prestada e pela oportunidade e condições proporcionadas para aprender engenharia e pelas pessoas com quem me fez interagir. Algumas delas considero amigos para a vida.

Agradeço à empresa *Gisgeo – Information System* e todos os seus colaboradores, por todo o apoio e disponibilidade prestada durante a realização do estágio.

Agradeço aos meus orientadores deste projeto, o Eng. Ricardo Baptista e o Professor Carlos Campos, por todo o acompanhamento, disponibilidade e partilha de conhecimento concedida durante este trabalho.

A nível pessoal, gostaria de agradecer à minha família, por todo o apoio prestado ao longo de todo o percurso académico que fez com que este dia fosse possível.

Por fim, agradeço a todas as pessoas que estiveram comigo até hoje, por me ensinarem a ser quem sou hoje.

Resumo

As inovações tecnológicas são por natureza impulsionadoras de modificações em diversas áreas. Um dos setores cujo desenvolvimento tem sido notório e muito destacada pela comunicação social nacional e internacional é o do Táxi, serviço público de transporte. Todavia nem todas as soluções existentes respondem às necessidades do pedido de serviço de táxi, especialmente para pequenas e médias localidades de Portugal. Nestes contextos específicos, a utilização de táxis tem características muito próprias que devem ser tidas em conta, como por exemplo: a gestão automática dos pedidos de serviços.

Todas as aplicações existentes para a requisição de um táxi inserem-se nas grandes áreas metropolitanas do país, o que faz com que nas pequenas e médias localidades se tenha de recorrer aos meios antigos de requisição, quer por chamada telefónica, quer por deslocação física à praça mais próxima.

A solução apresentada neste documento, denominada de GeoTáxi, tem como objetivo colmatar esta falta de soluções para as pequenas e médias localidades. Para isso foi desenvolvido um sistema de receção e atribuição de serviços de táxis, constituído por duas aplicações para dispositivos móveis (uma para o cliente e outra para o taxista) com o sistema operativo Android e uma aplicação *Web* para a administração da plataforma. Por último, com o intuito de interligar estas três aplicações (duas móveis e uma *Web*) e a base de dados tem-se a plataforma automática de gestão.

Os testes realizados no decorrer do desenvolvimento do produto e o *feedback* dado pelos colaboradores da empresa, validaram o seu correto funcionamento, estando pronta para a realização de testes piloto em carga na cidade de Chaves.

Palavras-Chave

Táxi, Android, PHP, PostgreSQL

Abstract

Technological innovations are conducive to changes in several areas. One of the sectors whose revolution has been notorious is the Taxi, public transport service. However, not all existing solutions respond to the needs of the taxi service request especially for small and medium cities in Portugal. In these specific contexts, the use of taxis has very specific characteristics that must be considered in the automatic management of service requests.

All applications for requesting an existing taxi are inserted in large metropolitan areas of the country, which means in small and medium cities have to resort to the old means of requisition, which were the telephone call, or else, the trip to the nearest square.

The solution presented in this document, called GeoTáxi, aims to address this lack of solutions in small and medium localities. For this purpose, a system for receiving and assigning taxi services was created, which also consists of two mobile applications with the Android operating system and a Web application for the platform administrator. One such application is for customers and the other for taxi drivers. Finally, to interconnect these three applications and the database is the management platform.

The tests performed during the development of the product and the feedback given by the company's employees, validated its correct functioning, being ready for a pilot test, in the city of Chaves that is where it will be begins integrated.

Keywords

Taxi, Android, PHP, *PostgreSQL*

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XV
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	2
1.2. EMPRESA	3
1.3. OBJETIVOS.....	4
1.4. ORGANIZAÇÃO DO RELATÓRIO	5
2. SERVIÇO DE TÁXIS	7
2.1. SETOR DOS TÁXIS	8
2.2. APLICAÇÕES DE REQUISIÇÃO DE TRANSPORTE.....	9
3. TECNOLOGIAS UTILIZADAS	13
3.1. ANDROID	13
3.2. <i>WEB SERVICES</i>	22
3.3. <i>HYPER TEXT MARKUP LANGUAGE (HTML)</i>	24
3.4. <i>PHP (HYPERTEXT PREPROCESSOR)</i>	25
3.5. <i>JAVASCRIPT</i>	25
3.4. <i>JQUERY</i>	25
3.5. <i>OPENLAYERS</i>	26
3.6. <i>BOOTSTRAP</i>	26
3.7. <i>PUSH NOTIFICATIONS</i>	27
3.8. BASE DE DADOS	28
4. ANÁLISE E MODELAÇÃO DO SISTEMA	31
4.1. ANÁLISE DE REQUISITOS	31
4.2. CASO DE USO.....	33
4.3. MODELO DE DADOS.....	41
5. IMPLEMENTAÇÃO	49
5.1. ARQUITETURA DO SISTEMA	49

5.2.	PLATAFORMA DE GESTÃO	50
5.3.	APLICAÇÃO DO GESTOR	57
5.4.	APLICAÇÃO DO CLIENTE	61
5.5.	APLICAÇÃO DO TAXISTA	73
6.	TESTES E RESULTADOS	87
6.1.	APLICAÇÃO DO GESTOR	87
6.2.	APLICAÇÃO DO CLIENTE	90
6.3.	APLICAÇÃO DO TAXISTA	95
6.4.	<i>FEEDBACK</i> DA EQUIPA	100
7.	CONCLUSÕES E TRABALHO FUTURO	101
7.1.	CONSIDERAÇÕES FINAIS	102
7.2.	TRABALHO FUTURO	103
	REFERÊNCIAS DOCUMENTAIS	105

Índice de Figuras

Figura 1 – Distribuição do SDK Android [14].....	14
Figura 2 – Arquitetura do Android [15].....	15
Figura 3 - Funcionamento da pilha de <i>Activities</i> [16].....	17
Figura 4 - Ciclo de vida de uma <i>activity</i>	18
Figura 5 - REST <i>Web Service</i> [18].....	23
Figura 6 - Modelos <i>Pull</i> e <i>Push</i>	27
Figura 7 - Funcionamento da FCM [27].....	28
Figura 8 – Casos de uso do Gestor.....	34
Figura 9 – Casos de uso da aplicação do cliente.....	37
Figura 10 – Casos de uso da aplicação do taxista.....	39
Figura 11 – Modelo de Dados Compactado.....	42
Figura 12 – Tabelas Táxi, Equipa, Calendário e Praça de Táxis.....	46
Figura 13 – Cliente, Pedido, Serviço, Táxi.....	47
Figura 14 – Arquitetura do Sistema Desenvolvido.....	50
Figura 15 - Modelo de registo e envio de notificações da FCM.....	51
Figura 16 – Estados de um Serviço.....	53
Figura 17 – Transferência de um serviço.....	55
Figura 18 – Estados do algoritmo de atribuição de serviços.....	55
Figura 19 – Página de <i>Login</i> da Aplicação do Gestor.....	57
Figura 20 – Página Principal da Aplicação do Gestor.....	58
Figura 21 – Menu lateral da página principal.....	58
Figura 22 – Formulário de Inserção de uma Praça.....	59
Figura 23 - Janela de consulta das praças.....	59
Figura 24 – Consulta dos <i>Reports</i> Efetuados pelos Taxistas.....	60
Figura 25 - Mudança no tipo de mapa a apresentar.....	61
Figura 26 – Arquitetura da Aplicação do Cliente.....	62
Figura 27 – <i>Login Activity</i> do Cliente.....	62
Figura 28 – Registo <i>Activity</i>	63
Figura 29 – <i>Main Activity</i> do Cliente.....	63
Figura 30 - Janela de confirmação do pedido.....	65
Figura 31 – Menu lateral da <i>main activity</i>	65
Figura 32 - Perfil <i>Activity</i>	66
Figura 33 - Campos de Alteração da <i>password</i>	67
Figura 34 - Visualização da informação da aplicação (<i>About Activity</i> e <i>Licencas Activity</i>).....	68

Figura 35 - Mensagens de erro ao comunicar com a plataforma.....	68
Figura 36 - Mensagens no momento do registo	69
Figura 37 - Mensagens no momento do <i>login</i>	69
Figura 38 - Requisito de permissões	70
Figura 39 – Mensagens no momento do pedido.....	70
Figura 40 – Mensagens na edição dos dados pessoais	71
Figura 41 – Mensagens na edição da <i>password</i>	71
Figura 42 – Janelas quando o cliente esta sem internet ou GPS.....	72
Figura 43 – Base Dados da Aplicação do Cliente	72
Figura 44 – Arquitetura da Aplicação do Taxista	74
Figura 45 – <i>Login Activity</i> do Taxista	74
Figura 46 – <i>First Login Activiy</i>	75
Figura 47 - <i>Main Activity</i> do Taxista.....	75
Figura 48 – Disponível <i>Activity</i>	76
Figura 49 – Janela de confirmação da transferência.....	77
Figura 50 – Tempo de Espera Superado (Disponível <i>Activity</i>)	77
Figura 51 - Menu Lateral da <i>Main Activity</i>	78
Figura 52 - Perfil <i>Activity</i> do Taxista	80
Figura 53 – Mensagens de erro ao comunicar com a plataforma.....	81
Figura 54 - Mensagens no <i>Login</i>	81
Figura 55 - Mensagens na alteração da <i>password</i> após o 1º <i>login</i>	82
Figura 56 – Requisitos de Permissões da Aplicação	82
Figura 57 - Mensagem na efetuação da queixa ao cliente.....	82
Figura 58 – Mensagens no momento da transferência de serviço.....	83
Figura 59 - Praça atualizada com sucesso	83
Figura 60 - Mensagem de Visualização de uma transferência	84
Figura 61 – Mensagens na edição dos dados pessoais do taxista.....	84
Figura 62 - Mensagens na alteração da <i>password</i>	85
Figura 63 – Janela quando o utilizador está sem <i>internet</i>	85
Figura 64 – Base Dados da Aplicação do Taxista.....	86
Figura 65 – Teste de inserção de uma praça.....	88
Figura 66 – Teste de consulta e edição de uma praça	89
Figura 67 - Teste de visualização das praças da cidade	89
Figura 68 - Teste de visualização da localização dos táxis existentes na cidade	90
Figura 69 - Teste de realização e acompanhamento de um pedido	91
Figura 70 - Teste de chegada de uma transferência	92
Figura 71 – Teste de consulta das praças da cidade	93
Figura 72 - Teste de consulta dos pedidos realizados num dia	94
Figura 73 – Teste de chegada de um pedido e consequente ação do taxista	95

Figura 74 - Teste de Transferência de Serviço Recebida	96
Figura 75 – Teste de consulta do calendário de praças do taxista.....	97
Figura 76 – Teste de visualização dos serviços realizados num determinado dia.....	98
Figura 77 – Teste de visualização do tempo de atividade num determinado dia	99
Figura 78 – Teste de uma queixa sobre o Cliente	99

Índice de Tabelas

Tabela 1 – Versões dos SDK Android [14].....	14
Tabela 2 - Limites do <i>PostgreSQL</i> (adaptado de [33])	30

Acrónimos

- ACID – Atomicidade, Consistência, Isolamento e Durabilidade
- AJAX – *Asynchronous JavaScript and XML*
- APK – *Android Package*
- ANTRAL – Associação Nacional dos Transportes Rodoviários em Automóveis Ligeiros
- API – *Application Programming Interface*
- ART – *Android Runtime*
- CERN – *European Laboratory for Particle Physics*
- CSS – *Cascading Style Sheets*
- DEX – *Dalvik Executable*
- DGAE – Direção Geral das Atividades Económicas
- DGTT – Direcção-Geral de Transportes Terrestres
- DOM – *Document Object Model*
- FCM – *Firebase Cloud Messaging*
- FPT – Federação Portuguesa do Táxi
- GCM – *Google Cloud Messaging*
- GNR – Guarda Nacional Republicana
- HTTP – *Hyper Text Transfer Protocol*
- HTTPS – *Hyper Text Transfer Protocol Secure*

HAL	–	<i>Hardware Abstraction Layer</i>
HTML	–	<i>Hyper Text Markup Language</i>
IMT	–	Instituto da Mobilidade e dos Transportes
JSON	–	<i>JavaScript Object Notation</i>
PSP	–	Polícia de Segurança Pública
REST	–	<i>Representational State Transfer</i>
SGML	–	<i>Standard Generalized Markup Language</i>
SIG	–	Sistemas de Informação Geográfica
SGBD	–	Sistema Gestor de Base de Dados
SO	–	Sistema Operativo
SOAP	–	<i>Simple Object Access Protocol</i>
SDK	–	<i>Software Development Kit</i>
UI	–	<i>User Interface</i>
WSDL	–	<i>Web Services Description Language</i>
XML	–	<i>Extensible Markup Language</i>
XMPP	–	<i>Extensible Messaging and Presence Protocol</i>

1. INTRODUÇÃO

Neste capítulo será realizada uma introdução ao trabalho realizado nesta dissertação onde será contextualizada a motivação para a elaboração do mesmo. Serão ainda definidos os objetivos propostos na realização do trabalho e por fim descrita a organização deste relatório.

O crescimento que a *Internet* tem tido nestes últimos anos, fez com que neste momento qualquer pessoa possa ter um acesso a mesma através de um *smartphone* ou computador. Com este avanço, deu-se a criação de diversos mercados sustentados em plataformas móveis e na *Web*.

Os mercados baseados em aplicações móveis surgiram do crescimento uso do *smartphone*. Surgindo novas perspectivas de mercado que até aqui não tinham muito por onde crescer, sendo um exemplo disto a requisição de um serviço de transporte que até ao surgimento das aplicações móveis era única e exclusivamente entregue aos táxis. Contudo, atualmente já existem diversas empresas com aplicações móveis para concorrer com estes como, por exemplo, a *Cabify* e a *Uber*. Este mercado encontra-se em grande crescimento e para isso, as empresas de táxis tiveram igualmente que aderir a estas novas tendências, criando também aplicações que permitissem aos utilizadores requisitar um táxi e acompanhar o desenvolvimento do serviço solicitado.

Todas estas aplicações, para o serviço de táxis e as restantes mencionadas, têm como foco principal as grandes cidades e é este problema que o sistema desenvolvido, e explicado nesta dissertação, vem colmatar, sendo um sistema de gestão e atribuição de táxis para cidades/localidades de menores dimensões. Este sistema contará com uma aplicação *Web*

para o administrador do sistema poder gerir e duas aplicações móveis Android, uma para os clientes e outra para os taxistas.

A aplicação do gestor tem o objetivo de inserir, consultar e editar todos os parâmetros que irão entrar na administração e atribuição dos serviços de táxis.

A aplicação móvel do cliente tem como objetivo principal a realização de pedidos de táxis. Quando o pedido for aceite por um taxista, o cliente deverá acompanhar o resultado do serviço, visualizando a deslocação do táxi.

Por outro lado, a aplicação do taxista tem como objetivos fundamentais a receção e resposta aos pedidos vindos da plataforma de gestão. Ao ser aceite o pedido, deverão ser apresentadas as características do cliente.

Por fim, para interligar estas aplicações com a base de dados, tem-se uma plataforma de gestão.

1.1. CONTEXTUALIZAÇÃO

Esta dissertação, elaborada no âmbito do Estágio Curricular para conclusão do Mestrado em Engenharia Eletrotécnica e de Computadores no ramo de Telecomunicações, tem como intuito a descrição das atividades desenvolvidas na *Gisgeo – Information System* durante os quatro meses e meio de estágio.

O mercado das aplicações tem-se desenvolvido de uma forma exponencial, isto deve-se grande parte ao sucesso dos *smartphones*. Só em 2010 foram vendidos mais de duzentos e noventa e seis milhões em todo o mundo. Este número não parou de crescer ao longo dos anos e no final de 2016 tinham sido vendidos cerca de 1,5 biliões de *smartphones* que corresponde a 28% da população mundial e tem-se previsto que até 2020, esta percentagem aumente para os 37% [1]. Este crescimento tão acentuado refletiu-se no mercado das aplicações, em junho deste ano, 2017, foram contabilizadas três milhões de aplicações, disponíveis na loja para dispositivos Android. Os números apresentados exibem o interesse dos utilizadores pelas aplicações móveis, o fácil acesso a dados móveis permite que muitas destas aplicações funcionem usando a *Internet* [2].

1.2. EMPRESA

A Gisgeo é uma empresa, fundada em 2008, que desenvolve Sistemas de Informação Geográfica (SIG), produzindo soluções *Web* e *mobile* capazes trabalhar e apresentar dados georreferenciados, especialmente com base em tecnologias *Open Source*, adaptáveis tanto na gestão operacional como no planeamento estratégico das empresas, para que se possa poupar custos e otimizar recursos [3].

A Gisgeo tem implementações das suas soluções em Portugal, Brasil e Angola, entre outros e o portfólio da empresa conta com um leque diversificado de soluções: a solução *GeoMed* possibilita agregar dados de saúde com diversas informações, possibilitando a criação e acompanhamento da implementação de intervenções de saúde, ou estratégias de prevenção; a solução *GeoSegur* permite que as informações de georreferenciação e caracterização recolhida em tempo real por agentes no terreno sejam transmitidas para um centro de decisão. Esta centralização de informação proporciona uma imagem abrangente e atualizada, possibilitando a tomada de decisões fundamentada em dados atualizados, bem como a distribuição inteligente pelos recursos disponíveis de ordens de operacionalização; a solução de *GeoMarketing* é uma ferramenta de *GeoIntelligence*, que concede dimensão geográfica aos dados empresariais habitualmente registados em CRM (*Customer Relationship Management*) / ERP (*Enterprise Resource Planning*) e analisados através de ferramentas de *Business Intelligence*, para além disto esta solução permite o registo de terrenos e prédios com ocupação comercial, industrial ou apoia a decisão de localização de novos comércios e indústrias. A informação recolhida pode ainda ser cruzada com dados demográficos, histórico de ocupação para permitir reconhecer locais de maior potencial de consumo; Uma das soluções da empresa é o *GeoIndoor* que proporciona o controlo e localização de máquinas com fonte elétrica ou recipientes/caixas sem fonte elétrica; A *GeoDecide* é uma tendência da sociedade atual a mobilidade dos recursos humanos por períodos incertos de tempo, enquanto ao serviço de uma empresa; Os Sistemas de Informação Geográfica permitem a gestão integrada do território (agrícola, industrial, habitacional), a unificação de dados de diversas origens, como informação cartográfica, topográfica e planos condicionantes, e a interligação com as áreas comercial e financeira. Para uma correta gestão dos recursos existentes e compreensão das carências no território, é fundamental a utilização de ferramentas SIG que garantam consulta e análise da informação georreferenciada e assim surge a solução *GeoSpatial*; Para quem não vive sem a adrenalina dos desportos radicais, ou para os que escolhem o sossego de um longo passeio por paisagens remotas, a segurança

deve estar sempre em primeiro lugar e para isso tem-se a solução de *GeoSport*; Realidade Aumentada é a tecnologia que amplifica a percepção da realidade circundante acrescentando informação virtual relativamente aos elementos da realidade envolvente, tornando-a interativa e manipulável digitalmente, usando som, vídeo, interfaces gráficos ou coordenadas geográficas, assim sendo surgiu a solução *GeoTravel* [4].

Por último, há a solução *GeoCar*, produto que a solução desenvolvida irá interligar-se. Esta é uma solução completa de gestão de frotas, possibilitando a localização e estado, em tempo real, de viaturas e respetivos condutores, potenciando o aumento da produtividade e da segurança através da gestão eficaz de recursos. Com o *Geocar* é possível economizar combustível, comunicações com os condutores, recursos, tempo, manutenção das viaturas e emissão de CO₂ (Dióxido de Carbono)[5].

1.3. OBJETIVOS

O objetivo principal deste projeto consiste no desenvolvimento de um sistema de gestão de serviços de táxi. Deverá existir uma aplicação *Web* para o administrador da plataforma gerir as características da gestão e atribuição dos serviços de táxis. Deverão ainda existir duas aplicações móveis Android, uma para o taxista, poder trocar a sua disponibilidade e responder a pedidos do cliente e uma outra para os clientes poderem realizar os pedidos de serviço de táxi. Por fim, para se interligar as aplicações com a base de dados, o sistema necessita que exista uma plataforma de gestão. Pretende-se desenvolver outras funcionalidades na plataforma tais como: o cálculo aproximando do tempo de espera do cliente, seguir na aplicação a deslocação do veículo, consultar os serviços realizados por utilizador e por data, por fim, contabilizar o acesso dos utilizadores tanto à plataforma de gestão, como também, às aplicações móveis. Dada a complexidade inerente a estes objetivos, surgiu a necessidade de os subdividir em múltiplas tarefas de realização mais simples, tais como:

- Pesquisa do estado da arte em sistemas de gestão e atribuição de serviços de transporte;
- Pesquisa do estado da arte em plataformas *Open Source* vocacionadas para o desenvolvimento de aplicações de “*push notification*”;
- Estudo da plataforma Android;

- Desenvolvimento dos protótipos: sistema de gestão e aplicações móveis (cliente e taxista);
- Integração com a plataforma de gestão de frotas;
- Implementação e teste dos protótipos aplicativos;
- Documentação da aplicação (manual de configuração e utilização).

1.4. ORGANIZAÇÃO DO RELATÓRIO

Esta dissertação está estruturada em sete capítulos. No capítulo 1 é realizada uma introdução ao trabalho, uma apresentação do problema, é feita uma apresentação da empresa para quem foi desenvolvida a solução descrita e são definidos os objetivos do trabalho.

No capítulo 2 é feito um enquadramento do que é o setor dos táxis, em que é apresentada uma breve história dos táxis em Portugal e de como eles estão regulamentados. É realizada igualmente uma rápida descrição das aplicações concorrentes ao serviço de táxis tradicional.

No capítulo 3 são abordadas as tecnologias usadas para desenvolver a solução que são descritas nesta dissertação, para que o leitor consiga ter uma correta análise à solução apresentada.

No capítulo 4 é realizada a análise de requisitos que o sistema deverá apresentar que são levantados através de casos de uso.

No capítulo 5 é descrita a implementação do sistema onde é detalhada a solução desenvolvida, indo ao encontro da especificação definida através dos casos de uso.

No capítulo 6 são apresentados os testes realizados nas aplicações a quando do desenvolvimento da solução.

No capítulo 7 estão apresentadas as principais conclusões retiradas do trabalho realizado nesta dissertação e perspetivados os desenvolvimentos futuros.

2. SERVIÇO DE TÁXIS

O desenvolvimento das redes móveis e dos *smartphones*, permitiu que fossem criados e desenvolvidos novos serviços [1]. Isto possibilitou não só auxiliar os seus utilizadores em tarefas do dia a dia, mas também permitiu a conceção de alternativas a serviços que já existiam. Devido ao crescimento exponencial das tecnologias móveis, muitas das grandes indústrias mais antigas não acompanharam este desenvolvimento e novos serviços de fácil uso para os consumidores surgiram. Estes, em diversos casos, são uma melhor opção aos serviços tradicionais já existentes e são uma concorrência direta.

O táxi é um dos serviços que enfrenta este problema. Manteve-se um pouco estático desde a sua criação até há bem pouco tempo. As grandes alterações efetuadas foram a evolução dos carros usados e dos serviços de despacho.

A seguir será apresentada a história e evolução dos táxis e como eles estão regulamentados em Portugal. Por último, serão apresentadas aplicações que foram criadas para fazer concorrência ao táxi e também as que foram criadas para que este se adaptasse às novas tecnologias.

2.1. SETOR DOS TÁXIS

O táxi é um meio de transporte com uma larga história sendo que em finais do século XIX surgiram os primeiros registos do uso de veículos motorizados, tendo a sua divulgação colaborado para a perda de relevância dos meios predominantes da altura, que era o transporte de pessoas por meios animais [6].

O setor dos táxis teve o seu desenvolvimento e consolidação na sociedade devido ao desenvolvimento tecnológico existente no século XX. As diversas inovações no setor, tais como, o taxímetro, o rádio e os computadores permitiram que a produtividade do setor ampliasse. Segundo o livro publicado por *Michael Beesley* [7], os fatores que influenciam a oferta de táxis são a procura turística, a procura local, a comparação entre a tarifa praticada pelos táxis e os seus mais diretos concorrentes e a taxa de emprego/desemprego. O autor refere estes fatores para o caso dos táxis em Londres, mas pode-se generalizar os mesmos aplicando à generalidade das cidades.

Em Portugal, os primeiros veículos a motor que efetuaram o serviço de táxi surgiram no princípio do século XX e este setor encontra-se sob a jurisdição do IMT (Instituto da Mobilidade e dos Transportes). No que diz respeito às licenças, os seus números são definidos pelas Câmaras Municipais e a homologação das viaturas é igualmente da garantia destes organismos públicos. As entidades responsáveis pela fiscalização do setor em Portugal são a DGTT (Direção Geral de Transportes Terrestres), as Câmaras Municipais, a GNR (Guarda Nacional Republicana) e a PSP (Policia de Segurança Pública) [8].

No que diz respeito aos preços praticados num serviço de táxi, a forma mais comum de cobrança do serviço é pela distância percorrida e pelos tempos de espera, mas pode também ser realizada a cobrança à hora, ao percurso ou ao contrato, desde que devidamente acordado entre cliente e prestador de serviço. A 1 de janeiro de 2013 entraram em vigor os preços impostos por convenção entre a DGAE (Direção Geral das Atividades Económicas), a ANTRAL (Associação Nacional dos Transportes Rodoviários em Automóveis Ligeiros) e a FPT (Federação Portuguesa do Táxi), e tem valores distintos de acordo com o serviço prestado. Outros fatores que podem influenciar o valor a cobrar são: a circulação dentro ou fora de meio urbano, a altura do dia e os tipos de prestação de serviço apresentados anteriormente. Para o serviço urbano, o preço fixo com que se inicia o trajeto e que abrange os primeiros 1800 metros de dia, ou 1440 metros à noite, designada de bandeirada, varia entre 3,25€ e 3,90€, respetivamente. O preço por quilómetro varia entre 0,47€ e 0,56€ (nos

casos apresentados o preço mais baixo equivale ao serviço diurno e o valor mais elevado ao serviço noturno). Se o cliente transportar alguns extras, tais como, bagagem ou animais de companhia, estes terão um preço fixo e será acrescido ao valor final do serviço [9].

Ultimamente, surgiu uma concorrência aos táxis que afetou ainda mais uma vez um setor em crise. A evolução da tecnologia possibilita a oferta de serviços, por formas que implicam menores custos e torna o serviço tradicional menos atrativo quer para os consumidores quer para alguns profissionais. Será explicada na secção seguinte a concorrência que surgiu no táxi e o que este setor criou para concorrer com as novas formas de serviço de transporte.

2.2. APLICAÇÕES DE REQUISIÇÃO DE TRANSPORTE

O desenvolvimento da *Web* possibilitou o desenvolvimento em massa da partilha de conteúdos, pois cada utilizador pode contribuir com conteúdos e ligarem-se uns aos outros. Com esta nova simplicidade de partilha nasceu uma nova economia de partilha e um novo conceito de consumo designado consumo colaborativo [10].

Esta economia é um fenómeno que apareceu, não só, a partir do progresso tecnológico da informação e das comunicações, mas também das comunidades e comércios *Web*. O consumo colaborativo faz parte da economia diária e tem como base a partilha entre duas entidades de bens ou serviços. Este modelo é empregue em mercados *online* através de plataformas, onde os utilizadores produzem e partilham o seu próprio conteúdo ou então colaboram de forma positiva no conteúdo de outros [10].

O novo formato de consumo explicado anteriormente é realizado através de plataformas tecnológicas, como um *Website* ou uma aplicação de *smartphone*. No entanto está muito sujeito à dinâmica social para existir a partilha e a colaboração [10].

Esta nova forma de partilha tornou-se nesta última década num fenómeno mundial, no que diz respeito à partilha de carros, sendo uma ótima opção à compra do mesmo. Esta consiste num grupo de pessoas que pagam o acesso a uma frota de automóveis, são usados praticamente apenas para viagens curtas [11].

Uma das primeiras empresas a aproveitar a potência do *smartphone* combinando com esta nova forma de partilha foi a *Zipcar*. Em que, para se servir da aplicação é necessário realizar o registo no site e após este ser validado recebe-se um cartão que serve de chave automática para abrir as portas de cada veículo. Sempre que se quiser alugar um carro, através da

aplicação, este pode ser escolhido o local de entrega mais próximo e dirigir-se a ele com o cartão que será associado a essa viatura. No que diz respeito aos custos dos veículos, estes podem ser diários, mensais ou anuais e todas as despesas estão abrangidas. O processo é todo ele automático através da aplicação, sem envolver nenhum responsável da empresa [11].

O grande problema com este tipo de aplicações é que obrigam à aquisição e manutenção dos veículos, e é neste ponto que empresas como a *Cabify*, *Uber*, ou a *Blablacar* inovam, em que são os donos dos carros que os alugam e conduzem, pagando os utilizadores apenas o percurso efetuado.

A *Blablacar*, tem um conceito um pouco diferente, sendo uma plataforma, cujo objetivo é conectar condutores e passageiros que viajarão para o mesmo sítio e podendo desta forma partilhar os custos da viagem. Não existe nenhum retorno monetário para o condutor ou para a empresa. Esta plataforma é uma prova real do desenvolvimento da economia dos transportes. Foi das primeiras empresas a surgir no mercado e desde então cresceu bastante, contando com uma comunidade de mais de 25 milhões de membros espalhados por 22 países [12].

A *Cabify* ou *Uber*, são atualmente as grandes concorrentes do serviço de táxis pois ambas fornecem preços mais baixos e outras vantagens comparando com os táxis. Estas aplicações são muito semelhantes no seu funcionamento e interação com os utilizadores. O passageiro ao querer efetuar uma deslocação, usa uma das aplicações para submeter o pedido de uma viagem com os pontos de início e destino pretendidos. Ao selecionar estes pontos a aplicação dará uma estimativa do preço que o utilizador irá pagar. O pedido ao ser recebido na central de despacho da aplicação, irá reencaminha-lo para o condutor disponível mais próximo. O passageiro é informado que o condutor está a caminho e enquanto não chegar até si, pode ver na aplicação vários dados relativos ao condutor, tais como o seu nome, o tipo de carro que conduz, a sua matrícula, a sua localização em tempo real e ainda uma estimativa do tempo que demorará a chegar.

As empresas que gerem os serviços de táxis, para não se deixarem ficar atrás da nova concorrência que surgiu no mercado, tiveram de realizar parcerias com empresas de desenvolvimento de aplicações para que fossem criados sistemas para a gestão e atribuição de serviços de táxis. Alguns exemplos de aplicações criadas para a requisição de um táxi são

a *Kabx* e a *Taxi-Link*. Estas aplicações têm parcerias com empresas de táxis das áreas metropolitanas do Porto, Lisboa e Coimbra e na zona do Algarve. O modo de funcionamento destas aplicações assemelha-se às apresentadas anteriormente em que o cliente ao realizar um pedido na aplicação, este é enviado para a central e esta enviará o serviço para o taxista mais próximo, enquanto este está a chegar, o cliente poderá observar na aplicação as características do táxi e onde ele se encontra.

Em suma, o serviço de táxis em Portugal tem vindo a ter uma forte concorrência, por parte de outras empresas de serviço de transporte (*Cabify*, *Uber*), que têm aplicações móveis para realizar os pedidos. Para colmatar esta perda de interesse por parte dos clientes, as empresas de serviço de táxi criaram igualmente aplicações móveis, para se irem adaptando aos tempos e para se tornarem competitivos com as novas formas de serviço de transporte. Estas aplicações têm o objetivo de realizar uma requisição de um serviço de táxi e irem acompanhando a posição atual em que o táxi se encontra.

De seguida são apresentadas as tecnologias estudadas e utilizadas para o desenvolvimento de um sistema de gestão de pedidos de táxis para pequenas localidades.

3. TECNOLOGIAS UTILIZADAS

3.1. ANDROID

O Android *SDK (Software Development Kit)* é uma plataforma de desenvolvimento para dispositivos móveis com um sistema operativo (SO) livre e *open-source* com o mesmo nome baseado no *kernel* do *Linux*, este SO permite o desenvolvimento de aplicações usando a linguagem de programação *Java*.

Para o desenvolvimento de aplicações Android é imprescindível utilizar o *Software Development Kit*, constituído pela *Application Programming Interface (API)* de classes *Java* e pelo emulador, isto é, um dispositivo móvel virtual em que se pode testar as aplicações criadas [13]. Além disto, é necessário igualmente um ambiente de desenvolvimento, nomeadamente o *Android Studio*. No momento do desenvolvimento da aplicação é essencial seleccionar a versão *SDK* sobre a qual se vai trabalhar. Para isso é necessário conhecer alguns pressupostos, entre eles, as versões existentes, a distribuição das versões pelos dispositivos, bem como ter em conta se as aplicações criadas para versões de *SDK* inferiores são compatíveis com versões mais recentes. Na página do Android é disponibilizada a informação que está representada na Tabela 1 em que se pode observar as versões do *SDK*.

Tabela 1 – Versões dos SDK Android [14]

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.7%
4.1.x	Jelly Bean	16	2.7%
4.2.x		17	3.8%
4.3		18	1.1%
4.4	KitKat	19	16.0%
5.0	Lollipop	21	7.4%
5.1		22	21.8%
6.0	Marshmallow	23	32.3%
7.0	Nougat	24	12.3%
7.1		25	1.2%

Na Figura 1 pode-se observar a sua distribuição [14].

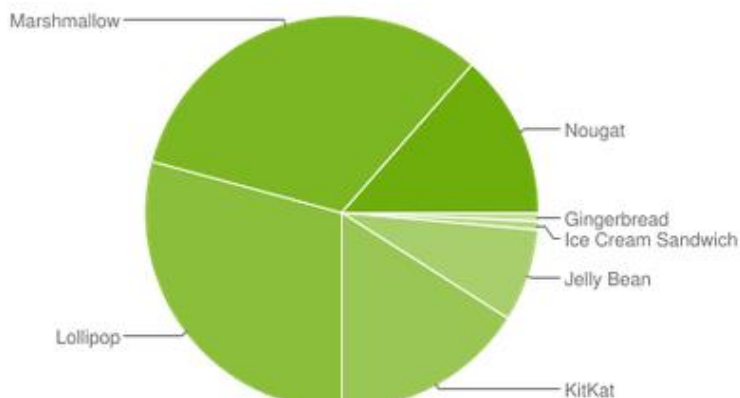


Figura 1 – Distribuição do SDK Android [14]

O desenvolvimento da aplicação Android foi realizada com o SDK 4.03 – 4.0.4, pois é a versão mais baixa compatível com uma das funcionalidades descritas mais à frente a ser considerada no estudo de distribuição e garantindo assim que a aplicação é compatível para todas as versões superiores.

3.1.1. ARQUITETURA

A arquitetura do Android encontra-se dividida em seis partes, como se pode observar pela Figura 2, são elas: *Kernel Linux*, camada de abstração do *hardware*, ambiente de execução (*Android Runtime*), bibliotecas, *Java API framework* e aplicações [15].

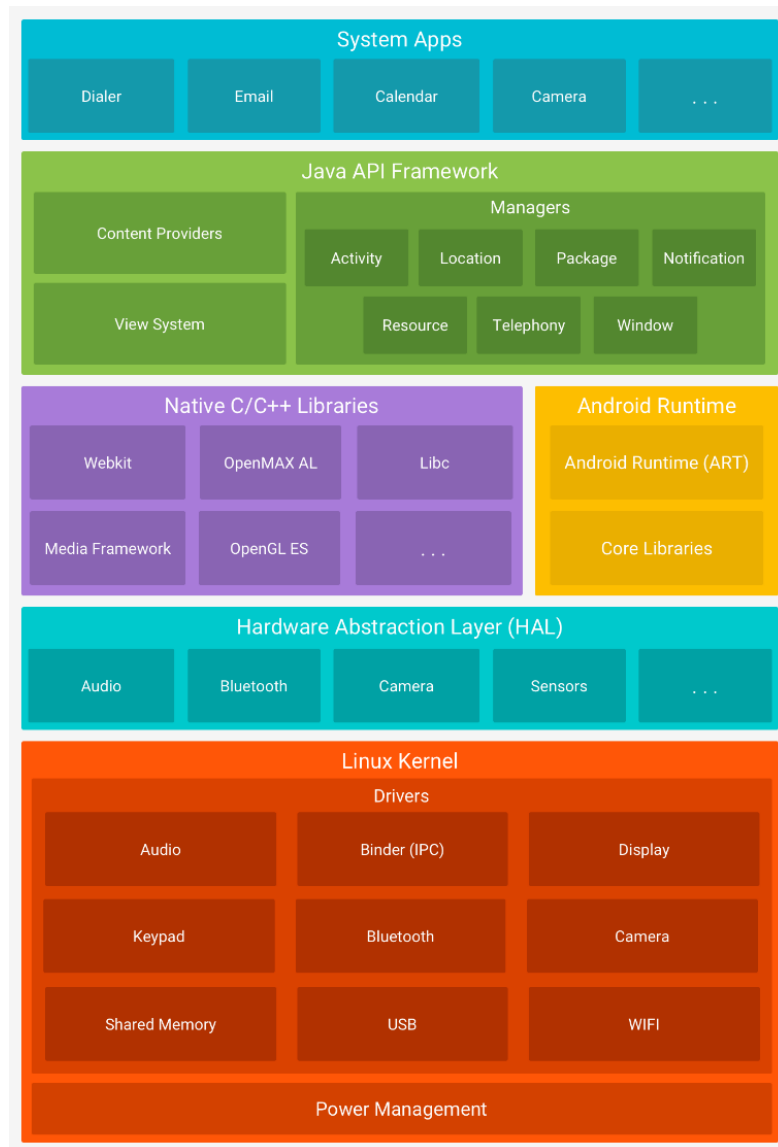


Figura 2 – Arquitetura do Android [15]

- *Kernel Linux* – O Android usa internamente o *kernel* do *Linux* para tarefas fundamentais de segurança, gestão de processos, memória, protocolos de rede, entre outros. O Android ao utilizar isto permite que este aproveite com que os fabricantes dos dispositivos desenvolvam *drivers* de *hardware* para um *kernel* conhecido.
- **Hardware Abstraction Layer (HAL)** – Esta camada providencia interfaces padrão que revela as capacidades do hardware do dispositivo para a camada de *Java API framework*. A HAL implementa uma interface para um tipo específico de hardware do dispositivo, como o módulo de câmara ou *bluetooth*., que posteriormente pode ser invocado pela camada *Java API framework*. Ao ser invocado o sistema Android carrega o módulo da biblioteca para este componente de *hardware*.

- Bibliotecas – Acima das camadas do *kernel* e *HAL* estão as bibliotecas nativas do Android, escritas em C/C++ e especializadas para cada arquitetura de hardware. Diversos componentes e serviços principais do Android, como o HAL ou o ambiente de execução, são implementados por código que requerem estas bibliotecas nativas. Algumas destas bibliotecas têm a sua funcionalidade exposta na camada *Java API framework*, explicada mais à frente. Nesta camada existe um sistema gestor de base de dados (SGBD), o *SQLite* que é uma base de dados *open source* que armazena os dados num arquivo de texto no dispositivo.
- Ambiente de Execução (ART - *Android Runtime*) – Para dispositivos com a versão do Android superior à 5.0 (API 21), cada aplicação executa um processo próprio com uma instância própria do ART. O ART é projetado para executar em várias máquinas virtuais em dispositivos de baixa memória, executando ficheiros *.DEX (Dalvik Executable)*, um formato de *bytecode* especialmente projetado para o Android e otimizado para oferecer consumo mínimo de memória. Antes desta versão, o Dalvik era o *Android Runtime* existente.
- *Java API framework* – Esta camada proporciona o acesso a blocos de programação empregues nas aplicações desenvolvidas para este sistema operativo, o que simplifica a reutilização de componentes. Entre as APIs disponíveis, destacam-se:
 - *Activity Manager* – Controla o ciclo de vida das aplicações e mantém uma “back stack” de forma a permitir a navegação ao utilizador.
 - *Resource Manager* – Oferece acesso a recursos que não sejam código-fonte, tais como, *strings, layouts*, imagens, entre outros.
 - *Notification Manager* – Permite que as aplicações exibam alertas personalizados na barra de estados do dispositivo.
 - *Location Manager* – Permite que a aplicação obtenha a localização geográfica do dispositivo, entre outras funcionalidades, oferecendo o acesso a serviços de localização do sistema.

- Aplicações – Esta é a camada mais alta na arquitetura da plataforma Android e é composta, tal como o nome indica, por todas as aplicações contidas no dispositivo, como por exemplo, e-mail, navegador de internet, calendário, entre outros.

3.1.2. ELEMENTOS DA APLICAÇÃO

De seguida serão apresentados, através de uma breve explicação, os vários componentes que uma aplicação desenvolvida em Android pode instanciar e executar.

- **Activities**

Uma *activity* é um componente da aplicação que proporciona uma janela de interface com o utilizador e que lhe permite interagir com a aplicação. Normalmente uma aplicação é constituída por diversas *activities* ligadas entre si. Um exemplo de uma aplicação com mais do que uma *activity* é uma aplicação de *email*, em que tem de existir uma para exibir a listagem dos *emails* e outra para a visualização dos mesmos.

As *activities* são tratadas como uma pilha (*stack*), quando uma *activity* é iniciada, esta é colocada no topo da pilha, passando a ser a *activity* em execução. A *activity* anterior é então colocada em *background* e só sairá deste estado quando estiver de novo no topo da pilha. Em qualquer altura, o utilizador pode pressionar o botão de retroceder, e ao realizá-lo ser-lhe-á mostrada a *activity* que estiver em *background* no topo da pilha (Figura 3).

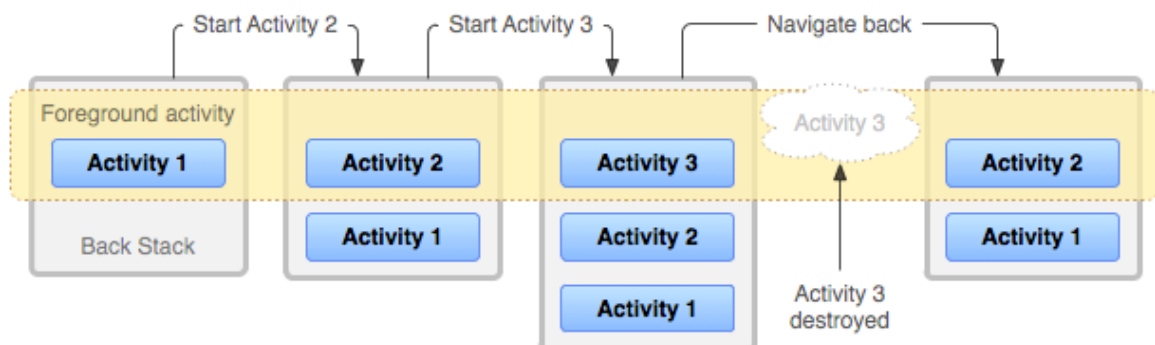


Figura 3 - Funcionamento da pilha de Activities [16]

As *activities* têm um ciclo de vida (Figura 4) que pode ser definido em três tipos:

Em execução: a *activity* está no topo da pilha, ou seja, está visível no ecrã do dispositivo;

Em Pausa: a *activity* está em execução, mas no momento não é a *activity* principal. Esta *activity* pode estar visível de uma forma mais transparente ou não ocupando todo o ecrã, pois

a atenção do utilizador recai sobre outra *activity*, um exemplo disto é quando se recebe uma chamada. Neste caso o sistema guarda todas as características da *activity*, apenas libertando esses recursos caso sistema requeira mais memória.

Parada: a *activity* foi colocada neste modo depois de perder a sua posição no topo da pilha para outra *activity* encontrando-se então em *background*. O sistema armazena todas as características de uma *activity*, mesmo quando ela está neste estado. No entanto, a *activity* não estando visível para o utilizador pode ser destruída a qualquer momento caso o sistema precise de memória.

A passagem entre os estados do ciclo de vida de uma *activity* (Figura 4) pode ser controlado através de métodos apropriados que serão a seguir explicados.

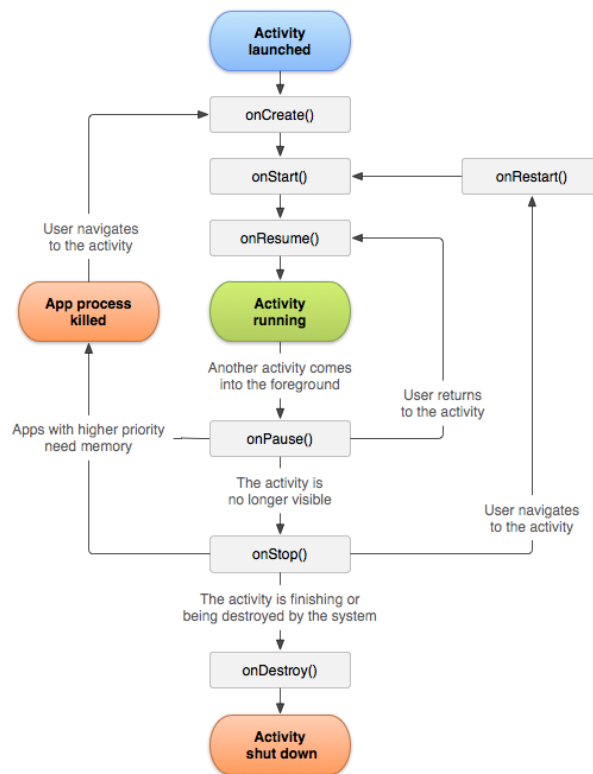


Figura 4 - Ciclo de vida de uma *activity*

onCreate(): Este método é invocado quando a *activity* é criada pela primeira vez, é aqui que se deve iniciar a interface de utilizador. Este método pode receber por parâmetro um *Bundle*, que é um conjunto de informação sobre o estado anterior desta *activity*, caso tenha existido.

onStart(): Este método é invocado quando a *activity* passa a estar visível para o utilizador.

onResume(): Este método é executado quando a *activity* está a iniciar a interação com o utilizador. É neste momento que esta está no topo da pilha de *activities*.

onPause(): Método invocado quando a *activity* vai ser colocada em *background*; este método deve ser usado para gravar definitivamente modificações de informação persistente, parar animações e todo outro tipo de computação que consuma muito processador. A próxima *activity* não será lançada enquanto este método não terminar, por isto, ele tem que ser de muito rápida execução.

onStop(): Executado quando a *activity* passa a não estar visível para o utilizador, devido a outra *activity* ter sido colocada em cima desta na pilha de *activities* ou simplesmente por esta ter sido concluída pelo utilizador. Um facto importante é que este método pode nunca ser invocado, isto deve-se a que quando o sistema carece de mais memória os processos podem ser interrompidos, sem existir uma notificação por parte do sistema.

onRestart(): Método invocado após a *activity* ter estado parada e quando está a ser lançada novamente.

onDestroy(): Este método é o último a ser executado antes da *activity* ser destruída, pode ser porque está a terminar (através do uso do método *finish()*) ou então porque o sistema está temporariamente a destruí-la de forma a obter espaço de memória.

Estes métodos anteriormente apresentados devem ser *overridden* na classe *Activity* para que o Android os execute na altura apropriada.

- **Services**

Os serviços no Android são idênticos aos serviços que existem nos sistemas operativos habituais, como é o caso do *Windows*. Estes são processos que são executados em *background*, segundo plano, e são normalmente utilizados em aplicações que necessitam de uma execução por um longo período de tempo, como a verificação de utilizações, um leitor de música, entre outros.

O Android define dois tipos de serviços, os *unbounded* e os *bounded*. Os serviços *unbounded* são um tipo de serviço que independente da *activity* que o iniciou, não existindo comunicação entre elas, por outro lado, existem os serviços *bounded* que são um tipo de serviço que oferece comunicação e interação entre este e a *activity* que o iniciou. Utilizando

mensagens e *BroadcastReceivers* (explicados mais à frente) a *activity* pode monitorizar o seu estado ou o progresso da tarefa que o serviço está a realizar a qualquer momento.

- ***Content Providers***

Um *Content Provider* é uma API que possibilita a disponibilização de dados a outras aplicações ou *activities*. O Android possui um *Content Provider* para os contactos (nomes, endereços, número de telefone, etc.), permitindo assim que estes possam ser acedidos por qualquer aplicação.

- ***Broadcast Receivers***

Um *broadcast receiver* é um componente que responde a determinados de eventos. Muitos dos eventos são enviados pelo próprio sistema, como por exemplo com a indicação de que o dispositivo está com pouca bateria disponível ou que a ligação *WiFi* está disponível. Todavia, é exequível que as aplicações transmitam eventos entre elas ou entre *activities* da mesma aplicação. Por exemplo, temos o caso de uma *activity* que expõe a evolução da transferência de um certo ficheiro, enquanto um serviço é responsável pela respetiva transferência, a *activity* implementa a classe *BroadcastReceiver* de forma a receber a informação do estado da transferência, enquanto o serviço de transferência transmite um evento com essa informação.

- ***Intents***

Uma *Intent* é um mecanismo usado para descrever uma ação abstrata que deve ser invocada quando se pretende, por exemplo, enviar um email, escolher um contacto, lançar outra *activity*, etc. Cada *Intent* está associada a um componente, que é chamado quando é realizado o tratamento da *Intent*. Uma *Intent* pode ser utilizada para lançar outra *activity*, iniciar ou comunicar com um serviço e também enviar mensagens de *broadcast* (que serão recebida pelos *broadcast receivers*).

Existem dois tipos de *intents*, as explícitas e as implícitas. As *intents* explícitas são aquelas em que se conhece o componente de destino, são usadas maioritariamente para iniciar *activities* dentro de uma aplicação, por outro lado, tem-se as *intents* implícitas que são aquelas que não se conhece o componente de destino e deixa o sistema Android tomar a decisão de que aplicação iniciar, um exemplo disto é quando se quer visualizar um ficheiro

PDF, usa-se este tipo de *intent* para que seja da responsabilidade do Android determinar quais as aplicações que estão aptas a executar a ação pretendida.

- **Resources**

Um recurso pode ser uma *string*, um *bitmap* ou qualquer outro pedaço de informação que a aplicação necessite, e deve estar separado do código fonte desta. Ao efetuar esta divisão, torna-se possível utilizar diferentes recursos para distintos dispositivos, tendo em conta por exemplo o tamanho do ecrã ou a linguagem do mesmo, desta forma, torna-se mais simples a criação de, por exemplo, aplicações multilinguagem.

Todos os recursos existentes nas aplicações devem ser colocados no diretório *res/*, e dentro de subdiretórios que agrupem os recursos por tipos e por configurações.

Quando a aplicação é compilada, todos os recursos são carregados para a aplicação, após a compilação de todos os recursos, é gerada uma classe de seu nome “R”, que contém os identificadores de todos os recursos existentes na aplicação.

- **Android Manifest**

Todas as aplicações criadas devem de ter no seu diretório raiz um ficheiro designado de *AndroidManifest.xml*. Este ficheiro apresenta ao sistema operativo Android a informação essencial sobre a aplicação que este tem de possuir antes de compilar e executar o código. Ele tem de apresentar a seguinte informação:

- Dar o nome ao package da aplicação. Este nome funciona como um identificador único da aplicação.
- Indicação das permissões que a aplicação irá requer, como por exemplo, o acesso à internet, ou o acesso à localização do dispositivo, entre outros.
- Descreve quais os componentes que a aplicação possui: *Activities*, *Services* e *Broadcast Receivers*.

3.1.3. VOLLEY

O Volley é uma biblioteca HTTP (*Hyper Text Transfer Protocol*) que torna as conexões de rede para aplicações Android mais fáceis e mais rápidas.

O Volley permite a programação automática de pedidos de rede, várias conexões de rede simultâneas para além de transparência na resposta. O Volley é também capaz de suportar pedidos de priorização ou cancelamento de “*requests*” permitindo uma boa organização, o que torna mais fácil preencher corretamente a interface do utilizador com dados obtidos de forma assíncrona a partir da rede [17].

Desta forma, esta biblioteca integra-se facilmente com qualquer protocolo e suporta o formato JSON (*JavaScript Object Notation*), no entanto não é adequado para grandes operações de download, uma vez que detém todas as respostas na memória durante a análise.

3.2. WEB SERVICES

Os *Web Services* proporcionam a interoperabilidade entre aplicações diferentes, que são executadas em várias plataformas e sistemas. Esta propriedade provém do facto dos serviços *Web* serem independentes de qualquer linguagem de programação, sistema operativo e plataforma [17].

No sentido contrário dos modelos cliente/servidor habituais, os *Web services* não fornecem uma interface gráfica de utilizador. Em vez disso, partilham dados e processos através de uma interface de programação numa rede. Os programadores podem, depois, se necessário adicionar a interface gráfica (como uma página *Web* ou um programa executável) para oferecer funcionalidades específicas aos utilizadores [17].

Para se utilizar um *Web service* tem de se escolher uma das arquiteturas existentes, SOAP (*Simple Object Access Protocol*) ou REST (*Representational State Transfer*). No caso do projeto desenvolvido e explicado nesta dissertação, usou-se a arquitetura REST que será explicada a seguir.

3.2.1. REST

Um *Web service* REST é um simples serviço web implementado através de HTTP. Este serviço ganhou uma grande aceitação, pois é uma alternativa mais simples que o SOAP e a *Web Services Description Language* (WSDL). Uma das forças desta mudança foi empresas como o *Facebook*, *Google* e *Yahoo* terem adotado este serviço, em detrimento do SOAP e WSDL.

REST foi apresentado em 2000 por *Roy Fielding* na Universidade da Califórnia, mas com um impacto inicial pouco significativo. Se for feita uma análise atual ao número de *Web services* que empregam estes princípios, verifica-se de facto que estes têm vindo a aumentar e mostram já um impacto significativo no plano geral [17].

REST esclarece um conjunto de princípios de arquitetura, pela qual é possível estruturar *serviços Web* que se destacam nos recursos do sistema, abrangendo, como as informações são endereçadas e transferidas através de HTTP por distintos clientes e com linguagens diferentes. Em suma, uma arquitetura do estilo REST consiste numa comunicação entre clientes e servidores, como se pode observar na Figura 5. No qual o cliente envia um pedido (*HTTP Request*) que inclui as informações indispensáveis para realizar uma determinada operação. O servidor processa o pedido e devolve uma resposta (*HTTP Response*) contendo um arquivo XML (*Extensible Markup Language*) ou uma *string* em JSON que pode representar desde uma simples mensagem a um conjunto de informações complexas.

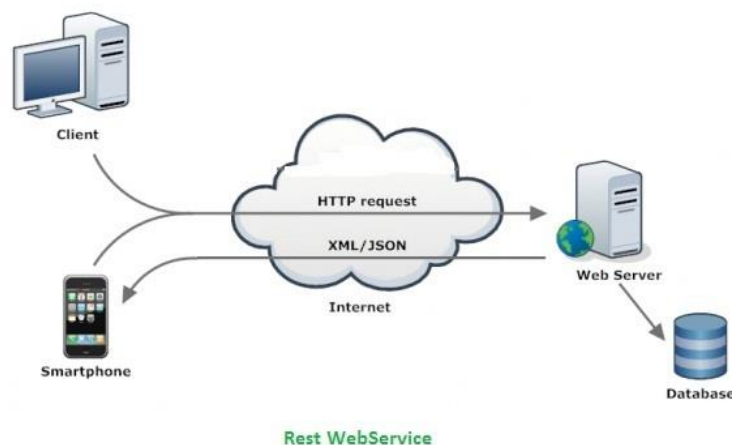


Figura 5 - REST Web Service [18]

Na arquitetura REST, não existe uma descrição do serviço, partindo do princípio que a mesma conhece o que deve ser enviado [19].

Os principais atributos da arquitetura REST são as pequenas quantidades de texto em XML ou JSON, atuar sobre protocolo HTTP e não requerer de um suporte específico de linguagem, pois, os dados são transferidos usando um XML simples ou uma *string* JSON.

3.2.2. JSON (*JAVASCRIPT OBJECT NOTATION*)

O formato JSON é um formato leve de intercâmbio de dados computacionais. Este é de fácil leitura e escrita por humanos e de fácil geração e análise por parte de computadores. JSON é representado em formato de texto e é completamente independente de linguagens de programação [17].

O formato JSON é composto por duas estruturas, onde a primeira é uma coleção de pares de nomes e respetivos valores, muito semelhante ao representado através de uma matriz associativa. Por outro lado, a segunda estrutura é uma lista de valores ordenada, muito semelhante aos conhecidos vetores ou matrizes não associativas [17].

O formato JSON é exposto como sendo a alternativa “sem gordura” do XML, invocando que a mesma informação representada nos dois formatos requer mais metadados no formato XML. Com base neste facto, o JSON intitula-se como sendo mais eficiente de analisar por computadores em comparação com o XML [17]

3.3. *HYPER TEXT MARKUP LANGUAGE (HTML)*

A linguagem HTML serve para compor, através de *tags*, a apresentação de páginas web. Criada em 1989 no laboratório da CERN (*European Laboratory for Particle Physics*) na Suíça por Tim Berners-Lim, o conceito surgiu devido à necessidade de partilhar documentação com pessoas de todo o mundo [14]. O HTML foi produzido com base em SGML (*Standard Generalized Markup Language*), usando as *tags* predefinidas pelo SGML, incluindo a *tag* <A> com o atributo HREF que permite a navegação no documento web. Encontra-se praticamente toda a informação em termos de *tags* da tecnologia na referência [20].

Esta linguagem pode ser consultada em navegadores *Web*. É essencial que, quando se constrói uma página *Web* em HTML, esta seja conciliável com todos os navegadores de feição a que a funcionalidade não esteja submissa do navegador que o utilizador tenha instalado no seu PC (*Personal Computer*). A última versão existente desta tecnologia, denominada HTML5, traz importantes alterações do papel do HTML no mundo *Web*, através de novas funcionalidades em termos semânticos e de acessibilidade [21].

Um editor de texto simples é o suficiente para a conceção de um documento *Web* em HTML, contudo existem aplicações que ajudam o tratamento deste tipo de texto. São exemplos o *Notepad++* e o *NetBeans*.

3.4. PHP (*HYPertext PREPROCESSOR*)

PHP é uma linguagem de *script open-source* empregue no desenvolvimento de conteúdos dinâmicos na elaboração de páginas *Web*. Esta foi uma das primeiras linguagens a privilegiar o processamento do lado do servidor, usando *scripts* diretamente nas páginas HTML (*Hyper Text Markup Language*), em vez de invocar ficheiros externos [22].

O código PHP é demarcado num ficheiro HTML para *tags* `<?php` e `?>`, que correspondem respetivamente ao início e ao final do script.

O PHP ficou popular pelo facto de ser uma linguagem de perceção acessível para iniciantes e apresentar recursos mais avançados para programadores mais experientes. O PHP pode ser utilizado na generalidade dos sistemas operacionais e é suportado pelos principais servidores *web*. O PHP para além de gerar páginas HTML, pode igualmente criar imagens, arquivos PDF (*Portable Document Format*) e animações *Flash*. Todavia, um dos principais pontos fortes do PHP é ser suportado pela vasta gama de bases de dados existentes [22].

3.5. JAVASCRIPT

JavaScript é uma linguagem de *scripting* (permite que se escreva código relacionado a elementos específicos de uma aplicação), baseada em Java, do lado do cliente, orientada aos objetos, que é suportada pela maior parte dos navegadores existentes, não gerando assim qualquer questão de compatibilidade. Suporta a criação de objetos, resposta a eventos, funções, métodos e propriedades.

3.4. JQUERY

O *JQuery* é uma biblioteca de *JavaScript*, que tem como objetivo simplificar a maneira como se desenvolvem aplicações em JavaScript. Desde logo facilita o acesso aos elementos / *tags* HTML que constituem o DOM (*Document Object Model*), alterando as suas propriedades através dos métodos existentes em *JavaScript*, quer através das propriedades dos elementos HTML. Esta biblioteca contém funções que possibilitam o rápido acesso ao DOM e outras funções, que além facilitar a compreensão de quem está a desenvolver, permite fazer mais

com muito menos linhas de código. Uma das funcionalidades mais usadas é mostrar ou esconder um certo elemento através de um evento, ou alterar o tamanho do elemento quer através propriedades de *JavaScript* ou dos atributos CSS (*Cascading Style Sheets*) [23].

A biblioteca *jQuery* veio melhorar a forma como se realizavam os pedidos assíncronos ao servidor, de seu nome *AJAX* (*Asynchronous JavaScript and XML*), que permite realizar pedidos ao servidor sem ser necessário recarregar a página ou carregar uma nova, e assim podem ser realizadas validações de dados e atualização de informação mais rápida do lado do cliente.

A *jQuery* é a biblioteca *Javascript* mais utilizada no desenvolvimento de websites, sendo atualmente utilizada por cerca de 73% dos dez mil sites com maior número de visitas [23]

Existem variantes como o *jQuery UI* usado para o desenvolvimento da UI (*User Interface*) para *Web* de forma facilitada, e o *jQuery Mobile*, usado para o desenvolvimento *Web* móvel, tentando colmatar algumas falhas existentes no *jQuery UI* quando se trata de dispositivos tácteis.

3.5. OPENLAYERS

O *OpenLayers* é uma biblioteca *open source* de *Javascript* que tem como objetivo exibir mapas interativos num navegador *Web*. É uma biblioteca de alto desempenho e com diversas funcionalidades tais como, a exibição de tiles de mapas de diferentes fornecedores como a *Google maps*, *Open Street Maps* ou a *Here maps*, pode exibir no mapa dados vetoriais ou marcadores carregados a partir de uma qualquer fonte em qualquer página *Web*. O *OpenLayers* foi desenvolvido para promover o uso de todo o tipo de informação geográfica e foi lançado sob a licença *2-clause BSD License* (também conhecida por *FreeBSD*) [24].

3.6. BOOTSTRAP

O *Bootstrap* é uma das *frameworks* mais utilizadas no desenvolvimento web, oferecendo *templates de design* baseados em HTML, CSS e *JavaScript* para a apresentação de diferentes componentes da interface com o utilizador, tais como formulários, caixas de texto, botões, menus de navegação, cabeçalhos e rodapés. O uso deste tipo de *frameworks* acelera substancialmente o desenvolvimento de páginas e aplicações com grafismo moderno e

atrativo. De referir que todos os componentes JavaScript do Bootstrap requerem o recurso à biblioteca jQuery [25].

3.7. *PUSH NOTIFICATIONS*

A maior diferença na comunicação com aplicações móveis em relação a aplicações *Web*, prende-se com a capacidade que a aplicação móvel terá para poupar energia ao dispositivo. Num parque informático clássico, por exemplo, a grande parte dos terminais a gerir encontram-se ligados em rede de forma estacionária sendo os seus endereços conhecidos e quase sempre inalteráveis. O mesmo não se pode dizer dos terminais móveis.

Num sistema de gestão e atribuição de serviços de táxis, quando chegar um pedido ao sistema este deve enviar a informação à aplicação do taxista e não ser esta a verificar num certo período de tempo se chegou algum pedido para si. Na prática, numa perspetiva onde a aplicação pedisse repetidamente ao servidor informação para verificar se tem novos pedidos, isto sobrecarregaria o sistema e representaria uma sobre utilização da ligação à rede, bem como o poder e tempo de processamento do servidor.

Um modelo desta natureza, onde o começo da comunicação é realizado pelo servidor é conhecido como modelo *push*. Este modelo apresenta vantagens face a interações do tipo pedido-resposta, também conhecido como modelo *pull*. Embora ambos sejam variações do modelo cliente/servidor, a diferença fundamental que existe entre eles é quem inicializa a comunicação. Na Figura 6 estão representadas as formas de inicialização do fluxo de informação em cenários *pull*, à esquerda, e *push*, à direita.

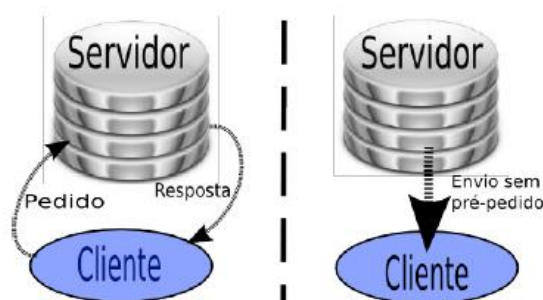


Figura 6 - Modelos *Pull* e *Push*

Ainda que a informação flua sempre no mesmo sentido, no sistema tipo *push*, os dados são enviados diretamente sem um pedido prévio, enquanto no modelo *pull* será sempre necessário realizar o pedido [26].

Para ser possível implementar o sistema de *push notifications* entre o servidor e as aplicações móveis existentes neste projeto, foi necessário utilizar a API fornecida pela *Google* designada *Firebase Cloud Messaging* (FCM).

3.7.1. FIREBASE CLOUD MESSAGING

O FCM é uma solução gratuita de envio de mensagens *push* entre plataformas. É a versão mais recente que ocupou o lugar do antigo GCM (*Google Cloud Messaging*) que é igualmente fornecido pela *Google*. A grande diferença entre o GCM e o FCM é que o último é um serviço multiplataforma, ou seja, funciona com *apps* Android, *iOs* e *Web*.

Na Figura 7 tem-se representado todas as partes necessárias para implementar o envio de *push notifications* através da FCM, em que esta necessitará de incluir um servidor que interaja consigo utilizando o protocolo HTTP ou o XMPP (*Extensible Messaging and Presence Protocol*) e ainda uma aplicação móvel ou Web que receberá a notificação.



Figura 7 - Funcionamento da FCM [27]

É possível enviar mensagens para a aplicação de duas formas: ou pelo servidor, ou então pela consola oferecida e disponibilizada pela *Firebase*. Estas mensagens podem ser enviadas no máximo com um *payload* de 4 KB [27].

3.8. BASE DE DADOS

As bases de dados são uma característica presente na maioria das arquiteturas de *software*. Algumas bases de dados como a *Oracle* [28] e a *Microsoft SQL Server* [29] são as mais populares no grupo das bases de dados comerciais. Nas derradeiras duas décadas a

comunidade *open-source* tem ampliado a qualidade das suas ferramentas computacionais, fazendo o *software open-source* mais apetecível para os empreendedores. Os sistemas gestores de bases de dados que se encontram na liderança deste segmento são o *PostgreSQL* [30] e o *MySQL* [31].

A empresa para a qual foi desenvolvido o projeto descrito nesta dissertação, utiliza o *PostgreSQL* em conjunto com a extensão existente para este, o *PostGIS*. Tanto os produtos da empresa, como também o sistema desenvolvido, necessita de tratar dados de informação geográfica. Este SGBD é o mais usado neste tipo de casos pois para além de ser *open source*, em comparação com a capacidade do *MySQL* interpretar e tratar este tipo de informação, possui uma maior robustez no processamento dos dados e mais funções na análise de informação geográfica [32].

3.8.1. POSTGRESQL

O *PostgreSQL* é uma base de dados *open-source* objecto-relacional. Possui mais de 15 anos de desenvolvimento ativo e uma arquitetura que ganhou um elevado relevo devido à sua fiabilidade e a correta integridade dos dados. É compatível com os principais sistemas operativos (*Windows, Linux, MacOS, etc.*) e é inteiramente aprovada pelo certificado ACID (Atomicidade, Consistência, Isolamento e Durabilidade). A base de dados inclui quase a totalidade dos tipos de dados SQL (*Structured Query Language*), incluindo *integer, numeric, boolean, char, varchar, date, interval e timestamp*, para além disto suporta também o armazenamento de grandes objetos binários, tais como, imagem, som ou vídeo. O *PostgreSQL* apresenta ainda interfaces nativas com as linguagens *C/C++, Java, .Net, Perl, Python, Ruby, PHP, ODBC*, entre outros [33].

O *PostgreSQL* ostenta processos de recuperação eficientes e eliminação de falhas, como o MVCC (*Multi-Version Concurrency Control*), pontos de restauro, replicação assíncrona e otimizador de query. É ainda muito escalável, tanto na quantidade de dados que pode gerir e o número de utilizadores que pode conter. Alguns limites gerais do *PostgreSQL* estão apresentados na Tabela 2 [33].

Tabela 2 - Limites do PostgreSQL (adaptado de [33])

Limite	Valor
Tamanho máximo da base de dados	Ilimitado
Tamanho máximo da tabela	32 TB
Tamanho máximo da linha	1,6 TB
Tamanho máximo do campo	1 GB
Número máximo de linhas por tabela	Ilimitado
Número máximo de colunas por tabela	250-1600, dependendo do tipo de dados da coluna
Número máximo de índices por tabela	Ilimitado

3.8.2. *POSTGIS*

O *PostGIS* é uma extensão espacial gratuita para o *PostgreSQL*. Ela adiciona ao SGBD suporte para tratamento de objetos geográficos permitindo que as *queries* de localização e o próprio tratamento dos dados sejam executadas no *SQL*.

Para além da percepção básica de localização, esta extensão oferece ainda muitos recursos que raramente são encontrados nos SGBD *open-source* concorrentes. Estes recursos podem ser encontrados nas bases de dados comerciais como é o caso da Oracle *Locator/Spatial* e o *SQL Server* [23].

4. ANÁLISE E MODELAÇÃO DO SISTEMA

A análise e definição dos requisitos e a modelação do sistema a ser desenvolvido é uma tarefa de extrema importância uma vez que o sucesso do sistema é dependente desta fase, sendo que uma análise e modelação escassa pode ditar o fracasso da aplicação.

Após o referido, nesta fase será levantada a análise de requisitos que o sistema terá de seguir, posteriormente serão especificados os casos de uso levantados para cada uma das partes constituintes do sistema e por fim será exposto e explicado sucintamente o modelo de dados criados para suportar a plataforma de gestão e atribuição de serviços de táxis.

4.1. ANÁLISE DE REQUISITOS

A plataforma que será apresentada neste documento terá como objetivo o pedido e atribuição de táxis na cidade de Chaves. Para entender o problema foi analisado juntamente com um taxista local todo o processo de serviço de táxi nesta localidade, desde a forma de distribuição dos táxis nas praças do concelho, bem como as regras de atribuição dos serviços pela plataforma. Os requisitos identificados foram os seguintes:

- Na cidade de Chaves existem cinco praças de táxis, sendo que se dividem em duas centrais e três periféricas.
- Os taxistas durante a semana de trabalho, passam por todas as praças, segundo uma sequência que é definida tendo em conta que eles se dividem em equipas de cinco táxis, onde cada equipa tem uma sequência de praças que se repete semanalmente.
- Há um princípio de distribuição de serviços entre todos os taxistas da cidade, com a atribuição de um serviço diário, sendo definido o objetivo pelo gestor da plataforma, que pode ser distinto para outras localidades a considerar de futuro, por isso é um parâmetro de configuração da plataforma. Esta distribuição tem influência na atribuição de serviços, pois, quem ainda não realizou o seu objetivo diário tem sempre prioridade sobre aqueles que já o tenham realizado.
- A atribuição de um pedido de táxi na cidade de Chaves será definida da seguinte maneira: deve ser atribuído ao táxi que está na praça mais próxima de onde foi feito o pedido, na condição de parado e disponível, sendo que os que ainda não realizaram o seu objetivo diário têm prioridade. Devem ser esgotadas todas as hipóteses dos táxis na praça mais próxima que ainda não tenham realizado o objetivo e posteriormente se nenhum dos táxis anteriores tiver disponível, ou aceite o pedido, a plataforma deve então escolher um dos restantes táxis nessa praça selecionada. Se porventura nenhum táxi dessa praça estiver disponível, a plataforma deverá entregar o serviço ao táxi que estiver mais próximo do local onde foi realizado o pedido, tendo em conta se cumpriu ou não o objetivo diário.
- A aplicação deve de permitir que quando um taxista aceitar o serviço e não o consiga realizar por um qualquer motivo, este possa realizar a transferência do mesmo, sendo esta da responsabilidade do taxista pois será contabilizado nos seus serviços realizados. Ao realizar a transferência, o taxista deve colocar o número de porta do taxista para quem será transferido o serviço (pois o taxista pode não ter aderido à plataforma e é pelo número de porta, que o cliente será informado do táxi que irá chegar).
- Por fim, a plataforma terá de permitir o registo de viaturas (número de porta, matrícula, número de registo e lotação) que estarão disponíveis para a distribuição de serviços de táxi, e a aplicação do cliente deverá de ter a opção para escolher o tipo de táxi, normal (até 4 pessoas) ou superior (6 ou 8 pessoas no máximo).

A forma como os táxis se dividem na cidade de Chaves é diferente das cidades vizinhas, em que, por exemplo, em Vila Real, os taxistas não tem praça pré-definida ou fixa.

4.2. CASO DE USO

Um caso de uso é uma descrição narrativa de uma sequência de eventos que ocorre quando um ator (agente externo) usa o sistema para realizar uma tarefa [34]. Um ator é uma entidade externa que interage com o sistema participando num caso de uso [35]. Uma representação gráfica dos casos de uso são os chamados diagramas de caso de uso, cujo objetivo é representar um requisito do sistema que será automatizado [35].

Serão apresentados nas seguintes subseções os casos de uso para cada um dos atores deste sistema.

4.2.1. APLICAÇÃO DO GESTOR

A aplicação que irá ser criada para o gestor da plataforma terá como objetivo a gestão de atribuição de pedidos de táxis. O gestor terá a possibilidade de decisão de inserção de vários elementos necessários para a plataforma. A Figura 8 tem representado o caso de uso identificado para o gestor da plataforma.

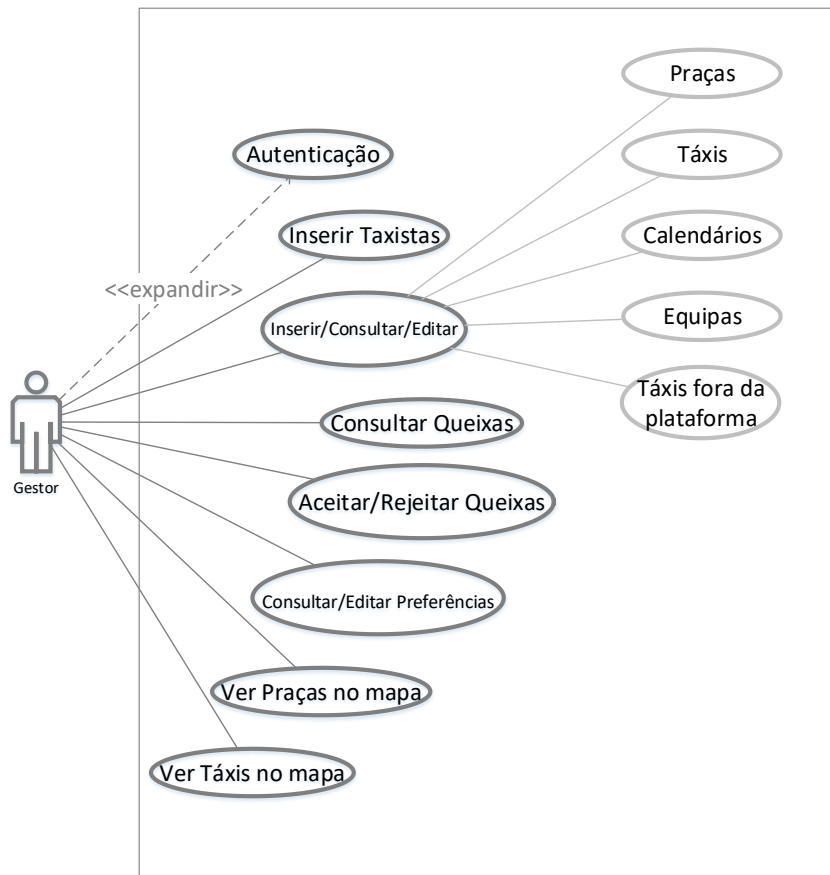


Figura 8 – Casos de uso do Gestor

Serão apresentados de seguida os detalhes de cada uma das funcionalidades para o utilizador gestor.

- Autenticação

Para se ter acesso ao sistema o utilizador deverá introduzir o seu *username* e a sua *password* para este possa efetuar a validação dos campos.

Ao validar corretamente, deve encaminhar o utilizador para a plataforma, se por outro lado, a validação estiver incorreta deverá apresentar um *pop-up* ao utilizador com uma mensagem de erro.

- Inserção

O utilizador poderá optar por selecionar a opção de inserir, ao realizá-lo, terá de escolher uma de seis opções: praças, taxistas, táxis, calendários, equipas e táxis de fora da plataforma. Ao ser escolhida a opção terá de ser apresentado um formulário

de inserção de dados. Quando este for submetido o sistema irá validá-lo e se ocorrer algum erro, abrir um *pop-up* a informar o utilizador.

- Consulta

O utilizador poderá optar por seleccionar a opção de consultar, ao fazê-lo, terá de escolher uma de cinco opções: praças, táxis, calendários, equipas e táxis de fora da plataforma. Quando escolhida a opção serão apresentadas todas as inserções feitas nessa categoria e o utilizador poderá seleccionar um registo. Quando o fizer serão apresentados todos os campos relativos ao registo seleccionado.

- Edição

Quando apresentada a consulta referida no ponto anterior, se o utilizador alterar algum dos campos do formulário terá de ser possível guardar essa alteração. Quando submeter o formulário alterado o sistema irá validá-lo e se ocorrer algum erro, abrir um *pop-up* a informar o utilizador.

- Consulta de Queixa

Deverá ser possível ao utilizador optar pela opção de consulta de queixas. Ao realizá-lo será apresentada uma janela com todas as queixas realizadas pelos taxistas sobre os clientes. Deverá proporcionar ao utilizador visualizar qual o nome do cliente que foi reportado, o nome do taxista que efetuou a queixa, os identificadores destes dois e ainda qual o motivo da queixa por parte do taxista.

- Aceitação/Rejeição da Queixa

Todas as queixas visualizadas pelo utilizador terão de conter duas opções, a aceitação e a rejeição da queixa.

A aceitação da queixa que terá a consequência de ocultar todas as queixas sobre o cliente a que se refere e ainda bloqueá-lo da plataforma.

A rejeição da queixa terá a consequência de ocultar aquela que foi acionada e não terá nenhuma consequência para o cliente

- Consulta e Edição das Preferências

O utilizador poderá optar por selecionar a opção de consultar e editar as preferências. Ao realizá-lo será apresentado um formulário com as preferências atuais da plataforma (objetivo diário, tempo de aceitação do taxista e como se deve atribuir a praça/táxi mais próximo). Ao alterar alguma destas características terá de ser exequível a edição das mesmas. Quando submeter o formulário editado este irá ser validado e se existir algum erro terá de ser apresentado um *pop-up* e informar o utilizador deste.

- Visualização de Praças no mapa

Deverá ser possível ao utilizador selecionar a opção de visualização das praças. Ao fazê-lo será mostrado no mapa da página principal um marcador para cada uma das praças que ele inseriu anteriormente.

Na consulta das praças deverá ser possível ao utilizador ver a informação sobre uma delas, terá também a opção de a visualizar no mapa, que ao seleciona-la mostrará apenas essa praça.

- Visualização de Táxis no mapa

Deverá ser possível ao utilizador selecionar a opção de visualização dos táxis existentes. Ao executá-lo será mostrado no mapa da página principal um marcador para cada um dos táxis inseridos, com a sua localização atual.

4.2.2. APLICAÇÃO DO CLIENTE

A aplicação criada para o cliente terá o objetivo principal de este poder efetuar um pedido de táxi e ir acompanhando o estado do pedido/serviço. A Figura 9 tem representado o caso de uso identificado para o utilizador cliente do serviço de táxis.

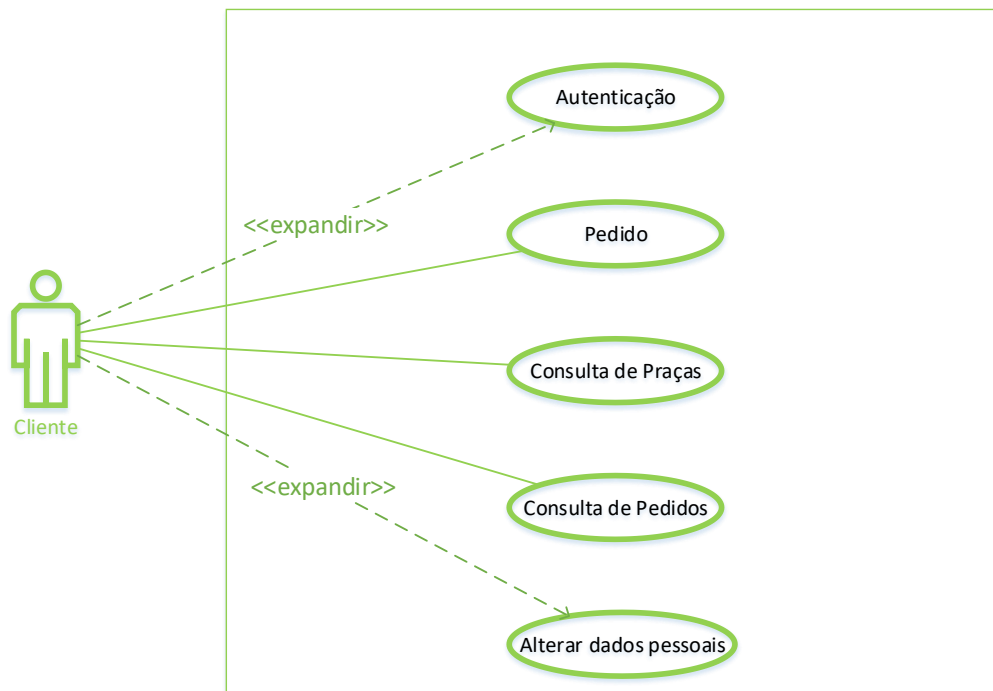


Figura 9 – Casos de uso da aplicação do cliente

- Pedido de um Táxi

Deverá ser possível ao utilizador selecionar a opção de pedido de um táxi. O sistema deverá enviar a informação para a plataforma de gestão e atribuição de serviços de táxis. Esta por sua vez deve retornar ao cliente a informação sobre o estado do pedido efetuado. Esta informação deverá ser atualizada num determinado intervalo de tempo, para que o cliente esteja ocorrente do estado do pedido.

- Consulta de Praças

O utilizador poderá optar por selecionar a opção de consultar as praças, em que o sistema deverá identificar em que cidade este se encontra. Se estiver dentro de uma cidade gerida pela plataforma deverá apresentar as praças desta, se por outro lado, não se encontrar deverá mostrar todas as praças existentes na plataforma. Em cada um dos cenários anterior, ao ser selecionada uma das praças apresentadas, deverá de ser apresentado um mapa com um marcador nas coordenadas da praça selecionada.

- Consulta de Pedidos

O utilizador poderá optar por selecionar a opção de consultar os pedidos de um certo dia. Ao seleciona-la o sistema deverá apresentar por defeito os pedidos efetuados no

dia atual. Se o utilizador selecionar um dos pedidos, deverá aparecer a informação personalizada sobre o pedido escolhido, ou então uma mensagem de erro se o pedido não obteve resposta de nenhum táxi.

Deverá também ser apresentado um calendário para que o utilizador possa escolher o dia em que pretende visualizar os seus pedidos.

- Alteração de Dados Pessoais

Deverá ser possível ao utilizador selecionar a opção de alterar os seus dados pessoais, em que, ao selecioná-la o sistema deverá apresentar um formulário próprio. Quando o formulário for alterado e submetido, o sistema deverá validá-lo e apresentar uma mensagem de erro no caso em que ocorrer alguma inconformidade na validação.

Quando o utilizador necessitar de alterar a sua *password* deverá aparecer um formulário, em que necessitará de introduzir a *password* atual e posteriormente por duas vezes a nova *password*. Ao submeter o formulário o sistema deverá validá-lo e apresentar uma mensagem de erro no caso em que ocorrer alguma inconformidade na validação.

4.2.3. APLICAÇÃO DO TAXISTA

A aplicação que irá ser criada para o taxista terá o objetivo principal de este poder responder aos pedidos que chegam para si. A Figura 10 tem representada o caso de uso identificado para o utilizador taxista.

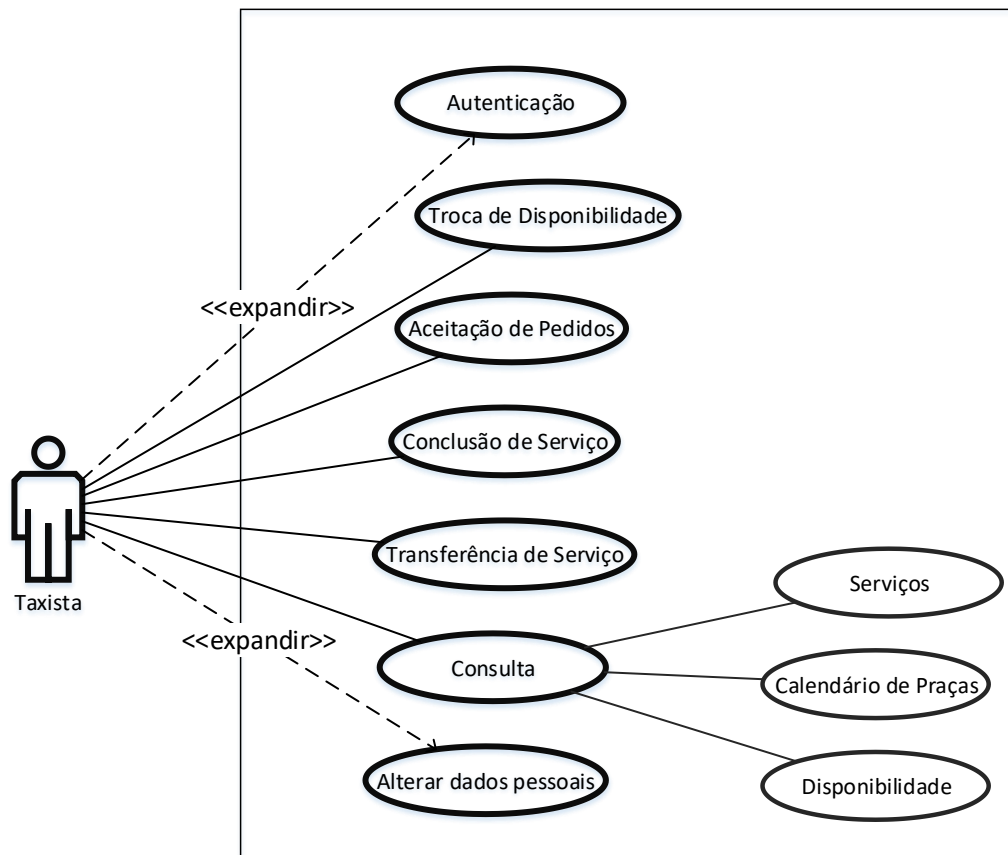


Figura 10 – Casos de uso da aplicação do taxista

- Troca de Disponibilidade

Deverá ser possível ao utilizador seleccionar a opção de alterar a sua disponibilidade, em que o sistema deverá verificar qual é o seu estado atual e encaminhá-lo para a página do estado oposto, ou seja, se estiver indisponível deverá encaminhar para a página do estado disponível e vice-versa.

- Aceitação/Rejeição de Pedidos

O utilizador ao estar disponível, está recetível à chegada de pedidos, em que se o sistema enviar um pedido para o taxista, este deve ser mostrado ao utilizador, com todas as características definidas. Este tem um tempo definido pelo gestor para responder ao pedido.

Se o pedido for rejeitado ou não respondido dentro do tempo definido, o sistema deverá trocar a disponibilidade ao taxista e enviar a informação do pedido para o

seguinte. Por outro lado, se o pedido for aceite o utilizador poderá visualizar todas as características do pedido e do cliente.

- Conclusão de Serviço

O utilizador quando está com um serviço deverá ser possível selecionar a opção de conclusão. Em que o sistema deverá enviar a informação para a plataforma de gestão, efetuando a troca de disponibilidade (de em serviço para disponível) do taxista e encaminha-lo para a página de disponível.

- Transferência de Serviço

O utilizador quando está com um serviço deverá ser possível selecionar a opção de transferi-lo. O sistema deverá avisá-lo que a transferência contará na mesma para os seus serviços diários e apresentar uma janela para introduzir o número de porta do táxi para o qual quer realizar a transferência. Ao submeter a transferência, a plataforma de gestão deverá validá-la e ao ocorrer com sucesso, o sistema deverá de trocar a disponibilidade (de em serviço para indisponível) do taxista e encaminhá-lo para a página principal. Por outro lado, se a validação não tiver sucesso deverá ser apresentada uma mensagem de erro ao utilizador

- Consulta do Calendário de Praças

O utilizador poderá optar por selecionar a opção de consultar o seu calendário semanal de praças, em que o sistema apresenta uma lista com as praças definidas para cada um dos dias da semana. Ao ser carregada numa das praças, deverá ser apresentado um mapa com um marcador nas coordenadas da praça selecionada.

- Consulta de Serviços

O utilizador poderá optar por selecionar a opção de consultar os seus serviços de um certo dia. Ao selecioná-la o sistema deverá apresentar os serviços efetuados no dia atual em que, se o utilizador selecionar um deles deverá aparecer a informação personalizada sobre o pedido escolhido, ou então uma mensagem de aviso se o serviço foi transferido.

Deverá também ser apresentado um calendário para que o utilizador possa escolher o dia em que pretende visualizar os seus pedidos.

- Consulta do Tempo de Disponibilidade

O utilizador poderá optar por selecionar a opção de consultar a sua disponibilidade num determinado dia. Quando selecionado um dia o sistema deverá apresentar uma janela com o tempo que o taxista esteve em cada um dos estados existentes (indisponível, disponível e em serviço). Se o estado do utilizador não for alterado no dia selecionado, será apresentada na janela, o estado em que ele esteve todo o dia

Deverá também ser apresentado um calendário para que o utilizador possa escolher o dia em que pretende visualizar a sua disponibilidade.

4.3. MODELO DE DADOS

Após realizada a análise de requisitos e especificados os casos de uso foi pensado o modelo de dados para o sistema, ele é apresentado de forma compactada na Figura 11.

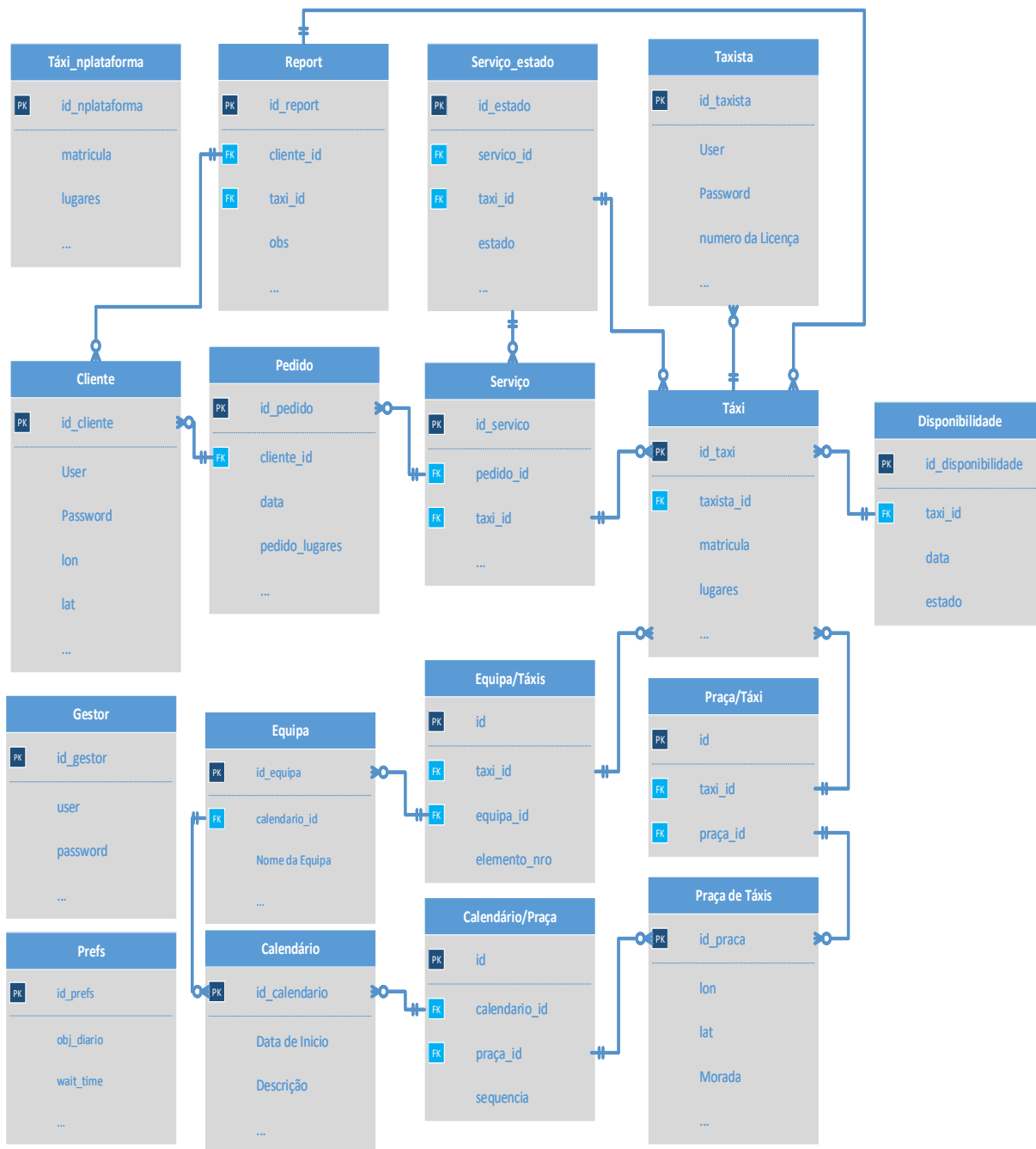


Figura 11 – Modelo de Dados Compactado

Serão agora explicadas cada uma das tabelas principais e seguidamente, subdividido este modelo de dados em diversas partes para uma explicação mais pormenorizada.

- Taxista

Esta tabela tem o objetivo de registar todos os taxistas inseridos pelo gestor. Os taxistas inseridos terão a capacidade de realizar o caso de uso da Figura 10 (secção 4.2.3). Um taxista está associado a um táxi.

- Táxi

Esta tabela tem a funcionalidade de registrar os táxis existentes. É a tabela central da solução, pois ao se realizar um serviço é necessário ter um táxi e também o táxi necessita de ter um calendário de praças. Um táxi para ser inserido necessita que a viatura esteja presente na base de dados da empresa, para que seja possível aceder à sua localização em tempo real. Este táxi está associado a uma disponibilidade. As disponibilidades existentes são:

- Indisponível
- Disponível
- Em Serviço
- Com Pedido Recebido

- Praça de Táxis

A tabela praça de táxis tal como o nome indica, tem a funcionalidade de registrar todas as praças existentes nas diversas cidades que a plataforma suportará.

- Calendário

Esta tabela tem a funcionalidade de registrar todos os calendários existentes. Cada calendário tem a si associado uma sequência de praças de táxis que serão inseridas na tabela calendário/praça. Um calendário está associado a uma equipa.

- Equipa

Esta tabela tem o objetivo de registrar todas as equipas de táxis das diversas cidades em que a plataforma esteja inserida. Cada equipa tem associada a si um conjunto de táxis que serão inseridos na tabela equipa/táxi.

- Cliente

Esta tabela tem a funcionalidade de registrar os clientes que poderão efetuar pedidos de serviços de táxis, para isso ela terá de conter toda a informação que o sistema necessitará.

- Pedido

Nesta tabela são inseridos todos os pedidos solicitados por parte dos clientes, por isto um pedido tem de estar associado a um cliente e a um serviço, mesmo que não obtenha resposta por parte de nenhum táxi.

- Serviço

Esta tabela tem o objetivo de registrar todos os serviços efetuados pelos táxis existentes. Um serviço necessita de estar associado a um táxi, mas se este táxi recusar o serviço, este será associado a outro táxi e assim sucessivamente até que um aceite, ou não existam mais disponíveis.

- Estado do Serviço

A tabela serviço_estado tem a funcionalidade de registrar todos estados que um serviço poderá ter, que poderá ser de um só táxi, ou então de diversos. Por estas razões o estado de serviço tem de estar associado a um serviço e poderá estar associado a diferentes táxis. Os estados existentes serão:

- Sem Táxis Disponíveis
- Aceite
- Concluído
- Transferido
- Recusado

- Report

Esta tabela tem a funcionalidade de registrar todas as queixas efetuadas pelos taxistas aos clientes e ainda lidar com as decisões do gestor sobre estas. Uma queixa está associada a um taxista/táxi e a um cliente.

- Gestor

Esta tabela tem a funcionalidade de registrar todos os gestores da plataforma das diferentes cidades suportadas. Os registos apenas serão inseridos nesta tabela se estiverem presentes na base de dados da empresa.

- Preferências

Nesta tabela tem-se apenas um registo por cada cidade administrada e tem como objetivo definir o modo como a gestão de atribuição do serviço de táxis é feita na cidade (objetivo diário que os taxistas têm de cumprir, intervalo de tempo que os taxistas têm para decidir o pedido e o modo como é realizado a atribuição da praça/táxi mais próximos).

- Táxis Fora da Plataforma

Esta tabela tem como objetivo registar todos os táxis de uma cidade gerida pela plataforma, mas que não tenham aderido a esta. Necessita de ter as informações necessárias para que se um serviço for transferido para um táxi desta categoria o cliente consiga saber qual será o táxi que irá chegar.

Na Figura 12 representam-se as tabelas que serão utilizadas na atualização da praça do dia que os taxista terão de estar, segundo o seu calendário.

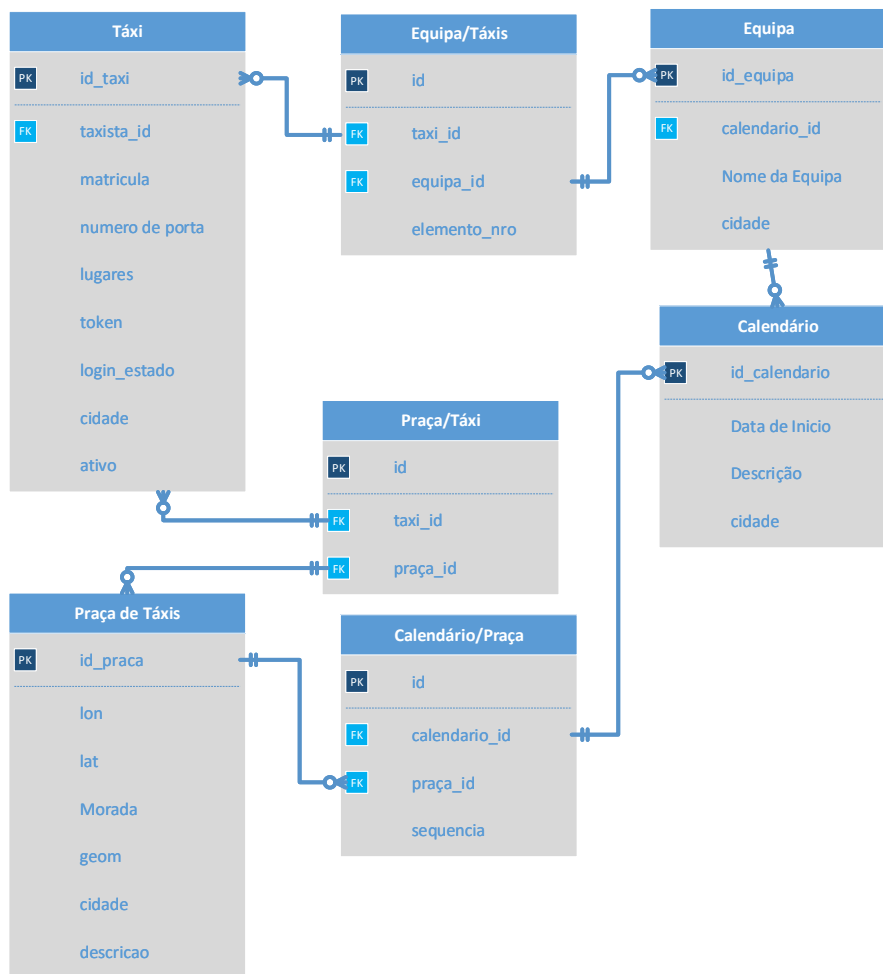


Figura 12 – Tabelas Táxi, Equipa, Calendário e Praça de Táxis

Um calendário tem de conter sete praças de táxis (uma por cada dia da semana), para isso, a tabela relacional calendário/prança tem a função de responder à relação de muitos para muitos que existe entre estas duas tabelas. Nesta tabela relacional tem-se ainda um campo “sequencia” que terá o objetivo de verificar qual o dia da semana em que o calendário se encontra.

Posto isto, tem-se uma equipa que tem um calendário a si associado através de uma chave externa (`calendário_id` na tabela equipa), aqui tem-se uma relação de 1 para muitos, pois uma equipa só tem um calendário, mas este pode pertencer a várias equipas. Uma equipa tem ainda um conjunto de táxis associados, que tal como na relação entre os calendários e as praças, foi criada a tabela relacional equipa/táxi para suportar esta relação.

Por fim tem-se a relação entre os táxis e as praças através da tabela praça/táxi, que tem como objetivo, no algoritmo de atribuição de serviços, facilitar a verificação de quais serão os táxis que estarão numa praça naquele dia. Esta tabela deverá ser atualizada diariamente, sempre

que o taxista entre pela primeira vez na aplicação, ou então, que requisite a atualização da praça.

A Figura 13 apresenta as tabelas envolvidas no pedido e atribuição de serviços de táxi.

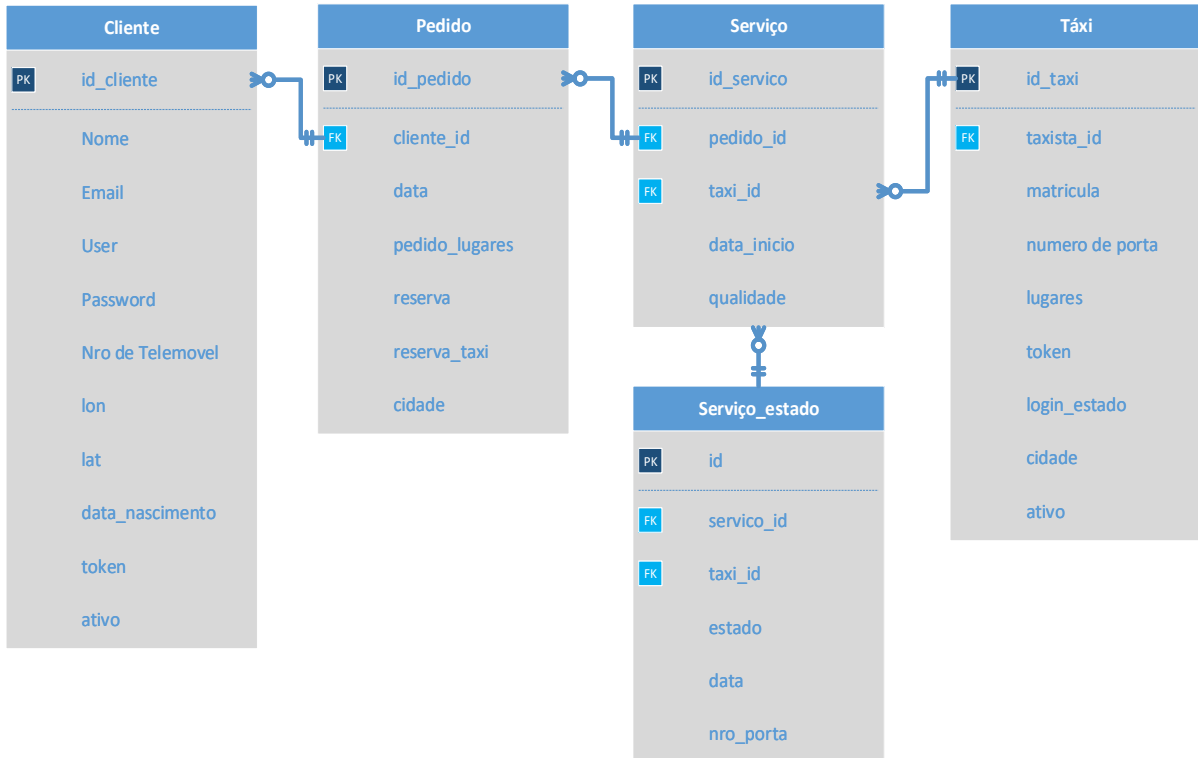


Figura 13 – Cliente, Pedido, Serviço, Táxi

Quando o cliente efetua um pedido, as características deste serão inseridas na tabela pedido e posteriormente, quando for atribuído o táxi mais próximo ao pedido, este é inserido na tabela serviço como uma chave estrangeira (táxi_id), tem ainda outra que é o identificador do pedido (pedido_id). Quando um táxi executar uma ação sobre o pedido que chegou, este deverá ser inserido na tabela serviço_estado que é uma tabela relacional entre o serviço e o táxi. Quando o pedido for rejeitado pelo taxista, o registo na tabela serviço deverá ser editado para o identificador do táxi seguinte.

Este processo será explicado no capítulo seguinte (Implementação), na secção da plataforma de gestão.

5. IMPLEMENTAÇÃO

Neste capítulo será apresentada a arquitetura do sistema e posteriormente descrita a implementação de cada um dos componentes desta arquitetura.

5.1. ARQUITETURA DO SISTEMA

O sistema concebido é composto por três aplicações e uma plataforma de gestão. Na Figura 14 está representada a arquitetura do sistema desenvolvido, em que se tem duas aplicações móveis para dispositivos com o SO Android, a do cliente e a do taxista, uma aplicação *Web* para o gestor, e uma plataforma de gestão onde estão criados os *Web Services* de interligação, entre as diferentes aplicações e a base de dados apresentada na secção 4.3.

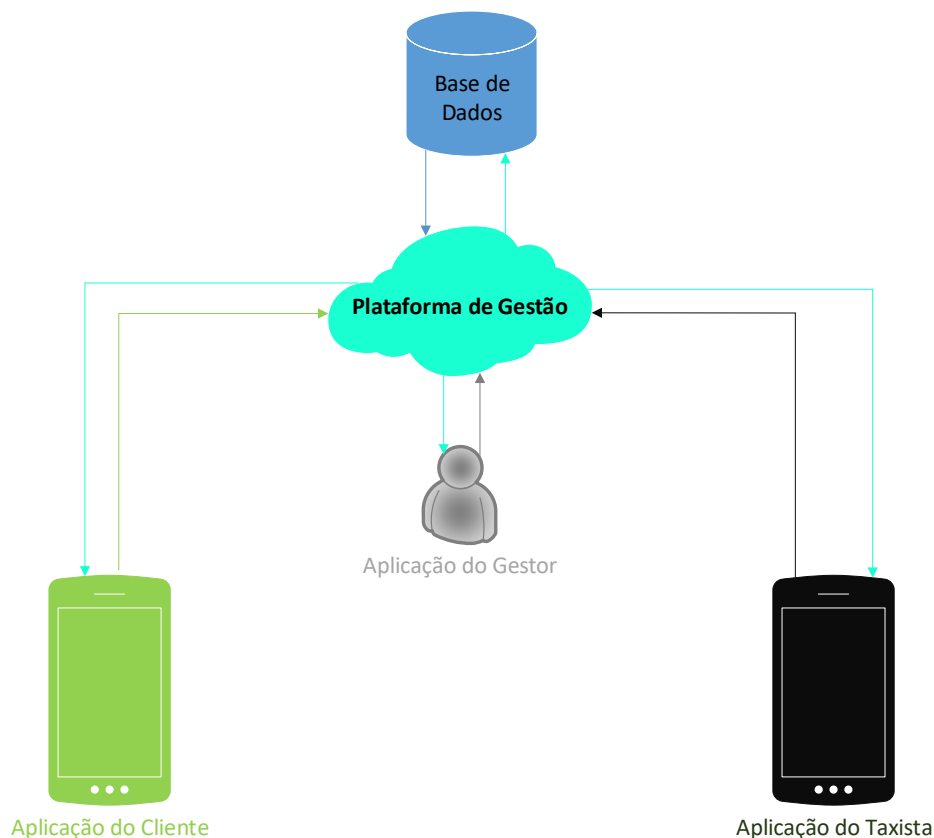


Figura 14 – Arquitetura do Sistema Desenvolvido

Serão explicados ao pormenor nas secções seguintes cada um dos elementos constituintes desta arquitetura.

5.2. PLATAFORMA DE GESTÃO

A plataforma de gestão é o centro de todas as operações realizadas no sistema. É nesta parte do sistema que são criados os *Web Services* (secção 3.2) que as aplicações necessitarão. Estes *Web Services* são utilizados pelas aplicações para interagir com a base de dados, para inserir, editar e consultar dados e ainda a validação dos dados vindos das aplicações desenvolvidas.

Esta parte do sistema foi desenvolvido em linguagem PHP (descrita na secção 3.4) e é nela que estão escritos e estruturados todos os métodos necessários para comunicar com as aplicações.

Para ser possível a plataforma de gestão enviar mensagem para as aplicações móveis (*push notification*, explicada na secção 3.7) é necessário configurar a *Firebase Cloud Messaging*. Para isso é necessário possuir uma conta Google e posteriormente criar um projeto e associar as aplicações a este projeto. Após esta configuração inicial, é necessário retirar das

configurações do projeto a chave do servidor (Configurações do Projeto→*Cloud Messaging*), que é uma chave identificativa do projeto criado.

Na Figura 15 tem-se representados os procedimentos necessários para o funcionamento do envio das *push notifications* da plataforma de gestão para as aplicações móveis do cliente e do taxista.

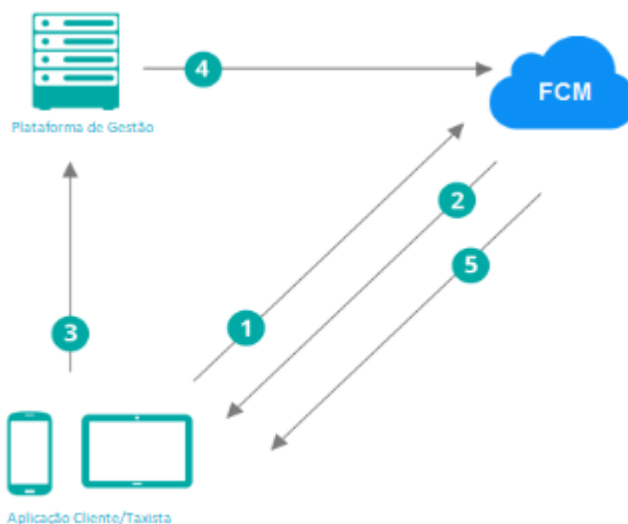


Figura 15 - Modelo de registo e envio de notificações da FCM

Este funcionamento segue uma ordem que será detalhada a seguir:

1. Quando é instalada a aplicação num dispositivo móvel, este conecta-se ao servidor da FCM e regista-se.
2. Após receber o registo, a FCM emite um *token* de registo para o dispositivo. Este *token* é único e exclusivo do dispositivo que se registou.
3. Sempre que uma das aplicações móveis submeter o *login*, a plataforma ao verificar que corresponde a um utilizador, verificará se já tem um *token* associado na base de dados, se já existir verifica se é igual ao que chegou no *login*, se não se verificar, irá editar o *token* que tem na base de dados pelo que chegou, se por outro lado ainda não tem nenhum irá associa-lo ao utilizador que realizou o *login*.
4. Quando a plataforma necessitar de enviar uma notificação, é realizado um pedido ao servidor da FCM contendo a mensagem de envio, juntamente com o *token* de registo do utilizador a quem será enviada a notificação.

5. O servidor FCM identifica o dispositivo usando o *token* de registro e envia a mensagem.

Quando a plataforma tem de realizar uma decisão para definir qual a praça ou táxi que estão mais próximas, ela utiliza a API de *routing* da *Here Maps* para realizar essa decisão. Com esta API consegue-se calcular rotas que combinam os seus próprios critérios de cálculo, baseando-se em dados atualizados do mapa e tem a funcionalidade de incluir informação de trânsito em tempo real. Ao realizar um pedido a esta API, o resultado apresentado retorna informações como, as ruas de início e fim, a distância entre os dois pontos, o tempo com e sem tráfego, entre outros.

Foi criado um *Web Service*, que engloba o *login* das três aplicações, que validará as credenciais de acesso e permitirá que o gestor verifique se é o primeiro *login* realizado por este, numa nova cidade. Ao comprová-lo inserirá as definições de *default* para a nova cidade, preferências essas que o gestor poderá posteriormente editar. Estas preferências de *default* são: sem ter nenhum objetivo diário (*obj_diario* = 0), procurar os táxis mais próximos em termos de distância (*short_distance* = *true*) e por fim, o taxista tem 10 segundos para responder ao pedido (*wait_time* = 10).

No caso do taxista, ao validar as credenciais, o *Web Service* de *login* verifica se o utilizador já realizou alguma alteração de disponibilidade no corrente dia, se isto não se verificar a plataforma irá atualizar automaticamente a praça do dia ao taxista.

Para ser possível ao gestor realizar as inserções das diferentes características que serão essenciais para a gestão e atribuição de serviços de táxis (praças, taxistas, táxis, equipas e calendários), foram criados *Web Services* para cada uma destas inserções. Estes *Web Services* têm como objetivo a validação do formulário submetido. No caso da inserção de um táxi necessita de validar se o *email* introduzido corresponde ao de um taxista válido e também tem de verificar se a matrícula introduzida existe na base de dados da empresa.

Quando se trata da visualização das características anteriormente inseridas, a plataforma ao conhecer qual a cidade que o gestor está a administrar, apenas apresentará os dados dessa cidade.

Na Figura 16 tem-se representado um esquema de como a plataforma processa um serviço de táxi.

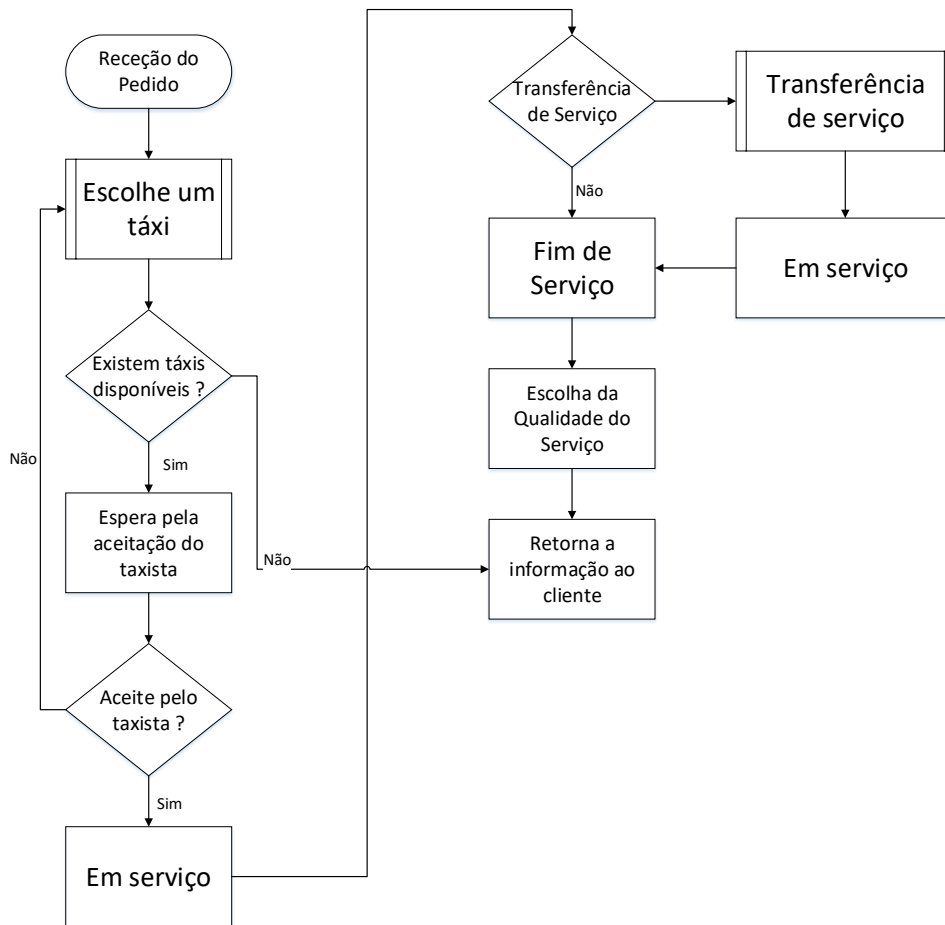


Figura 16 – Estados de um Serviço

Quando um cliente efetua um pedido, este é enviado para o *Web Service* correspondente onde é inserida um novo registo na tabela “pedido” da base de dados e posteriormente executado o algoritmo de atribuição de um serviço, que será explicado posteriormente.

Ao terminar o algoritmo pode acontecer um de três cenários: existem táxis disponíveis e o algoritmo atribui um táxi, não existem táxis disponíveis, por último, o cliente não se encontra em nenhuma cidade abrangida pela plataforma. No primeiro caso, a plataforma envia uma *push notification* para o taxista com as informações do pedido e retorna a informação ao cliente, avisando-o que lhe foi atribuído um táxi, ficando este à espera de uma resposta. No segundo cenário a plataforma retorna o pedido enviado pelo cliente com a informação que o pedido não obteve resposta, por fim no último cenário a plataforma informa o cliente que não se encontra em nenhuma cidade abrangida por si, enviando-lhe todas as cidades em que

a plataforma se encontra disponível, para que o cliente possa escolher uma para realizar o seu pedido.

O taxista ao aceitar o pedido, esta aceitação é enviada para o *Web Service* de aceitar pedido, que inserirá na tabela *serviço_estado* a aceitação e é enviada uma *push notification* para o cliente informando-o que o seu pedido foi respondido e o táxi está a caminho. Até ao fim do serviço ambas as aplicações, cliente e taxista, vão pedindo informação regularmente sobre esse serviço à plataforma, que para isto foi criado o *Web Service* atualizações.

Se o taxista rejeitar o pedido, ou não respondendo em tempo útil, esta é enviada para o *Web Service* *rejeitar_pedido* que irá trocar a disponibilidade ao taxista e inserir na tabela *serviço_estado* o estado da rejeição do pedido. De seguida irá de novo realizar o algoritmo, enviando uma *push notification* para o novo taxista. Se porventura o algoritmo retornar que não existem mais táxis disponíveis, a plataforma envia uma *push notification* ao cliente dando esta informação.

Quando o taxista enviar a notificação de fim do serviço, este irá para o *Web Service* de *concluir_serviço*, que irá trocar a disponibilidade do taxista, inserir a conclusão na tabela *serviço_estado* e informar o cliente enviando-lhe uma *push notification*. Ao recebê-la o cliente terá de escolher a qualidade que pretende atribuir ao taxista/serviço. Quando a classificação do cliente é realizada, a informação será enviada para o *Web Service* de *qualidade_serviço* que inserirá a qualidade atribuída na base de dados e terminar o processo de atribuição de um serviço de táxi.

Se o taxista porventura transferir o serviço para outro táxi, introduz o número de porta deste, sendo enviada a informação para o *Web Service* de *transferir_serviço*. Este terá de verificar se o número existe na base de dados (na tabela *táxi* ou então na tabela *táxi_nplataforma*, táxis que não aderiram à plataforma). Se o número de porta existir é verificado se o táxi se encontra na plataforma e se tem o *login* efetuado na sua aplicação, enviando-lhe a informação da transferência. Por outro lado, se não se encontrar com o *login* realizado ou se tratar de um táxi de forma da plataforma o cliente é informado do estado atual da transferência do serviço. Se o número de porta não existir em nenhuma tabela, o taxista é informado desse erro e o serviço continua a ser seu. Como se pode observar pelo fluxograma da Figura 17

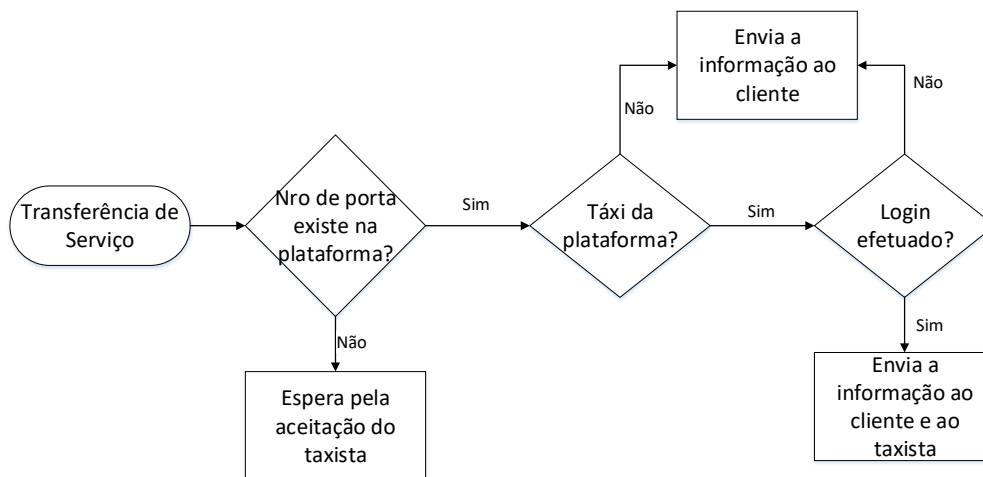


Figura 17 – Transferência de um serviço

A Figura 18 apresenta os passos seguidos na atribuição de um serviço de táxi, segundo a análise de requisitos elaborada na secção 4.1.

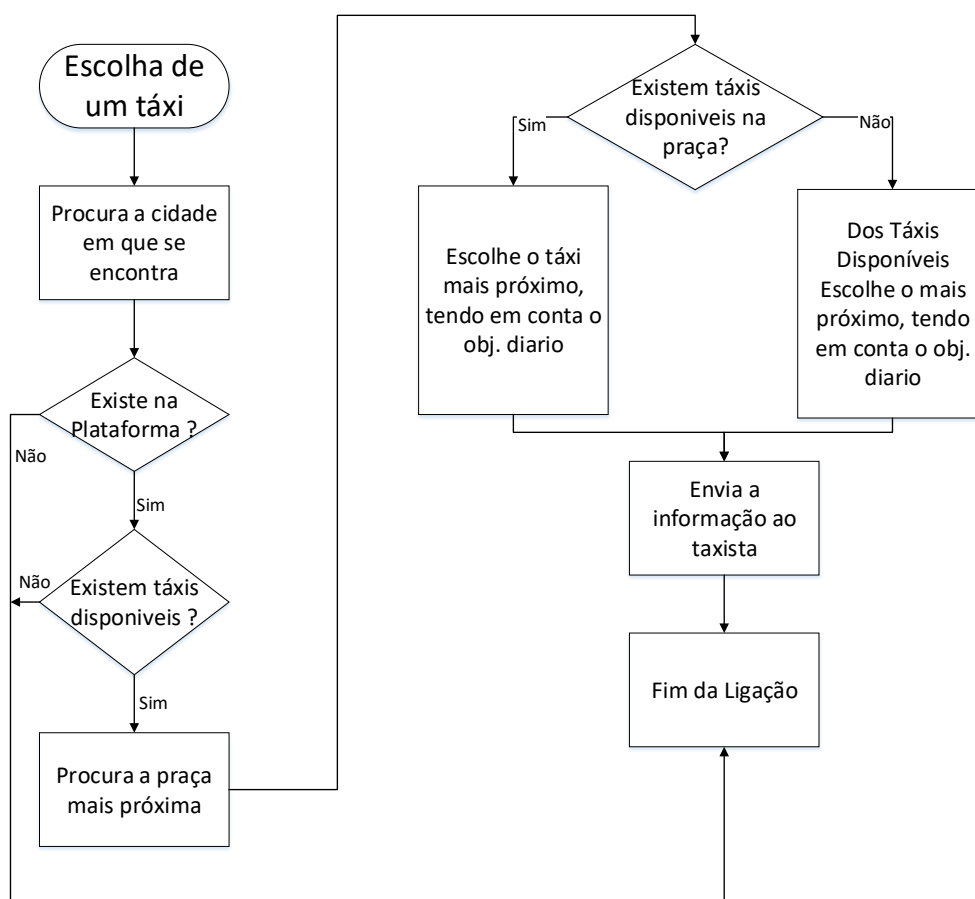


Figura 18 – Estados do algoritmo de atribuição de serviços

Ao receber o pedido, a plataforma em primeiro lugar verifica, segundo as coordenadas do cliente, em que cidade este se encontra. Ao se encontrar dentro de uma cidade abrangida

pela plataforma, esta é guardada juntamente com as preferências a ela associada. Por outro lado, ao não se inserir em nenhuma das cidades abrangidas pela plataforma o pedido será cancelado, enviando ao cliente esse erro e a informação das cidades existentes para que este posteriormente consiga realizar o pedido. Quando o cliente voltar a efetuar o pedido escolhendo uma das cidades abrangidas pela plataforma, este seguirá o seu processo normal.

Após este passo, é verificado se existe algum táxi disponível na cidade em que o cliente se encontra, ou então na cidade escolhida. Se estiver algum táxi disponível será guardado para posteriormente se necessário ser utilizado. Não estando nenhum disponível o cliente é informado que o pedido não obteve reposta.

Posto isto, é verificado qual das praças da cidade é aquela que está mais próxima do cliente e depois de obter o resultado desta verificação, apura-se quais os táxis que estão nesta praça no corrente dia. Destes táxis verifica-se quais estão disponíveis, de seguida se estão dentro da zona geográfica da praça. Ao se verificar isto, serão apurados o número de serviços realizados por cada um dos táxis no corrente dia. Dos táxis que ainda não realizaram o seu objetivo diário é verificado qual está mais próximo do cliente. Se todos os táxis já tiverem realizado o objetivo diário é apurado qual o mais próximo do cliente.

Se nenhum táxi estiver disponível na praça mais próxima, é verificado dos táxis inicialmente disponíveis, aqueles que ainda não tenham cumprido o seu objetivo diário e se encontrar pelo menos um táxi nesta situação é verificado qual está mais próximo do cliente sendo-lhe atribuído esse serviço. Se todos já tiverem cumprido o objetivo é escolhido o que estiver mais próximo do cliente.

O táxi que tiver sido atribuído ao cliente receberá uma *push notification* com as características do pedido, que são a localização do cliente, a distância e o tempo que demorará a chegar ao cliente. Posto isto o algoritmo é terminado ficando a plataforma a aguardar pela resposta do taxista.

Uma das grandes dificuldades na realização deste projeto foi o processo de contagem do tempo de resposta ao pedido por parte de um taxista. Esta contagem, para uma séria abordagem da plataforma, tem de se encontrar do lado do servidor e não ser a aplicação do taxista a realizar essa contagem e passado o intervalo de tempo definido, enviar a informação para a plataforma a informar que não respondeu a tempo ao pedido.

Como a linguagem PHP é síncrona, ou seja, é difícil a criação de *threads* e consequentemente a criação de processos paralelos ao principal. Inicialmente tentou-se instalar a extensão do PHP chamada *pthreads*, mas nunca se conseguiu por a funcionar corretamente. A solução encontrada para solucionar este problema foi a troca de sistema operativo, do servidor que se encontrava num SO *Windows* e passou para o SO *Linux*. No momento de começar a contar o tempo de espera foi executado um *script* de PHP em que o *Web Service* que o inicia não necessita de esperar pela resposta, enviando logo a resposta ao cliente. Este era um dos problemas com a solução em *Windows*, pois ao executar o *script* a resposta não era enviada ao cliente até que o intervalo de tempo de espera tivesse sido cumprido.

5.3. APLICAÇÃO DO GESTOR

Esta parte do sistema, tem como objetivo principal o registo, consulta e edição de todas as componentes envolvidas na gestão e atribuição de táxis. Esta aplicação *Web* foi desenvolvida, tal como a plataforma de gestão, em linguagem PHP, pois o gestor terá acesso a ela através de um navegador. Na Figura 19 apresenta-se a página inicial da aplicação do gestor.

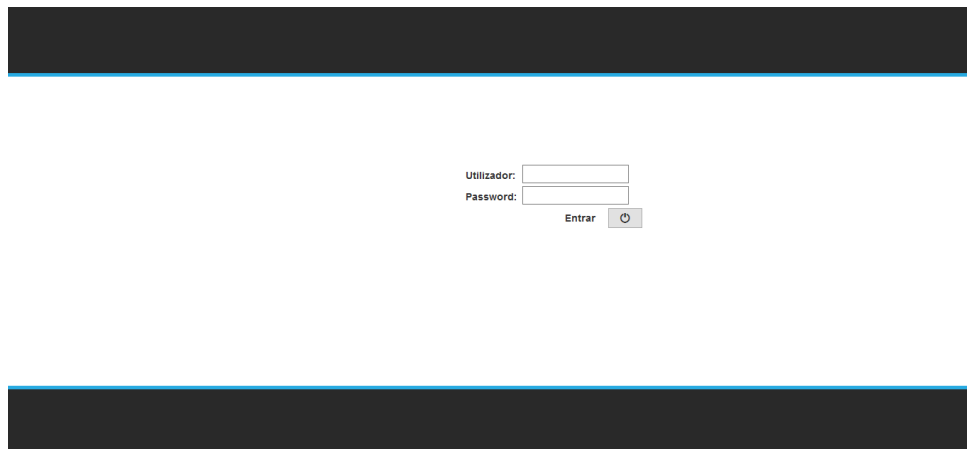


Figura 19 – Página de *Login* da Aplicação do Gestor

Inicialmente o gestor ao aceder à aplicação é-lhe mostrada a página de *login*, em que tem de introduzir as suas credenciais, que serão iguais às utilizadas na plataforma da empresa, o *GeoCar*, pois este utilizador necessita de estar também registado nesta plataforma da empresa. Ao submeter as credenciais, elas serão enviadas para o *Web Service* de *login*, que irá validadas e se estiverem corretas a aplicação será reencaminhada para a página principal, que está apresentada na Figura 20.

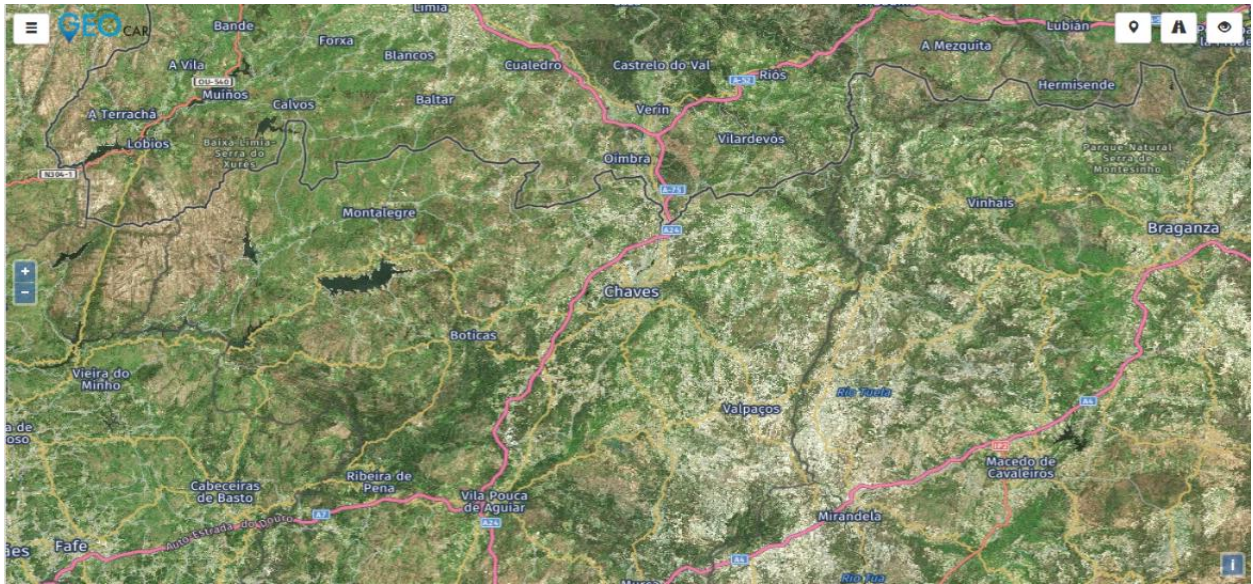


Figura 20 - Página Principal da Aplicação do Gestor

Nesta página ao se carregar de menu existente no canto superior esquerdo, será aberto um menu lateral, que está apresentado na Figura 21.

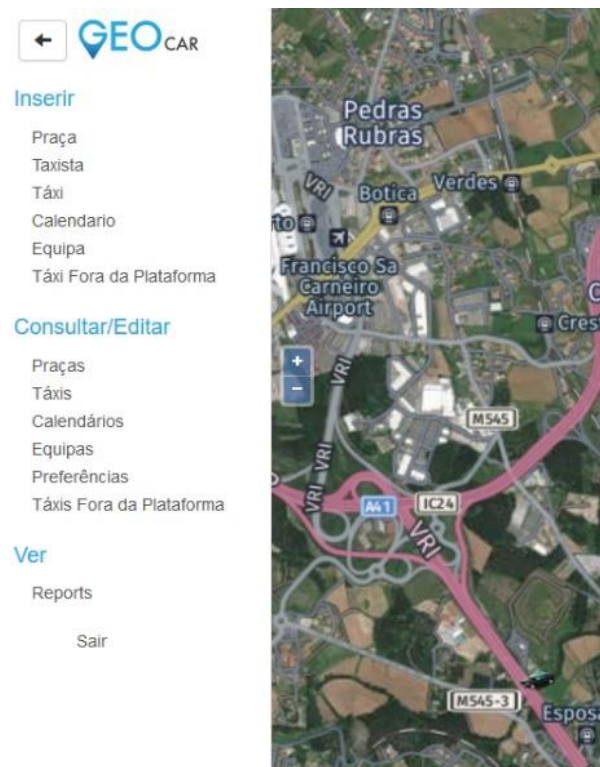
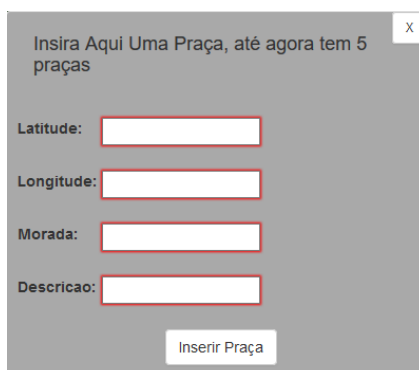


Figura 21 – Menu lateral da página principal

Este menu possibilita a escolha das características de gestão e atribuição de serviço de táxis que se quer inserir ou consultar. Pode ser igualmente terminada a sessão neste menu.

Ao se selecionar uma inserção, neste caso de uma praça, será exibido o formulário apresentado na Figura 22.



Insira Aqui Uma Praça, até agora tem 5 praças

Latitude:

Longitude:

Morada:

Descricao:

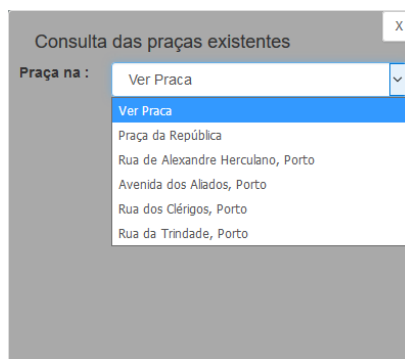
Inserir Praça

Figura 22 – Formulário de Inserção de uma Praça

Quando o formulário for submetido, a plataforma no *Web Service* correspondente à característica inserida, irá validar os dados e retornar um sucesso, a aplicação irá ser reencaminhada para a página principal, ou então, apresentada uma mensagem de erro.

Isto acontece igualmente para qualquer outro caso de inserção (taxistas, táxis, calendários, equipas e táxis que não aderiram à plataforma).

Após se realizar uma inserção é possível consultá-la e se necessário alterar qualquer um dos campos. Para isto, no menu lateral apresentado na Figura 21 seleciona-se uma opção de consultar/editar. Em que a aplicação irá pedir informação ao *Web Service* de consulta das características de gestão de serviços de táxis, neste caso de praças que existem na cidade administrada pelo gestor. Na Figura 23 tem-se apresentada a janela de consulta das praças inseridas.



Consulta das praças existentes

Praça na :

- Ver Praca
- Praça da República
- Rua de Alexandre Herculano, Porto
- Avenida dos Aliados, Porto
- Rua dos Clérigos, Porto
- Rua da Trindade, Porto

Figura 23 - Janela de consulta das praças

Ao ser selecionada uma praça, será realizado um pedido à plataforma para se obter as características dela e apresentar ao utilizador, em que se for editado algum campo, aparecerá um botão para confirmar a edição. Ao submeter a edição, a plataforma validará os novos campos e redirecionar a aplicação de volta à página principal no caso de sucesso da inserção, ou então, apresentar uma mensagem de erro no caso da validação for incorreta.

Tal como esta consulta/edição de uma praça, todas as outras inserções realizadas anteriormente, excetuando o caso dos taxistas, pois cabe a estes (na sua aplicação móvel) a alteração dos seus dados pessoais, podem ser igualmente consultadas e se pretendido alteradas. Para isso segue-se os passos realizados anteriormente.

No caso da consulta de visualizar e alterar praça, tem a particularidade de se poder visualizar no mapa onde é que se encontra a praça, carregando no botão “Ver no mapa” na janela aberta para consulta. A plataforma com as coordenadas que tem da praça mostra um marcador no mapa.

A Figura 24 tem representada a página de consulta e atuação sobre as queixas efetuadas pelos taxistas.



Cliente		Taxi			
Id do Cliente	Nome do Cliente	Id do Taxi	Nome do Taxista	Observações	Bloquear Cliente
1	Tiago	5	1234	Fxo	<input checked="" type="checkbox"/> <input type="checkbox"/>

Figura 24 – Consulta dos *Reports* Efetuados pelos Taxistas

Ao aceder a esta página, a aplicação efetuará um pedido ao *Web Service report* que retornará e mostrará ao gestor todas as queixas que ainda não tiverem sido respondidas. Em cada linha é apresentada a informação do identificador do cliente e o seu nome. São apresentados os

mesmos dados para o taxista. Por fim, tem-se o motivo da queixa e os botões de atuação sobre esta.

Ao ser carregado no botão de rejeitar a queixa (✘), a plataforma irá ocultar esta linha das queixas visíveis e reencaminhar a aplicação para a página principal. Por outro lado, ao ser carregado no botão de aceitar a queixa (✓) a plataforma irá bloquear o cliente e remover todas as queixas visíveis do cliente que acabou de ser bloqueado. Após esta ação a aplicação será direcionada para a página principal.

Por fim, na página principal (Figura 20) existem três botões no canto superior direito, o da direita tem o objetivo de mudar o tipo de mapa como se pode observar pela Figura 25.

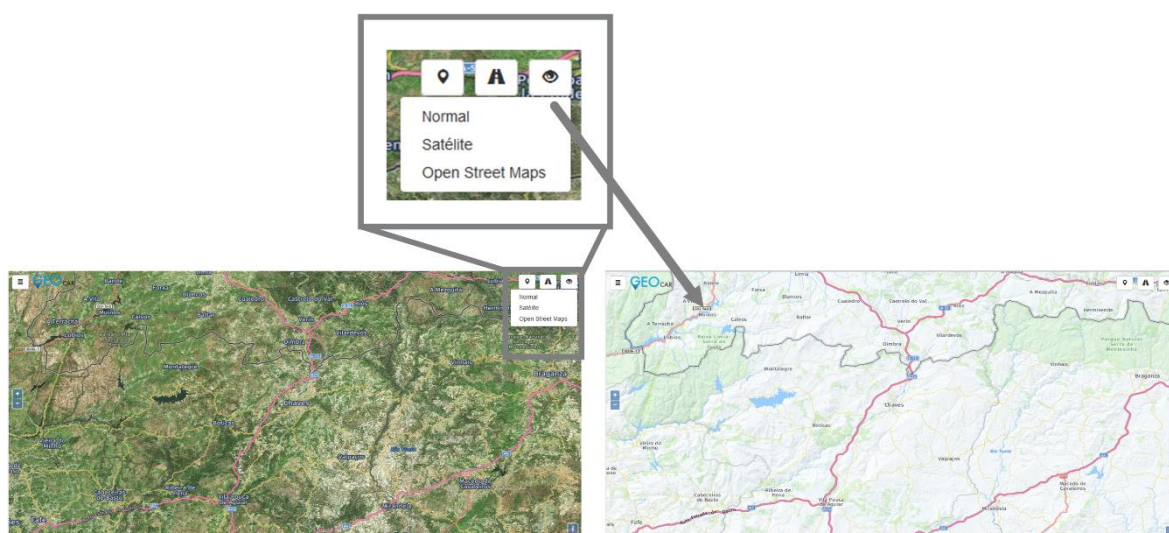


Figura 25 - Mudança no tipo de mapa a apresentar

Os restantes botões têm a função de assinalar no mapa a localização das praças (centro) e a localização dos táxis (esquerda).

5.4. APLICAÇÃO DO CLIENTE

Nesta seção será explicada a aplicação Android criada para o cliente. A aplicação criada tem como principal objetivo efetuar pedidos de táxis por parte dos clientes e o acompanhamento deste até à sua conclusão. Tem ainda outras funcionalidades como a consulta das praças existentes na cidade, os pedidos realizados num determinado dia e também a visualização do perfil do utilizador.

Na Figura 26 estão representadas as *activities* (funcionalidades) criadas para esta aplicação, em que será explicada cada uma delas a seguir.

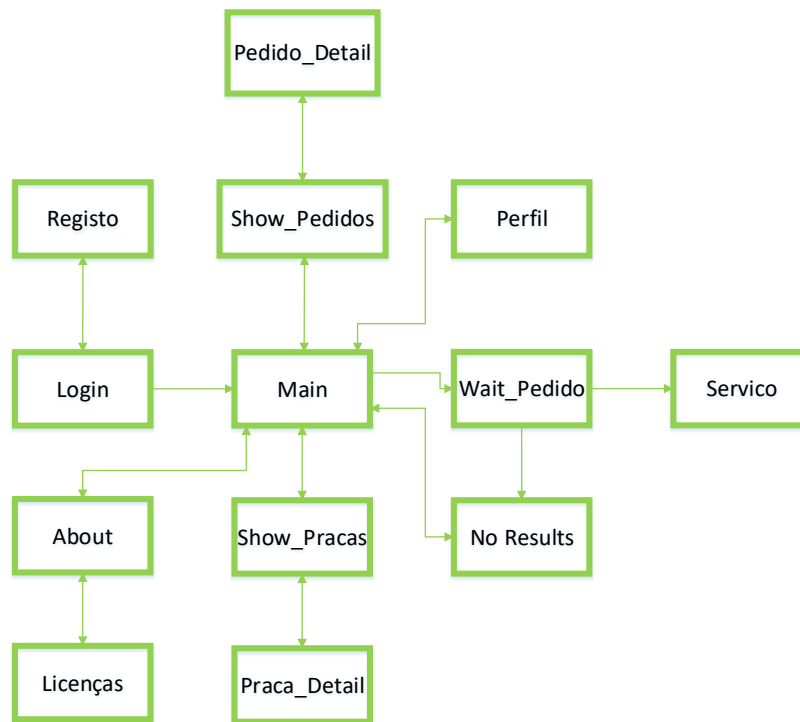


Figura 26 – Arquitetura da Aplicação do Cliente

Quando a aplicação é instalada e iniciada, é mostrado o ecrã de *login*, apresentado na Figura 27.

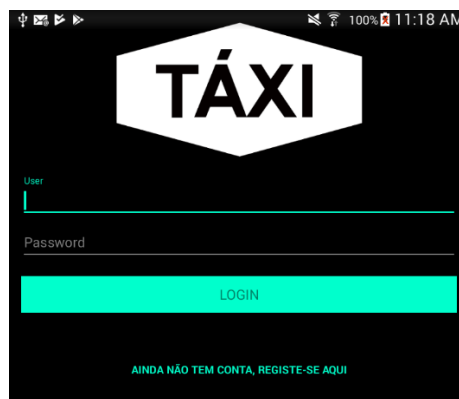


Figura 27 – Login Activity do Cliente

O cliente se ainda não estiver registado na aplicação, pode realizá-lo em que é apresentada a *activity* presente na Figura 28.

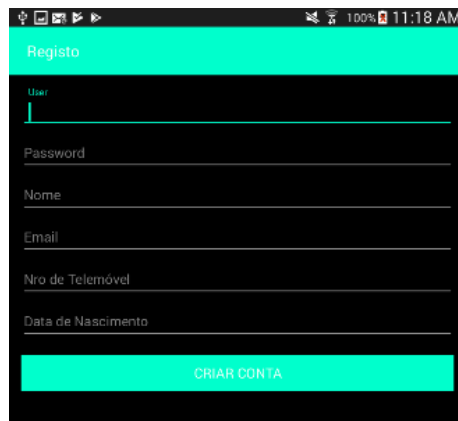


Figura 28 – Registo Activity

Ao preencher os campos do formulário de registo e submetê-lo, será enviado um pedido para o *Web Service* de registo, que irá validar os dados inseridos e se for um sucesso a aplicação é direcionada para a *activity* de *login* (Figura 28). Nesta *activity*, ao efetuar o *login* com sucesso a aplicação é redirecionada para a *activity* principal da aplicação, que é apresentada na Figura 29.

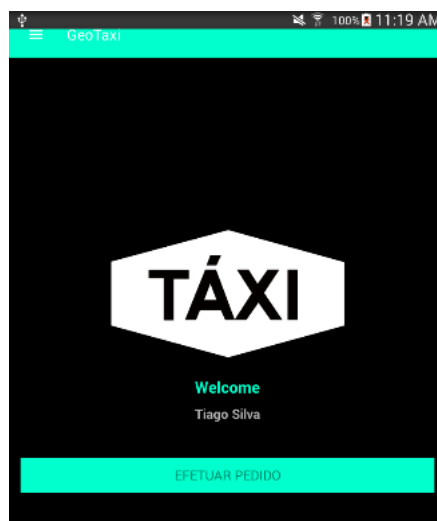


Figura 29 – Main Activity do Cliente

Nesta *activity*, o objetivo principal do cliente é realizar um pedido. O cliente ao carregar no botão “Efetuar Pedido”, será aberta uma janela para este escolher o tipo de táxi que pretende (4, 6 ou 8 lugares). Ao escolher o táxi pretendido, o pedido será enviado para a plataforma de gestão, para o *Web Service* pedido. Este pode retornar que o pedido foi entregue a um taxista e a aplicação esta será reencaminhada para uma *activity* de espera da aceitação do taxista (*Wait_Pedido Activity*). Por outro lado, ao retornar que não existem táxis disponíveis a informação será mostrada ao cliente numa nova *activity*. Para retornar à *main activity*, o

cliente necessita de carregar no botão de voltar (existente no canto superior esquerdo da página).

Quando um taxista aceitar o pedido a aplicação receberá uma *push notification* do servidor e será reencaminhada para a *activity* de serviço, em que é apresentada a informação sobre o serviço. Quando o taxista informar a plataforma do fim do serviço, esta enviará uma nova *push notification* para o cliente informando-o da nova situação. Ao recebe-la será mostrada na aplicação uma janela para classificar a qualidade do serviço finalizado. Quando submetida a avaliação de qualidade a aplicação será encaminhada para a *main activity* da aplicação (Figura 29).

Se porventura aquando da espera da resposta do pedido, o taxista rejeitá-lo e não houver mais táxis disponíveis, a plataforma enviará uma *push notification* para o cliente e a aplicação será encaminhada para a página de informação que o pedido não obteve resposta (*No Results Activity*).

Se o cliente estiver bloqueado pelo gestor da plataforma, na realização do pedido a plataforma retornará essa informação à aplicação e esta será reencaminhada para a *activity* de *login*.

No momento em que o cliente está com um serviço atribuído, logo na Serviço *Activity*, este serviço pode ser transferido por parte do taxista para outro táxi. Quando isto acontecer, dependendo do táxi para o qual foi transferido o serviço, visualiza-se uma janela com a informação da transferência e as características do novo táxi.

Quando o serviço for transferido para um táxi da plataforma, a informação apresentada ao cliente tem as mesmas características de um serviço normal. Por outro lado, quando o táxi para o qual foi transferido o serviço não aderiu à plataforma, ou então não tenha, no momento da transferência, o *login* efetuado é exposta ao cliente apenas a matrícula e o número de porta do táxi que irá chegar.

Se no momento em que o cliente efetua o pedido, a plataforma verificar que o utilizador está fora das cidades abrangidas, essa informação será retornada à aplicação e será aberta uma das janelas apresentadas na Figura 30. A janela da direita aparecerá ao cliente se a plataforma apenas estiver presente numa cidade, com a informação da cidade onde irá ser realizado o pedido caso o utilizador quiser confirmar o pedido.

Por outro lado, se a plataforma estiver em mais do que uma cidade será apresentada a janela da esquerda em que terá um *dropdown* com todas as cidades existentes na plataforma. Ao seleccionar uma cidade, o pedido será realizado com os táxis da cidade escolhida. Se escolher a opção “INDIFERENTE” serão procurados os táxis existentes em todas as cidades abrangidas pela plataforma, sendo escolhido o que se encontra mais perto do cliente.

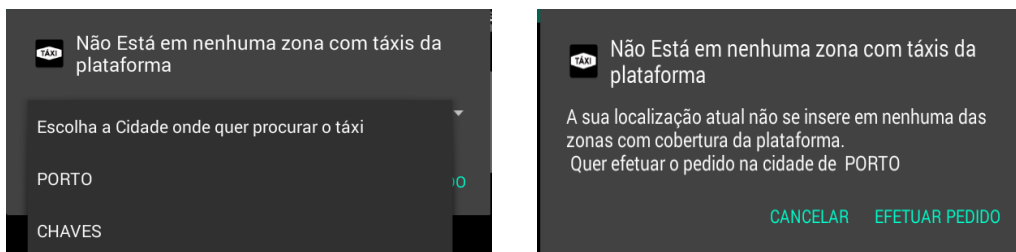


Figura 30 - Janela de confirmação do pedido

Na *main activity* (Figura 29) ao carregar-se no botão existente no canto superior esquerdo, ou então deslizando o dedo no ecrã, aparecerá um menu lateral que esta representado na Figura 31.

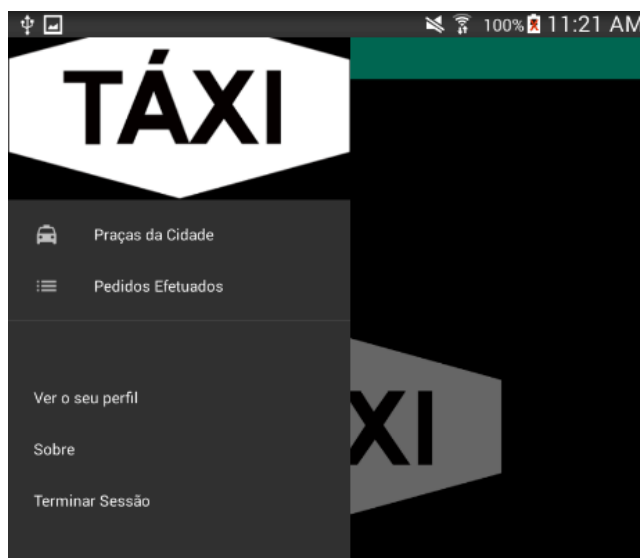


Figura 31 – Menu lateral da *main activity*

Neste menu é possível a visualização das praças da cidade em que o cliente se encontra, consultar os pedidos efetuados num determinado dia, ver o seu perfil e consultar a informação sobre a aplicação. Também é exequível realizar neste menu o termino da sessão, em que ao executá-lo a aplicação será reencaminhada para a *activity* de *login*.

Para o cliente consultar as praças da cidade é necessário no menu lateral da *main activity* carregar-se em “Ver Praças”. Ao realizá-lo a aplicação é reencaminhada para a *activity* de visualização das praças existentes na cidade, ou então de todas as praças da plataforma se o cliente não se encontrar em nenhuma cidade controlada por esta. Ao ser carregada numa praça da lista apresentada, a aplicação é direcionada para a *activity* de visualização das características da praça selecionada, em que mostra ao cliente a rua onde se encontra a praça e apresenta também um mapa com a localização desta. Para se regressar à *activity* anterior, é necessário carregar no botão no canto superior esquerdo.

O cliente poderá visualizar os pedidos realizados num determinado dia carregando no menu lateral da *main activity* em “Ver Pedidos”. Ao executá-lo a aplicação é reencaminhada para a *activity* de visualização dos pedidos efetuados no corrente dia, em que, se ainda não tiver realizado nenhum pedido é apresentada uma mensagem com esta informação. Por outro lado, se já tiver realizado pedidos é apresentada uma lista com estes e ao carregar num pedido a aplicação é encaminhada para a *activity* de mostrar os detalhes do pedido selecionado. Para se regressar à *activity* anterior, é necessário carregar no botão voltar no canto superior esquerdo.

Na consulta dos pedidos realizados, é possível a alteração do dia pretendido, para isso tem-se um botão no canto superior direito, em que quando pressionado é apresentado um calendário em que se pode realizar a alteração para o dia pretendido.

No menu lateral da *main activity* ao carregar-se em “Ver o seu Perfil” a aplicação será encaminhada para a *activity* apresentada na Figura 32.

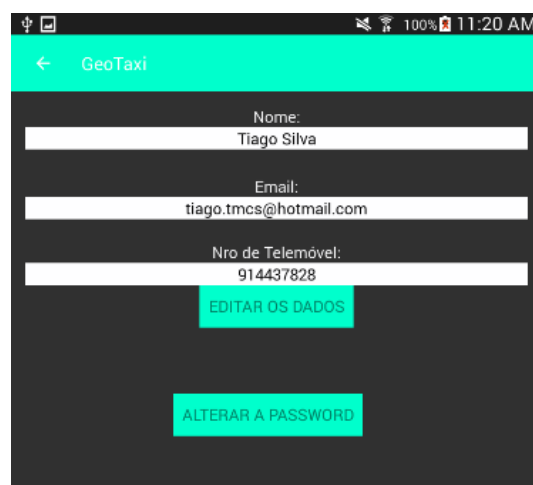


Figura 32 - Perfil Activity

Nesta *activity*, o cliente pode consultar os dados do seu perfil e se porventura necessitar de alterá-los, terá de carregar no botão “Alterar Dados”. Os campos passam a estar disponíveis para edição. Ao validar a edição é verificado antes de enviar os dados para a plataforma de gestão se algum dos campos foi alterado. Ao verificar-se a edição de alguns dos campos será enviado para o *Web Service* de alteração de dados apenas os campos que tiverem sido alterados.

Se o utilizador precisar de alterar a sua *password* terá de carregar no botão “Alterar a *Password*” e ao realiza-lo apareceram os campos necessários preencher para realizar a alteração (Figura 33).

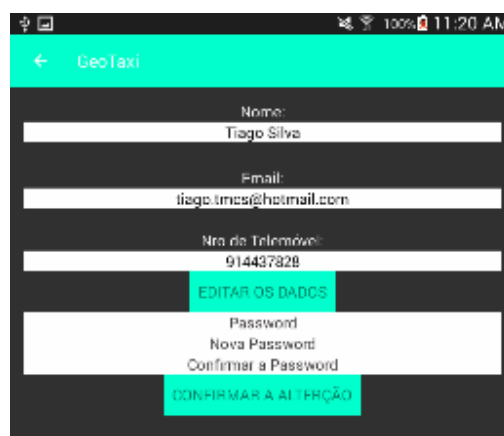


Figura 33 - Campos de Alteração da *password*

Ao ser submetido o formulário, antes de ser enviado para a plataforma de gestão é verificado se alguns todos os campos foram preenchidos e se a nova *password* esta coincide nos dois campos.

Na Figura 34 estão representadas as *activities* Sobre e Licenças, que terão o objetivo de mostrar ao cliente uma pequena descrição sobre a aplicação desenvolvida. É apresentado um botão para mostras as licenças de código aberto usadas no desenvolvimento da aplicação. Esta *activity* Sobre é acedida na parte superior esquerda, através do menu lateral da *main activity*.

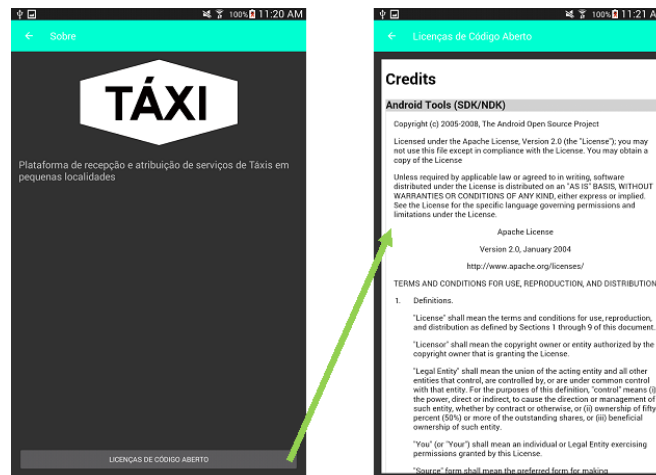


Figura 34 - Visualização da informação da aplicação (*About Activity* e *Licencas Activity*)

5.4.1. MENSAGENS DA APLICAÇÃO

Nesta secção será explicada cada uma das mensagens que poderão aparecer na aplicação, seja por qualquer erro de utilização ou de comunicação com a plataforma de gestão. Na Figura 35 estão representadas as mensagens erro que serão mostradas ao cliente quando existe algum erro na comunicação entre a aplicação e plataforma de gestão.

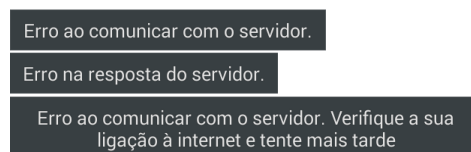


Figura 35 - Mensagens de erro ao comunicar com a plataforma

A primeira mensagem na figura será mostrada quando a aplicação não consegue comunicar com a plataforma, seja por esta estar em baixo ou por qualquer outro erro de comunicação. A segunda mensagem aparecerá quando a resposta vinda da plataforma, que tal como explicado anteriormente virá no formato JSON, trouxer algum erro. Por fim, a mensagem de baixo, da Figura 35, surgirá quando a aplicação tenta comunicar com a plataforma, mas esta está sem acesso à internet.

A Figura 36 apresenta as mensagens que poderão aparecer no momento da submissão do formulário de registo, presente na *activity* Registo (Figura 28).

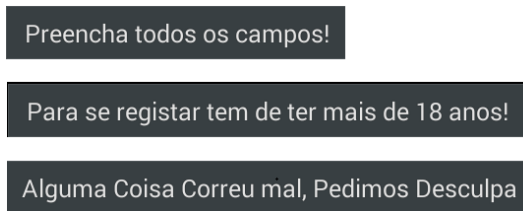


Figura 36 - Mensagens no momento do registo

A mensagem que se encontra mais acima da figura será exibida quando a submissão for efetuada, mas o utilizador ainda não preencheu todos os campos do formulário. A mensagem central aparecerá quando o utilizador que realizou o registo for de menor idade. Por fim, a última mensagem será apresentada quando a plataforma detetar algum erro nos campos do formulário, ou então alguns dos campos, como o email ou o número de telemóvel, já existirem na base de dados. Na Figura 37 estão exibidas as mensagens que poderão aparecer no momento do *login*.

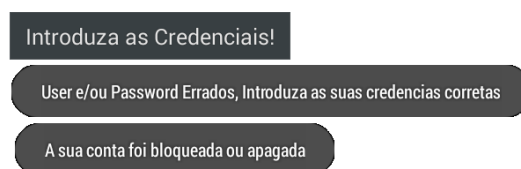


Figura 37 - Mensagens no momento do *login*

A primeira mensagem, mais acima na figura será apresentada quando o cliente efetuou o *login*, mas não preencheu todos os campos. A mensagem central será exibida quando a aplicação ao efetuar o *login* na plataforma verificar que os dados introduzidos estão errados. Finalmente, a mensagem mais abaixo na figura será apresentada quando o utilizador tentar fazer *login*, mas a plataforma retornou que a sua conta foi bloqueada (pelo gestor).

A partir da versão 5.0 do sistema operativo Android é necessário que o utilizador de uma aplicação dê acesso para que essa possa aceder aos recursos do Android, neste caso a localização do dispositivo. Por isto, na Figura 38 está representada a janela que aparecerá quando o utilizador entrar pela primeira vez na *main activity* da aplicação, em que se este recusar o acesso a aplicação será fechada e só poderá visualizar a *activity* quando a permissão for dada.

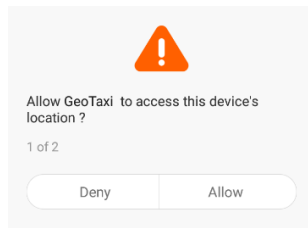


Figura 38 - Requisito de permissões

A Figura 39 apresenta todas as mensagens que poderão ser apresentadas quando o cliente efetua o pedido.

- 1 A obter a localização, o pedido não foi efetuado. Pedimos Despulpa pelo incomodo
- 2 Pedido Não Efetuado, tem de escolher o numero de pessoas
- 3 Aconteceu Alguma Erro durante o pedido!!!
- 4 A sua conta foi bloqueada, não pode efetuar mais pedidos
- 5 Cancelou o Pedido
- 6 A sua localização atual não se insere em nenhuma das zonas com cobertura da plataforma !!!

Figura 39 – Mensagens no momento do pedido

A mensagem número 1 será apresentada quando a aplicação não consegue aceder à localização do dispositivo, logo o pedido não será enviado para a plataforma. A mensagem 2 aparecerá quando o utilizador submeter o pedido e não tenha escolhido o tipo de táxi que pretende. A terceira mensagem (3) será exibida quando o pedido ao ser enviado para a plataforma, esta encontre ou tenha algum erro e não consiga finalizar a atribuição do pedido. A quarta mensagem (5) será mostrada quando o cliente for bloqueado pelo gestor da plataforma. A mensagem número 5 aparecerá quando o cliente ao escolher o tipo de táxi cancelar o pedido. Por último, a mensagem 6 será apresentada quando a localização do utilizador não se encontra em nenhuma das cidades cobertas pela plataforma.

Na Figura 40 estão representadas as mensagens que poderão aparecer quando o utilizador efetua a alteração dos seus dados pessoais.

- 1 Alteração Efetuada com Sucesso
- 2 O numero de telemovel introduzido já existe
- 3 O email introduzido já existe
- 4 O email e o numero de telemovel introduzidos já existem
- 5 Ocorreu alguma erro na edição
- 6 Tem de alterar alguma dos campos

Figura 40 – Mensagens na edição dos dados pessoais

A mensagem 1 é apresentada quando a edição é realizada com sucesso. A segunda (2) aparecerá quando o utilizador ao editar o seu número de telemóvel, a plataforma já tenha um cliente com o mesmo número que ele introduziu. A terceira mensagem (3) da figura será exibida quando o utilizador ao alterar o seu *email*, a plataforma já tenha um cliente com o mesmo email inserido. A mensagem número 4 é mostrada quando o cliente ao editar o número de telemóvel e o *email* a plataforma já tenha estes dois campos introduzidos em outros clientes. A mensagem 5 será apresentada quando após o utilizador submeter os dados, a plataforma detete algum erro nestes. Por fim, a última (6) mensagem é exibida caso o utilizador submeta a edição de dados para a plataforma, mas não alterou nenhum dos campos. Em todo o caso, é sempre possível regressar ao menu anterior sem alterar os dados pessoais. A Figura 41 apresenta todas as mensagens existentes na alteração da *password*.

- 1 Alteração Efetuada com Sucesso
- 2 Erro a password antiga não está correta
- 3 A nova password tem de ser igual em ambos os campos
- 4 Preencha todos os campos!
- 5 Ocorreu alguma erro na edição

Figura 41 – Mensagens na edição da *password*

A mensagem número 1 é exibida quando a edição foi completada com êxito. A segunda (2) é mostrada quando a plataforma verifica que a *password* antiga não está correta. A mensagem central da figura (3) é exposta quando a nova *password* e a sua confirmação não são iguais. A mensagem 4 é apresentada quando o utilizador submete a edição de dados e não preencheu todos os campos. Por fim, a última mensagem (5) é apresentada quando a edição foi submetida, mas a plataforma encontrou algum erro nos dados introduzidos.

Na Figura 42 estão representadas duas janelas que serão mostradas ao cliente se este não se encontrar com uma conexão à internet (janela de cima na figura) ou então quando a aplicação não consegue aceder à localização do dispositivo (janela mais abaixo na figura).

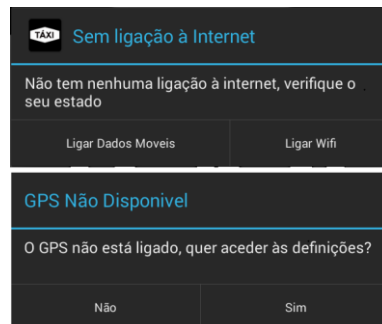


Figura 42 – Janelas quando o cliente esta sem internet ou GPS.

5.4.2. BASE DE DADOS DA APLICAÇÃO

Tal como descrito na apresentação do Android, este tem uma base de dados interna para cada aplicação, para ser possível guardar um maior conjunto de informações necessárias à aplicação.

Na Figura 43, pode-se observar as tabelas criadas nesta aplicação, que irão servir para guardar informações sobre a sessão do utilizador e também sobre o serviço quando este estiver a decorrer.

user	
PK	id
	nome
	cliente_id
	cliente_lat
	cliente_lon
	login_at

serviço	
PK	id
	taxista_nome
	taxi_ rua
	cliente_ rua
	taxi_ tempo
	taxi_ matricula
	taxi_ lat
	taxi_ lon
	transfer

Figura 43 – Base Dados da Aplicação do Cliente

Na tabela *user* tem-se as características do cliente, onde se encontra o seu nome o seu identificador (*id*), as suas coordenadas, a última vez que realizou um pedido e a data e hora do último *login*.

Por outro lado, a tabela *serviço* contém o nome do taxista, a rua onde se encontra o táxi, a rua onde se encontra o cliente, o tempo que o táxi demorará a chegar, a matrícula e as coordenadas do táxi, e por fim um campo para identificar se o serviço sofreu alguma transferência. Os dados desta tabela serão alterados sempre que for pedido à plataforma uma atualização da posição do táxi, ou então quando a plataforma enviar uma notificação com a transferência do serviço.

5.5. APLICAÇÃO DO TAXISTA

A aplicação desenvolvida tem como principal objetivo a receção e resposta de pedidos vindos da plataforma e se for aceite deve de o concluir ou transferir. Tem ainda outras funcionalidades como a troca do estado de disponibilidade, a consulta do seu calendário semanal de praças, os serviços efetuados num determinado dia e ainda a sua atividade num certo dia. Tem também a funcionalidade de visualização do perfil.

A Figura 44 apresenta as *activities* (funcionalidades) criadas para a aplicação do taxista, em que elas serão agora explicadas e como se relacionam umas com as outras.

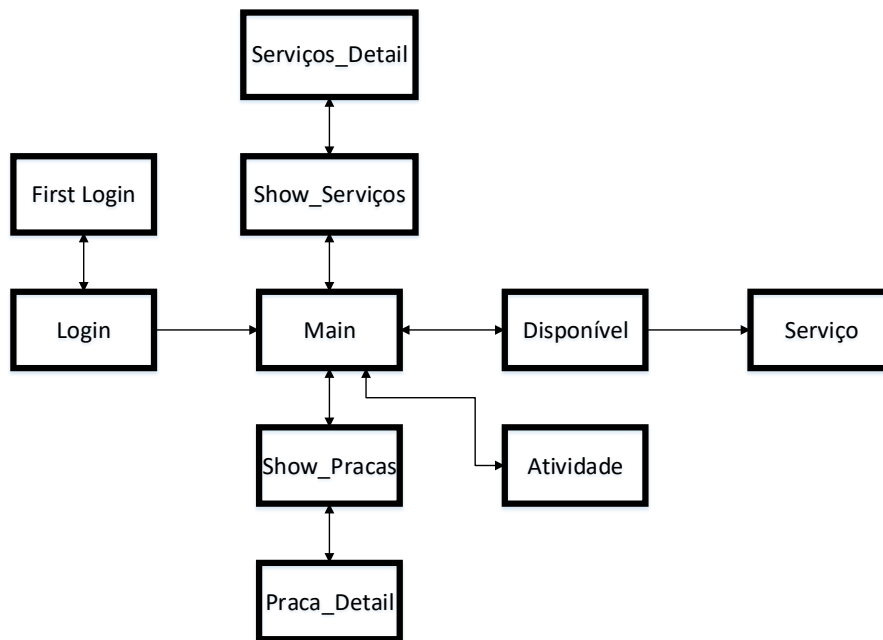


Figura 44 – Arquitetura da Aplicação do Taxista

O taxista ao instalar e iniciar a aplicação ser-lhe-á apresentada a *activity* de *login* (Figura 45).

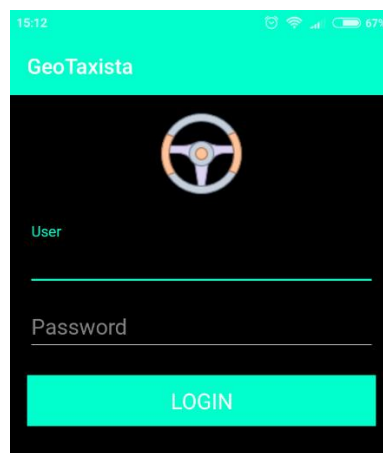


Figura 45 – Login Activity do Taxista

Nesta *activity* o utilizador deve introduzir as credencias de acesso dadas pelo gestor da plataforma. Ao serem validadas com sucesso, a plataforma irá verificar se é a primeira vez que o taxista efetua o *login*. Ao confirmar a plataforma retornará à aplicação que será encaminhada para a *activity* da Figura 46. Por outro lado, se não for a primeira vez que o taxista faz o *login* após o gestor o ter inserido na plataforma, a aplicação será reencaminhada para a *activity* principal, que está apresentada na Figura 47.

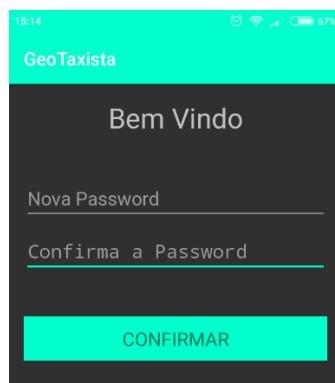


Figura 46 – First Login Activiy

Esta *activity* existe, pois, o gestor da plataforma registou o taxista com uma *password* e para esta ser uma *password* pessoal, o utilizador nesta *activity* necessita de introduzir por duas vezes uma nova *password*. Ao submeter o formulário, a aplicação irá verificar se os campos introduzidos coincidem. Ao verificar-se é enviado o pedido para a plataforma que irá realizar a edição na base de dados, e ao retornar o sucesso na edição, a aplicação será reencaminhada para a *activity* de *login* (Figura 45). Ao realizar o *login* com as novas credenciais a aplicação é encaminhada para a *activity* principal, que se pode observar na Figura 47.

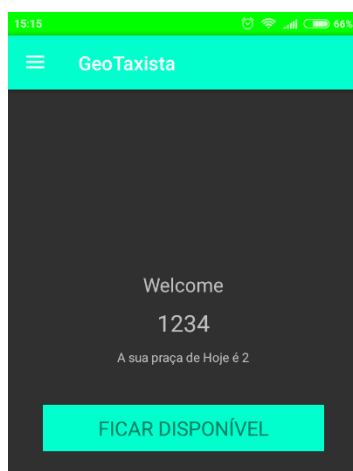


Figura 47 - Main Activity do Taxista

O utilizador nesta *activity* pode realizar diversas consultas e ainda editar os seus dados pessoais, mas o seu objetivo principal é ficar disponível para poder receber pedidos e como consequência realizar serviços. Para isto acontecer é necessário que o taxista pressione o botão central da janela (“Ficar Disponível”), ao realiza-lo a aplicação é encaminhada para a *activity* em que o utilizador está disponível, que pode ser observada na Figura 48.

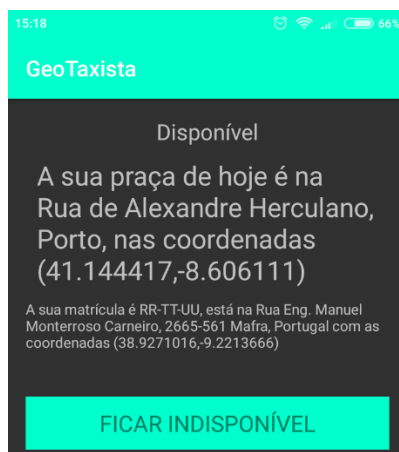


Figura 48 – Disponível Activity

Sempre que o utilizador for iniciar esta *activity*, será realizado um pedido à plataforma para ser retornado e apresentado ao taxista, a localização atual do táxi e a praça em que ele se encontra no corrente dia. Estes dados irão sendo atualizados regularmente.

Quando a plataforma enviar um pedido para o taxista, este é informado do mesmo e apresentadas as características (rua onde se encontra o cliente, a que distância esta de si e quanto irá demorar a chegar). O pedido pode ser rejeitado, em que é realizada a troca de disponibilidade para “Não Disponível” na plataforma e na aplicação, sendo esta reencaminha para a *activity* de *login* (Figura 45).

Por outro lado, se o pedido que chegou da plataforma for aceite pelo utilizador a aplicação enviará a aceitação e será reencaminhada para a *activity* de serviço. Nesta terá todas as características do mesmo e ainda a possibilidade de telefonar para o cliente que realizou o pedido e ver a localização do mesmo numa aplicação de navegação GPS instalada no dispositivo. Para estas duas funcionalidades é necessário o taxista pressionar um dos botões centrais desta *activity* (esquerda para telefonar e o da direita para ser reencaminhado para um navegador GPS).

Quando o serviço estiver concluído, o taxista deverá de carregar no botão no canto inferior esquerdo da *activity*, em que ao realiza-lo será enviada a informação para a plataforma e a aplicação encaminhada para a *activity* de disponível (Figura 48).

Se o taxista estiver com um serviço, mas não o conseguir realizar, é possível transferi-lo para outro taxista para isso é necessário pressionar o botão no canto inferior direito do ecrã, em que aparecerá uma janela para confirmar a transferência, ao realiza-lo aparecerá uma nova

janela para introduzir o número de porta do táxi para o qual quer transferir o serviço, ao submeter a transferência, será enviado um pedido para o *Web Service* de transferência de serviço, que irá seguir os passos apresentados na Figura 17. Se a transferência tiver sido realizada com sucesso, a aplicação será redirecionada para a *main activity* (Figura 47).

No momento da transferência do serviço se a plataforma verificar que o táxi com o número de porta introduzido tiver maior capacidade do que o cliente especificou no pedido, a plataforma enviar a informação ao taxista que terá de confirmar para terminar a transferência. A janela de confirmação é apresentada na Figura 49.



Figura 49 – Janela de confirmação da transferência

Se porventura aquando da chegada do pedido à aplicação do taxista, este demorar mais do que o intervalo de tempo de aceitação definido pela plataforma, esta enviará uma notificação para aplicação que apresentará a janela exibida na Figura 50.



Figura 50 – Tempo de Espera Superado (Disponível Activity)

Passado um pequeno intervalo de tempo a aplicação será reencaminhada para a *main activity* (Figura 47).

Enquanto o taxista está com um serviço a decorrer, este pode efetuar uma queixa sobre um cliente, por um qualquer motivo, ao gestor da plataforma da sua cidade, para isso necessita de carregar no botão que se encontra no canto superior direito da serviço *activity*.

Quando a queixa for submetida, será enviada para a plataforma, para o *Web Service de report*, que irá validá-lo e inserir na tabela *report* da base de dados. Esta a partir deste momento fica disponível para o gestor consultar e atuar em conformidade.

Quando o utilizador não estiver com nenhum serviço a decorrer, é possível que chegue uma transferência de serviço por parte de outro taxista para si. Ao isto acontecer, ou seja, a plataforma informar a aplicação que tem uma transferência, esta será redirecionada para *activity* de serviço, mas com algumas alterações, em que é informado que está com uma transferência e sem o botão de realizar uma nova transferência, pois o serviço só pode ser transferido de táxi para outro uma única vez.

Na *main activity* (Figura 47) ao carregar-se no botão existente no canto superior esquerdo, aparecerá um menu lateral que esta representado na Figura 51.

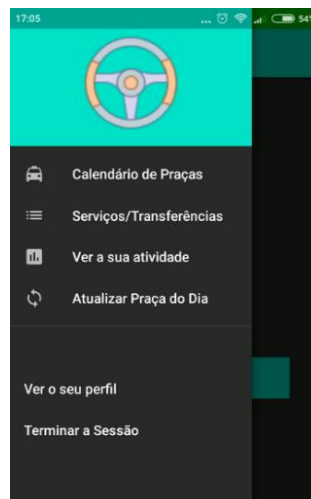


Figura 51 - Menu Lateral da *Main Activity*

Este menu tem a função de permitir o taxista ver o seu calendário de praças, visualizar os seus serviços realizados, ver a sua atividade diária, atualizar a praça do corrente dia, ver o seu perfil e por fim terminar a sessão.

Ao selecionar a última opção, “Terminar Sessão”, a aplicação será redirecionada para a *activity* de *login* (Figura 45).

Quando o utilizador selecionar a quarta opção, “Atualizar Praça do Dia”, será enviado um pedido à plataforma para que esta atualize a praça do dia do táxi, quando isto terminar será retornada à aplicação a praça do corrente dia e atualizada a informação a mostrar ao utilizador.

Para o taxista visualizar o seu calendário semanal de praças é necessário que no menu lateral da *main activity* se selecione a opção “Calendário de Praças”. Ao realizá-lo a aplicação é encaminhada para *activity* de visualização do calendário semanal de praças, em que tem uma lista com os diferentes dias da semana e a praça associada ao dia. Ao selecionar um dos dias, a aplicação é direcionada para a *activity* que tem a informação sobre a praça e um mapa com a sua localização. Para se regressar à *activity* anterior, é necessário carregar no botão no canto superior esquerdo.

Quando o utilizador pretender observar os serviços realizados num determinado dia, é necessário no menu lateral da *main activity* selecionar a opção “Serviços/Transferências”. Ao ser executada a aplicação é encaminhada para a *activity* de visualização dos serviços ou transferências realizadas no corrente dia, em que é apresentada uma lista com os diversos serviços e transferências realizadas, em que as transferências são assinaladas a vermelho, pois não pode visualizar os detalhes dela. Quando ainda não existirem serviços realizados no dia é apresentada uma mensagem com essa informação. Ao se selecionar um serviço realizado a aplicação é reencaminhada para a *activity* de detalhes do serviço, em que são apresentados todos os detalhes do serviço selecionado. Para se regressar à *activity* anterior, é necessário carregar no botão voltar existente no canto superior esquerdo.

Na visualização dos serviços e transferências, é possível alterar o dia em que se pretende visualizar os serviços realizados, para isso carrega-se no botão do canto superior direito e aparecerá um calendário para escolher o dia pretendido. Nesta *activity*, quando se selecionar uma transferência é mostrada uma mensagem a avisar que o serviço foi transferido e que não pode visualizar os detalhes do mesmo.

Na visualização dos detalhes do serviço é possível efetuar uma queixa do cliente para quem foi realizado o serviço, para isso é necessário pressionar o botão existente no canto superior direito, em que ao realiza-lo será mostrada uma janela para realizar a queixa. Ao submeter o formulário, esta fica visível para o gestor da plataforma poder tomar uma decisão sobre o cliente.

Quando o utilizador pretender ver a sua atividade diária é necessário no menu lateral da *main activity* selecionar-se a opção “Ver a sua Atividade”, ao realiza-lo a aplicação é redirecionada para a *activity* de visualização do tempo de disponibilidade no corrente dia, em que pode aparecer três caixas de texto com cores diferente com o tempo que o taxista este em cada

uma das disponibilidades existentes na plataforma, ou então, apenas um texto se ainda não houve mudança de disponibilidade.

Aqui tal como na visualização dos serviço e transferências, é possível alterar o dia em que se quer ver a atividade, para isso carrega-se no botão do canto superior direito e aparecerá um calendário para escolher o dia pretendido.



Figura 52 - Perfil Activity do Taxista

Ao carregar-se em “Ver o seu Perfil” no menu lateral da *main activity* a aplicação será direcionada para a *activity* exibida na Figura 52, em que o taxista pode ver os dados do seu perfil, e se porventura precisar de modifica-los terá de pressionar o botão “Alterar Dados”, em que os campos passam a estar disponíveis para a edição. Se o utilizador necessitar de alterar a sua *password* terá de carregar no botão “Alterar a *Password*” e o realiza-lo aparecerão os campos necessários de preencher para realizar a alteração, que serão introduzir a *password* antiga e por duas vezes uma nova.

5.5.1. MENSAGENS DA APLICAÇÃO

Serão explicadas nesta secção cada uma das mensagens que poderão aparecer na aplicação, seja por qualquer erro desta ou na comunicação com a plataforma de gestão, ou então uma outra informação.

As mensagens apresentadas na Figura 53, de erro na comunicação com a plataforma de gestão, são idênticas às apresentadas na aplicação do cliente.

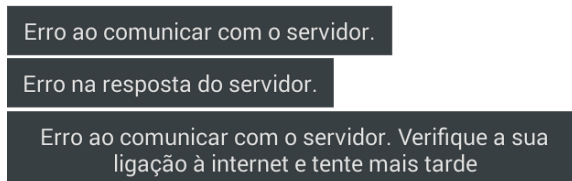


Figura 53 – Mensagens de erro ao comunicar com a plataforma

A mensagem mais acima na figura aparecerá quando a resposta vinda da plataforma, que tal como explicado anteriormente virá no formato JSON, vier com algum erro. A mensagem do centro será exibida quando a aplicação não consegue comunicar com a plataforma, seja por esta estar em baixo ou por qualquer outro erro na comunicação. Por fim, a mensagem em baixo na figura aparecerá quando a aplicação tentar comunicar com a plataforma, mas está sem um acesso à internet.

A Figura 54 apresenta as mensagens de erro que podem aparecer no momento de submissão do formulário de *login*.

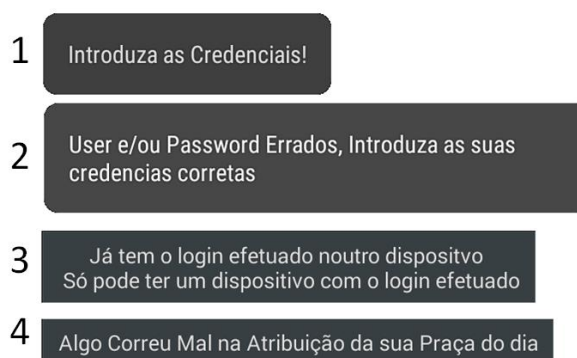


Figura 54 - Mensagens no *Login*

A mensagem 1 será apresentada quando o utilizador submeteu o formulário, mas não preencheu todos os campos. A segunda mensagem (2) será exibida quando a aplicação enviar o formulário para a plataforma e esta retornar que não encontrou os campos base de dados. A mensagem 3 é mostrada quando o taxista que está a realizar o *login* já tem uma sessão iniciada noutro dispositivo. Por fim, a mensagem 4 é exposta quando a plataforma encontrou algum erro na atualização da praça do dia do táxi que efetuou o *login*.

Estão representadas na Figura 55 as mensagens que poderão aparecer quando o utilizador altera a *password* após o primeiro *login*.

As duas passwords tem de ser iguais

Algo Correu Mal!!!

Figura 55 - Mensagens na alteração da *password* após o 1º login

A mensagem de cima é exibida quando as *passwords* não coincidem e a debaixo é apresentada quando a plataforma ou a aplicação têm algum erro no processamento do formulário.

Tal como explicado anteriormente, a partir da versão 5.0 do Android é necessário que o utilizador dê acesso para que a aplicação possa aceder aos recursos do Android, neste caso a localização do dispositivo e a possibilidade de gerir as chamadas, pois a aplicação pode ter a capacidade de iniciar uma chamada. Na Figura 56 está representada as janelas que aparecerão, uma a seguir à outra, quando o utilizador entrar pela primeira vez na *main activity*, em que se este rejeitar o acesso a aplicação será encerrada e só poderá visualizar a *activity* se a autorização for dada.

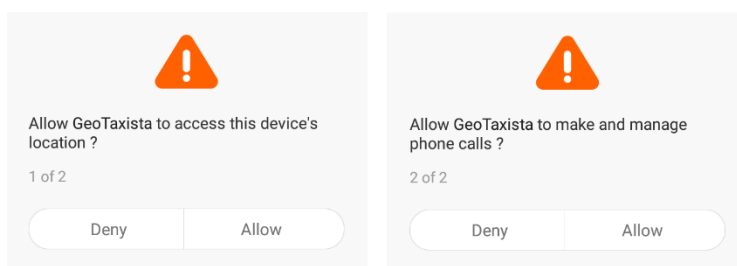


Figura 56 – Requisitos de Permissões da Aplicação

As mensagens que poderão aparecer no momento da efetuação de uma queixa estão representadas na Figura 57.

Reportar Problema sobre o cliente

Cancelou a Queixa

Report Efetuado

Figura 57 - Mensagem na efetuação da queixa ao cliente

A mensagem mais acima na figura é mostrada quando o utilizador carrega no botão para efetuar a queixa. A mensagem central da figura é exibida quando o taxista cancela a queixa

e a última mensagem é apresentada quando o utilizador envia para a plataforma a queixa e esta valida-a.

Na Figura 58 estão representadas as mensagens que poderão aparecer no momento em que o taxista efetua a transferência do serviço

- 1 Introduza o nro de porta, do táxi para o qual pretende transferir
- 2 Algum erro ao transferir o serviço
- 3 Transferência Não Efetuada
O nro de porta inserido, não existe!!!
- 4 Transferência Não Efetuada
O taxi para o qual transferiu o serviço não tem capacidade suficiente!!!
- 5 Transferência Não Efetuada
Não pode transferir o serviço para si mesmo!!!!
- 6 Cancelou a Transferência

Figura 58 – Mensagens no momento da transferência de serviço

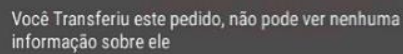
A mensagem 1 aparecerá quando o utilizador confirmar que quer realizar a transferência. A segunda mensagem (2) será exposta quando a aplicação ou a plataforma encontrarem ou tiverem algum erro com a transferência e o mesmo não foi transferido. As seguintes três mensagens são de erro na transferência, em que a mensagem 3 é apresentada quando o número de porta introduzido não existe na cidade, a mensagem 4 é exibida quando o táxi para o qual foi transferido o serviço tem capacidade inferior à requisitada pelo cliente, por ultimo, a mensagem 5 é mostrada quando o número de porta introduzido é o do taxista que esta a transferir o serviço. Por fim, a última mensagem (6) é apresentada quando em qualquer altura da transferência o utilizador cancelar a mesma.

Na Figura 59 está representada a mensagem exibida quando o utilizador envia o pedido à plataforma para atualizar a praça do dia e esta retorna à aplicação a praça atualizada.

Praça atualizada

Figura 59 - Praça atualizada com sucesso

A Figura 60 apresenta a mensagem exibida quando o taxista, na consulta dos serviços e transferências de um certo dia, carregar num serviço que ele transferiu.



Você Transferiu este pedido, não pode ver nenhuma informação sobre ele

Figura 60 - Mensagem de Visualização de uma transferência

Na Figura 61 estão patenteadas as mensagens que poderão surgir quando o utilizador efetua a alteração dos seus dados pessoais.

- 1 Tem de alterar alguma dos campos
- 2 Alteração Efetuada com Sucesso
- 3 O email introduzido já existe
- 4 O numero da licença introduzido já existe
- 5 O email e o numero da licença introduzidos já existem
- 6 Ocorreu alguma erro na edição

Figura 61 – Mensagens na edição dos dados pessoais do taxista

A mensagem 1 é exibida quando o utilizador submete a edição, mas não modificou nenhum dos campos. A segunda mensagem (2) é exposta quando a edição é executada com sucesso. A mensagem 3 aparecerá quando o utilizador ao alterar o seu email, a plataforma já possua um taxista com o email que ele inseriu. A mensagem 4 será mostrada quando o utilizador ao alterar o seu número de licença, a plataforma já detenha um taxista com o número de licença que foi inserido. A mensagem 5 é apresentada quando o cliente ao editar o número de licença e o email a plataforma já tenha estes dois campos introduzidos em outros taxistas. Por fim, a última mensagem (6) será mostrada quando o utilizador ao submeter a edição dos dados a plataforma haja algum erro nesta.

Estão apresentadas na Figura 62 todas as mensagens existentes na alteração da *password*

- 1 Alteração Efetuada com Sucesso
- 2 Preencha todos os campos!
- 3 Erro a password antiga não está correta
- 4 A nova password tem de ser igual em ambos os campos
- 5 Ocorreu alguma erro na edição

Figura 62 - Mensagens na alteração da *password*

A mensagem 1 é exibida quando a edição foi completada com êxito. A segunda mensagem (2) é mostrada quando o utilizador submete a edição e não preencheu todos os campos. A mensagem 3 é exposta quando a plataforma verifica que a *password* antiga não está correta. A mensagem 4 é apresentada quando a nova *password* e a sua confirmação não são iguais. Por fim, a última mensagem (5) é exposta quando a edição foi submetida, mas a plataforma encontrou ou tenha algum erro nesta.

Na Figura 63 está representada a janela que será mostrada ao utilizador quando este inicia a aplicação e esta verificar que o dispositivo está sem uma conexão à internet.

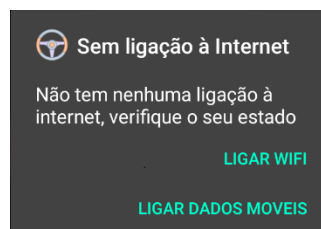


Figura 63 – Janela quando o utilizador está sem *internet*

5.5.2. BASE DE DADOS DA APLICAÇÃO

Tal como elucidado na apresentação do Android, e nesta mesma secção na aplicação do cliente, o SO tem uma base de dados interna para cada aplicação, para ser possível guardar um maior conjunto de informações necessárias à aplicação.

Na Figura 64 pode-se observar as tabelas criadas nesta aplicação, que irão servir para guardar informações sobre a sessão do utilizador e também sobre o pedido quando este chegar à aplicação.

user	
PK	id
	nome
	taxista_id
	taxi_id
	praca_id
	cidade
	login_at
	disponibilidade
	estado_id

pedido	
PK	id
	pedido_id
	taxi_distancia
	cliente_rua
	cliente_lat
	cliente_lon
	taxi_tempo

Figura 64 – Base Dados da Aplicação do Taxista

Na tabela *user* tem-se as características do taxista, em que se tem o seu nome o seu identificador (*id*), o identificador do táxi e praça, a cidade em que se encontra, a última vez que realizou o *login* (este campo verifica se é necessário atualizar a praça do dia), a disponibilidade atual do taxista e o estado em que se encontra o serviço.

Por outro lado, a tabela *pedido* contem o identificador do pedido, a distância a que o táxi se encontra do cliente, a rua onde se encontra o cliente, as suas coordenadas e o tempo que o utilizador demorará a chegar ao cliente.

6. TESTES E RESULTADOS

Neste capítulo apresentam-se os testes que foram realizados a quando do desenvolvimento da solução. Estes são testes sob as funcionalidades implementadas no sistema desenvolvido.

6.1. APLICAÇÃO DO GESTOR

Nesta secção apresentam-se os testes realizados após o desenvolvimento da aplicação do gestor. Os testes nesta aplicação foram realizados por uma pessoa, para ser simulado um caso real de implementação. Na Figura 65 apresenta-se o teste realizado para a inserção de uma praça.

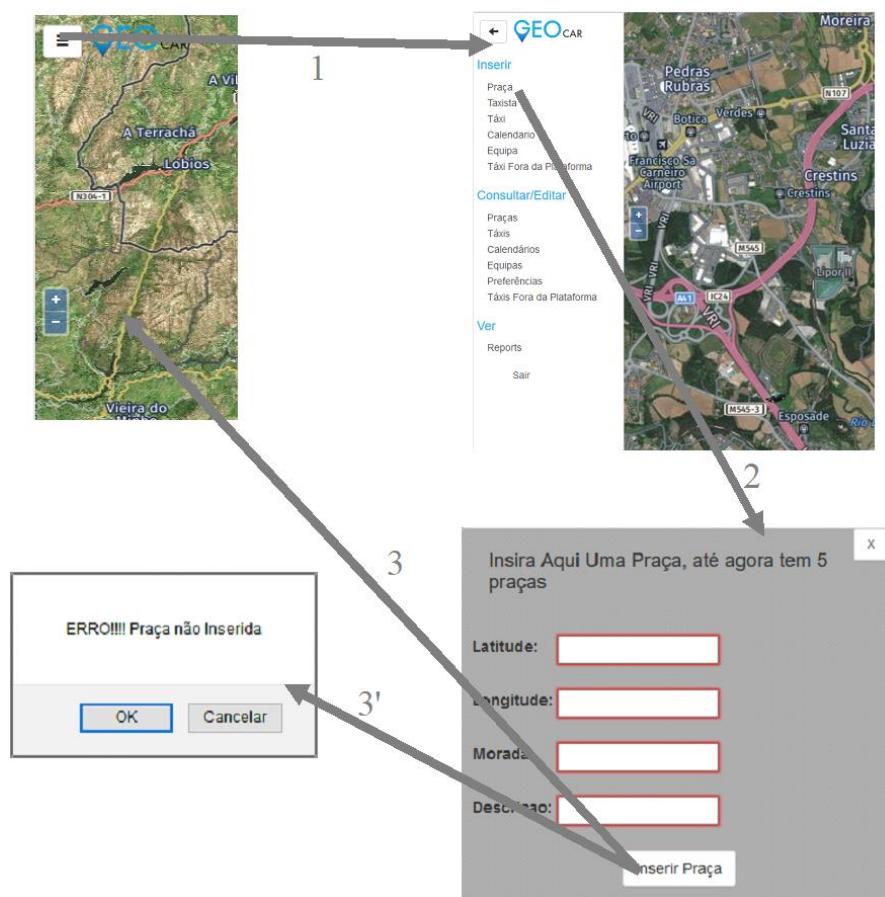


Figura 65 – Teste de inserção de uma praça

Para se realizar este ensaio é necessário aceder ao menu lateral da página principal carregar na opção de inserir praça, ao realizá-lo será apresentada uma janela com um formulário com os campos necessários para se efetuar a inserção. Ao submeter o formulário, no caso de sucesso da inserção a plataforma irá validar e redirecionar a aplicação de volta à página principal, ou então, apresentar uma mensagem de erro no caso da falha na validação.

Na Figura 66 apresenta-se os passos necessários para a realização da consulta e edição de uma praça anteriormente inserida.

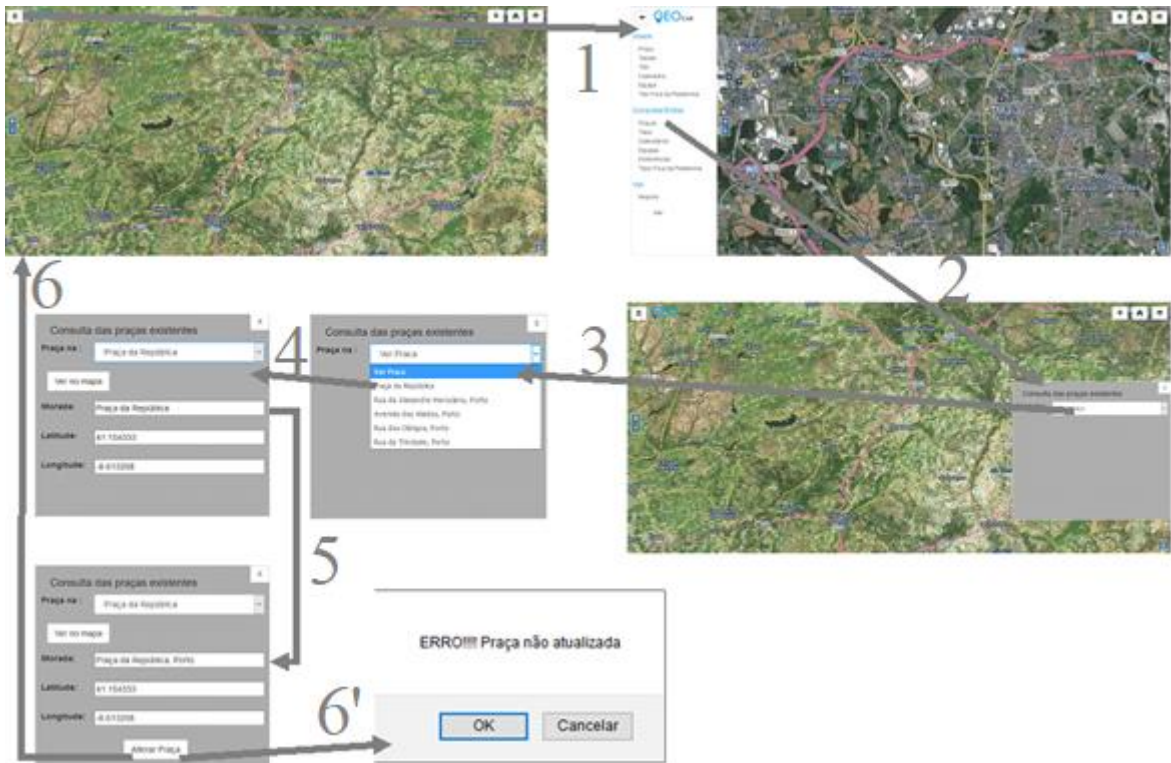


Figura 66 – Teste de consulta e edição de uma praça

Tal como no caso anterior, é necessário em primeiro lugar aceder ao menu lateral e aí carregar na opção de consultar/editar praça, em que irá aparecer uma janela com uma *dropdown* com as diferentes praças inseridas. Ao selecionar uma delas aparecerá as características da praça selecionada, em que ao ser editado algum campo, aparecerá um botão para confirmar a edição. Ao submeter a edição a plataforma validará os novos dados e redireciona a aplicação de volta à página principal no caso de sucesso da edição, ou apresenta uma mensagem de erro no caso de insucesso na validação.

A Figura 67 apresenta a verificação realizada, de visualização no mapa de todas as praças da cidade administrada pelo gestor.

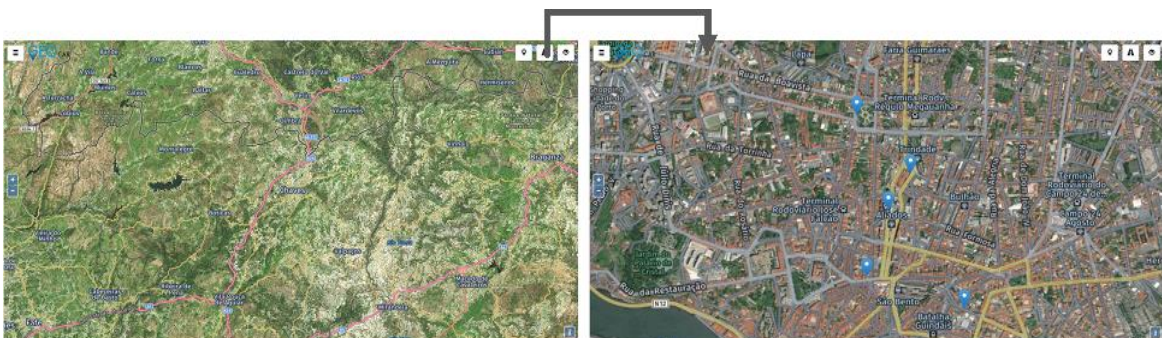


Figura 67 - Teste de visualização das praças da cidade

Para se realizar a verificação, tem que se carregar no botão do central do grupo de botões do canto superior direito da página principal. Ao realizá-lo será apresentado no mapa um marcador com a localização de cada uma das praças existentes na cidade. Ao carregar novamente neste botão, os marcadores colocados no mapa desaparecerão.

Na Figura 68 tem-se representado o ensaio realizado na consulta da localização atual dos táxis existentes na cidade.

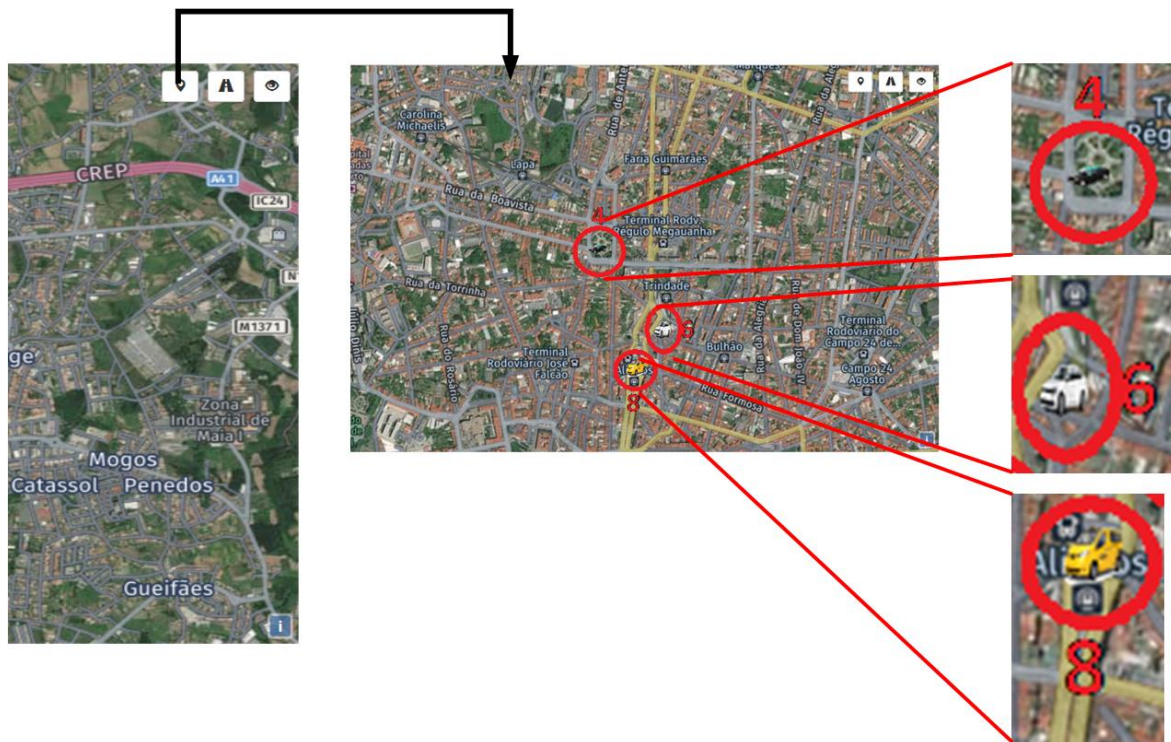


Figura 68 - Teste de visualização da localização dos táxis existentes na cidade

Para se realizar o ensaio, tem que se carregar no botão da esquerda do grupo de botões do canto superior direito da página principal. Ao ser pressionado será apresentado no mapa um marcador com a localização de cada um dos táxis existentes na cidade. Existem três tipos de marcadores diferentes, para se distinguir os tipos de táxis existentes (4, 6 ou 8 lugares).

6.2. APLICAÇÃO DO CLIENTE

Nesta secção apresentam-se os testes realizados após o desenvolvimento da aplicação do cliente. Estes testes foram realizados por todos os colaboradores da empresa, para se simular o teste do protótipo em ambiente real.

Na Figura 69 tem-se representada a verificação de realização de um pedido e consequente acompanhamento do mesmo. Esta verificação engloba tanto a aplicação do cliente como também a plataforma de gestão, na atribuição de serviços de táxis.

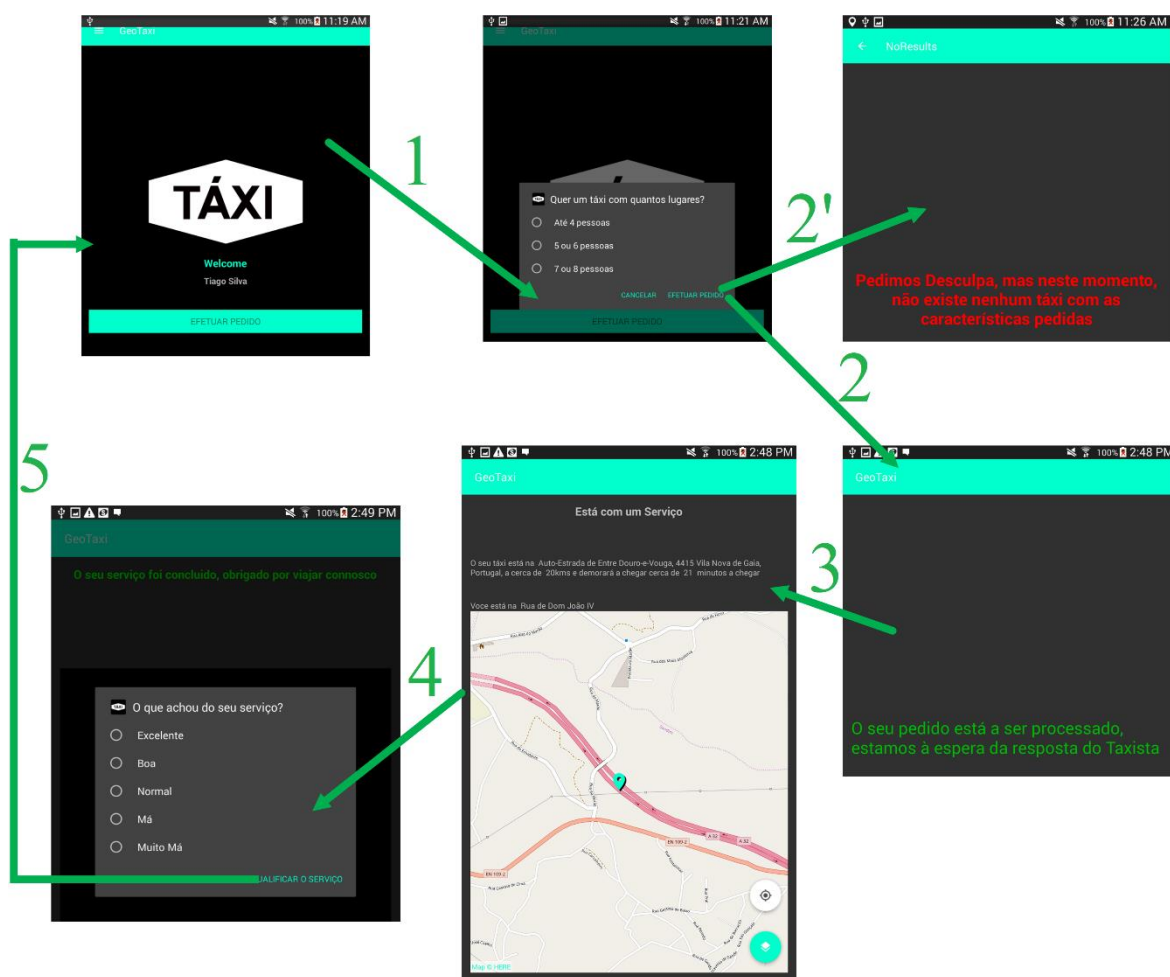


Figura 69 - Teste de realização e acompanhamento de um pedido

Para se realizar esta verificação, tem-se que na *activity* principal da aplicação pressionar o botão “Efetuar Pedido”. Ao pressioná-lo será aberta uma janela para este escolher o tipo de táxi que pretende (4, 6 ou 8 lugares). Quando escolhido o pedido será entregue à plataforma e a aplicação encaminha para uma janela de espera, enquanto a plataforma não informar que o taxista aceitou o pedido. Ao ser informado pela plataforma da aceitação do taxista, será encaminhado para o ecrã com as características do serviço (rua onde se encontra o táxi, distância e tempo até à sua chegada e a localização do táxi no mapa). Estas informações serão atualizadas regularmente. Quando o taxista terminar o serviço, será apresentada uma janela ao cliente para este escolher a qualidade do serviço prestado, após realiza-lo será redirecionado para a *activity* principal.

Foi testado igualmente a transferência de serviço em que pode ocorrer um de dois cenários, como se pode observar pela Figura 70.

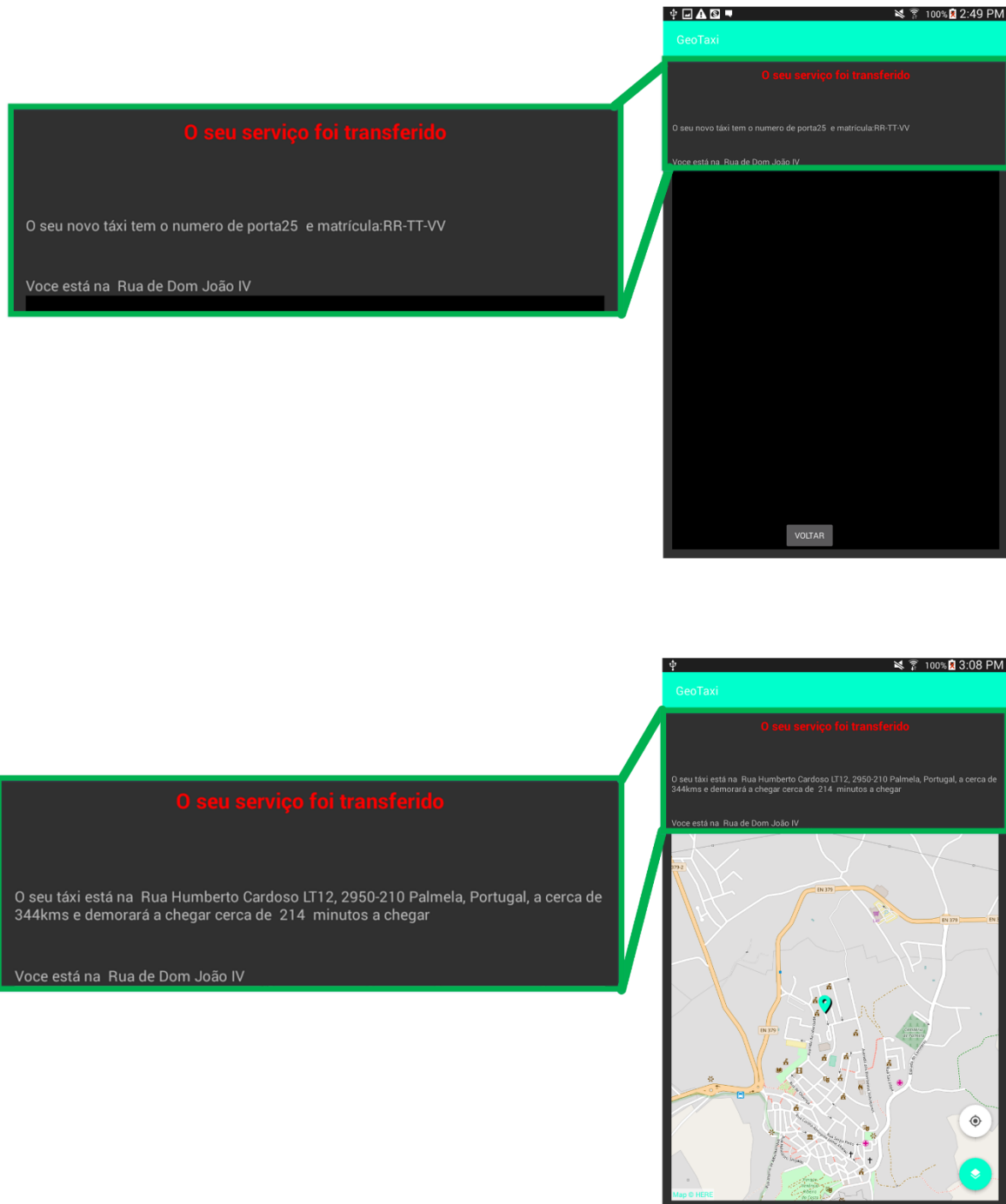


Figura 70 - Teste de chegada de uma transferência

A janela da esquerda é apresentada quando chegou uma transferência e o taxista não se encontra na plataforma, sendo apresentado ao cliente apenas o número de porta e a matrícula do táxi que irá chegar. A janela da direita é apresentada quando a transferência foi realizada para um táxi existente na plataforma, ficando com as mesmas características de um serviço

normal. Na Figura 71, apresenta-se o ensaio a quando da consulta das praças existentes na cidade.

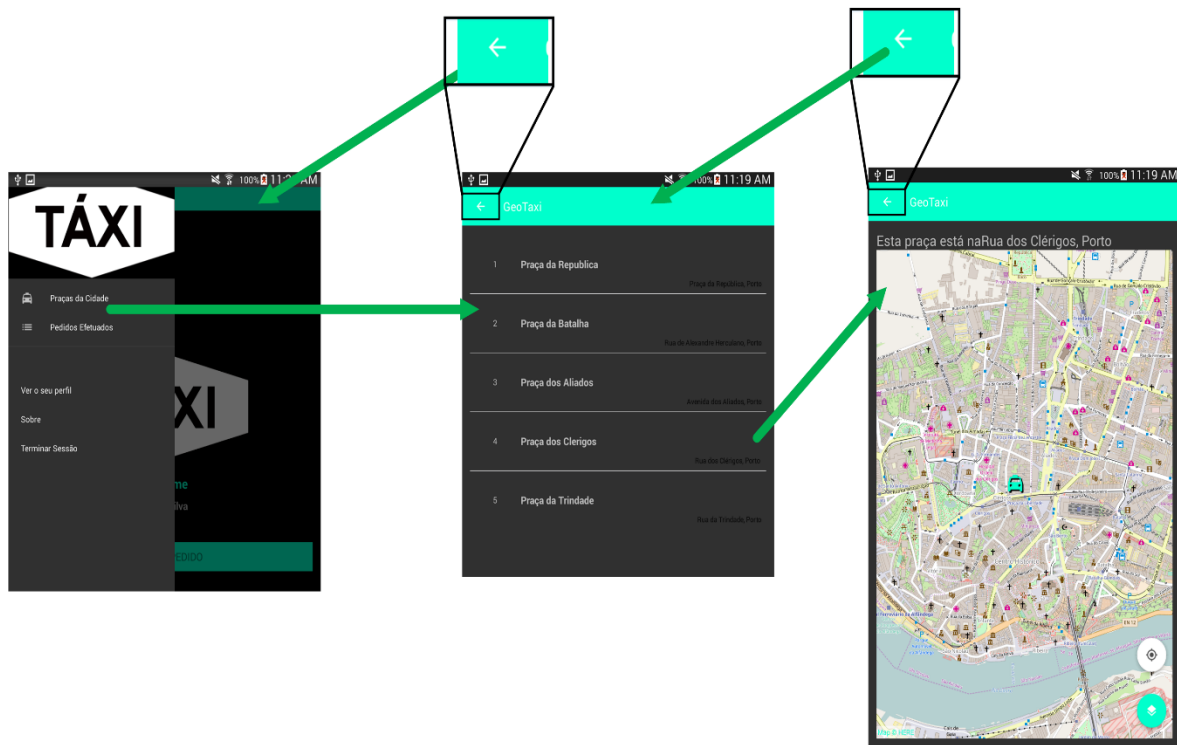


Figura 71 – Teste de consulta das praças da cidade

Para se realizar este ensaio, no menu lateral da *activity* principal tem que se carregar em “Praças da Cidade”. Ao realizá-lo é mostrado ao cliente uma lista com as praças existentes na cidade onde se encontra, ou então de todas as praças existentes na plataforma. Ao seleccionar uma das praças, é apresentado um mapa com a localização da praça e as suas características. Para se regressar à página anterior pressiona-se o botão existente no canto superior esquerdo do ecrã (←). Na Figura 72 apresenta-se o teste realizado para a visualização dos pedidos realizados num determinado dia.

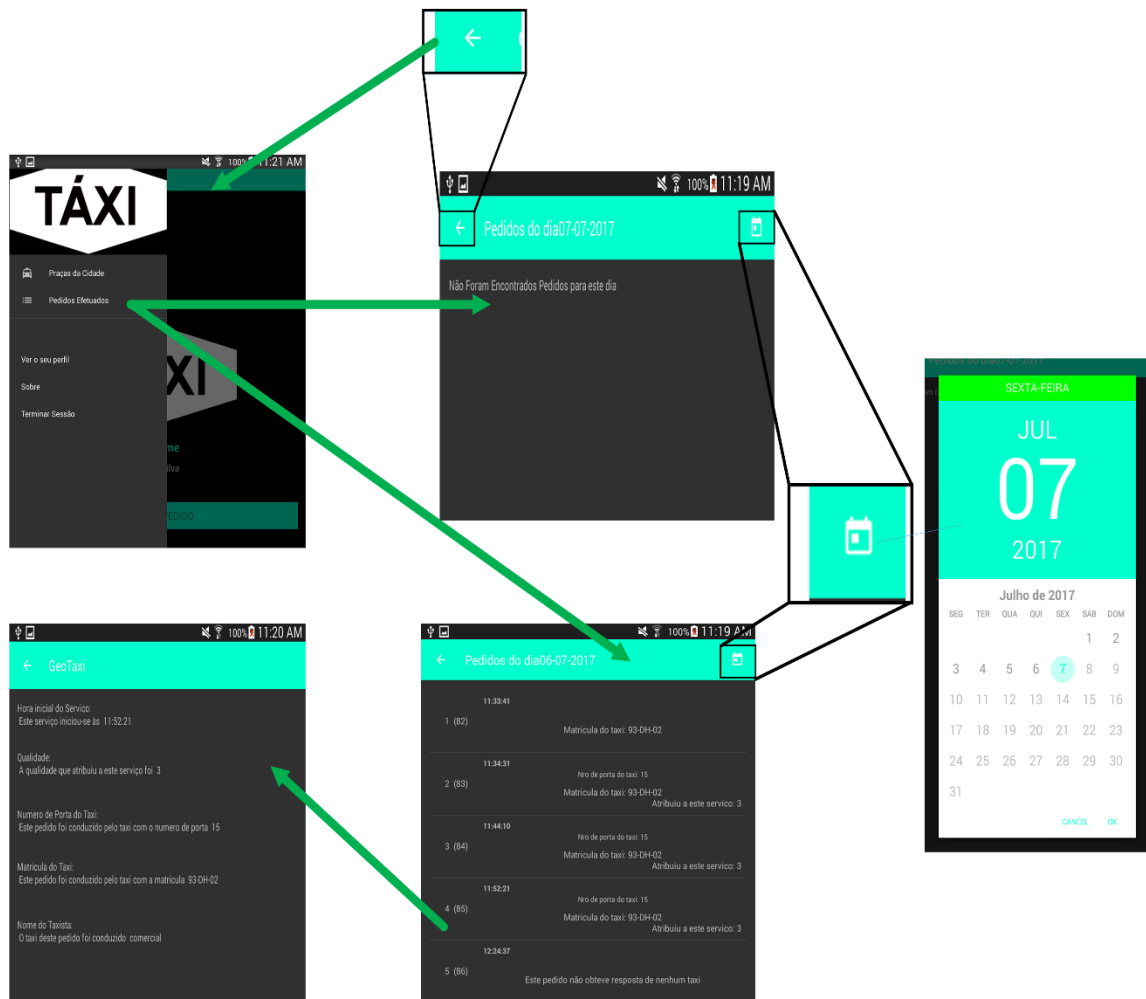


Figura 72 - Teste de consulta dos pedidos realizados num dia

Para que este teste seja realizado, tem-se que no menu lateral da *activity* principal pressionar em “Pedidos Realizados”, em que será apresentado ao cliente todos os serviços realizados no corrente dia, ou então uma mensagem a informar que ainda não tem pedidos realizados nesse dia. Ao se seleccionar um dos pedidos serão apresentadas as características do mesmo ao utilizador. Para se regressar à página anterior pressiona-se o botão existente no canto superior esquerdo do ecrã.

Foi ainda testado a alteração do dia da consulta, em que se tem de pressionar o botão existente no canto superior direito, ao realiza-lo é apresentado um calendário para escolher o dia pretendido.

6.3. APLICAÇÃO DO TAXISTA

Nesta secção apresentam-se os testes realizados após o desenvolvimento da aplicação do taxista. Tal como no caso da aplicação do cliente, os testes foram realizados por todos os colaboradores da empresa, para se simular o teste do protótipo em ambiente real.

Na Figura 73 tem-se representado o teste de chegada de um pedido à aplicação do taxista e consequente ação do taxista. Este ensaio engloba tanto a aplicação do taxista como também a plataforma de gestão, na atribuição de serviços de táxis.



Figura 73 – Teste de chegada de um pedido e consequente ação do taxista

Quando a plataforma envia um pedido para a aplicação do taxista, este é informado do mesmo e das suas características (rua onde se encontra o cliente e distância e tempo que demora ao cliente). O taxista pode aceitar o pedido sendo-lhe apresentada uma janela com

as características do serviço, em que tal como explicado no capítulo implementação (secção 5.5), poderá telefonar para o cliente e visualizar num navegador a localização do cliente. Para concluir o serviço, o taxista pressiona o botão existente no canto inferior esquerdo do ecrã (“Concluir”) em que a aplicação será encaminhada para o ecrã anterior.

Foi verificado ainda se transferência do serviço para outro taxista estava a funcionar corretamente. Para isso o taxista que irá transferir o serviço necessita de carregar no botão do canto inferior direito do ecrã (“Transferir”), em que lhe é apresentada uma janela para confirmar a transferência. Ao realizar a confirmação aparece outra janela para introduzir o número de porta do táxi para quem vai ser transferido o serviço. Ao submeter a transferência será validada pela plataforma e a aplicação redirecionada para a *activity* principal.

Se o taxista rejeitar um pedido que chegou, é comunicado automaticamente à plataforma essa rejeição e a aplicação é redirecionada para a *activity* principal. Por fim se o taxista não responder em tempo útil ao pedido (tempo definido pelo gestor da plataforma) é apresentada a Figura 50 que se encontra na secção 5.5 (Tempo de Espera Superado).

Por fim, foi testada a chegada da plataforma de uma transferência de serviço por parte de outro taxista, em que ao ser recebida a aplicação do taxista é reencaminhada para a janela da Figura 74.

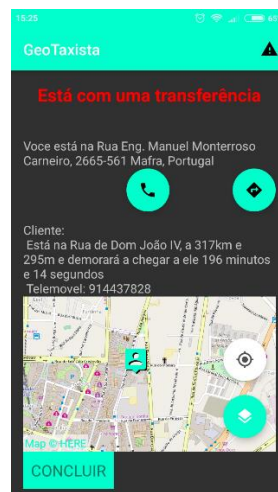


Figura 74 - Teste de Transferência de Serviço Recebida

Na Figura 75 tem-se representado o ensaio realizado na visualização do calendário de praças do taxista.

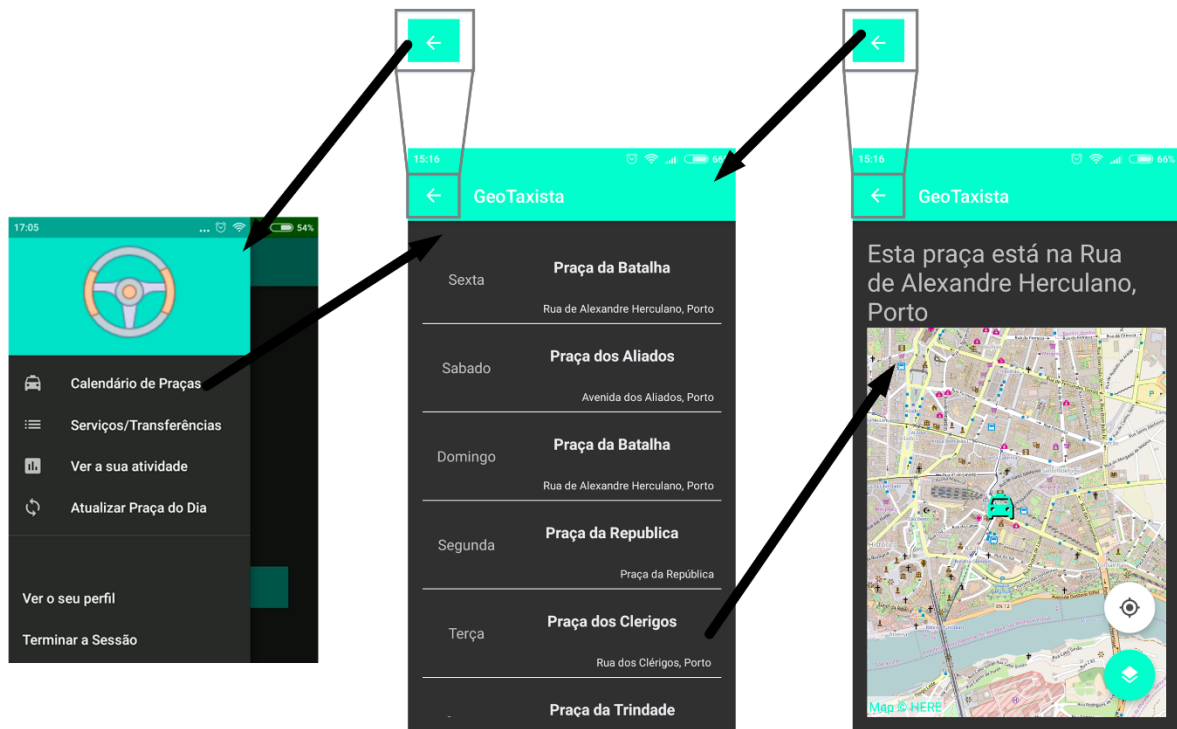


Figura 75 – Teste de consulta do calendário de praças do taxista

Para se realizar este ensaio, no menu lateral da *activity* principal tem de se carregar em “Calendário de Praças”. Ao executá-lo é mostrado ao cliente uma lista das praças em que este tem de estar em cada um dos dias da semana. Ao selecionar um dia da semana, é apresentado um mapa com a localização da praça correspondente e as suas características. Para se regressar à página anterior pressiona-se o botão existente no canto superior esquerdo do ecrã (←). A Figura 76 apresenta o teste realizado para a consulta dos serviços realizados num determinado dia.

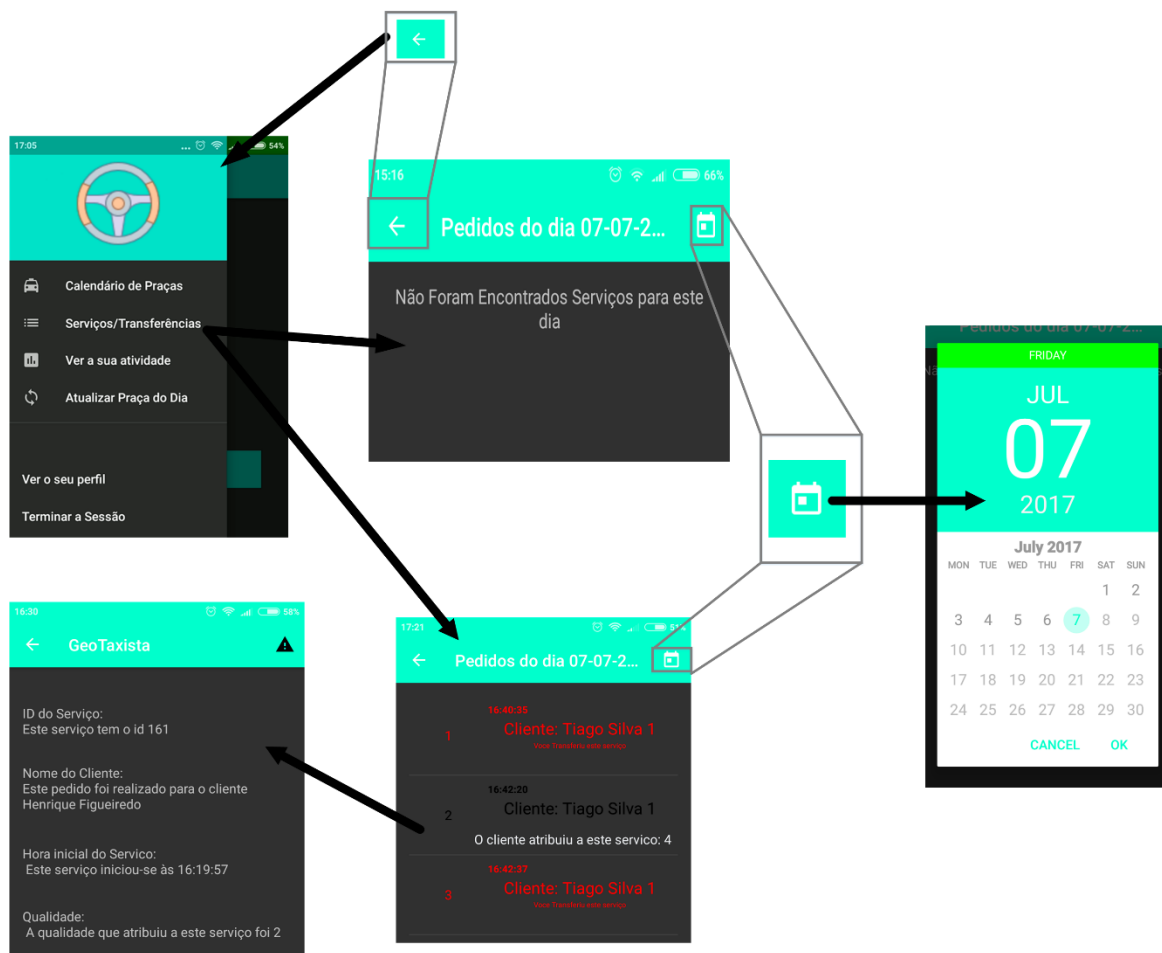


Figura 76 – Teste de visualização dos serviços realizados num determinado dia

Ao realizar este teste, pressionou-se em “Serviços/Transferências” no menu lateral da *activity* principal, em que serão apresentados ao utilizador todos os serviços realizados no corrente dia, ou então uma mensagem a informar que ainda não tem pedidos concretizados nesse dia. Ao se seleccionar um dos serviços são apresentadas as características do mesmo ao utilizador. Na lista de serviços, aparecem com a cor vermelha aqueles serviços que foram transferidos por parte do taxista, não podendo este visualizar as suas características.

Para se regressar à página anterior pressiona-se o botão existente no canto superior esquerdo do ecrã. Foi ainda testado a alteração do dia da consulta, em que se tem de pressionar o botão existente no canto superior direito, ao realiza-lo é apresentado um calendário para ser escolhido o dia pretendido.

Na Figura 77 está representado a verificação realizada para a consulta do tempo de atividade do taxista num determinado dia.

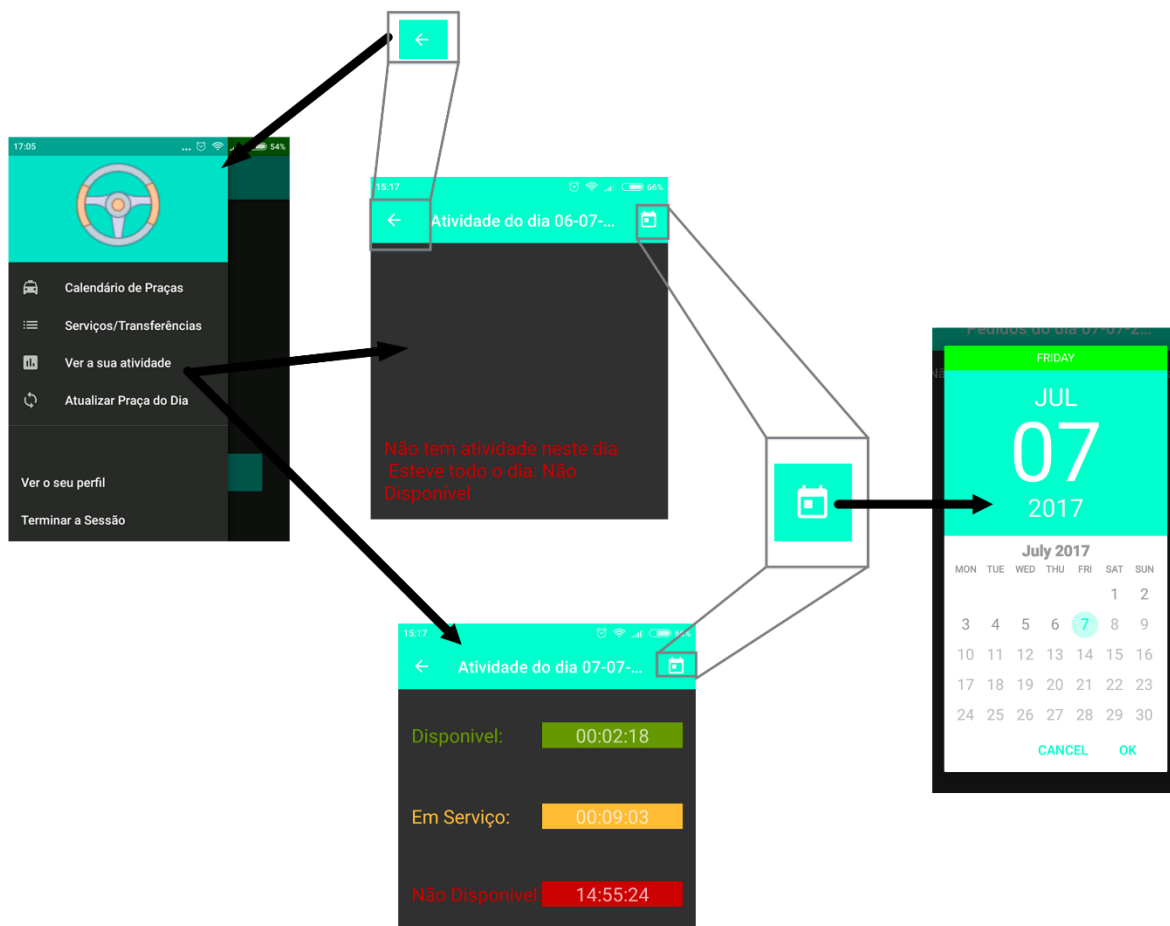


Figura 77 – Teste de visualização do tempo de atividade num determinado dia

Para se realizar esta verificação necessita-se de entrar no menu lateral da *activity* principal e pressionar em “Tempo de atividade”. Ao realiza-lo é exibido ao taxista uma janela com o tempo de atividade que este esteve em cada um dos estados existentes (não disponível, disponível e em serviço) no corrente dia. Tal como nos casos anteriores, para se regressar à página anterior pressiona-se o botão existente no canto superior esquerdo do ecrã. Foi igualmente testado a alteração do dia da consulta.

Por fim, o último teste realizado e apresentado na Figura 78 foi o de apresentação de uma queixa.

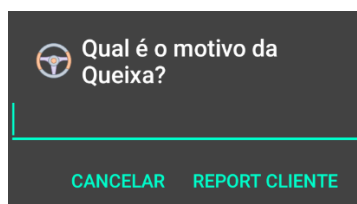


Figura 78 – Teste de uma queixa sobre o Cliente

Este teste foi realizado, tanto no momento em que o taxista está em serviço, como também, quando o mesmo se encontra a consultar as características do serviço realizado. Nos dois casos o utilizador taxista carrega no botão existente no canto superior direito do ecrã e aparece a janela da Figura 78. Ao concluir a escrita da queixa e ao submetê-la, é apresentada uma mensagem de sucesso.

6.4. FEEDBACK DA EQUIPA

Após terminar a implementação de todas as partes constituintes da arquitetura do sistema (apresentada na Figura 14 da secção 5.1), foi iniciada a fase de testes ao sistema desenvolvido. Para isto acontecer foram gerados os APKs (*Android Package*) das aplicações móveis e distribuídos pelos colaboradores da empresa.

As aplicações foram testadas por sete colaboradores da empresa Gisgeo, para a qual foi desenvolvido o projeto apresentado nesta dissertação.

Os utilizadores das aplicações referiram que estas se encontravam fluídas e intuitivas, de rápida compreensão e utilização. Deram algumas ideias para melhorar a qualidade das aplicações que foram implementadas, tais como, a inclusão na aplicação do taxista da chamada para cliente e o reencaminhamento para um navegador GPS com a localização de recolha do cliente, quando estiver em serviço. Na aplicação do cliente deram a ideia de o registo ser apenas para utilizadores de maior idade.

Numa versão anterior à apresentada neste documento, o aspeto negativo apontado pelos colaboradores da empresa focou-se no aspeto gráfico, que não se encontrava muito adequado em termos de cores e identidade das aplicações produzidas pela empresa. Para colmatar esta falha foi mudado o aspeto das aplicações móveis para as apresentadas atualmente neste documento. Para este novo aspeto, escolheu-se as cores atribuídas aos táxis em Portugal (preto e verde).

Por fim, indicaram alguns aspetos a implementar no futuro, tais como, incorporar uma secção na aplicação do cliente para este simular o custo da viagem, ou então, dar mais funcionalidades à aplicação do gestor, incluindo os serviços a decorrer em tempo real.

7. CONCLUSÕES E TRABALHO FUTURO

Neste capítulo serão apresentadas as conclusões obtidas através das análises, escolhas e implementações realizadas. Serão ainda discutidos os melhoramentos que poderão ser realizados no futuro ao projeto desenvolvido.

Este projeto pretende responder ao problema que foi apresentado pela empresa Gisgeo. Problema esse que era a ausência de uma plataforma de gestão e atribuição de serviço de táxis em pequenas e médias cidades, que foi colmatado com a solução apresentada nesta dissertação.

Com a solução definida e implementada, os taxistas, daqui em diante, possuem uma ferramenta onde poderão aceitar e consecutivamente realizar um serviço, que até agora estava dependente de uma chamada telefónica ou da chegada de uma pessoa à praça de táxis. Há outros aspetos também relevantes para o taxista entre os quais: a indicação da sua praça no corrente dia.

No que diz respeito aos clientes, daqui em diante, eles possuem uma aplicação que permite requisitar um serviço de táxi, que até aqui só existiam nos grandes centros urbanos, sem desembolsar qualquer valor monetários com a chamada telefónica para a central de táxis,

nem tempo com a deslocação à praça mais próxima. Uma outra vantagem desta aplicação é permitir ao cliente visualizar onde se encontram as praças na cidade.

Por último, o gestor da plataforma, tem uma aplicação para gerir o serviço de táxis na sua cidade, perdendo menos tempo a atribuir os serviços que chegam através das chamadas telefónicas. Tem ainda a responsabilidade de preservar o bem-estar dos taxistas, tomando conhecimento das suas queixas. As aplicações foram criadas com o intuito de serem intuitivas e de fácil utilização, o que permitirá que qualquer um dos três atores da solução tenha uma rápida adaptação, não sendo, portanto, este um obstáculo à sua divulgação.

Esta solução, apesar de ter sido desenvolvida tendo em vista a distribuição dos serviços de táxis na cidade de Chaves, não está, de maneira alguma, limitada à sua forma de atribuição dos serviços daquela cidade, podendo ser utilizada noutra cidade. A solução foi preparada para conseguir responder à maioria das pequenas e médias localidades. Esta solução pode ser perfeitamente reajustada para permitir que outras cidades com serviço de táxis possam aderir à plataforma, mesmo que não tenham as mesmas regras da cidade para a qual foi desenvolvida.

7.1. CONSIDERAÇÕES FINAIS

Em termos de conclusão, os objetivos inicialmente propostos foram atingidos com sucesso e o *feedback* dado pelos colaboradores da empresa para a qual se realizou este projeto, foi positivo.

O objetivo principal do projeto era desenvolver uma plataforma de gestão e atribuição de serviços de táxis. Partindo deste objetivo, foi necessário criar uma aplicação móvel para os clientes, onde estes pudessem efetuar uma requisição de um serviço de táxi. A plataforma de gestão atribuir-lhe um táxi e enviar uma notificação com o pedido para uma aplicação criada para os taxistas. O taxista pode aceitar ou rejeitar o serviço. Ao aceitar o serviço, a plataforma deve informar o cliente que o seu táxi está a caminho, dando tanto ao cliente como ao taxista, a informação sobre o serviço que está a decorrer. Ao ser rejeitado pelo taxista, a plataforma terá de escolher outro táxi. Para administrar a plataforma de gestão, foi criada a aplicação do gestor, onde este pode configurar as características da gestão e atribuição dos serviços de táxis.

Para concluir, em termos académicos, enveredar pelo desenvolvimento Android abriu caminho para um estudo específico e pormenorizado de uma plataforma em constante crescimento no mundo empresarial. Em termos pessoais, a realização deste estágio na *Gisgeo – Information System*, permitiu um crescimento pessoal em áreas fundamentais no universo empresarial tais como rotina, planeamento e trabalho de equipa.

7.2. TRABALHO FUTURO

Sendo o produto desenvolvido um protótipo que prevê colmatar uma lacuna no mercado do táxi, é essencial realizar testes piloto mais exigentes que permitam identificar outras limitações da ferramenta. No entanto, de seguida são listadas propostas desenvolvimentos futuros com vista a complementar as funcionalidades implementadas e a ultrapassar as atuais limitações:

Aplicação do Gestor:

- Consultar os serviços realizados e o tempo de atividade de cada um dos táxis existentes na cidade administrada.
- Mostrar no mapa os pedidos e serviços que estão a decorrer no momento.

Aplicação do Cliente:

- Permitir que o cliente escolha o local onde quer ser recolhido pelo táxi, selecionando num mapa esse ponto.
- Permitir que o cliente possa reservar um serviço de táxi para uma data e hora posterior, e se porventura quiser, com um taxista preferencial.
- Permitir ao cliente efetuar uma queixa sobre o taxista e esta ficar visível para o administrador da plataforma.

Aplicação do Taxista:

- Permitir que o taxista visualize quem são os elementos da sua equipa, com toda a informação sobre estes.
- Criar uma lista que o taxista pudesse aceder para mostrar todos os números de porta dos táxis da cidade onde se encontra.

Referências Documentais

- [1] Statista, “Cell phone sales worldwide 2007-2016.” [Online]. Available: <https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/>.
- [2] Statistic, “Google Play Store: number of apps 2009-2017.” [Online]. Available: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>.
- [3] GISGEO - Information Systems, “GISGEO :: information systems.” [Online]. Available: <http://www.gisgeo.pt/index.php>.
- [4] GISGEO - Information Systems, “GISGEO :: information systems | soluções de apoio à decisão.” [Online]. Available: <http://www.gisgeo.pt/solucoes.php>.
- [5] GISGEO - Information Systems, “GeoCar | Gestão de Frotas, Localização de Viaturas, Sistemas de Informação Geográfica - Frotas.” [Online]. Available: <http://www.geocar.info/pages/gestao-de-frotas/>.
- [6] A. T. Bond, “An App for That: Local Governments and the Rise of the Sharing Economy,” *Notre Dame Law Rev.*, vol. 90, no. 2, pp. 77–96, 2015.
- [7] M. Beesley, *Privatization, Regulation and Deregulation*. Routledge, 1997.
- [8] G. Constitucional, “Decreto-Lei n.º 251/98 - Regulamenta o acesso à actividade e ao mercado dos transportes em táxi,” pp. 1–15, 2017.
- [9] ANTRAL, “Tabela de Preços.”
- [10] R. Belk, “You are what you can access: Sharing and collaborative consumption online,” *J. Bus. Res.*, vol. 67, no. 8, pp. 1595–1600, 2014.
- [11] F. Bardhi and G. M. Eckhardt, “Access Based Consumption: The Case of Car Sharing,” *J. Consum. Res.*, vol. 39, no. 4, pp. 881–898, 2012.
- [12] M. De Novellis, “Social Impact Lures MBA Grads To Sharing Economy Pioneers Like Airbnb | BusinessBecause.” [Online]. Available: <http://www.businessbecause.com/news/sharing-economy/3734/social-impact-lures-mbas-to-sharing-economy-pioneers>.
- [13] K19 Treinamentos, “Desenvolvimento Mobile com Android,” p. 152, 2012.
- [14] L. Addison Wesley, “A history of HTML,” 1998. [Online]. Available: <https://www.w3.org/People/Raggett/book4/ch02.html>.
- [15] “Platform Architecture | Android Developers.” [Online]. Available: <https://developer.Android.com/guide/platform/index.html>.
- [16] “Tasks and Back Stack | Android Developers.” [Online]. Available: <https://developer.Android.com/guide/components/activities/tasks-and-back-stack.html>. [Accessed: 09-Aug-2017].
- [17] Android Developers, “Transmitting Network Data Using Volley.” [Online]. Available: <https://developer.Android.com/training/volley/index.html>.
- [18] Parvez, “Types of Web Services SOAP,XML-RPC and Restful.” [Online]. Available: <https://www.phpflow.com/php/web-service-types-soapxml-rpcrestful/>.
- [19] C. D. F. FERREIRA and R. D. MOTA, “Comparando Aplicação Web Service Rest e Soap,” *UniPar*, 2006.
- [20] R. Data, “HTML Reference,” *W3School*, 2015. [Online]. Available: https://www.w3schools.com/tags/ref_byfunc.asp.
- [21] T. A. Powell, *The Complete Reference HTML & CSS*, vol. 1. 2011.
- [22] “PHP: Hypertext Preprocessor.” [Online]. Available: <http://be2.php.net/>.

- [23] PostGIS Project Steering Committee (PSC), “PostGIS — Spatial and Geographic Objects for PostgreSQL,” 2013. [Online]. Available: <http://postgis.net/>.
- [24] “OpenLayers - Welcome.” [Online]. Available: <http://openlayers.org/>.
- [25] “Bootstrap · The world’s most popular mobile-first and responsive front-end framework.” [Online]. Available: <http://getbootstrap.com/>.
- [26] R. L. Jacinto, “SuusMDM : Gestão de Parques Informáticos de Terminais Móveis Engenharia Informática e de Computadores,” 2012.
- [27] Firebase, “Firebase Cloud Messaging.” [Online]. Available: <https://firebase.google.com/docs/cloud-messaging/>.
- [28] Oracle, “Database.” [Online]. Available: <https://www.oracle.com/database/index.html>.
- [29] Microsoft, “SQL Server 2016.” [Online]. Available: <https://www.microsoft.com/pt-pt/sql-server/sql-server-2016>.
- [30] The PostgreSQL Global Development Group, “PostgreSQL.” [Online]. Available: <https://www.postgresql.org/>. [Accessed: 10-Aug-2017].
- [31] Oracle Corporation, “MySQL.” [Online]. Available: <https://www.mysql.com/>. [Accessed: 10-Aug-2017].
- [32] M. C. Hansen, A. Krylov, and A. Tyukavina, “International Conference on Recent Trends in Physics 2016 (ICRTP2016),” *J. Phys. Conf. Ser.*, vol. 755, p. 11001, 2016.
- [33] The PostgreSQL Global Development Group, “About PostgreSQL.” [Online]. Available: <https://www.postgresql.org/about/>.
- [34] S. Ceria, “Casos de Uso,” p. 18, 2001.
- [35] G. R. De Souza, “Diagrama de Caso de Uso,” *Instituto Fed. Educ. Cienc. e Tecnol.*, 2010.