



LEONARDO
FERNANDES
PINHEIRO

SMART PARKING: UMA APLICAÇÃO MÓVEL PARA
GESTÃO INTELIGENTE DE LOCAIS DE
ESTACIONAMENTO

SMART PARKING MOBILE APPLICATION

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor José Manuel Matos Moreira do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho aos meus pais pela força que me deram em todos os momentos difíceis que tive ao longo deste percurso. Também lhes quero agradecer pelos sacrifícios que fizeram para me darem um futuro melhor. Sem eles hoje não estaria aqui.

Além disso quero dedicar todo este trabalho ao meu amigo Jorge Fernandes que ficará para sempre no meu coração.

o júri

presidente

Prof. Doutor Joaquim Arnaldo Carvalho Martins
Professor Catedrático, Universidade de Aveiro

vogais

Prof. Doutor Fernando Joaquim Lopes Moreira
Professor Associado, Departamento de Inovação, Ciência e Tecnologia da Universidade
Portucalense

Prof. Doutor José Manuel Matos Moreira
Professor Auxiliar, Universidade de Aveiro

agradecimentos

Quero começar por agradecer aos meus pais, que são as pessoas mais importantes da minha vida, pelos mesmos motivos que lhes dedico este texto. Gostaria também de incluir a minha irmã neste agradecimento por ter sido um apoio incondicional durante toda a minha vida.

Gostava de deixar um agradecimento à minha madrinha por ter sido o meu maior apoio nesta minha aventura em Portugal.

De seguida, gostaria de agradecer imenso ao meu orientador Prof. Doutor José Manuel Matos Moreira, pelo seu apoio e por toda a ajuda que me deu nesta última etapa do meu percurso académico.

Quero agradecer ao Rui Costa e ao Nuno Ribeiro por me terem deixado realizar esta dissertação em conjunto com a Ubiwhere. Em relação aos colaboradores da Ubiwhere quero começar pelo Bruno Silva, o meu orientador dentro da Ubiwhere, que foi fundamental para chegar a esta meta. Esteve sempre disponível para tudo que precisei, deu-me todo o apoio necessário para concluir esta fase, além de ter sido a pessoa que mais me ensinou. Para além deles, quero agradecer ao André Duarte, Francisco Monsanto, Ricardo Vitorino e ao Pedro Diogo por me terem ajudado com todas as dificuldades que encontrei de forma a integrar a aplicação no Citibrain. Quero também lhes agradecer por me terem integrado na equipa deles.

Por último, mas não menos importante, quero agradecer a todos os meus amigos que fiz durante este maravilhoso percurso. Quero-lhes dizer obrigado por me terem apoiado e ajudado a chegar ao final deste percurso. Sem querer individualizar quero deixar um especial agradecimento ao Filipe Reis, Joca, Cristiano, Ana Ortega, Diogo Gaitas, Bruno Costa, Filipa Carpinteira, Rui Filipe, Bruno Mendes, Vítor, Jéssica Silva, Dimos, Sílvia Patrícia e a todos os outros que não mencionei mas que também foram importantes.

Quero destacar o João Simões e o Nuno Barbosa por terem sido as pessoas que mais me ajudaram durante todo este percurso, que estiveram sempre presentes em todos os momentos que atravessei e pela grande amizade que construímos.

palavras-chave

Smart Parking, Smart Cities, Estacionamento, Internet of Things

resumo

Um veículo passa grande parte do seu tempo de vida estacionado e uma parte significativa do tempo que se encontra em movimento é à procura de um lugar para estacionamento. Com o número de veículos no mundo a aumentar de ano para ano, o trânsito nas estradas também tende a crescer e assim a procura de lugares de estacionamento também se torna mais difícil. Assim sendo, a gestão do tráfego e de lugares de estacionamento adquire uma importância cada vez maior, na medida em que permite diminuir os custos, acidentes e congestão de tráfego, ao mesmo tempo que contribuem para a redução dos problemas relacionadas com poluição.

Atualmente o investimento em Smart Cities tem aumentado, havendo assim mais cidades ligadas à rede que são monitorizadas em tempo real, seja por câmaras ou sensores. Com isto em mente, a Ubiwhere formou um consórcio de três empresas especializado em soluções inteligentes para as cidades, denominado Citibrain. O Citibrain foca-se em duas áreas principais: Mobilidade e Ambiente, nas quais detém quatro aplicações verticais (Trânsito, Estacionamento, Ambiente e Resíduos). Foi neste contexto que surgiu o Smart Parking, uma solução tecnológica de gestão de espaços de estacionamento que se enquadra no vertical dos estacionamentos do Citibrain.

No âmbito do Smart Parking, foi desenvolvida uma aplicação móvel, foco desta dissertação de mestrado, destinada quer a condutores, quer a entidades de gestão urbana (técnicos e agentes de fiscalização). De forma a ajudar os condutores foi criada uma forma de pesquisa de áreas de estacionamento com lugares livres e respetivo pagamento do tempo de estacionamento. A gestão de estacionamento não se resume só à procura de um lugar livre, mas também ao cumprimento do respetivo pagamento associado ao tempo de ocupação. Assim foi criada uma monitorização do estado de ocupação e de pagamento de cada lugar individual, rentabilizando assim a fiscalização dos estacionamentos para os agentes de fiscalização.

Por último, foi acrescentada uma forma de gestão de lugares, que permite aos técnicos proceder à instalação, reparação e alterações de cada sensor relacionado com cada lugar individual.

Todas estas funcionalidades estão acessíveis usando um smartphone onde os utilizadores podem correr a aplicação.

Após concluir o desenvolvimento da aplicação e de serem realizados testes por parte de alguns utilizadores, conclui-se que a aplicação vai ao encontro das necessidades encontradas no dia-a-dia, sendo ainda fácil de usar. Em relação aos fiscais consegue-se reduzir o tempo de fiscalização, rentabilizando assim o trabalho dos mesmos. Por último, a autonomia oferecida na questão de gestão de sensores, é outra mais-valia, bastando os técnicos de instalação recorrerem à aplicação.

keywords

Smart Parking, Smart Cities, Parking, Internet of Things

abstract

A vehicle spends much of its lifetime parked, and considerable part of the time it spends on the road it is for looking for a parking spot. As the number of vehicles in the world increases from year to year, road traffic also tends to grow and so the demand for parking spaces becomes more difficult. Consequently, traffic management and parking spots are becoming increasingly important as they reduce costs, accidents and traffic congestion, while at the same time they contribute to the reduction of pollution-related problems.

Currently, as the investment in Smart Cities has increased, more cities are connected to the world wide web and monitored in real time, either by cameras or by sensors. With this in mind, Ubiwhere has created a consortium of three companies specialized in smart cities solutions, called Citibrain. Citibrain focuses on two main areas: Mobility and Environment, in which it has four vertical applications (Traffic, Parking, Environment and Waste). In this context, the Smart Parking Mobile Application was created, a technological solution for the management of parking spaces, which fits in the vertical of Citibrain's parking.

In the scope of Smart Parking focusing on this master's dissertation, a mobile application has been developed, intended for drivers and the urban management entities (technicians and inspection agents). In order to assist drivers, a tool to search parking areas and manage payments was created. As explained, parking management is not just about finding an available place, but also about fulfilling the payment of the occupation time. To make possible for the inspection agents to control the parking slots in real time, a monitoring system of the state of occupation and payment was created.

To allow technicians to proceed with the installation, reparation and modifications of each sensor related to each individual place, a management component was also added to the application.

All these features are at a distance of a simple smartphone where users can run the application.

After completing the development of the application and testing with some users, the results show that the application meets the needs found in everyday life and stills easy to use. In relation to the inspection agents, the application makes it possible to reduce inspection time, thus making their work profitable. Finally, the autonomy offered in the issue of sensor management, is another benefit, just by accessing the mobile application by the technicians.

Índice

1. Introdução	1
1.1 Descrição da aplicação	2
1.2 Motivação e Objetivo Geral.....	3
1.3 Público-alvo.....	3
2. Estado de Arte	5
2.1 Gestão de estacionamentos	5
2.1.1 Estacionamentos off-street	6
2.1.2 Estacionamentos on-street	8
2.2 Sensores	10
2.3 Métodos de Pagamento	11
2.3.1 Soluções existentes.....	12
2.3.2 Segurança nos pagamentos	15
2.3.3 Ferramentas de desenvolvimento	16
3. Objetivos e Especificações	19
3.1 Objetivos	19
3.2 User Stories	21
3.2.1 Perfil Condutor	21
3.2.2 Perfil Agente de fiscalização.....	26
3.2.3 Perfil Técnico de instalação de sensores	31
4. Definição da Arquitetura do Sistema	33
4.1 Backend	33
4.1.1 API RESTful	35
4.1.2 OAuth2.0	36

4.1.3 WebSockets	37
4.2 Frontend	37
5. Desenvolvimento da Solução	39
5.1 Requisitos de desenvolvimento	39
5.1.1 Ambiente de Desenvolvimento	39
5.1.2 Planeamento de Projeto e Controlo de Versões	39
5.2 Backend	42
5.3 Aplicação	50
5.3.1 Registo e Login	53
5.3.2 Filtro de indicativos	55
5.3.3 Loading Screen	56
5.3.4 Menu	57
5.3.5 Perfil	58
5.3.6 Áreas de Estacionamento	58
5.3.7 Carros	60
5.3.8 Pagamentos e Histórico	62
5.3.9 Verificação	64
5.3.10 Atualização de estado dos sensores	70
5.3.11 Instalação de sensores	71
5.3.12 Gestão de sensores	72
5.3.13 Definições	73
5.4 Desenvolvimento em Ionic	74
5.4.1 Integração de Ion-datetime-picker	75
5.4.2 Integração de Google Maps API	76
5.4.3 Integração de Geofence	78
5.4.4 Integração de marker-clustering	79
5.4.5 Integração de angularjs-slider	80
5.4.6 Integração de BarcodeScanner	81
5.5 Testes	83
5.5.1 Testes unitários	83

5.5.2 Testes de utilizadores	84
5.6 Dificuldades.....	85
6. Conclusão	87
7. Trabalho Futuro	89
8. Referências	91

Índice Figuras

Figura 1: Tabela de tempo perdido por dia em cada um dos bairros estudados [12]	6
Figura 2: Sensor usado para comunicar com a aplicação	11
Figura 3: Google Wallet Card [35]	15
Figura 4: Arquitetura do sistema	34
Figura 5: Funcionamento do Meshblu[51]	35
Figura 6: APIs dos backend.....	36
Figura 7: Tarefas do Gitlab	40
Figura 8: Controlo de versões do Gitlab	41
Figura 9: Users Api e JSON Object.....	42
Figura 10: Parking_events Api e JSON Object	43
Figura 11: Infractions Api e JSON Object	44
Figura 12: Occupancies Api e JSON Object	45
Figura 13: Payments Api e JSON Object.....	45
Figura 14: Parking_spots Api e JSON Object	46
Figura 15: Parking_areas Api e JSON Object	47
Figura 16: Vehicles Api e JSON Object	48
Figura 17: Parking_spots_distances Api e JSON Object	49
Figura 18: Routing Api e JSON Object	49
Figura 19: Tabela de cores relativas aos estacionamentoos	52
Figura 20: Processo de Registo e/ou Login.....	54
Figura 21: Seletor de indicativos	55
Figura 22: Ecrã de carregamento.....	56
Figura 23: Menu.....	57
Figura 24: Perfil de utilizador	58
Figura 25: Fluxo de navegação para área de estacionamento	59
Figura 26: Método de adição de carros pelos condutores	60
Figura 27: Método de adição de carros pelos agentes de fiscalização.....	61
Figura 28: Workflow de pagamento	62
Figura 29: Associação de pagamento efetuado por condutor	65

Figura 30: registo de pagamento efetuado usando parquímetro	67
Figura 31: Emissão de coima	69
Figura 32: Mensagem enviada pelo Meshblu.....	70
Figura 33: Instalação de sensores	71
Figura 34: Ecrãs de gestão de sensores	72
Figura 35: Definições.....	73
Figura 36: Funcionamento do Ionic [67]	74
Figura 37: Ion-datetime-picker.....	75
Figura 38: Pesquisa usando o Google Maps	76
Figura 39: Rota até ao centro de Aveiro	77
Figura 40: Street View dos Estacionamentos.....	77
Figura 41: Geofence [64]	78
Figura 42: Notificação usando o Geofence	78
Figura 43: Marker-clustering.....	79
Figura 44: Integração de angularjs-slider	80
Figura 45: Leitura de bilhete	82
Figura 46: Teste unitário de login.....	83

1. Introdução

Atualmente as tecnologias estão cada vez mais presentes no nosso dia-a-dia. No nosso mundo tudo começa a ser controlado de forma remota com um dispositivo móvel. As cidades não fogem a esse padrão e começam a ser controladas e monitorizadas por dispositivos. Os estacionamentos nas cidades são um dos pontos que se está a tentar aceder remotamente, de forma a ser possível conseguir encontrar um lugar de estacionamento sem ter de se perder muito tempo à procurar do mesmo e assim descongestionar o trânsito nas cidades.

Assim sendo, estão a ser implantados sensores de deteção de veículos em zonas destinadas a estacionamento. Graças a estes sensores já é possível conhecer o estado de um determinado lugar de estacionamento, se está livre ou ocupado. Percebendo que circulam mais de 1 bilião de veículos no mundo segundo um artigo de 2011 do jornal Le Monde [1], se se conseguir ter a informação relativa aos estacionamentos disponível para consulta em tempo real, então será possível facilitar a vida diária de todos os condutores.

Como as Smart Cities vieram para ficar, a Ubiwhere em conjunto com a Micro I/O [2] e a Wavecom [3] criaram o Citibrain [4], projeto focado nas Smart Cities. O Citibrain atualmente já tem quatro verticais, o Smart Parking, que foi apresentado nesta dissertação, o Smart Waste [5] que consiste em monitorizar os contentores do lixo públicos e definir rotas até ao mesmo para ser feita a recolha de forma rápida e eficaz, o Smart Air Quality [6] usado para medir fatores do ambiente de uma cidade, entre níveis de poluição, temperaturas, qualidade de ar, etc. e o Smart Traffic [7] que permite medir o trânsito das diversas estradas e tentar distribuir o mesmo pela restante cidade de forma a facilitar a circulação na mesma.

Esta dissertação apresenta uma solução para o problema anteriormente descrito, integrando-o no que hoje chamamos de Smart Cities. Este trabalho foi desenvolvido na Ubiwhere [8] e foi integrado no vertical do Smart Parking apresentado anteriormente.

1.1 Descrição da aplicação

A aplicação móvel que foi desenvolvida tem o nome de “Smart Parking: uma aplicação móvel para gestão inteligente de locais de estacionamento” e tem como principal objetivo a consulta do estado dos lugares de uma zona selecionada para a qual o condutor se quer deslocar. É oferecida ao utilizador a opção de selecionar uma área de estacionamentos, ver o número de lugares vagos que se encontram na mesma e obter a rota até à área selecionada. Assim a procura de um lugar vago deixa de ser uma preocupação e perda de tempo. Com esta aplicação pretende-se facilitar as deslocações dos utilizadores para locais que desconhecem, monitorizando em tempo real onde se encontram lugares vagos.

A aplicação também está destinada às entidades reguladoras dos estacionamentos, que também poderão fiscalizar os pagamentos dos estacionamentos que se encontram ocupados através da aplicação.

Por último, com a ajuda da aplicação os técnicos vão poder integrar novos lugares de estacionamento e corrigir erros que poderão existir em alguns sensores.

1.2 Motivação e Objetivo Geral

Uma vez que já existem algumas aplicações que vão ao encontro da pesquisa de estacionamentos, ainda são muito poucas as que permitem juntar a fiscalização à pesquisa de um estacionamento. Além disso, das aplicações que já existem ainda não houve nenhuma que conquistou o mercado por unanimidade. Assim sendo consegue-se oferecer uma única aplicação que permite que um condutor consiga encontrar um lugar com facilidade e proceder ao pagamento do mesmo, que permita que um agente de fiscalização consiga saber através do seu smartphone onde se encontram todos os veículos que ainda não foram controlados e que permita que um técnico faça a gestão de sensores de estacionamento.

Com esta aplicação pretende-se principalmente os seguintes objetivos:

- Reduzir o tempo de procura de um estacionamento.
- Facilitar o pagamento e prolongamento de tempo de um estacionamento.
- Rentabilizar as fiscalizações de estacionamentos localizando todos os veículos não controlados.
- Gerir os sensores instalados de forma a proceder à instalação, trocas e mesmo à sua remoção caso necessário.

1.3 Público-alvo

Os principais alvos da aplicação são os seguintes:

- Condutor – pessoa que está a procura de um lugar vago numa localização e pretende efetuar o pagamento do mesmo.
- Agente de fiscalização – pessoa que vai fiscalizar os pagamentos dos estacionamentos e emitir coimas caso não tenha sido efetuado nenhum pagamento por parte do condutor do veículo.
- Técnico – pessoa que vai gerir os sensores de estacionamento.

2. Estado de Arte

2.1 Gestão de estacionamentos

Atualmente vivemos num mundo onde o conceito de Smart City se começa a desenvolver. Cada vez mais tudo é controlado de forma remota e através de um simples clique num dispositivo estamos ao alcance de tudo, e as Smart Cities não são exceção. Atualmente existem muitos estudos sobre como controlar as cidades e o que se pode implementar nas mesmas para facilitar a vida dos seus moradores, das pessoas que lá trabalham e de quem por outro motivo se desloca às mesmas.

O estudo [9] revela que um carro passa 95% do seu tempo de vida estacionado e apenas 5% em circulação, sendo por isso os estacionamentos uma das vertentes visadas pelas Smart Cities. Se analisarmos esse trânsito concluímos que grande parte não é causado por quem se encontra no seu trajeto mas sim por quem já chegou ao seu destino, sendo que 30% é causado pela procura de estacionamentos como nos revela D. C. Shoup em “Cruising for parking” [10]. Com esta informação conseguimos facilmente perceber que para reduzir a circulação dos carros dentro das cidades, os estacionamentos são um fator importante a ter em conta.

Esta dissertação é sobre esse mesmo tema, o Smart Parking, que consiste na forma de estacionar de modo inteligente e sem preocupações. Citando a Citroën “Ao longo da sua vida vai passar 4 anos e 1 mês num carro, por isso aproveite!” [11]. A procura de estacionamento é um dos pontos mais desagradáveis desse tempo, e assim sendo, importa explorar uma forma de tornar todo esse tempo o mais agradável possível. Nesta dissertação vai-se tentar ir ao encontro da melhor maneira de estacionar, entenda-se mais rápida, sem ser necessário andar à procura de um parqueamento durante um período indeterminado de tempo. Pela investigação feita em [12], 100% dos habitantes de Paris (dos bairros de Saint-Germain e Commerce) já desistiram no mínimo uma vez de uma deslocação por falta de estacionamento. Somando o tempo diário perdido à procura

de um lugar vago em locais autorizados no bairro de Saint-Germain chega-se a 294 horas e 462 horas no bairro de Commerce como nos ilustra a tabela da Figura 1. Analisando de forma geral o país inteiro, chega-se a acumular cerca de 70 milhões de horas perdidas todos os anos à procura de um estacionamento o que representa cerca de 700 milhões de euros perdidos todos os anos, isto tudo sem olhar para o trânsito, poluição e efeito de estufa provocados por todos estes veículos.

	Total time lost	Lost time per hectare	Lost time per authorized space
Lyon district Presquîle	434h	14h 14min	41min
Grenoble district Vaucanson	157h	6h 21min	14min
Paris district Commerce	462h	9h 57 min	20min
Paris district Saint-Germain	294h	13h 40 min	42min

Figura 1: Tabela de tempo perdido por dia em cada um dos bairros estudados [12]

Segundo D. C. Shoup [10], com base nos vários tipos de estacionamento que existem pode-se concluir que as duas grandes vertentes são off-street e on-street.

2.1.1 Estacionamentos off-street

Estacionamentos off-street é o nome atribuído às casas de estacionamento, edifícios ou áreas criadas com a única utilidade de fornecer estacionamento para os veículos. Grande parte destas zonas de estacionamento está restrita por um gradeamento ou mesmo cancelas.

Sabendo que mais de metade da população vive em cidades, a utilidade de ter uma aplicação que disponibiliza o tempo médio de ocupação de um lugar, a sua taxa de ocupação diária e a disponibilidade em tempo real dos parques de estacionamento facilitaria a vida diária dos condutores. Em [13], após ser analisado se há falta de lugares de estacionamento ou se apenas é necessário uma melhor gestão dos estacionamento disponíveis, a conclusão foi que basta melhorar a gestão dos mesmos. Neste último artigo são sugeridas duas formas de estacionamento, uma estática e outra dinâmica. A grande diferença entra as duas é o facto de

ser necessário efetuar uma pré-reserva de um lugar de estacionamento ou chegar e estacionar no lugar vago mais próximo. O estacionamento estático é feito com antecedência e assim obriga o condutor a efetuar a reserva de um lugar mesmo antes de o ocupar. O utilizador acede à sua aplicação e verifica para o espaço de tempo que necessita onde se encontra um lugar desocupado e faz a reserva e o pagamento do mesmo, tendo desta forma o lugar garantido à hora desejada. O método dinâmico, que é o mais usado e também explorado nas restantes soluções apresentadas consiste em chegar a um local, consultar os lugares vagos onde o condutor pode parquear o seu veículo por um tempo indeterminado e efetuar o pagamento do tempo pretendido, podendo ainda ser oferecido ao utilizador um método de conseguir prolongar o tempo de estacionamento se necessário e assim efetuar o pagamento do mesmo.

WeGo&Park é a solução oferecida por parte Wellness Telecom [14], uma empresa espanhola que tem vários projetos visando as Smart Cities. O WeGo&Park é um sistema de procura de estacionamentos baseado em câmaras de vigilância. O uso das câmaras de vigilância é justificado pelo facto de ser possível abranger um maior espaço por um menor preço comparando com os sensores individuais em cada lugar. Cada câmara tem o objetivo de abranger um determinado espaço onde se encontram um número definido de lugares. Esses espaços captados pelas câmaras são processados de forma a detetar se se encontram carros nos lugares definidos que estão a ser observados pelas câmaras.

Além das soluções apresentadas existem outras soluções que permitem explorar os parques de estacionamento limitados por cancelas, consultar as taxas de ocupação dos mesmos e ajudar o utilizador a encontrar o parque com vagas disponíveis, mais próximo de um determinado local para onde se pretende deslocar. Com todas estas características, o Parkwhiz [15], que se apresenta com as seguintes frases “Find it. Book it. Park it.”, pretende oferecer uma solução que consiste na pesquisa de áreas de estacionamento perto do destino do utilizador. O Parkme [16], além destes serviços, vai ainda mais longe e fornece a opção de estacionamento estático, que consiste numa reserva de lugar de estacionamento mesmo antes de o utilizador ter chegado ao lugar. Estes dois concorrentes dão ao utilizador a opção de poder consultar os estacionamentos de um determinado local, ver a que distância o mesmo fica do destino final, obter uma estimativa de tempo médio que terá de percorrer a pé até ao destino e ainda reservar o lugar com antecedência para o garantir no dia e hora pretendida. Estas duas soluções ainda oferecem o método de pagamento online para facilitar o serviço ao cliente. Em termos de avaliações o ParkWhiz parece ter mais sucesso junto dos utilizadores conseguindo ter no Google Play [17] uma avaliação de 4.3/5 contra 3.8/5 do Parkme (em 31.05.2017) e assim ficando à frente apenas na avaliação, já que segundo os dados fornecidos pelo Google Play ambas as aplicações estão no intervalo de “100 000 – 500 000” instalações.

Pelo mundo fora já existem várias empresas a tentar explorar a vertente do Smart Parking, sendo que muito poucas se estão a conseguir destacar. “Less time parking means less stress and happier customers.”, uma das frases com a qual a empresa Smart Parking Limited (ASX:SPZ) [18] se tenta destacar, sendo que são uma das entidades existentes que prestam o serviço do Smart Parking ao cliente. A aplicação móvel da mesma fornece o número de lugares disponíveis num dado espaço dedicado ao estacionamento e assim guia o utilizador através de indicações simples como “vire à esquerda”, “vire à direita” ou “siga em frente” até atingir o lugar pretendido. Alguns dos objetivos da empresa Smart Parking Limited (ASX:SPZ) é diminuir o trânsito nas estradas, reduzir poluição do ambiente, ter condutores menos stressados nas estradas, dar maior uso aos espaços de estacionamento, entre outras. Todos estes pontos explorados pela Smart Parking Limited (ASX:SPZ) são para ter em consideração nesta dissertação, mas pelas críticas deixadas tanto nos comentários no Google Play [17] (e tendo uma avaliação de 2.6/5 em 28.09.2016 e 1.8/5 em 31.05.2017) como no iTunes [19] a aplicação ainda precisa de ser muito melhorada ou mesmo totalmente alterada. Sendo uma aplicação que está a ser usada um pouco por todo mundo e que foi apenas transferida entre mil a cinco mil vezes em três anos, mostra que não está a conseguir recolher o quórum do público-alvo para já. A Smart Parking Limited (ASX:SPZ) além deste serviço ainda oferece a pesquisa de estacionamentos on-street aos seus utilizadores.

2.1.2 Estacionamentos on-street

Os estacionamentos on-street, assim como o nome indica, são estacionamentos junto à estrada, um local onde um condutor pode estacionar sem ter de abandonar a estrada onde se encontra.

Estes lugares estão frequentemente sujeitos a alterações temporárias, ou em certos casos acabam por ser removidos definitivamente devido às mais diversas razões com por exemplo, obras na via pública. Assim, conclui-se que é necessário criar um sistema onde um administrador consegue atualizar estas modificações nas bases de dados, disponibilizando as mesmas alterações em tempo real ao utilizador, conseguindo assim facilitar a rotina de todos os condutores e reduzir o trânsito das cidades, é o que defende Basavaraju S. R. em [20].

Utilizando um sistema de nuvem para partilha de informação resolve o problema de consultar os lugares em tempo real. Desta forma consegue-se visualizar os lugares vagos e ocupados em tempo real, tornar tudo mais ágil, facilitar a procura de lugares e reduzir o tempo de procura de lugares de estacionamento tal como é explicado em [21]. A aplicação móvel é apenas uma interface para o utilizador interagir com a plataforma de estacionamento. O utilizador pode efetuar o pagamento do estacionamento e prolongar o mesmo caso necessário. Os agentes de fiscalização podem igualmente utilizar a aplicação para uso profissional e assim consultar os locais onde estão a ser cometidas infrações. Portanto além de terem o seu trabalho facilitado conseguem torná-lo mais eficiente e rentável para as entidades reguladoras.

Como referido anteriormente a aplicação móvel da Smart Parking Limited (ASX:SPZ) além de disponibilizar os dados sobre os estacionamentos off-street também já fornece o serviço de procura de estacionamentos on-street. Estudando a solução oferecida pela mesma, são usados sensores chamados SmartEye implementados nos locais de estacionamentos, que são sensores individuais para transmissão de informação de gestão, pagamento e monitorização de conformidade. Para calcular o valor que o utilizador terá de pagar é usado um sensor de identificação por radiofrequência (RFID), tal como os identificadores que podemos colocar hoje em dia nos carros para os pórticos das autoestradas, que é lido pelo SmartEye e assim contabiliza o tempo que um veículo esteve estacionado num determinado local. Quando o condutor retirar o carro do estacionamento, será calculado, através desse mesmo tempo e o preço fixado em tabela para a duração referida, o montante que o utilizador terá de pagar. Essa mesma importância será debitada automaticamente da conta do utilizador associada ao sensor RFID, conseguindo desta forma o utilizador pagar apenas o valor correspondente ao tempo certo que ocupou realmente o estacionamento. O SmartRep é o software usado para processar todos estes dados recebidos pelos SmartEye e gerir todos os dados relacionados com os utilizadores e estacionamentos. Finalmente o SmartGuide poderá ser usado pelo utilizador final afim de o direcionar até ao lugar pretendido, tendo o mesmo a opção de personalizar a pesquisa dos lugares, tal como estacionamentos destinados a pessoas com mobilidade reduzida ou grávidas entre outros.

A aplicação Smart Parking da companhia Mobile City [22] é outra aplicação já existente que pode ser usada em cerca de cinquenta cidades na Alemanha, também vai ao encontro de alguns dos objetivos desta dissertação. Esta aplicação fornece ao utilizador a lista de estacionamentos pagos para os quais o utilizador poderá efetuar o pagamento. No entanto não oferece a listagem das localizações dos lugares vagos. Mas tal como pretendido o utilizador pode efetuar o pagamento do estacionamento do seu veículo ao indicar em que cidade e rua se encontra parqueado, selecionando o tempo que pretende lá ficar, introduzindo a sua matrícula e o seu número de telefone onde será notificado via mensagem de texto caso exceda o tempo solicitado.

Desta forma consegue também prolongar a sua sessão, pagando a mesma através da aplicação móvel. Além disso a aplicação fornece uma lista com parques de estacionamento existentes para uma determinada cidade e o seu tarifário em vigor, podendo o utilizador adicionar os parques mais utilizados por si à sua lista de favoritos. A classificação da mesma no Google Play é superior à da aplicação oferecida pela Smart Parking Limited (ASX:SPZ) tendo sido avaliada pelos utilizadores em 2.9/5 (em 28.09.2016).

Todos estes trabalhos vão ao encontro da aplicação Smart Parking Mobile Application, mas visando mais a parte off-street, sendo que o grande objetivo do trabalho desenvolvido nesta dissertação é o estacionamento on-street, vertente que ainda não está muito explorada.

2.2 Sensores

Os sensores usados, como ilustrado na Figura 2, servem para a deteção de veículos nos espaços determinados para estacionamento de forma a ser possível conhecer o atual estado dum determinado lugar. Estes sensores magnéticos estão incorporados numa cápsula resistente que se encontra colada ou mesmo ligeiramente enterrada no alcatrão. Os sensores, alimentados por uma bateria, conseguem atingir uma longevidade de cerca de seis anos e meio, havendo a possibilidade de serem alimentados através de um painel solar, tendo um custo superior e complexidade mais elevada.

Estes dispositivos têm a capacidade de comunicar por LoRa ou por SIGFOX, que são gateways de comunicação, e têm algoritmos de deteção de veículos otimizados. Cada vez que existir uma alteração no campo magnético, entenda-se entrada ou saída de um veículo, será enviado um evento pelos mesmos. Nesta dissertação não será abordado o funcionamento nem as especificações técnicas destes sensores, tendo em conta que só serão utilizados para recolher informação e transmitir a mesma para que a aplicação possa consumir esses dados. Atualmente estes sensores já estão montados em certos lugares de estacionamento e encontram-se prontos para serem utilizados.



Figura 2: Sensor usado para comunicar com a aplicação

2.3 Métodos de Pagamento

Nesta dissertação não foi implementado nenhum método de pagamento. Atualmente está a ser desenvolvido um vertical de pagamentos pela Ubiwhere, que quando estiver pronto será integrado na aplicação desenvolvida no âmbito desta dissertação. No entanto vamos analisar o que existe e o que é preciso ter em conta antes destes serem integrados na aplicação.

A aplicação vai necessitar de uma forma de pagamento de modo a que um utilizador possa usufruir na totalidade de todas as funcionalidades oferecidas. Desta forma o utilizador não precisa de se preocupar em procurar um parquímetro, podendo assim estacionar e efetuar o pagamento durante o percurso do carro até ao destino, enquanto caminha ou usa outro meio de transporte. Caso o utilizador exceda o tempo para o qual efetuou o pagamento, poderá prolongar o mesmo à distância de um clique sem necessitar de se dirigir até ao seu veículo para trocar o bilhete em papel.

Tendo em conta que os pagamentos relacionados com os estacionamento se elevam a pequenos montantes, a maioria dos métodos de pagamento online não compensaria ter estacionamento pagos devido às taxas quase sempre serem maiores que o próprio montante. Nesta situação os micropayments acabam por ser a melhor solução, conseguindo em muitos casos evitar taxas importantes.

O micropayment [23] é uma transação financeira de um montante monetário bastante baixo que geralmente ocorre online. Não existe nenhuma norma que defina um montante máximo até ao qual uma transação é considerada um micropayment, mas sim cada entidade define o seu montante máximo para as mesmas.

2.3.1 Soluções existentes

Atualmente já existem diversas soluções de micropayment implementadas que estão a ser usadas [24], podendo ser utilizadas com diversas tecnologias e de várias formas:

- Chamada com valor acrescentado, sendo que o utilizador recebe um código por mensagem de texto que lhe dá acesso a zonas pagas de uma determinada página web.
- Envio de mensagem de texto com uma determinada palavra, que será taxada com um valor acrescentado.
- Solução Internet+ que consiste em que o micropayment seja acrescentado pela operadora que fornece o serviço de internet ao cliente na emissão da fatura do serviço.
- Solução MPME (Micro Payment Mobile & Enablers) que permite que o montante seja debitado do saldo ou acrescentado à fatura do serviço de telemóvel. Esta solução tem a vantagem de conseguir chegar a todas as pessoas que dispõem de telemóvel (mais de 5 mil milhões em todo mundo [25]).
- Cripto moeda (por exemplo Bitcoin [26]).
- Cartões pré-pagos que são debitados a cada compra.
- PayPal [27], sendo que é preciso ter em conta as taxas acrescentadas.

As soluções apresentadas têm a vantagem de serem acessíveis a todas as pessoas e oferecem segurança. Tendo em conta que se está a lidar com dinheiro é sempre importante conseguir oferecer o máximo de segurança aos utilizadores. Todos estes serviços têm a vantagem de apenas poderem ser utilizados em número limitado (na maior parte é gerado um código que pode ser usado apenas uma vez). Todos estes serviços são confidenciais sendo que mantêm o seu utilizador em total anonimato. Pode-se concluir que são uma forma rápida, fácil e cómoda de efetuar pagamentos, sendo que não há necessidade de haver uma pré-inscrição e conseguindo oferecer uma visão dos gastos que se está a ter.

Estas soluções trazem bastantes vantagens aos fornecedores de serviços, sendo já soluções desenvolvidas e fáceis de incluir em qualquer serviço web. Desta forma o fornecedor consegue oferecer um serviço simples que funciona de forma instantânea. Após o utilizador autorizar o pagamento, o mesmo é debitado diretamente da conta do utilizador. A entidade que está a cobrar o serviço obtém diretamente o valor sem correr o risco de o pagamento ser cancelado pelo utilizador. Comparadas com outras soluções, as taxas existentes neste tipo de serviço são relativamente baixas, às vezes nulas para os clientes. Em relação ao fornecedor de serviço, as taxas raramente são nulas mas podem rondar apenas alguns cêntimos, assim este também consegue restringir os seus gastos tendo em conta que não necessita de criar nenhuma infraestrutura para suportar os pagamentos.

Hoje em dia já existem várias aplicações prontas a serem usadas para efetuar pagamentos online. A Caixa Geral de Depósitos já tem uma aplicação que simula um TPA (Terminal de Pagamento Automático) virtual, o Netcaixa Online [28]. Este serviço pode ser integrado em qualquer solução web simulando um TPA virtual afim que o utilizador possa efetuar pagamentos com o seu cartão, seja débito ou crédito (nacionais e estrangeiros, sendo estes Mastercard ou Visa). Todas as transações são feitas usando o protocolo de segurança 3D Secure [29], que é um protocolo de segurança de pagamentos online de forma a reduzir o risco de fraudes ligando o processo de autorização financeira com a autenticação online. Assim as transações podem ser efetuadas em qualquer parte do mundo sem necessidade de recorrer a outros recursos além do número do cartão que será usado pra efetuar o pagamento. Esta solução é bastante vantajosa sendo que não existem montantes mínimos obrigatórios de pagamento, além disso não é necessário nenhum montante mínimo para a abertura de uma conta na Caixa Geral de Depósitos e existe isenção de comissão de manutenção.

Tanto a Google [30] como a Apple [31] têm as suas aplicações de pagamentos, respetivamente o Android Pay [32] e a Apple Pay [33]. Cada uma das aplicações só está disponível para os dispositivos que tenham os sistemas operativo das mesmas, o que limita o uso dos utilizadores. Tendo em conta que a maioria dos smartphones usados pelas pessoas corre em iOS ou Android, para conseguir abranger todas estas pessoas teria de se integrar ambas as opções na aplicação desenvolvida. Ambas as aplicações foram criadas com o mesmo âmbito, funcionando da mesma forma. Acedendo a aplicação e fazendo o login o utilizador consegue gerir os seus cartões de pagamento, bastando para isso introduzir as informações relativamente aos mesmos, ficando assim a aplicação pronta a ser usada. Estando assim tudo pronto para ser usado, o utilizador pode passar ao próximo passo, o pagamento. Para realizar um pagamento de um bem ou serviço prestado numa loja física, o utilizador só precisa de aproximar o seu dispositivo a um leitor NFC (Near Field Communication), caso o seu dispositivo disponha dessa funcionalidade. Em termos de efetuar um pagamento online tem de ser disponibilizada a opção Android Pay ou Apple Pay, para dispositivos Android ou Apple respetivamente pela entidade que está a cobrar o serviço.

Google Wallet [34] foi desenvolvido pela Google com o âmbito de poder transferir e receber dinheiro de forma simples e cómoda sem ter de pagar taxas ou recorrer a uma entidade financeira. O utilizador consegue armazenar os dados dos seus cartões de pagamento, tal como acontece com o Android Pay e o Apple Pay e assim efetuar o pagamento através dos mesmos. A grande vantagem é esta aplicação ter sido desenvolvida para representar uma carteira virtual, permitindo assim efetuar pagamentos com a mesma ou mesmo receber dinheiro através de transferências. A fim de poder usufruir desse mesmo dinheiro o utilizador associa a sua conta bancária à aplicação, podendo assim transferir o dinheiro para a mesma sempre que o desejar. Até à criação do Android Pay, a Google Wallet também disponibilizava pagamentos por NFC aos seus utilizadores e disponibilizava um cartão físico, como podemos ver na Figura 3, com o qual o utilizador podia efetuar pagamentos e levantamentos tanto em TPA como em ATM (Automated Teller Machine ou simplesmente caixa multibanco) de todos os cartões e contas associadas ao seu Google Wallet.



Figura 3: Google Wallet Card [35]

“Blockchain is far and away an early leader in Bitcoin. They’re actually enabling the entire ecosystem.” [36], é desta forma que o The New York Times [37] destaca o Blockchain [38]. Blockchain é uma base de dados distribuída que é usada para manter uma lista contínua crescente de registos. Cada bloco contém um timestamp e um link para o bloco anterior [39]. Além do The New York Times, também outros prestigiados meios de comunicação como The Telegraph [40], Bloomberg [41], The Wall Street Journal [42], entre outros já realçaram a mesma. Computerworld [43] define o Blockchain como uma estrutura de dados que representa uma entidade de contabilidade financeira ou até mesmo um registo de transações de Bitcoins. De forma a garantir que ninguém comprometa a integridade ou até mesmo a autenticidade das transações, cada uma delas é digitalmente assinada. Uma transação, ou mesmo uma correção de uma transação já existente, provoca a execução de vários algoritmos, que consistem na avaliação e verificação do histórico individual do bloco do Blockchain que é recebido, de forma a averiguar se o próprio histórico e a assinatura são válidos.

2.3.2 Segurança nos pagamentos

A fim de ter uma ideia sobre como funcionam alguns destes processos de pagamento, vão ser analisados os protocolos e trocas de mensagens usadas pelo NetBill [44], um dos sistemas já existentes de micropayments online. Segundo o estudo efetuado por Monica Fengqiu Wu sobre o Micropayment Protocol [45] a autenticação é feita usando uma chave pública baseada em Kerberos [46] de forma a garantir uma autenticação distribuída (PDKA), a cifração é baseada no algoritmo RSA public key e DES (Data Encryption Standard) e finalmente a assinatura é alcançada através do algoritmo de assinatura digital (DSA). Todos estes processos explicados são efetuados por um broker, que serve de intermédio entre o vendedor e o utilizador, o qual representa a entidade financeira que vai fazer chegar o dinheiro de uma conta para a outra.

2.3.3 Ferramentas de desenvolvimento

Existem duas formas de desenvolver aplicações mobile, nativas ou multiplataforma. As aplicações nativas são desenvolvidas para um sistema operativo específico e recorrem às ferramentas dessa plataforma (como por exemplo os SDK que são específicos de cada plataforma). Por outro lado, com o desenvolvimento de aplicações multiplataforma, apesar do código, ou grande parte dele, ser reaproveitado na geração das aplicações para as diferentes plataformas, algumas vezes este tem um desempenho inferior em relação às aplicações nativas. A grande vantagem de desenvolver aplicações multiplataformas é o reaproveitamento de grande parte ou todo o código para todas as plataformas. Para começar vão ser apresentadas formas de desenvolver aplicações nativas em Android e iOS. O desenvolvimento para telefones que correm outros sistemas operativos não vai ser abordado, tendo em conta que detinham apenas 0,7% do mercado dos smartphones no terceiro trimestre de 2016 segundo um estudo revelado pelo IDC [47].

- Android nativo:

Uma aplicação nativa Android pode ser desenvolvida usando o Android Studio ou o Eclipse, entre outros. Android nativo é desenvolvido em Java e é necessário usar o Android SDK. Com a ajuda destas ferramentas torna-se mais fácil criar aplicações que vão ao encontro especificamente de sistemas operativos Android.

- Swift e Objective-C:

Swift e Objective-C são duas linguagens que permitem programar para iOS. Swift foi criado pela Apple para programar especificamente para iOS e OS X. Ambas servem para desenvolver aplicações nativas para iOS, sendo desenvolvidas usando o X-Code da Apple. De forma a desenvolver uma aplicação em qualquer uma destas linguagens é necessário ter uma máquina a correr um sistema operativo OS X, sendo praticamente impossível fazê-lo em Windows ou Linux, sendo esta a maior desvantagem.

Assim sendo se o utilizador optar por uma das soluções apresentadas apenas conseguirá oferecer uma aplicação a um tipo de utilizadores específico. Com as soluções que vão ser apresentadas a seguir é possível desenvolver aplicações multiplataforma de forma a conseguir abranger mais utilizadores. Qualquer uma das soluções permite desenvolver código a fim de criar uma aplicação que vai correr tanto em Android como iOS.

- Xamarin:

Xamarin [48] é uma companhia de software criada em 2011 e adquirida pela Microsoft em 2016. Xamarin permite desenvolver aplicações tanto para iOS, como Android ou até mesmo Windows, o que é classificado como aplicação híbrida. O código é escrito em C# usando .Net e bibliotecas nativas. Segundo o Xamarin, as aplicações são nativas e híbridas ao mesmo tempo, híbridas pelo facto de funcionarem para os vários sistemas operativos e nativas porque apenas 75% do código é partilhado entre os diversos sistemas operativos, sendo que os restantes 25% são considerados UI.

- React-Native:

React-Native é um framework criado pelo Facebook no início de 2015, continuando a evoluir com a ajuda de diversos utilizadores através do repositório do Github [49]. O principal objetivo do React-Native é a reutilização de grande parte do código entre as duas plataformas (iOS e Android). O código é escrito em Javascript de forma a facilitar o deploy para ambos os sistemas operativos.

- Ionic

Ionic [50] é um framework gratuito que foi desenvolvido no final do ano de 2013, visando a criação de aplicações móveis híbridas, usando HTML5 e Javascript para dispositivos correndo iOS7+ e Android 4.4+. Comparado com os frameworks mencionados anteriormente todo o código escrito é usado para Android e iOS, não havendo assim necessidade de ter partes de códigos distintas destinadas a cada sistema operativo.

3. Objetivos e Especificações

3.1 Objetivos

Após o estudo e a análise dos vários artigos consegue-se facilmente concluir que a procura e o aproveitamento dos estacionamentos livres não são sempre evidentes. O foco do desenvolvimento da Smart Parking Mobile Application é resolver a grande parte destes problemas. Esta aplicação tem como objetivo ajudar os condutores a encontrar lugares de estacionamento e os agentes de fiscalização responsáveis a controlarem os mesmos, sem esquecer os técnicos que vão instalar os sensores a fim de os lugares poderem ser usados e caso necessário proceder à manutenção dos mesmos.

Neste sentido vai ser desenvolvida uma aplicação móvel cross-platform que se destina a três tipos de utilizadores, os condutores, os agentes de fiscalização e por fim os técnicos. Vão ser apresentadas as funcionalidades base da mesma e os principais objetivos que foram considerados.

Funcionalidades base:

- Aceder ao mapa em tempo real
- Ver estado de lugares (ocupado ou livre)
- Login/logout

Condutores:

- Registrar
- Editar/atualizar informação pessoal
- Encontrar um lugar livre
- Efetuar pagamento
- Notificações de tempo esgotado
- Prolongar sessão de estacionamento
- Aceder ao histórico de pagamentos

Agentes de fiscalização:

- Registo de pagamentos
- Associação de pagamentos a lugares
- Autuar veículos por falta de pagamento
- Introduzir informação de pagamento manual

Técnicos:

- Consultar sensores no mapa
- Ver estado dos sensores
- Instalar novos sensores
- Editar informação sobre os sensores
- Remover sensores

Vão ser analisados os objetivos mencionados começando pelas funcionalidades base da aplicação. Todos os utilizadores da aplicação têm acesso a um mapa em tempo real, conseguindo obter a sua própria localização no mesmo e ver o estado das várias áreas de estacionamento, que são compostas por conjuntos de lugares incorporados com os nossos sensores.

Um utilizador com o perfil de condutor é um utilizador comum, que vai usar a aplicação de forma a procurar um lugar de estacionamento e pagar o mesmo. O utilizador pode pesquisar por uma localidade de forma a encontrar um lugar perto do destino. O condutor vai diferenciar entre áreas de estacionamento com lugares disponíveis e sem lugares disponíveis, de forma a perceber para onde se deve dirigir. Depois de estacionar o seu veículo numa zona paga o utilizador poderá efetuar o pagamento do lugar que ocupou e ter acesso à gestão do lugar durante o tempo de ocupação.

Um utilizador com perfil de agente de fiscalização pode visualizar o estado de ocupação de cada lugar individual. Para cada lugar pode-se efetuar a verificação do estado de pagamento, registar um pagamento de um bilhete em papel, associar um pagamento efetuado através da aplicação a um lugar, ou até mesmo registar uma infração caso nenhum dos cenários anteriores se verifique.

Por último, um técnico de instalação de sensores vai poder aceder a todos os sensores instalados nas ruas de forma a efetuar alterações aos mesmos, entre remover ou editar informações dos mesmos. Além disso pode instalar novos sensores e proceder ao registo dos mesmos através da aplicação de forma a ficarem imediatamente disponíveis para utilização.

A fim de perceber como estes objetivos foram integrados na aplicação e de os contextualizar com a utilização da aplicação, vão ser apresentadas as user stories.

3.2 User Stories

3.2.1 Perfil Condutor

US.01.01 – Como utilizador não autenticado posso registar-me para poder aceder a todas as funcionalidades disponíveis na aplicação.

- Quando o utilizador abre a aplicação pela primeira vez tem de proceder ao registo a fim de poder usar a aplicação.
- O utilizador preenche todos os campos obrigatórios, tal como o seu primeiro e último nome, email, número de telefone, data de nascimento e uma password à sua escolha com mínimo de seis caracteres. Após todos os campos obrigatórios preenchidos o utilizador submete o seu pedido.
- Em caso de sucesso o utilizador é redirecionado para o ecrã “Login”.
- Em caso de algum campo obrigatório em falta, o utilizador recebe uma mensagem de erro a indicar qual o campo em falta.
- Em caso de utilizar um email que já se encontra em uso o utilizador recebe uma mensagem de erro com essa informação.
- Em caso de outro erro, o mesmo será mostrado ao utilizador.

US.01.02 – Como utilizador não autenticado posso autenticar-me na aplicação de forma a ter acesso à aplicação.

- Quando o utilizador abre a aplicação procede ao login preenchendo os campos com as suas credenciais e submete o pedido para aceder à aplicação.
- Em caso de sucesso o utilizador é redirecionado para o ecrã principal da aplicação.
- O login permanece até ser feito o logout.
- Caso o utilizador introduza um email não registado, um texto sem formato de email no campo destinado ao mesmo ou uma password errada, o utilizador recebe uma mensagem de erro com a mesma informação.

US.01.03 – Como utilizador autenticado posso consultar os meus dados.

- O utilizador acede ao seu perfil e é direcionado para o ecrã com os seus dados pessoais.
- O utilizador pode consultar o seu primeiro nome, último nome, email e restantes dados registados.
- Para retroceder ao ecrã inicial o utilizador pode cancelar a operação ou usar o botão físico para regressar.
- O utilizador vai ser redirecionado para o ecrã principal.

US.01.04 – Como utilizador autenticado posso aceder aos meus dados de forma a alterar os mesmos.

- O utilizador acede ao seu perfil e é direcionado para o ecrã com os seus dados pessoais.
- O utilizador pode alterar os mesmos e submeter as alterações.
- O utilizador pode cancelar a operação ou retroceder com o botão físico.
- Em caso de os dados serem editados com sucesso, o utilizador obtém uma mensagem a informar que as alterações foram efetuadas e é redirecionado ao ecrã principal.
- Em caso de erro o utilizador é notificado do mesmo e as alterações serão descartadas.

US.01.05 – Como utilizador autenticado posso consultar os dados sobre os meus veículos.

- No ecrã vão aparecer todos os veículos registados pelo utilizador e a respetiva informação dos mesmos.
- Em caso de não haver veículos registados o utilizador só vai ver a opção para adicionar novas entradas.

US.01.06 – Como utilizador autenticado posso aceder aos meus veículos para adicionar novas entradas.

- O utilizador acede aos seus veículos e é direcionado para o ecrã de gestão de veículos.
- No ecrã vão aparecer todos os veículos registados e a uma opção de registo de um novo veículo.
- O utilizador vai selecionar a opção de adicionar e introduzir os dados do veículo e submeter os mesmos.
- Em caso de sucesso o veículo registado vai aparecer logo na lista dos seus veículos.
- Em caso de insucesso ou cancelamento, o veículo não vai aparecer na lista dos veículos nem será adicionado ao sistema.

US.01.07 – Como utilizador autenticado posso remover um veículo da lista para atualizar a minha lista de veículos.

- O utilizador acede aos seus veículos e é direcionado para o ecrã de gestão de veículos.
- No ecrã vão aparecer todos os veículos registados pelo utilizador.
- O utilizador vai remover o veículo que selecionar.
- O utilizador vai confirmar a sua escolha.
- Em caso de sucesso o veículo vai desaparecer da lista e será removido do sistema.
- Em caso de insucesso ou cancelamento, o veículo não vai desaparecer da lista de veículos, nem será removido do sistema.

US.01.08 – Como utilizador autenticado posso pagar o meu estacionamento de forma a gerir o mesmo.

- O utilizador acede ao ecrã de gestão de pagamentos.
- O utilizador define o tempo que pretende ficar estacionado.
- O utilizador pode definir um alarme, a fim de obter um aviso quando o seu tempo se estiver a esgotar e outro quando o tempo estiver esgotado.
- O utilizador seleciona a matrícula do carro que já registou anteriormente, para o qual quer efetuar o pagamento e valida o pagamento e a área na qual se encontra estacionado.
- Em caso de sucesso o utilizador obtém uma mensagem a informar que o pagamento foi efetuado com sucesso e é redirecionado para o histórico a fim de visualizar o pagamento.
- Em caso de não ter matrículas registadas no sistema o utilizador recebe uma mensagem para efetuar o registo com a opção de ser redirecionado para o ecrã de gestão de veículos.
- Em caso de o utilizador não selecionar um área ou um tempo válido recebe uma mensagem de erro a avisar sobre o sucedido.
- Em caso de já existir um pagamento a decorrer com a matrícula selecionada o utilizador recebe uma mensagem a avisar sobre o mesmo e a informar que o mesmo pode ser consultado no seu histórico.
- Em caso que ocorra outro erro, este será mostrado ao utilizador.

US.01.09 – Como utilizador autenticado posso aceder ao meu histórico para consultar os pagamentos anteriores e/ou a decorrer.

- O utilizador acede ao histórico, onde vai ver os pagamentos que ainda estão a decorrer seguidos dos pagamentos que já expiraram.
- Em relação a cada pagamento o utilizador pode consultar o montante que pagou e quantas vezes foram prolongados.

US.01.10 – Como utilizador autenticado posso aceder a um pagamento que está a decorrer para prolongar o tempo de estacionamento.

- O utilizador acede ao histórico e vai ver os pagamentos que ainda estão a decorrer.
- Associado ao pagamento vai estar a indicação de hora de fim do mesmo.
- O utilizador pode selecionar a opção de tempo e indicar o tempo adicional que pretende ficar.
- O utilizador terá de confirmar a operação.

- Em caso de optar por prolongar o seu pagamento, a hora de fim associada ao pagamento será então prolongada, adicionando o tempo que o utilizador indicou.
- Em caso de sucesso o utilizador recebe uma notificação sobre o mesmo e no seu histórico será mostrada a nova hora de fim associada ao pagamento que está a decorrer.
- Em caso de um utilizador não introduzir um tempo válido receberá a informação do mesmo e o tempo não será alterado.

US.01.11 – Como utilizador autenticado posso aceder ao mapa para consultar os estacionamentos.

- No ecrã principal o utilizador vai ver o mapa e as áreas de estacionamento representadas de acordo com o seu estado de ocupação.

US.01.12 – Como utilizador autenticado posso procurar um destino e obter a rota até ao mesmo.

- O utilizador vai aceder à opção de pesquisa.
- O utilizador pode escrever a morada do local de destino ou seleccionar uma área de estacionamento como ponto de destino.
- O utilizador irá submeter a sua pesquisa e irá obter a rota de carro do ponto de partida até ao estacionamento mais perto do seu destino e a rota a pé do respetivo estacionamento até ao ponto destino.
- Caso o ponto de destino seja uma área de estacionamento, o utilizador irá só obter a rota até à mesma.
- Em caso de sucesso o utilizador poderá consultar a área de estacionamento seleccionada a fim de navegar para a mesma.
- Em caso de insucesso por não encontrar nenhuma área de estacionamento com lugares vagos ou por não existir nenhuma área de estacionamento perto do destino introduzido, o utilizador irá obter essa informação.

US.01.13 – Como utilizador autenticado posso definir filtros para a pesquisa de estacionamento.

- O utilizador vai aceder ao mapa.
- No mapa irá seleccionar a opção de filtros.
- Desta forma o utilizador pode definir um intervalo de preço dentro do qual pretende visualizar áreas de estacionamento.
- O utilizador pode também definir uma distância máxima do ponto de destino até à área de estacionamento.

US.01.14 – Como utilizador autenticado posso fazer logout da aplicação de forma a sair da mesma.

- O utilizador acede à opção logout e é direccionado para o primeiro ecrã.
- O utilizador deverá voltar a ser um utilizador não autenticado.

3.2.2 Perfil Agente de fiscalização

US.02.01 – Como utilizador não autenticado posso autenticar-me na aplicação de forma a ter acesso à aplicação.

- Quando o utilizador abre a aplicação procede ao login preenchendo os campos com as suas credenciais e submete o pedido para aceder à aplicação.
- Em caso de sucesso o utilizador é redireccionado para o ecrã principal da aplicação.
- O login permanece até ser feito o logout.
- Caso o utilizador introduza um email não registado, um texto sem formato de email no campo destinado ao mesmo ou uma password errada, o utilizador recebe uma mensagem de erro com a mesma informação.

US.02.02 – Como utilizador autenticado posso aceder os meus dados para consultar os mesmos.

- O utilizador acede ao seu perfil e é direcionado para o ecrã com os seus dados pessoais.
- O utilizador pode consultar o seu primeiro nome, último nome, email e restantes dados registados.
- Para retroceder ao ecrã inicial o utilizador pode cancelar a operação ou usar o botão físico para regressar.
- O utilizador vai ser redirecionado para o ecrã principal.

US.02.03 – Como utilizador autenticado posso aceder aos meus dados e alterá-los.

- O utilizador acede ao seu perfil e é direcionado para o ecrã com os seus dados pessoais.
- O utilizador pode alterar os mesmos e submeter as alterações.
- O utilizador pode cancelar a operação ou retroceder com o botão físico.
- Em caso de os dados serem editados com sucesso, o utilizador obtém uma mensagem a informar que as alterações foram efetuadas e é redirecionado para o ecrã principal.
- Em caso de erro o utilizador é notificado e as alterações serão descartadas.

US.02.04 – Como utilizador posso visualizar os estados dos diversos lugares.

- Como agente de fiscalização ao aceder ao mapa vão ser apresentados os diversos lugares representados com as diversas cores consoante os seus respetivos estados.

US.02.05 – Como utilizador posso registar um pagamento de um bilhete.

- O utilizador vai selecionar o lugar desejado e a fim de proceder à verificação do mesmo.
- O utilizador irá introduzir a matrícula do veículo e caso o mesmo não se encontre registado na aplicação terá de ser feito o seu registo.
- O utilizador pode introduzir os dados do bilhete a fim de registar o pagamento na aplicação e associar o mesmo ao estacionamento.
- O utilizador também pode ler o código QR do bilhete a fim de obter os dados do pagamento e assim não necessita de os introduzir.
- Caso seja bem-sucedido o estado do estacionamento muda.

US.02.06 – Como utilizador posso editar um pagamento de um bilhete em papel.

- O utilizador ao verificar que se enganou nos dados de um pagamento que foi registado por um agente de fiscalização pode aceder ao mesmo a fim de editar os dados do mesmo.
- O utilizador irá seleccionar o lugar que está a verificar e obter a informação do pagamento, caso o pagamento registado seja de um bilhete em papel, o agente poderá editar os dados do mesmo.
- Em caso de sucesso o utilizador é redirecionado para o mapa, para continuar o seu trabalho.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.07 – Como utilizador posso apagar um pagamento de um bilhete em papel.

- O utilizador pode aceder a um pagamento que foi registado por um agente de fiscalização a fim de apagar o registo.
- O utilizador irá seleccionar o lugar e obter a informação do pagamento associado ao mesmo, e caso o pagamento registado seja de um bilhete em papel, o agente poderá apagar os dados do mesmo.
- Em caso de sucesso o utilizador é redirecionado para o mapa, para continuar o seu trabalho.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.08 – Como utilizador posso associar um pagamento efetuado por um condutor a um lugar.

- O utilizador irá seleccionar o lugar que deseja verificar.
- O utilizador irá introduzir a matrícula do veículo e obter a informação do pagamento, não estando este pagamento associado a nenhum lugar o utilizador poderá associar o mesmo ao lugar que está a verificar.
- Em caso de sucesso o utilizador é redirecionado para o mapa, para continuar o seu trabalho.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.09 – Como utilizador posso mudar o lugar associado a um pagamento.

- O utilizador irá selecionar o lugar que deseja verificar.
- O utilizador irá introduzir a matrícula do veículo e obter a informação do pagamento, estando este pagamento associado a outro lugar, o condutor pode ter trocado de lugar mantendo o pagamento anterior, tendo sido o pagamento efetuado através da aplicação ou através de um parquímetro.
- O utilizador terá ao seu dispor a opção de associar esse pagamento ao lugar que está a verificar.
- Em caso de sucesso o utilizador é redirecionado para o mapa para continuar o seu trabalho e a cor do lugar muda.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.10 – Como utilizador posso desassociar um pagamento e assim fica o lugar sem pagamento associado.

- O utilizador irá selecionar o lugar a fim de obter a informação do pagamento.
- O utilizador poderá desassociar o mesmo do lugar.
- Em caso de sucesso o utilizador é redirecionado para o mapa, para continuar o seu trabalho.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.11 – Como utilizador posso gerir um estacionamento que não está pago de forma a autuar um veículo.

- O utilizador ao verificar um lugar que está por verificar e não disponha de bilhete em papel pode selecionar o lugar para proceder à verificação do mesmo.
- O utilizador irá introduzir a matrícula do veículo e caso o mesmo não se encontre registado na aplicação terá de ser feito o registo do mesmo.
- Não encontrando nenhum pagamento associado ao veículo o utilizador irá selecionar a opção de autuar a fim de registar a infração por falta de pagamento.
- Em caso de sucesso o utilizador é redirecionado para o mapa para continuar o seu trabalho e o lugar muda de cor.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.12 – Como utilizador posso gerir um estacionamento para corrigir o lugar onde se encontra um carro já autuado.

- O utilizador ao verificar um lugar que está por verificar e não disponha de bilhete em papel pode seleccionar o lugar para proceder à verificação do mesmo.
- O utilizador irá introduzir a matrícula do veículo e obter a informação relativa à coima já emitida.
- Ao verificar que o veículo mudou de lugar e continua sem pagamento, o utilizador irá associar o infração ao lugar onde se encontra o veículo no respetivo momento.
- Em caso de sucesso o utilizador é redirecionado para o mapa, para continuar o seu trabalho.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.13 – Como utilizador posso apagar uma infração de forma a ser eliminado o registo.

- O utilizador pode seleccionar o lugar de forma a eliminar a coima associada ao mesma.
- Em caso de sucesso o utilizador é redirecionado para o mapa, para continuar o seu trabalho.
- Em caso de insucesso o utilizador irá obter uma mensagem de erro a informar do sucedido.

US.02.14 – Como utilizador autenticado posso fazer logout da aplicação de forma a sair da mesma.

- O utilizador acede à opção logout e é direccionado para o primeiro ecrã.
- O utilizador deverá voltar a ser um utilizador não autenticado.

3.2.3 Perfil Técnico de instalação de sensores

US.03.01 – Como utilizador não autenticado posso autenticar-me na aplicação de forma a ter acesso à aplicação.

- Quando o utilizador abre a aplicação procede ao login preenchendo os campos com as suas credenciais e submete o pedido para aceder à aplicação.
- Em caso de sucesso o utilizador é redirecionado para o ecrã principal da aplicação.
- O login permanece até ser feito o logout.
- Caso o utilizador introduza um email não registado, um texto sem formato de email no campo destinado ao mesmo ou uma password errada, o utilizador recebe uma mensagem de erro com a mesma informação.

US.03.02 – Como utilizador autenticado posso aceder os meus dados para consultar os mesmos.

- O utilizador acede ao seu perfil e é direcionado para o ecrã com os seus dados pessoais.
- O utilizador pode consultar o seu primeiro nome, último nome, email e restantes dados registados.
- Para retroceder ao ecrã inicial o utilizador pode cancelar a operação ou usar o botão físico para regressar.
- O utilizador vai ser redirecionado para o ecrã principal.

US.03.03 – Como utilizador autenticado posso aceder aos meus dados de forma a alterar os mesmos.

- O utilizador acede ao seu perfil é direcionado para o ecrã com os seus dados pessoais.
- O utilizador pode alterar os mesmos e submeter as alterações.
- O utilizador pode cancelar a operação ou retroceder com o botão físico.
- Em caso de os dados serem editados com sucesso, o utilizador obtém uma mensagem a informar que as alterações foram efetuadas e é redirecionado ao ecrã principal.
- Em caso de erro o utilizador é notificado do mesmo e as alterações serão descartadas.

US.03.04 – Como utilizador posso adicionar um sensor na sequência de um processo de instalação de sensores.

- O utilizador lê o identificador do sensor através do código QR e introduz a localização do mesmo.
- Em caso de sucesso, o utilizador vai ver o sensor identificado no mapa e o estado passa de “ready” para “active”.
- Em caso de erro irá aparecer a mensagem de erro.

US.03.05 – Como utilizador posso ver o estado dos sensores.

- O utilizador acede a lista de sensores.
- Em caso de sucesso serão listados todos os sensores registados.

US.03.06 – Como utilizador posso mudar o estado de um sensor.

- O utilizador acede a lista de sensores.
- Na lista de sensores o utilizador pode escolher o sensor que deseja gerir ou então pesquisar pelo mesmo lendo o código QR do sensor.
- O utilizador muda o estado e submete o pedido.
- Em caso de sucesso o estado do sensor na lista de sensores é alterado.
- Em caso de insucesso, a estado do sensor mantém se inalterada.

US.03.07 – Como utilizador posso alterar a localização de um sensor.

- O utilizador acede a lista de sensores.
- Na lista de sensores o utilizador pode escolher o sensor que deseja gerir ou então pesquisar pelo mesmo lendo o código QR do sensor.
- O utilizador muda a localização do sensor e submete o pedido.
- Em caso de sucesso a localização associado ao sensor na lista de sensores é alterada.
- Em caso de insucesso, a localização mantém-se.

US.03.08 – Como utilizador autenticado posso fazer logout da aplicação de forma a sair da mesma.

- O utilizador acede à opção logout e é direcionado para o primeiro ecrã.
- O utilizador deverá voltar a ser um utilizador não autenticado.

4. Definição da Arquitetura do Sistema

4.1 Backend

Atualmente já existe uma arquitetura implementada que define todo o processo de como a informação chega dos sensores e é tratada até chegar à aplicação. Como se pode verificar na Figura 4, olhando da esquerda para a direita consegue-se perceber que os sensores se encontram no início e a aplicação desenvolvida no fim desta arquitetura.

De forma a perceber todo este funcionamento, vai ser feita uma breve explicação sobre o mesmo. Atualmente já existem sensores instalados nos estacionamentos, sendo que o sistema de informação já existente vai consumir a informação gerada pelos sensores. A fim de conseguir receber e processar as mensagens recebidas de forma segura está a ser usado o Meshblu [51], sistema baseado em nuvem que é usado para comunicações entre sensores e a API, de forma a enviar alterações de estados em tempo real via WebSocket. Como ilustra a Figura 5 consegue-se perceber que o Meshblu recebe mensagens de sensores devidamente autorizados e faz o reencaminhamento para o próximo serviço. De seguida é usado o RabbitMQ [52], software de código aberto que implementa o Advanced Message Queuing Protocol (AMQP), onde está um sistema de filas que passa as mensagens para as camadas seguintes. Neste ponto entra o Celery [53], que é um sistema implementado em Python [54] que dispara “workers” sempre que o RabbitMQ recebe novas mensagens e guarda as mesmas numa base de dados do backend, de forma a garantir que a base de dados está sempre atualizada. Finalmente para conseguir ligar isto tudo com a aplicação é usada uma API RESTful que fornece toda esta informação para poder ser consumida.

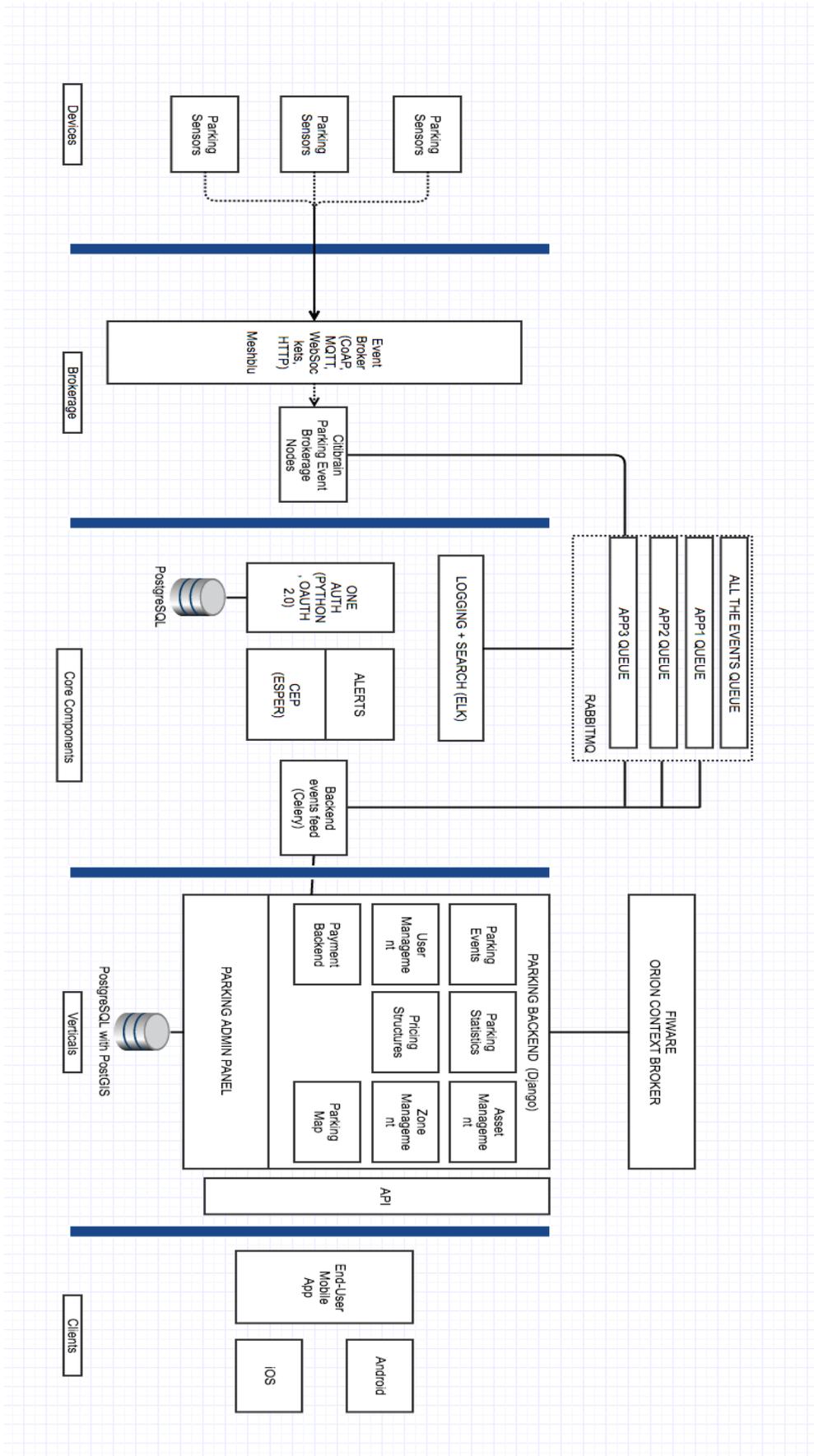


Figura 4: Arquitetura do sistema

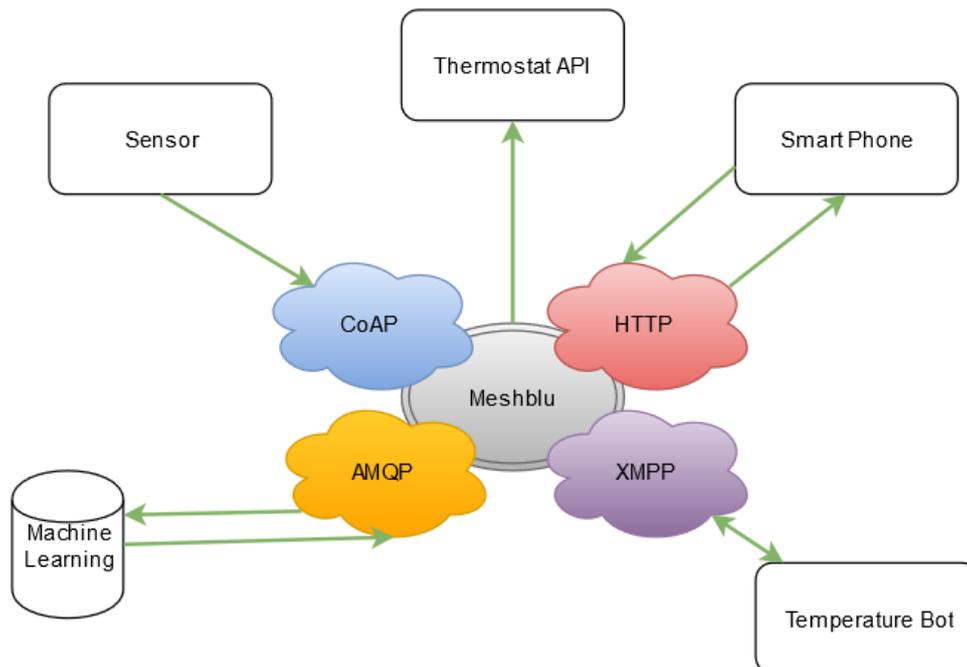


Figura 5: Funcionamento do Meshblu[51]

4.1.1 API RESTful

A plataforma de backend do Smart Parking disponibiliza uma API RESTful, que é consumida pela aplicação desenvolvida no âmbito desta dissertação. Este backend processa e armazena toda a informação recebida dos sensores de estacionamento em tempo real, permitindo a API aceder aos dados sobre os eventos de estacionamento. A API disponibiliza ainda outra informação complementar, relativa às áreas de estacionamento e respetivos horários de funcionamento, preços, informação sobre sensores, gateways, etc.

Todos os acessos à API são controlados usando um servidor que implementa OAuth 2.0 [55]. Existe um sistema de roles no qual se baseia a autorização dos diferentes tipos de utilizadores disponíveis na aplicação, permitindo especificar diferentes níveis de acesso à informação disponível.

Toda a informação em relação aos pedidos está documentada usando a framework Swagger [56]. Todos os pedidos e respostas a pedidos foram tratados em formato JSON. Assim sendo, cada utilizador consegue receber e enviar as informações necessárias para o backoffice de forma a garantir que mantém sempre os dados atualizados e que todas as alterações efetuadas são guardadas.

Auth_Users	
Citibrain users	+ Add Change
Events	
Park events	+ Add Change
Infractions	
Infractions	+ Add Change
Occupancy	
Occupancys	+ Add Change
Payments	+ Add Change
Spots	
Parking areas	+ Add Change
Parking spots	+ Add Change
Vehicles	
Vehicles	+ Add Change

Figura 6: APIs dos backend

4.1.2 OAuth2.0

Outra das vertentes que nos vai interessar será o OAuth2.0. O OAuth2.0 não é um protocolo de autenticação mas sim um protocolo que é usado pelo protocolo de autenticação para garantir certas autorizações dentro do sistema. Neste caso o OAuth2.0 está a ser usado de forma a garantir que a aplicação consiga aceder à API que detém a informação dos sensores.

Através do OAuth2.0 quando um utilizador faz o login na aplicação, é feito um pedido de forma a obter um token que tem um tempo de validade associado a si. Graças a este token são fornecidas as autorizações que um utilizador necessita de forma a interagir com o backend, tendo em conta que relativamente a cada perfil de utilizador serão necessárias autorizações diferentes para aceder a diferentes pedidos nas várias APIs. Desta forma garante-se que um utilizador

normal ao entrar na aplicação não vai conseguir aceder a dados de outros utilizadores nem poder fazer alterações significativas nos registos.

4.1.3 WebSockets

De forma a garantir que toda a informação chega em tempo real aos dispositivos móveis dos utilizadores do Smart Parking Mobile Application, é criada uma ligação através de WebSocket. Atualmente toda esta estrutura já existe, sendo necessário então integrar na aplicação uma ligação ao WebSocket existente a fim de receber todas as alterações de estados dos sensores. Assim sendo cada vez que um carro estaciona ou sai de um determinado lugar que tenha um sensor instalado, o sensor vai enviar uma alteração de estado. Esta alteração de estado vai ser enviada através do Meshblu por um canal WebSocket. A aplicação vai fazer uma ligação ao Meshblu e vai ficar a escuta de mensagens. Quando a aplicação recebe uma mensagem do Meshblu, vai ver qual o lugar que lhe está associado e vai atualizar o estado desse lugar na aplicação.

4.2 Frontend

Tal como relatado no estado de arte existem vários frameworks disponíveis para criar uma aplicação móvel. Com o Smart Parking pretende-se desenvolver uma aplicação móvel que poderá ser usada no dia-a-dia e chegar ao maior número possível de utilizadores. Desta forma será mais útil criar uma aplicação híbrida. Analisando as soluções existentes o Ionic destaca-se por conseguir usar todo o código escrito para iOS e Android, sem ser necessário fazer ajustes consoante o sistema operativo. Ionic destaca-se também por ser um framework gratuito, e oferece muitas bibliotecas externas fáceis de instalar à distância de um simples comando. As bibliotecas usadas podem ser alteradas ou ajustadas consoante as necessidades, sendo possível aceder ao código das mesmas de forma a fazer alterações, sejam elas grandes ou apenas de design.

5. Desenvolvimento da Solução

Neste capítulo vão ser explicados todos os aspetos de desenvolvimento e bibliotecas usadas. Vai ser também explicado como foi desenvolvida a aplicação de forma a chegar ao produto final e como foram usadas as bibliotecas.

5.1 Requisitos de desenvolvimento

5.1.1 Ambiente de Desenvolvimento

A aplicação foi desenvolvida no sistema operativo Windows10 e foi necessário instalar primeiro o Node.js [57] e o Cordova [58] de forma a correr o Ionic. Este processo está documentado no site do Ionic [59]. Desta forma foi criada a aplicação de forma híbrida, facilitando o deploy para qualquer um dos sistemas operativos móveis com um simples comando. O desenvolvimento do código foi feito usando o IDE da JetBrains [60], o WebStorm 2016 [61]. Todas as bibliotecas foram instaladas usando o bower [62]. A aplicação está destinada a smartphones que correm Android ou iOS.

5.1.2 Planeamento de Projeto e Controlo de Versões

Para organizar um plano de trabalho e definir metas foi usado o Gitlab [63]. Como representado na Figura 7 foram especificadas as tarefas a fazer, consoante o estado de uma tarefa, em desenvolvimento, em teste ou finalizada, a tarefa foi colocada numa coluna específica de forma a saber o que falta fazer em relação a cada tarefa.

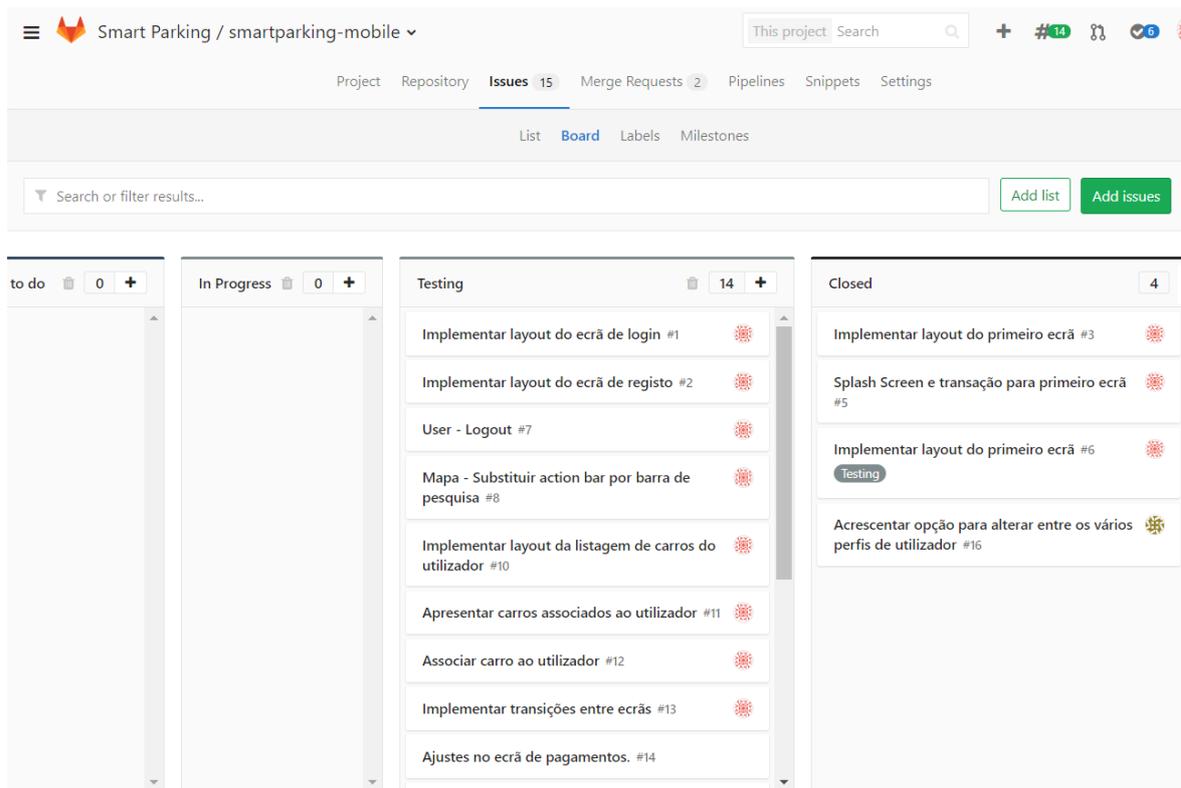


Figura 7: Tarefas do Gitlab

De forma a garantir uma cópia de segurança de todo o código e a manter versões estáveis foi usado o repositório do Gitlab. Com a ajuda deste repositório foi guardado todo o código desenvolvido e foram criadas versões de controlo.

Smart Parking / smartparking-mobile

This project Search

Project **Repository** Issues 15 Merge Requests 2 Pipelines Snippets Settings

Files **Commits** Branches Tags Contributors Graph Compare Charts

smartparking-mobile [View open merge request](#) [Filter by commit message](#)

29 Mar, 2017 3 commits

- inicialização de aplicação**
Leonardo Pinheiro committed 2 months ago [a8481ea9](#) [Browse Files](#)
- refs #2, #7: popup com erros de registo, logout implementado**
Leonardo Pinheiro committed 2 months ago [cea4d959](#) [Browse Files](#)
- screenSize fixed**
Leonardo Pinheiro committed 2 months ago [07dcd925](#) [Browse Files](#)

28 Mar, 2017 5 commits

- bugs corrigidos**
Leonardo Pinheiro committed 2 months ago [ac546e46](#) [Browse Files](#)
- refs #1, #2: Hardware Button fixed, map loadingscreen added**
Leonardo Pinheiro committed 2 months ago [6ac31dd5](#) [Browse Files](#)
- screen size ajust with scroll**
Leonardo Pinheiro committed 2 months ago [21d7b51a](#) [Browse Files](#)
- path to icons added**
Leonardo Pinheiro committed 2 months ago [19e43629](#) [Browse Files](#)

23 Mar, 2017 1 commit

- refs #5: edição de splash screen , ...**
Leonardo Pinheiro committed 2 months ago [141bf2c9](#) [Browse Files](#)

22 Mar, 2017 3 commits

- Merge branch 'devLeo' of ssh://gitlab.ubiwhere.com:2223/citibrain-smart-parking/...**
Leonardo Pinheiro committed 2 months ago [44833403](#) [Browse Files](#)

Figura 8: Controlo de versões do Gitlab

5.2 Backend

Como já foi referido no capítulo 4.1.1, todos os dados usados pela aplicação estão armazenados no backend. De forma a garantir uma estrutura sólida, cada um dos dados tem um cabeçalho atribuído a si a fim de se conseguir saber a que campo se refere.

- Users – serviço responsável por guardar os dados dos utilizadores. Quando é feito o registo são guardados todas as informações dos utilizadores e ao fazer login é feito um pedido de forma a obter as informações do próprio utilizador. Foi disponibilizado um POST e GET de todos os utilizadores registados, e usando a chave primária de um utilizador foi disponibilizado um GET, PUT, PATCH e DELETE.

users

POST	/api/users/
GET	/api/users/
PUT	/api/users/{pk}/
GET	/api/users/{pk}/
PATCH	/api/users/{pk}/
DELETE	/api/users/{pk}/

```
{
  "email": "lpinheiro@ubiwhere.pt",
  "username": "lpinheiro@ubiwhere.pt",
  "password": "123456",
  "first_name": "Leonardo",
  "last_name": "Pinheiro",
  "metadata": {
    "phone_number": "911234567",
    "birth_date": "1990-09-20T00:00:00.000000Z",
    "phone_code": "+351",
    "nationality": "PT"
  }
}
```

Figura 9: Users Api e JSON Object

- **Parking_events** – serviço responsável por guardar os dados dos eventos de estacionamento. Tendo em conta que do lado do utilizador vão ser lidos os eventos que ocorreram, vai ser usado um GET para obter todos os eventos ou um GET usando a chave primária para obter um evento específico.

parking_events

GET	/api/parking_events/
GET	/api/parking_events/{pk}/

```
{
  "id": 81178,
  "asset": "30-5097",
  "start_timestamp": "2017-05-26T09:05:19Z",
  "end_timestamp": null,
  "is_infraction": true,
  "payment_id": null,
  "notification": null,
  "user": null,
  "parking_spot": 42
}
```

Figura 10: Parking_events Api e JSON Object

- **Infractions** – serviço usado por um agente de fiscalização de forma a gerir as infrações cometidas pelos condutores. É disponibilizado o método de POST para criar e GET para obter todas as infrações cometidas. De forma a gerir uma infração individual, usando a sua chave primária, são disponibilizados os métodos GET, PUT, PATCH e DELETE.

infractions

POST	/api/infractions/
GET	/api/infractions/
PUT	/api/infractions/{pk}/
GET	/api/infractions/{pk}/
PATCH	/api/infractions/{pk}/
DELETE	/api/infractions/{pk}/

```
{
  "id": 84,
  "notes": "Parking",
  "severity": 1,
  "status": 0,
  "fine": 1,
  "type": "Parking",
  "creation_date": "2017-06-01T08:50:59.472538Z",
  "vehicle": 131,
  "user": null,
  "park_event": null,
  "parking_spot": 50,
  "parking_area": 1
}
```

Figura 11: Infractions Api e JSON Object

- Occupancies – método utilizado para disponibilizar toda a informação sobre a ocupação de um lugar de estacionamento. Sempre que um utilizador quiser efetuar um pagamento, o mesmo será associado a uma ocupação, todos os prolongamentos também vão ser associados ao mesmo evento. De forma a gerir todas as ocupações é disponibilizado um POST e um GET, e para uma ocupação individual usando a chave primária, é fornecido um GET, PUT, PATCH e DELETE.

occupancies

POST	/api/occupancies/
GET	/api/occupancies/
PUT	/api/occupancies/{pk}/
GET	/api/occupancies/{pk}/
PATCH	/api/occupancies/{pk}/
DELETE	/api/occupancies/{pk}/

```
{
  "id": 98,
  "extensions": {
    "length": 6,
    "data": {
      "0": "2017-05-26T11:10:41.652000Z",
      "1": "2017-05-26T11:15:41.652000Z",
      "2": "2017-05-26T11:20:41.652000Z",
      "3": "2017-05-26T11:30:41.652000Z",
      "4": "2017-05-26T11:35:41.652000Z",
      "5": "2017-05-26T11:40:41.652000Z"
    }
  },
  "start_time": "2017-05-26T11:05:41Z",
  "end_time": "2017-05-26T11:27:19Z",
  "control_time": null,
  "vehicle": 131,
  "spot": null,
  "parking_area": 1
}
```

Figura 12: Occupancies Api e JSON Object

- Payments – todos os pagamentos são registados usando este serviço, sejam pagamentos efetuados através da aplicação, ou pagamentos feitos comprando um bilhete numa máquina e depois registados por um agente de fiscalização. O tipo de pagamento serve para definir essa diferença. Foram disponibilizados POST e GET de forma a gerir todos os pagamentos. Para os registos individuais é disponibilizado um GET e DELETE.

payments

POST	/api/payments/
GET	/api/payments/
GET	/api/payments/{pk}/
DELETE	/api/payments/{pk}/

```
{
  "id": 4,
  "timestamp": "2017-05-26T11:05:41Z",
  "amount": 0.25,
  "currency": "€",
  "payment_type": "mobileApp",
  "payment_method": "virtualCredit",
  "occupancy": 98
}
```

Figura 13: Payments Api e JSON Object

- **Parking_spots** – serviço responsável por guardar todos os lugares de estacionamento no backend. Este serviço é usado sempre que a aplicação é iniciada por um agente de fiscalização de forma a obter todos os lugares de estacionamento existentes. Para que os técnicos possam criar novos lugares e registar os mesmos é disponibilizado o método POST, para obter todos os lugares, é fornecido um GET. De forma a gerir cada lugar individual pode ser usado um GET, PUT, PATCH e DELETE. Desta forma é possível corrigir qualquer erro relacionado com um lugar ou é mesmo eliminar o mesmo se por algum motivo for removido da lista de lugares.

parking_spots

POST	/api/parking_spots/
GET	/api/parking_spots/
PUT	/api/parking_spots/{pk}/
GET	/api/parking_spots/{pk}/
PATCH	/api/parking_spots/{pk}/
DELETE	/api/parking_spots/{pk}/

```
{
  "id": 40,
  "occupied": false,
  "last_state_change": "2017-04-20T14:11:54Z",
  "asset": "30-5080",
  "location": {
    "type": "Point",
    "coordinates": [
      -8.6357246,
      40.637412599999976
    ]
  },
  "parking_area": 1
}
```

Figura 14: Parking_spots Api e JSON Object

- **Parking_areas** – cada área de estacionamento é composta por vários parking spots e este serviço é usado para os utilizadores poderem localizar as áreas de estacionamento onde podem usar a Smart Parking Mobile Application. Quando a aplicação é iniciada por um condutor, é feito um GET por forma a obter todas as áreas de estacionamento onde pode usar a aplicação, seja para se dirigir ou para efetuar o pagamento da mesma. Na parte do backend, as áreas de estacionamento podem ser criadas, editadas ou até mesmo apagadas, usando POST, PUT, PATCH e DELETE.

Usando a chave primária associada ao veículo é dada a possibilidade de fazer obter um veículo individual (GET), fazer alterações (PUT e PATCH) e mesmo apagar a entrada (DELETE).

vehicles

POST	/api/vehicles/
GET	/api/vehicles/
PUT	/api/vehicles/{pk}/
GET	/api/vehicles/{pk}/
PATCH	/api/vehicles/{pk}/
DELETE	/api/vehicles/{pk}/

```
{
  "id": 131,
  "infractions": [
    84
  ],
  "license_plate": "AA-11-22",
  "description": "2016-12-25",
  "category": null,
  "type": null,
  "brand": "BMW",
  "model": "M3",
  "color": "white",
  "owner": 40
}
```

Figura 16: Vehicles Api e JSON Object

- Parking_spots_distances – serviço usado para obter os lugares próximos da localidade pesquisada. É fornecido um GET, com o qual o utilizador obtém todos os lugares na proximidade das coordenadas do ponto fornecido como argumento. Os resultados são apresentados por ordem de menor distância até ao ponto. Assim sendo é usado a área ao qual pertence o ponto de forma a obter a rota.

parking_spots_distances

GET /api/parking_spots_distances/

```
{
  "results": [
    {
      "id": 42,
      "occupied": true,
      "last_state_change": "2017-05-26T09:05:19Z",
      "asset": "30-5097",
      "location": {
        "type": "Point",
        "coordinates": [
          -8.635595853967288,
          40.6373067599409
        ]
      },
      "parking_area": 1
    },
    {
      "id": 40,
      "occupied": false,
      "last_state_change": "2017-04-20T14:11:54Z",
      "asset": "30-5080",
      "location": {
        "type": "Point",
        "coordinates": [
          -8.6357246,
          40.637412599999976
        ]
      },
      "parking_area": 1
    }
  ]
}
```

Figura 17: Parking_spots_distances Api e JSON Object

- Routing – serviço usado para desenhar as rotas no mapa. É sempre feito o pedido de rota do ponto de partida até à área de estacionamento e caso o destino seja diferente da área de estacionamento, então é feito um novo pedido da área de estacionamento até ao destino. É usado um GET, e são passados dois argumentos em cada pedido, o start_point, que representa as coordenadas do ponto de partida e o end_point, as coordenadas do destino.

routing

GET /api/routing/

```
{
  "route": "q{olAlwonOuBiA{C_EqCkEiDkFqCkEy@wI_AwIi@aLoA_CcDaGsFaEqEg@eVfBpBagA~Dgd@pJmk@tAiI_I[yReAkO{@sJk@}u@cEiv@mD{Jk@sTgBeD[~BqpBDcDp@kMnFsm@dD1W|EmVdHqStIyQpQ{]pLwU~KuQjDZhEiBpB`G~@dC1BzEjEvFfdJdMvIfUdP~N1ItOxF~TpEtFfA`GopK1BxLfDbD`DhDvEjA|C~@`DpYwKhBc@hFd@pK`CdGbAnFz@~HxAbK`BtKhBjwApVfo@vGTjDz@|CrAxBhBxAvBn@zBDjCm@vBcB|AmCv@kDL{D[wDcAcD`@gIxAojExBaCzG_CjIvDvDRvDa@xB}@jAwA`EYzDZpEhCbo@|h@bbAd|@`KtKkAhZiI~YrDNbDrCdGdF|NnMpPxNfjJI",
  "distance": 3131
}
```

Figura 18: Routing Api e JSON Object

5.3 Aplicação

Esta secção apresenta o funcionamento da aplicação de forma a explicar o que foi desenvolvido e como foram cumpridos os objetivos definidos.

Ao iniciar a aplicação pela primeira vez os nossos utilizadores poderão fazer um registo, sendo-lhes atribuído o perfil de condutor. Analisando então primeiro o perfil de condutor, e após efetuado o registo na aplicação, este poderá efetuar o login/logout a qualquer altura se assim o desejar. Existe uma área pessoal na aplicação que possibilita a alteração dos dados introduzidos pelo utilizador podendo mesmo completar o perfil com mais informação. Para usufruir de todas as funcionalidades da aplicação o utilizador deve proceder ao registo dos seus veículos. Para tal o utilizador precisa de introduzir os dados relacionados com o seu veículo, como a marca, modelo, cor, e outras informações relativas ao mesmo. Estas informações são apresentadas aos agentes de fiscalização quando pesquisam uma matrícula no momento de fiscalizar um veículo. Pretende-se assim reduzir as probabilidades de um agente de fiscalização emitir uma coima para o veículo errado. Caso o utilizador o desejar poderá eliminar qualquer registo do(s) seu(s) veículo(s) da base de dados apagando o(s) mesmo(s) através da aplicação.

No que diz respeito aos estacionamento, o condutor pode visualizar o mapa e consultar o estado das várias áreas de estacionamento, a fim de saber se as mesmas dispõem de lugares livres ou se estão todas ocupadas. Ao selecionar uma área de estacionamento que tem lugares livres, o condutor irá obter a melhor rota até essa área para poder dirigir-se à mesma. Tendo em conta que muitas vezes uma pessoa necessita de se deslocar a um local que não conhece, foi disponibilizada uma área de pesquisa ao utilizador, onde o mesmo poderá inserir a localidade para a qual se deseja deslocar e dessa forma irá também obter uma rota até a área de estacionamento mais próxima do destino introduzido, e uma rota do estacionamento até ao destino. Caso o utilizador deseje selecionar outra área de estacionamento que não a selecionada, basta clicar noutra área, mantendo-se assim o ponto de destino e apenas será recalculada outra rota com o mesmo ponto de partida e destino, alterando-se só a área de estacionamento. Desta forma consegue-se facilitar a procura de um lugar em qualquer zona para a qual um condutor se vai deslocar, conhecendo ou não a zona. Como existem muitos condutores na estrada à procura de estacionamento, não se consegue garantir uma reserva de um lugar com antecedência, havendo assim o risco de quando se chegar ao lugar selecionado, o mesmo já ter sido ocupado. Para evitar esta situação, foram introduzidas áreas de estacionamento a fim de agrupar vários lugares que pertencem à mesma área, conseguindo assim aumentar a probabilidade de haver lugares livres para a zona onde o condutor se vai deslocar. Quando o utilizador tiver uma área selecionada

poderá visualizar a mesma antes de se dirigir para lá. Ao visualizar a área o utilizador irá obter a informação se há lugares disponíveis, por quantos lugares é composta a área e quantos se encontram livres. Além disso o utilizador irá ver também o preço que é praticado nessa área para não ter uma desagradável surpresa na hora de pagamento. Como nem sempre é fácil uma pessoa se localizar num mapa, é dado um street view para ajudar o utilizador a identificar a área para a qual se vai dirigir. Esta última opção é interativa e o utilizador poderá interagir com o mapa que está a visualizar, sendo possível fazer todo tipo de zoom, ou até mesmo navegar.

Por fim, se o utilizador desejar navegar para esta localização será lançado um pedido pela aplicação a fim de abrir uma aplicação de navegação que o utilizador tenha instalado no telemóvel para poder se dirigir até ao lugar. Após estacionar o carro o seu condutor terá de efetuar o pagamento do mesmo caso se encontre numa zona paga. Sendo que o lugar faz parte dos disponibilizados pela aplicação, o seu pagamento também é facilitado ao utilizador. Como o utilizador tem os dados do seu carro introduzidos no seu perfil, ou caso tenha vários veículos registados, basta selecionar o veículo desejado para associar ao pagamento. No ato de pagamento o utilizador terá de indicar o tempo que pretende ocupar o lugar e assim será calculado com base na taxa associada ao mesmo o valor que será debitado da conta do utilizador. De forma a garantir que o utilizador não seja multado caso o tempo selecionado se esteja a esgotar, será enviada uma notificação para o dispositivo do utilizador, permitindo que seja prolongado o tempo, não necessitando assim o condutor regressar ao carro para trocar o bilhete. Além destas funcionalidades todas, o utilizador poderá aceder ao seu histórico, o que permite ter uma ideia do dinheiro que lhe foi debitado da conta em todos os pagamentos efetuados e controlar o mesmo.

As entidades que decidam optar por esta solução terão a possibilidade de criar perfis de agentes de fiscalização. Estes perfis como os mesmos indicam, destinam-se a agentes que fiscalizam os estacionamento. Como já indicado, a aplicação será um meio para as entidades responsáveis pela fiscalização dos lugares de estacionamento poderem ter uma visão geral sobre o estado dos estacionamento. Tendo acesso ao mapa da cidade e conseguindo visualizar os lugares vagos e ocupados em tempo real, tal como os condutores, os agentes irão visualizar o estado de pagamento dos lugares. Para facilitar este processo são usadas várias cores de forma a perceber o estado de um lugar. A seguinte tabela serve para relacionar as diversas cores com os diversos estados dos lugares:

Estado dos Lugares	Cor
Lugar livre	Cinzento
Lugar ocupado e verificado	Azul
Lugar ocupado com necessidade de verificação	Vermelho

Figura 19: Tabela de cores relativas aos estacionamentoos

Com base nas diversas cores o agente de fiscalização irá efetuar as devidas verificações. Começando por o caso onde um lugar se encontra ocupado e necessitando de ser verificado, o agente de fiscalização terá duas opções ao seu dispor. Em ambas, o agente irá pesquisar pela matrícula do veículo a fim de obter a informação da mesma para confirmar que é o carro certo. Caso o veículo não se encontre registado na aplicação o agente irá introduzir os dados do carro para que o mesmo fique registado na aplicação. Analisando agora os dois possíveis casos, o primeiro é o mais comum atualmente, o veículo tem um bilhete em papel colocado no interior, nesse caso o agente irá introduzir os dados do bilhete a fim que o pagamento fique registado na aplicação e associado ao devido lugar. No caso de não haver bilhete no interior do carro, o procedimento é similar ao anterior, ao efetuar a pesquisa da matrícula o agente irá obter a informação do estado de pagamento do veículo. No caso de haver um pagamento efetuado através da aplicação ou uma coima já associado ao veículo, o agente de fiscalização poderá associar a mesma ao lugar onde se encontra o carro. No caso de a aplicação não devolver nenhuma das informações anteriores, irá ser facultada a opção de poder autuar o veículo por falta pagamento. Em todos estes casos o estado do lugar irá passar para ocupado e verificado.

Caso um lugar se encontre no estado de ocupado e já tenha sido verificado, um agente pode consultar o estado, para poder corrigir algum erro que tenha feito no momento de verificação. No caso de o lugar estiver livre o agente de fiscalização não poderá efetuar nenhuma operação sobre o lugar.

Desta forma os agentes de fiscalização conseguem tornar o seu trabalho mais rentável, sabendo onde se encontram veículos que ainda não foram verificados e localizando os mesmos com facilidade, reduzindo assim significativamente o tempo de verificação do estado de pagamento de cada um dos veículos.

Por último é necessário um utilizador que trate da manutenção de todos os lugares existentes, o técnico. Como os lugares são monitorizados através de sensores que se encontram aplicados no chão, o utilizador consegue gerir esses mesmos sensores. Quando é feita a instalação de um sensor, o utilizador terá de o registar usando a aplicação fornecendo os dados do identificador e as coordenadas. A qualquer momento, também pode ser removido um lugar temporariamente ou definitivamente, usando a aplicação. Quando a bateria acabar o sensor vai deixar de emitir qualquer sinal e o utilizador vai conseguir identificar essa mesma anomalia ou outra qualquer que não deixe o sensor emitir sinal. Com a aplicação o utilizador pode localizar o sensor e proceder à reparação do mesmo ou à simples troca de bateria. A qualquer momento o utilizador pode alterar a localização do sensor e assim à distância de uns cliques atualizar a informação.

5.3.1 Registo e Login

Ao inicializar a aplicação o utilizador vai poder registar-se. Para cumprir este processo com sucesso, o utilizador terá de preencher todos os campos. Após validar o seu registo o utilizador poderá efetuar o seu login com o email e password que registou de forma a aceder a todas as funcionalidades que a aplicação tem para lhe oferecer, tal como ilustra a Figura 20.

Após fazer login o utilizador recebe um token e enquanto esse token for válido o utilizador pode abrir diretamente a aplicação sem precisar de preencher as suas credenciais. Caso o token expire o utilizador é redirecionado para a página de login onde terá de repetir o mesmo processo de forma a pedir um novo token.

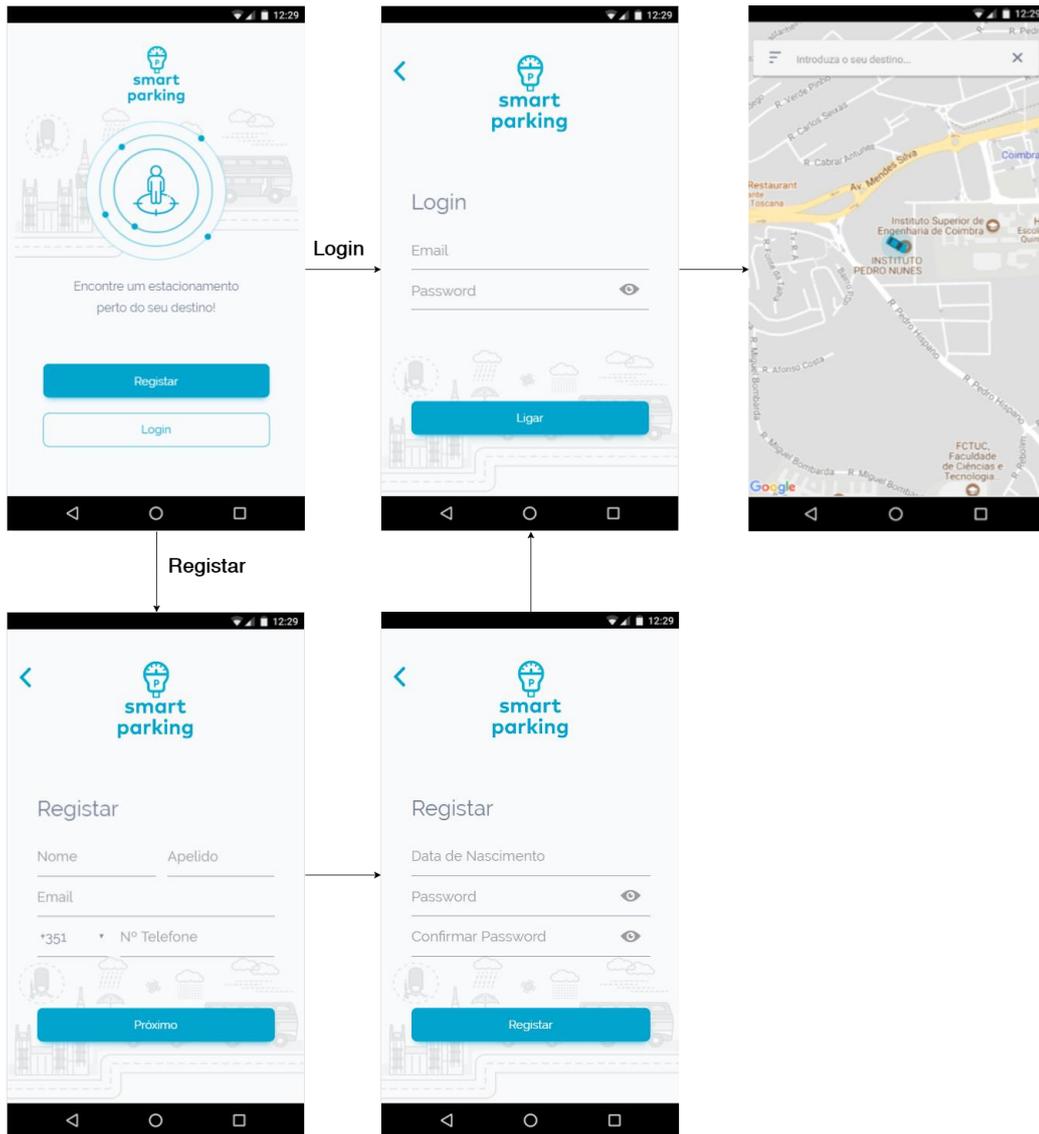


Figura 20: Processo de Registo e/ou Login

5.3.2 Filtro de indicativos

Ao fazer o registo na aplicação o utilizador necessita de introduzir os seus dados, um dos quais o número de telefone que é composto por um indicativo e um número único. Como o indicativo será sempre uma combinação de números restritos por uma lista já existente, a melhor opção foi restringir também o mesmo aos utilizadores de forma a garantir um indicativo válido. Após efetuar uma pesquisa foi encontrada uma lista completa com todos os países com os respetivos indicativos [64], que foi copiada para um ficheiro. Ao copiar a lista para um ficheiro não é necessário aceder sempre ao endereço de rede para obter a lista. Após isso foi necessário filtrar a lista de forma a obter cada indicativo uma única vez, tendo em conta que alguns países partilham os seus indicativos, como por exemplo o Canada é os Estados Unidos da América.

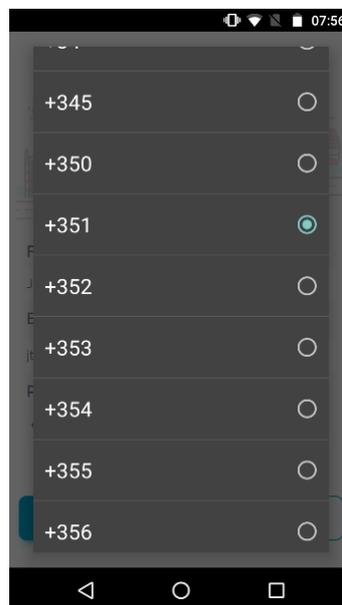


Figura 21: Seletor de indicativos

5.3.3 Loading Screen

Ao efetuar o login vai ser feito um pedido à API para obter um token válido, o perfil e os dados do utilizador. De seguida é feito outro pedido à API para obter os lugares e os seus estados. Como a duração de todo este processo pode durar alguns segundos dependendo da ligação do cliente, foi introduzido um indicador de atividade, representado na Figura 22, para o utilizador ter a perceção que há uma atividade a decorrer. Este ecrã foi inicialmente criado através de uma imagem animada, mas nos testes realizados a mesma não apresentava o melhor desempenho. A seguir foi estudado o caso de criar a animação através de código CSS a fim de ver as diferenças de desempenho e com esta opção foi possível obter uma imagem animada mais suave e natural a ser visualizada. Ao manter o ecrã de login ou apenas uma imagem estática podia dar a ideia ao utilizador que a aplicação tinha parado de funcionar, assim sendo a opção recaiu sobre uma animação entre o ecrã de login e o mapa, garantido que a ideia de carregamento passa para o utilizador.

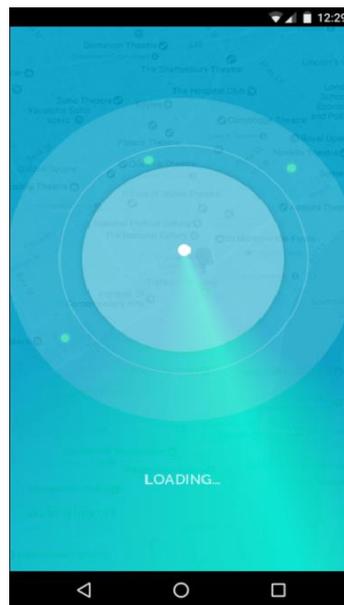


Figura 22: Ecrã de carregamento

5.3.4 Menu

Neste ecrã consegue-se ver todas opções oferecidas aos utilizadores da aplicação. Da parte dos utilizadores comuns, Figura 23 lado esquerdo, consegue-se ver que um utilizador pode pesquisar um lugar, ver o seu histórico, gerir os seus veículos, pagar os seus estacionamento, aceder às definições, gerir o seu perfil ou sair da aplicação. Já da parte de um agente de fiscalização as funcionalidades são mais restritas como nos mostra a imagem no centro da Figura 23. O agente de fiscalização pode verificar os estacionamentos, alterar as definições, gerir o seu perfil ou sair da aplicação. O técnico consegue aceder à gestão de lugares, definições e o próprio perfil, como ilustrado no lado direito da Figura 23.

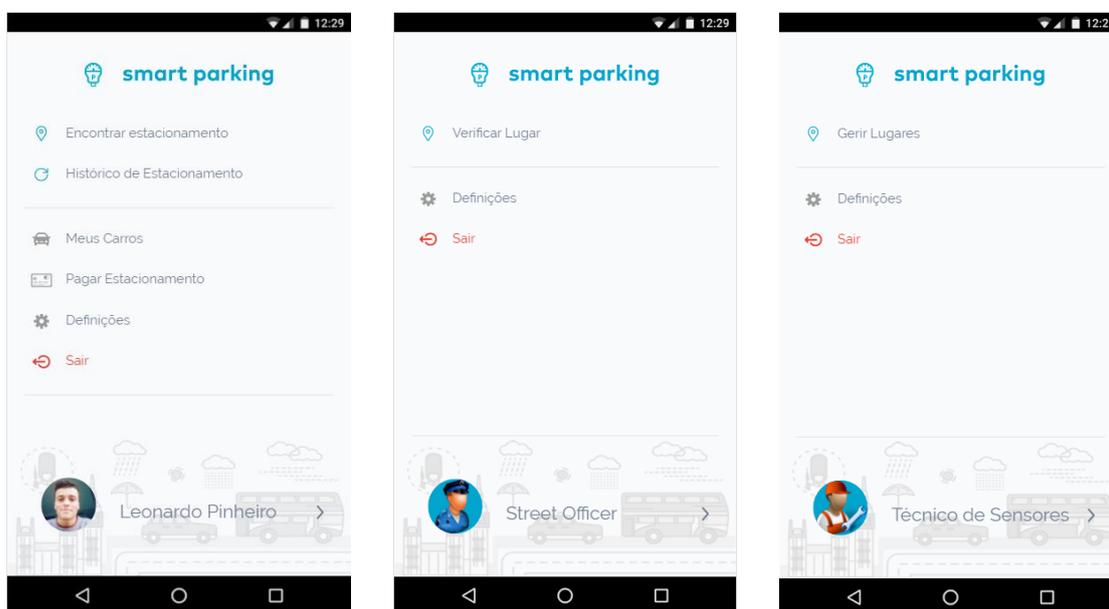


Figura 23: Menu

5.3.5 Perfil

A figura abaixo mostra a informação que um utilizador introduziu na altura do seu registo exceto a sua data de nascimento. Ao editar o seu perfil, o utilizador consegue alterar qualquer um dos campos visíveis incluindo a sua foto de perfil.

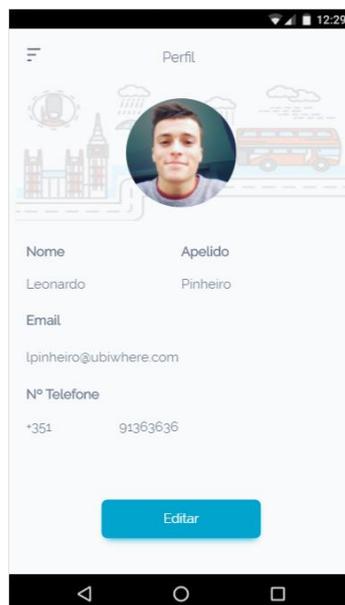


Figura 24: Perfil de utilizador

5.3.6 Áreas de Estacionamento

Nesta secção conseguem-se ver os mapas tal como são vistos da parte dos condutores. No mapa encontram-se as áreas de estacionamento representadas por um ícone azul com o P no interior. A cercar esse ponto o utilizador consegue ver um perímetro que define onde se encontram os lugares pertencentes à área de estacionamento. Estas áreas de estacionamento são desenhadas a partir das coordenadas registadas no backend. Pegando nas coordenadas da área é desenhado um polígono a partir da função fornecida pelo Google Maps [65], sendo este depois personalizado em termos de aparência.

Tal como será explicado em maior detalhe na secção 5.4.2, um utilizador pode pesquisar um lugar ou simplesmente clicar no ícone representado a área para o qual se pretende dirigir. Quando

um utilizador seleccionar o seu destino é desenhada a reta da sua localização até ao estacionamento e outra reta tracejada do estacionamento até ao destino. Estas retas também são desenhadas usando um função do Google Maps [66] e de seguida personalizadas.

Após isso o utilizador poderá visualizar a área de estacionamento, como já explicado anteriormente. Ao navegar para essa área é lançado o pedido para ser redirecionado para uma aplicação que consiga usar as coordenadas tal como mostra a Figura 25.

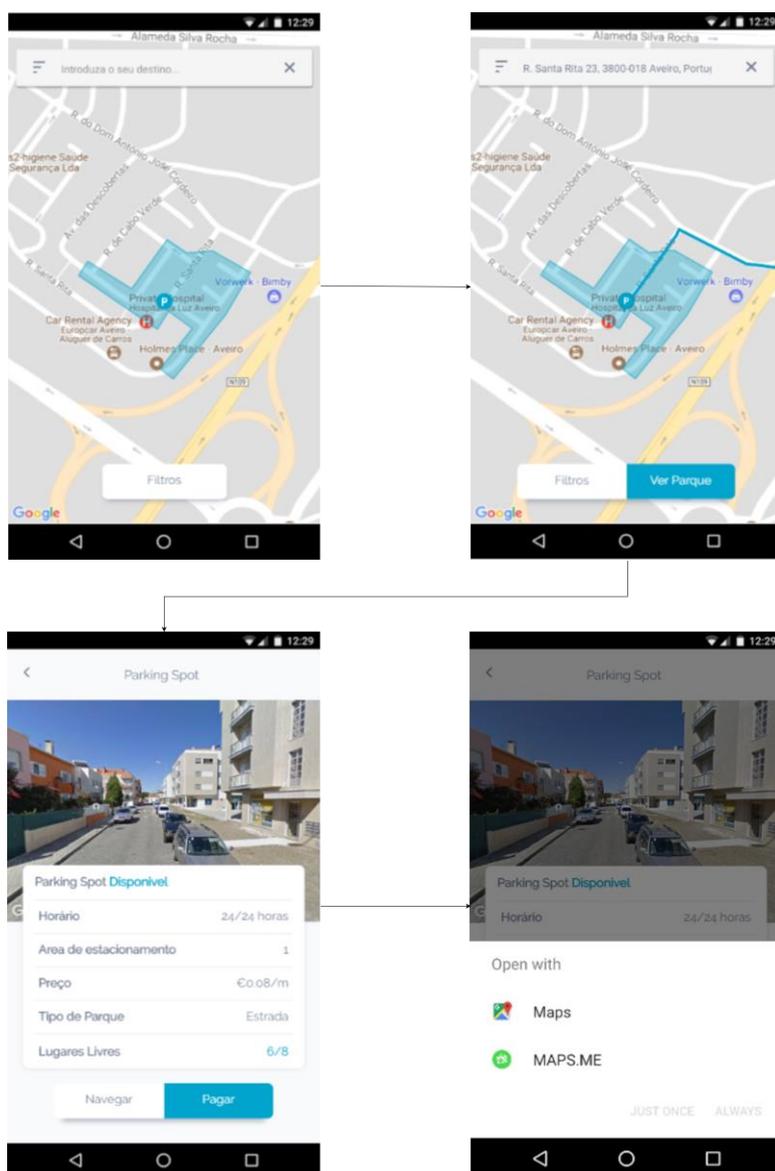


Figura 25: Fluxo de navegação para área de estacionamento

5.3.7 Carros

De forma a poder usar todas as funcionalidades da aplicação, o utilizador terá de registar os seus veículos na aplicação. Como representado na Figura 26, um utilizador com o perfil de condutor ao aceder pela primeira vez à área dos seus veículos, irá encontrar uma lista vazia. De forma a proceder ao registo do seu carro terá de adicionar o seu veículo. No ecrã seguinte será necessário a introdução dos dados do seu carro e, ao concluir a ação, será redirecionado para o ecrã anterior onde irá aparecer já o registo que acabou de efetuar. Nenhum utilizador está restrito a um número máximo de veículos que pode adicionar, no entanto para poder efetuar pagamentos de estacionamento terá de ter no mínimo um veículo para associar o pagamento ao mesmo.

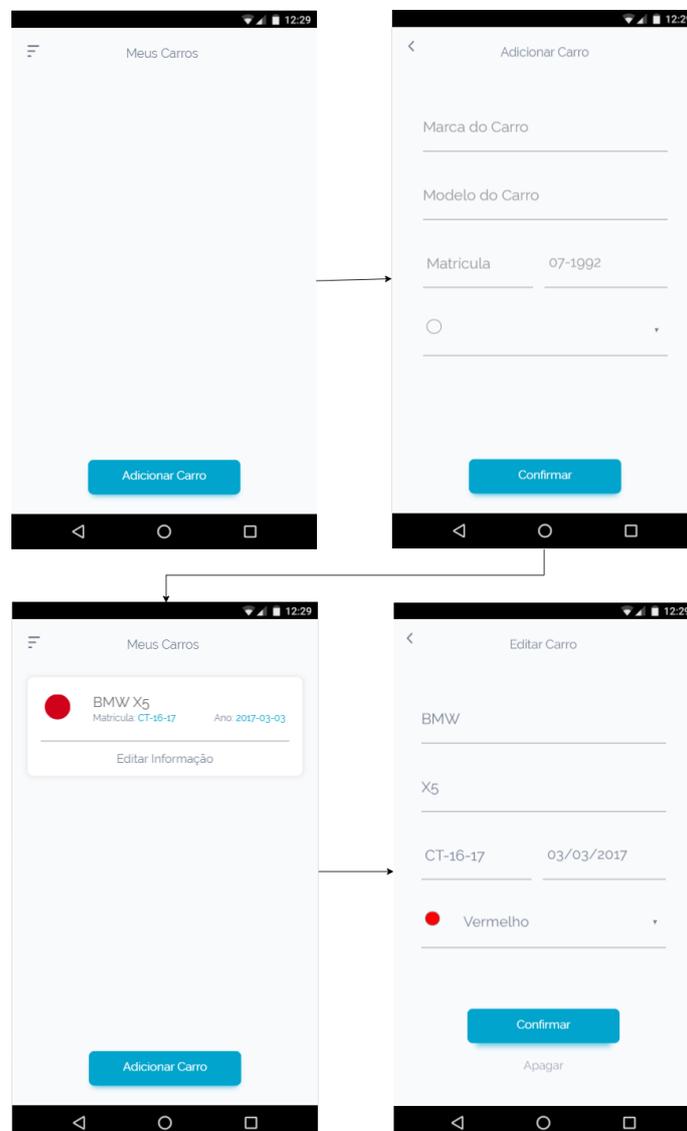


Figura 26: Método de adição de carros pelos condutores

A qualquer altura um utilizador pode aceder ao registo do seu veículo e eliminar ou editar a informação sobre o mesmo. Um utilizador não pode registar uma matrícula já registada no sistema. Se porventura o veículo do utilizador já se encontrar registado na aplicação seja por o automóvel já ter pertencido a outra pessoa ou por outro motivo, o utilizador terá de contactar a administração do sistema de forma a alterar o dono do veículo ou de eliminar o carro do sistema a fim que este utilizador possa proceder ao registo em seu nome.

Por outro lado, quando um agente de fiscalização estiver a controlar os veículos estacionados, se porventura encontrar um automóvel que não se encontre registado na aplicação, então terá de introduzir os dados do mesmo no momento do registo do ticket ou da emissão da coima como ilustrado na Figura 27. Não é obrigatório introduzir a data de registo do veículo nem os campos de marca e modelo, pois o agente pode desconhecer esses dados do veículo.

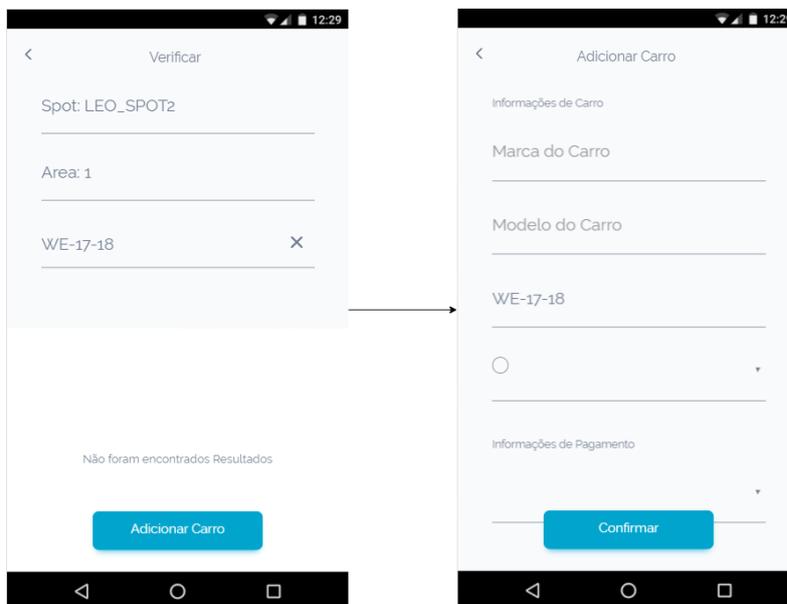


Figura 27: Método de adição de carros pelos agentes de fiscalização

5.3.8 Pagamentos e Histórico

Quando um utilizador chega ao lugar de estacionamento desejado, deve proceder ao pagamento da ocupação do lugar no qual se encontra. Nesta secção será explicado este procedimento e as opções oferecidas ao utilizador.

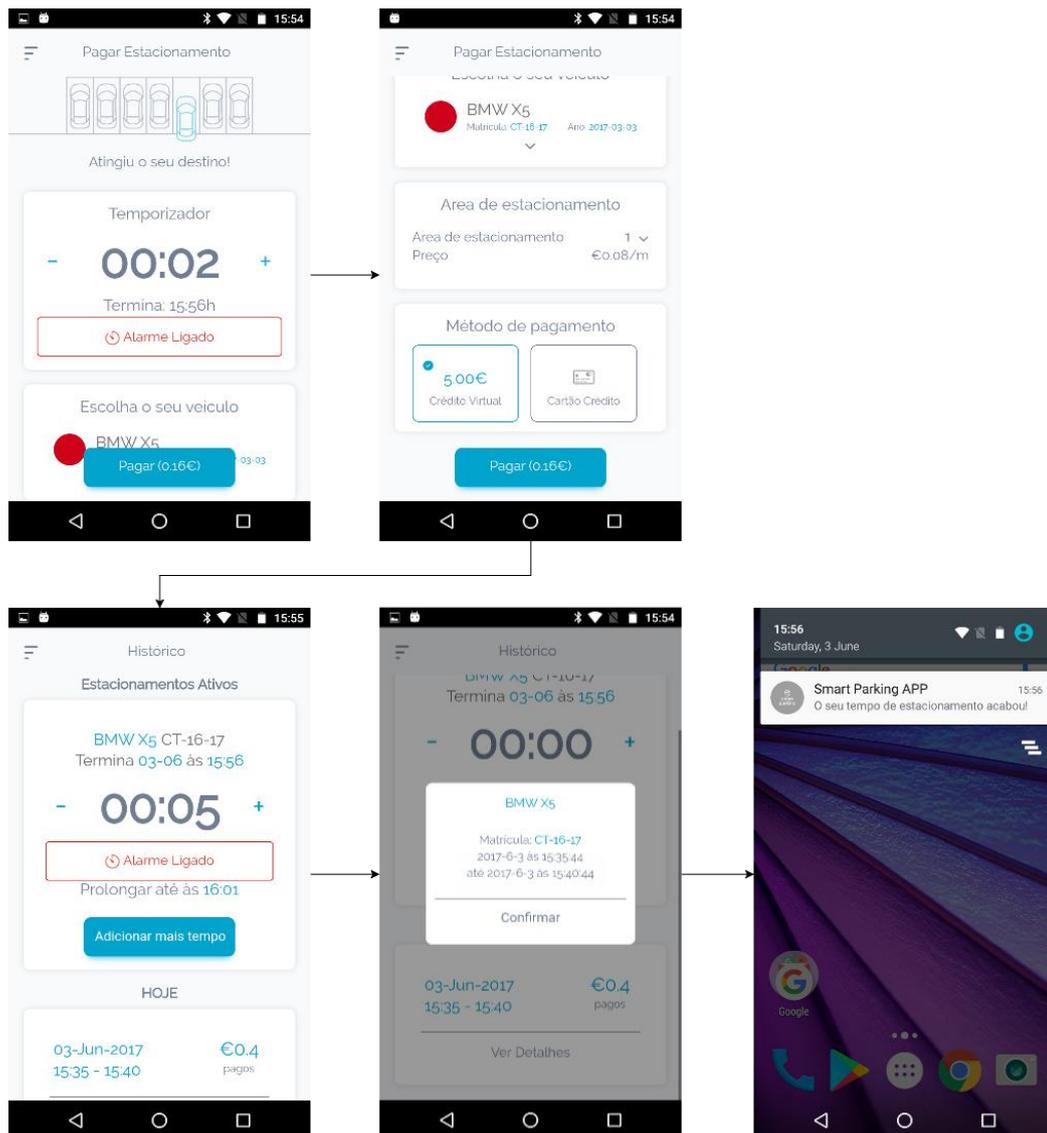


Figura 28: Workflow de pagamento

Se o utilizador estiver a usar a opção de navegação no momento que entrar na área de estacionamento desejada, a aplicação irá lançar um alerta e o seu funcionamento será explicado em detalhe na secção 5.4.3. Através deste alerta o utilizador irá logo entrar para o ecrã onde pode efetuar o pagamento do seu estacionamento. Caso não use a opção de navegação o utilizador também pode aceder ao ecrã de pagamentos de outras formas. Se o utilizador não tiver veículos registados não poderá efetuar nenhum pagamento e será assim direcionado para o ecrã de registo de veículos que foi explicado na secção anterior. Caso tenha no mínimo um automóvel registado, então neste ecrã o utilizador tem de selecionar o tempo para o qual pretende efetuar o pagamento. Como existe o risco de se atrasar, o utilizador pode ativar um alarme de forma a ser notificado quando o seu tempo estiver a acabar. De seguida, o utilizador seleciona o veículo ao qual pretende associar o pagamento e a zona, caso tenha acedido ao ecrã de estacionamento através do menu. Se tiver acedido pela notificação lançada pelo Geofence, o seu funcionamento será explicado mais à frente na secção 5.4.3, não necessita de selecionar a área para o qual quer efetuar o pagamento, esse campo já vem preenchido de forma a não induzir o utilizador em erro.

No histórico o utilizador pode consultar todos os pagamentos que efetuou, tendo estes já expirado ou estando ainda a decorrer. Nos pagamentos que ainda estão a decorrer, é oferecido ao utilizador a opção de prolongar o seu tempo e efetuar o pagamento desse mesmo tempo que está a adicionar. O utilizador também pode ativar ou desativar o alarme neste ecrã. Caso o utilizador tenha o seu veículo registado e se por falta de bateria no momento ou outro motivo não pode efetuar o pagamento do estacionamento usando a aplicação, pode efetuar o pagamento usando um parquímetro. Se o agente de fiscalização controlar o veículo e registar o pagamento, então é dada a opção ao utilizador de consultar o pagamento e depois de o gerir como se tivesse sido feito na aplicação, tendo assim a possibilidade de consultar, prolongar ou até mesmo ativar um alarme de forma a saber o tempo que lhe resta. Olhando para a Figura 28 consegue-se ver a sequência de ecrãs associados ao pagamento, representando o último uma notificação de fim de tempo emitida pelo alarme que foi ativado pelo utilizador. Se um utilizador tiver um pagamento a decorrer ao entrar na aplicação, em vez de ser direcionado para o mapa a fim de pesquisar um lugar é reencaminhado para a página de históricos, de forma a conseguir visualizar logo o seu pagamento e poder efetuar alterações caso o deseje.

5.3.9 Verificação

Nesta secção vai ser explicado como são processadas as verificações dos estacionamentos ocupados feitas pelos agentes de fiscalização. Só os utilizadores com o perfil de agente de fiscalização vão ter acesso a estes ecrãs.

Quando um utilizador com perfil de agente de fiscalização entra na aplicação as opções à sua disposição são diferentes das oferecidas aos restantes utilizadores. Assim como definido nos objetivos da aplicação, o agente de fiscalização vai querer fiscalizar os estacionamentos ocupados de forma a garantir que os pagamentos dos respetivos lugares estão a ser respeitados.

Ao aceder ao mapa, o utilizador poderá efetuar operações só sobre os estacionamentos ocupados. O utilizador vai aceder à aplicação e ver onde se encontram estacionamentos ocupados que ainda não foram verificados. Estes estacionamentos vão estar representados com a cor vermelha de forma a sinalizar um alerta para os utilizadores. Ao verificar um estacionamento vão ser oferecidas 3 opções ao utilizador, sendo estas, associação de um pagamento feito por um utilizador usando a aplicação de Smart Parking a um lugar em concreto, registo de um bilhete comprado num parquímetro ou registo de uma infração cometida por um utilizador e assim emitir uma coima. Os estacionamentos já verificados serão apresentados com um ícone de cor azul.

- Pagamento efetuado com aplicação:

Nesta subsecção vai ser analisado o caso de um utilizador ter efetuado o pagamento do seu estacionamento usando a aplicação e de que forma o agente de fiscalização vai tratar esse pagamento. Como o condutor já efetuou o pagamento, então o seu veículo já se encontra registado na aplicação. Assim sendo, o utilizador irá introduzir a matrícula do veículo que está a ocupar esse mesmo sítio e irá aparecer a informação do veículo associada a essa matrícula e o pagamento efetuado pelo seu dono. Junto à informação de pagamento, o utilizador terá a opção de associar este pagamento ao lugar que está a verificar para que o lugar mude de estado como ilustrado pela Figura 29. Se o utilizador se enganar a associar o pagamento pode desfazer a ação que fez acedendo ao lugar e desassociando o pagamento.

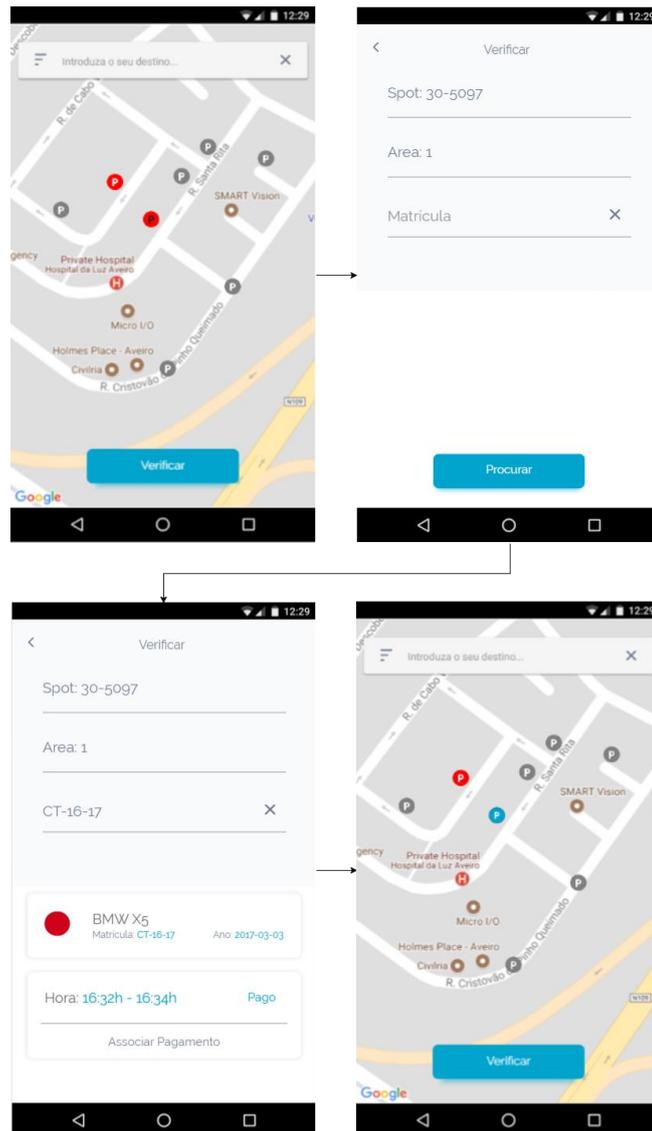


Figura 29: Associação de pagamento efetuado por condutor

Como o pagamento é feito por zona, o condutor pode decidir mudar o carro de sítio e assim manter o pagamento do estacionamento se estiver na mesma zona. O sensor ao detetar que o veículo abandonou o seu lugar passa a ser representado como estando livre. O lugar onde o condutor estacionou passa a estar ocupado. Ao verificar esse novo lugar, o utilizador pode voltar a associar o pagamento ao novo lugar da mesma forma que foi feito anteriormente.

- Pagamento efetuado em parquímetro

Sendo que a aplicação não vai revolucionar a forma de pagar os estacionamento nos primeiros tempos e ainda existem pessoas que não têm smartphones, vai continuar a haver pessoas que vão efetuar o pagamento do seu estacionamento usando os métodos tradicionais, os parquímetros. Assim sendo vai ser analisado esse mesmo caso.

Quando um agente de fiscalização encontrar um veículo com um bilhete comprado num parquímetro terá de o registar na aplicação de forma a mudar o estado do estacionamento para saber que já foi controlado. A Figura 30 mostra o processo a seguir. Sendo que pode haver uma vertente ligeiramente diferente, que será no ato de introduzir a matrícula na pesquisa do veículo em causa. Se o mesmo não se encontrar registado, o utilizador terá de introduzir os dados do veículo de forma a registar o automóvel na aplicação (fluxo veículo não registado). Se o mesmo estiver registado, o utilizador só terá de selecionar o veículo de forma a atualizar o seu estado (fluxo veículo registado da figura). Nos dois casos, de forma a lhe associar o pagamento, o utilizador terá de introduzir os dados do bilhete, nomeadamente, código de identificação de bilhete, hora de início, hora de fim e área associada ao bilhete. Outra maneira de introduzir estes dados é ler o código QR que se encontra no bilhete de forma a obter essas mesmas informações.

Se o utilizador decidir mudar de lugar e manter o seu bilhete, a verificação do mesmo será feita da forma que foi explicado no ponto anterior. Tendo em conta que o pagamento já se encontra registado na aplicação, não é necessário voltar a introduzir nenhum dos dados associados ao pagamento.

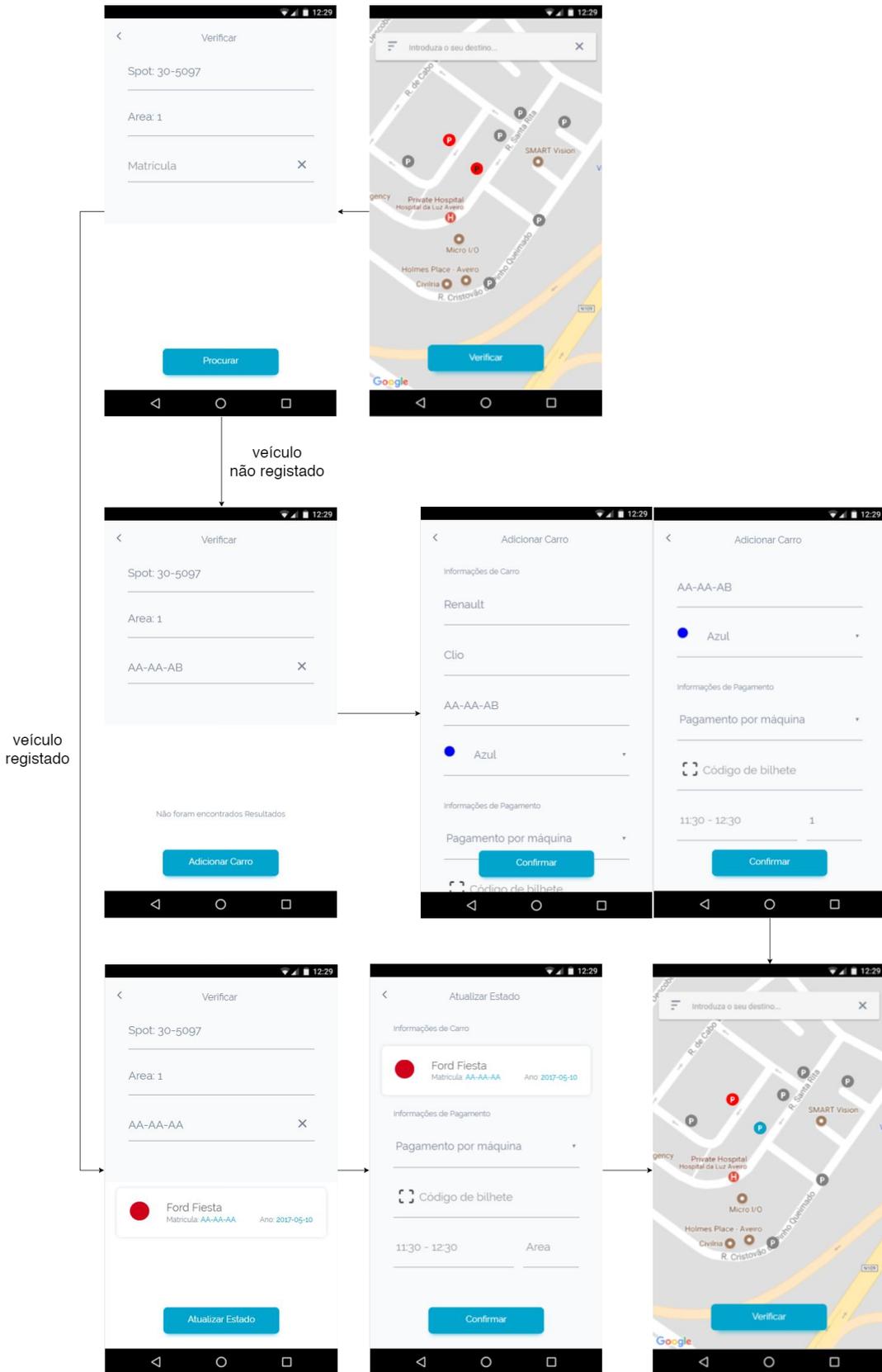


Figura 30: registo de pagamento efetuado usando parquímetero

- Sem pagamento

Por último irá sempre haver utilizadores que não efetuam o devido pagamento dos seus estacionamento ou que simplesmente deixam expirar os seus pagamentos. Nesta subsecção irá ser apresentada a solução proposta para esses casos.

Quando um agente de fiscalização detetar um lugar por verificar na sua aplicação e verificar que o mesmo não tem nenhum bilhete visível no seu para-brisas, então irá proceder à verificação do mesmo. Ao pesquisar a matrícula, se o veículo não se encontrar registado, significa que não efetuou pagamento. No caso de encontrar um veículo registado mas sem pagamento associado então está na mesma situação. No primeiro caso, como podemos ver no fluxo de veículo não registado da Figura 31, o utilizador terá de introduzir os dados do veículo para que este seja registado na aplicação e marcar como não pago. No segundo caso, o utilizador só terá de atualizar o estado do estacionamento como não pago tendo o veículo já sido selecionado, tal como ilustrado no fluxo de veículo registado da mesma figura. Após isso será registada a infração no sistema para que possa ser emitida uma coima.

Caso um veículo se encontre nesta situação poderá acontecer o mesmo que nos casos anteriores, o veículo mudar de lugar. Essa situação será tratada tal como as duas anteriores sendo que no momento que é feita a pesquisa sobre a matrícula, irá aparecer uma infração associada ao automóvel. Assim sendo, o agente de fiscalização pode associar esta infração ao novo lugar que o carro está a ocupar.

Em geral o agente de fiscalização pode tratar estes três casos, sendo que no caso de registo de bilhete ou de infração será oferecida a opção de eliminar a mesma ou até mesmo editar a informação relativa ao bilhete. Caso o agente tenha associada uma infração ou um pagamento a um lugar por engano, pode corrigir essa situação.

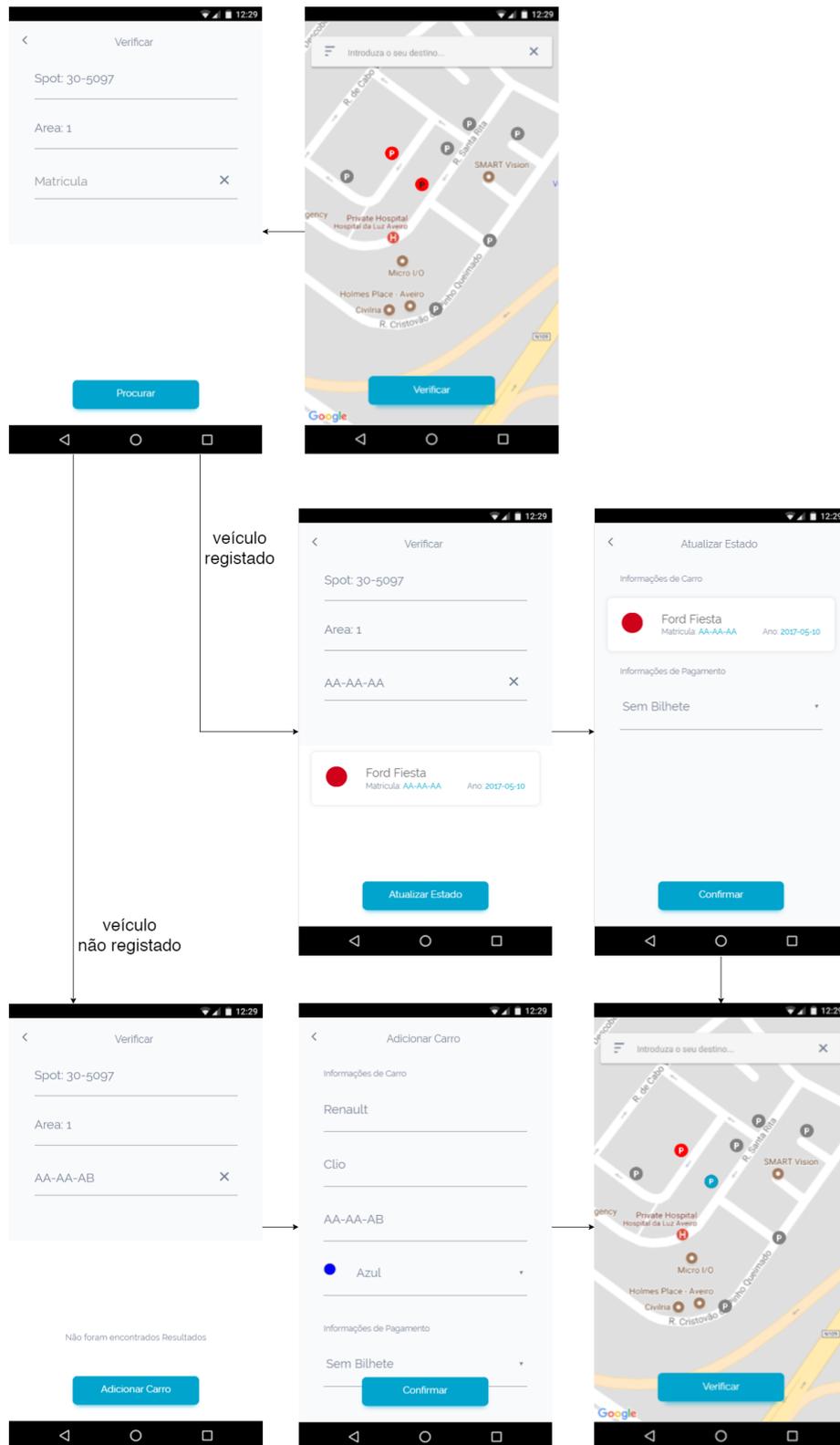


Figura 31: Emissão de coima

5.3.10 Atualização de estado dos sensores

Cada vez que um sensor deteta uma alteração de estado envia uma mensagem para o servidor. De forma a receber todas estas alterações de estados, ao iniciar a aplicação é feita uma ligação ao Meshblu usando um canal websocket, que fica à escuta de novas mensagens. Olhando para a Figura 32, que representa o JSON recebido em cada mensagem, da parte da aplicação são usados os campos “uuid” para identificar o sensor ao qual se refere a mensagem e o campo “status” para conhecer o estado do sensor, sendo que 0 representa livre e 1 ocupado.

A gestão da mensagem recebida depende do perfil de utilizador que se encontra ativo. Caso seja um agente de fiscalização é atualizado o estado de cada sensor em tempo real, conseguido assim o agente de fiscalização perceber sempre onde se encontra um veículo, mesmo que tenha chegado há instantes. Caso o utilizador tenha o perfil de condutor, então estas mensagens são processadas de forma a atualizar o estado de ocupação de cada área, seja para retirar ou acrescentar um lugar livre à área à qual pertence o sensor.

```
{
  "payload":{
    "application":"parking",
    "gateway_timestamp": "2017-07-05T10:27:00",
    "messages":[
      {
        "timestamp": "2017-07-05T10:34:00",
        "uuid": "30-5037",
        "type": "reading",
        "status": 0,
        "sequence_number": 1,
        "firmware_version": "1.124.0.2",
        "algorithm_version": "1.124.0.15",
      }
    ]
  }
}
```

Figura 32: Mensagem enviada pelo Meshblu

5.3.11 Instalação de sensores

De forma a proceder à instalação de um sensor, o asset associado ao mesmo já tem de estar registado na API com o estado de “ready”. O utilizador com o perfil de técnico terá de ler o código QR que se encontra no sensor e introduzir a localização onde vai instalar o mesmo.

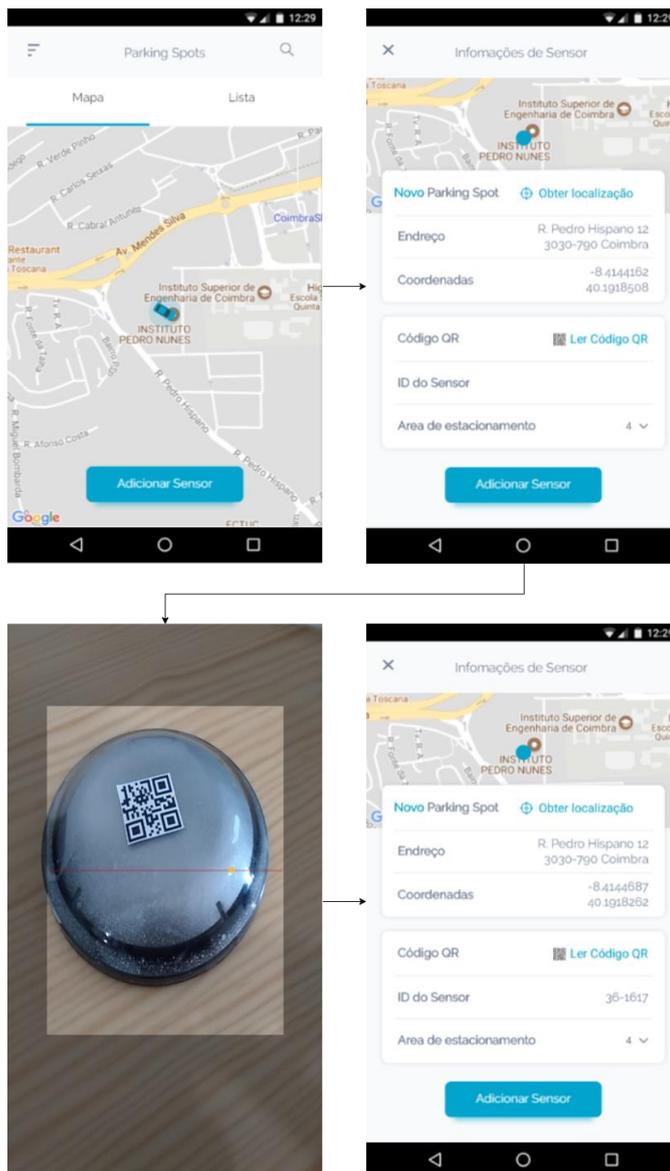


Figura 33: Instalação de sensores

Como podemos ver na Figura 33, após ter lido o código QR e introduzindo o local onde vai ser instalado o sensor, o utilizador consegue ver a localização do ponto no mapa, antes de confirmar o registo. Ao confirmar o registo, é feito um POST de forma a associar o sensor ao asset já registado. Após esse registo é feito um PATCH do estado do asset para “active” de forma a poder ser usado pelos restantes utilizadores.

5.3.12 Gestão de sensores

De forma a gerir todos os sensores instalados o utilizador pode alterar entre a aba do mapa e da lista de sensores instalados. Associado a cada sensor o utilizador irá ver um ícone com a cor representando o estado do asset associado. Azul representa um asset com o estado “active”, não necessitando assim qualquer intervenção técnica. Já o vermelho representa um sensor cujo asset associado se encontra no estado de “inactive”. Caso o sensor não tenha nenhuma anomalia e tenha apenas sido desativado por questões de logística, o utilizador pode alterar o estado do mesmo para “active”, no caso de ter de remover o sensor para ir para o centro de reparações então altera o estado para “under_repair”. Após um sensor ser reparado no centro de reparações deve ser feita uma alteração de estado de “under_repair” para “ready” de forma a poder voltar a ser instalado. Finalmente quando um sensor voltar a ser instalado é feita uma alteração de estado de “ready” para “active”. Todos estes pedidos são feitos através de um PATCH na API do asset associado ao sensor.

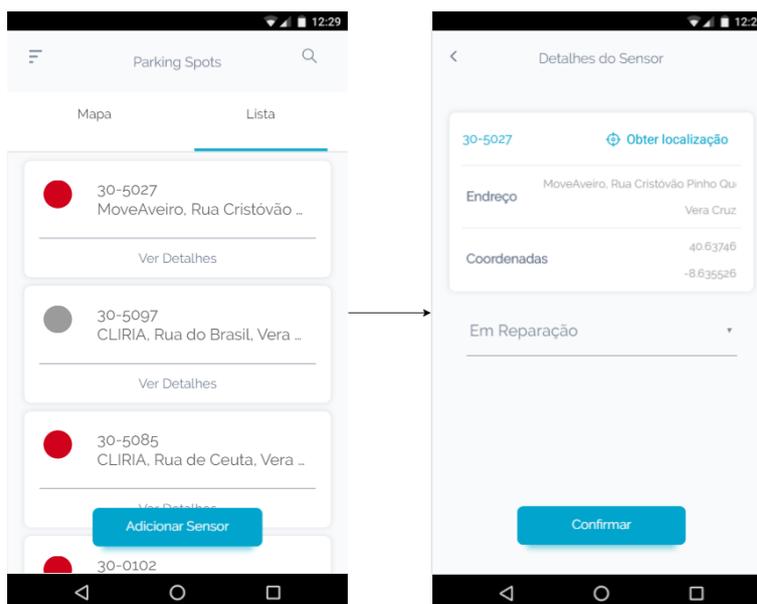


Figura 34: Ecrãs de gestão de sensores

Na Figura 34 conseguimos verificar que além de mudar o estado, o utilizador tem a possibilidade de corrigir a localização associada ao sensor.

5.3.13 Definições

Por último, como em todas as aplicações o utilizador pode ajustar as definições consoante as suas preferências. No caso de um utilizador comum, o condutor, este poderá mudar a língua da aplicação ou até mesmo a moeda em que vai querer ver os preços apresentados e efetuar o pagamento dos estacionamento. Ficou também já implementada uma opção para mais tarde integrar os registos dos métodos de pagamento, tal como se pode ver no lado esquerdo da Figura 35.

No caso de o utilizador ser um agente de fiscalização ou um técnico, só poderá alterar a língua na qual a aplicação é apresentada, tendo em conta que não vai efetuar pagamentos nem consultar preços, assim como se pode verificar no lado direito da Figura 35.

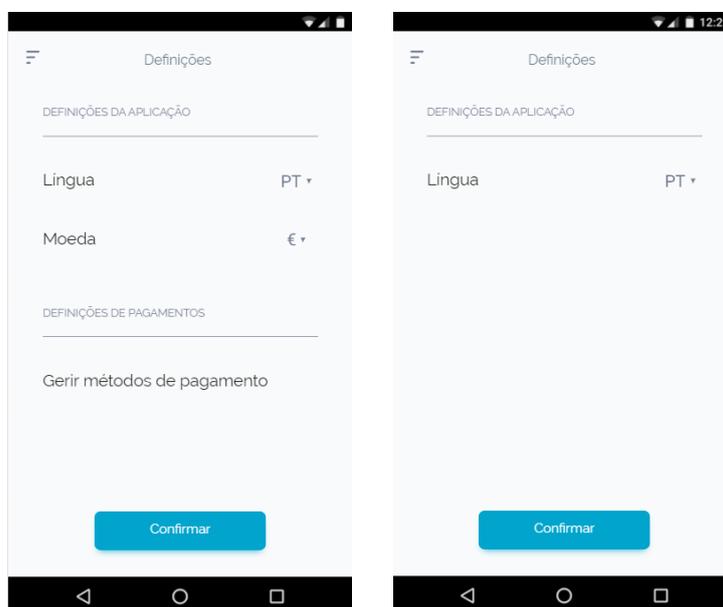


Figura 35: Definições

5.4 Desenvolvimento em Ionic

A Figura 36 serve para dar uma visão geral sobre o Ionic, que vai ser explicado no seguinte texto. Ao iniciar o projeto é criado um ficheiro chamado `app.js` onde são injetados todos os módulos, declaradas as constantes globais e variáveis correspondentes às bibliotecas instaladas. Todos os comportamentos da aplicação também são definidos neste ficheiro, entre botões físicos, idioma, etc. Quando o utilizador efetuar o login, consoante o perfil de utilizador é redirecionado para o ecrã com o mapa adequado, seja condutor, técnico ou mesmo fiscal. Também é feito o routing, que é a declaração dos estados com os respetivos templates HTML, para que o utilizador possa navegar nos devidos ecrãs e são declarados os argumentos que podem ser passados entre ecrãs.

Todos os controladores são desenvolvidos em Javascript. Nestes ficheiros são definidas todas as funções correspondentes à view. Sejam funções que vão ser chamadas quando clicado algum botão ou funções executadas ao aceder ao ecrã, como por exemplo no mapa o pedido para obter todas as áreas disponíveis que vão ser apresentadas aos condutores.

No `index.html` são declarados todos os controladores criados anteriormente e bibliotecas a serem usadas. Após isso todos os restantes ecrãs são desenvolvidos em HTML. Em cada ficheiro HTML é desenvolvido o código da parte visual tal como deve ser apresentada aos utilizadores na respetiva view, é declarado qual o controlador correspondente ao ecrã onde se encontram as funções do mesmo e são associadas as funções já criadas no controlador dos respetivos botões. Finalmente toda a parte estética da aplicação, entre estilos de botões, cores, formas, etc. é desenvolvida em ficheiros do tipo CSS.

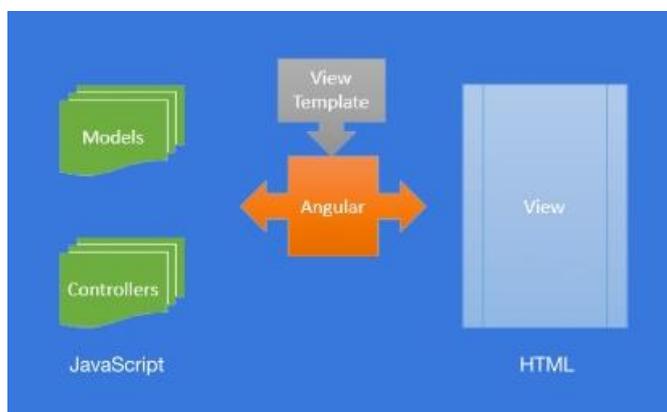


Figura 36: Funcionamento do Ionic [67]

5.4.1 Integração de Ion-datetime-picker

Começando com o registo de um utilizador é necessário introduzir a data de nascimento e, para facilitar a tarefa ao utilizador, foi introduzido o ion-datetime-picker [68]. Esta biblioteca permite fornecer 3 tipos de seccionadores, data, hora ou ambos. No caso do registo foi usado o seccionador de data, Figura 37 lado esquerdo. Na mesma figura do lado direito, está representado o seccionador de hora, que foi usado para selecionar o tempo de estacionamento para o qual um utilizador quer efetuar o pagamento do estacionamento.

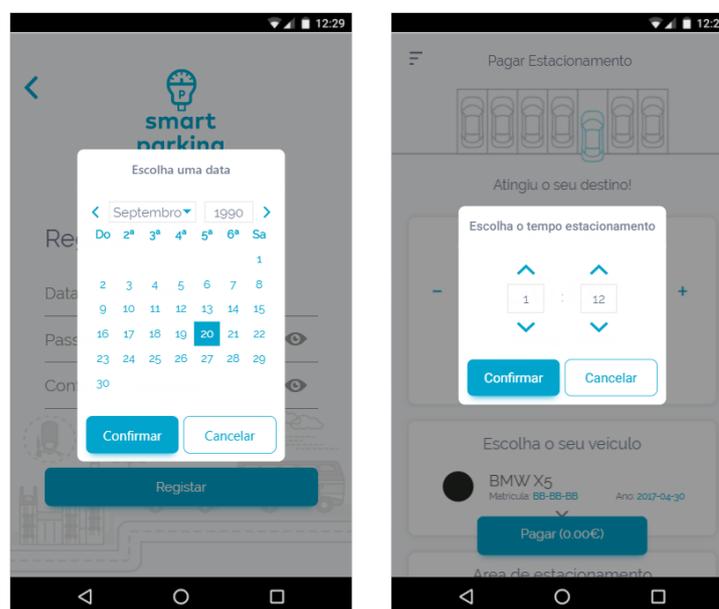


Figura 37: Ion-datetime-picker

Como é possível ver na figura acima o ion-datetime-picker foi personalizado de forma a corresponder ao layout da restante aplicação. Foram criados os modelos das várias línguas que a aplicação aceita, tanto as cores, a fonte, os tamanhos como os botões foram alterados.

5.4.2 Integração de Google Maps API

- Mapa:

O foco principal da aplicação é o mapa. Para garantir que um utilizador possa procurar um lugar de estacionamento, este terá de usar o mapa para saber para onde se encontra, onde fica o estacionamento e para onde se deve dirigir. A principal vantagem de usar a Google Maps API [69], é que é familiar para a maioria das pessoas e corre em todos os dispositivos. Dá para optar por vários tipos de visão de mapas, seja mapa de estradas, terreno, satélite ou mesmo híbrido.

- Barra de pesquisa:

Como é pretendido encontrar um lugar perto de uma localização, nada melhor que usar a barra de pesquisa do Google Maps. Já tendo a API para visualizar o mapa, aproveita-se para integrar a barra de pesquisa. Ao introduzir o nome do destino, a API vai já dando sugestões de forma a auto completar a pesquisa sem ser necessário escrever tudo por extenso, como é possível ver na Figura 38. Também é dada a opção de introduzir coordenadas ao utilizador.

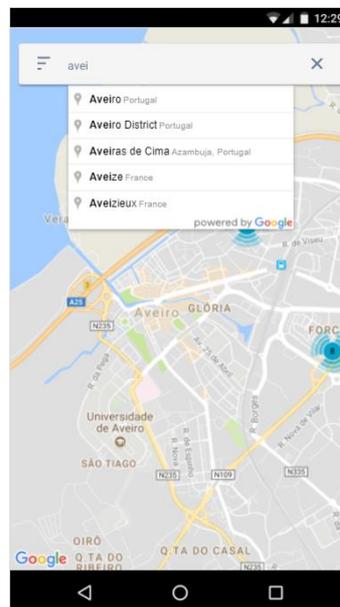


Figura 38: Pesquisa usando o Google Maps

- Geocoder:

Usando o Geocoder [70] do Google Maps foi possível traduzir o destino introduzido pelo utilizador em coordenadas. Estas coordenadas são usadas para fazer um pedido para a API de forma a obter as coordenadas do estacionamento mais perto do destino como ilustrado na Figura 39.

- Street View:

Para dar a possibilidade ao utilizador de visualizar o parque de estacionamento que selecionou foi usado o Google Maps Street View. Aproveitando a interatividade do Street View, o utilizador pode fazer todo tipo de operações sobre o mapa, por exemplo zoom ou avançar na rua.

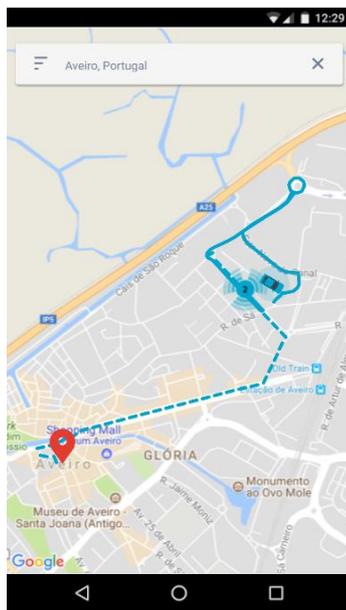


Figura 39: Rota até ao centro de Aveiro

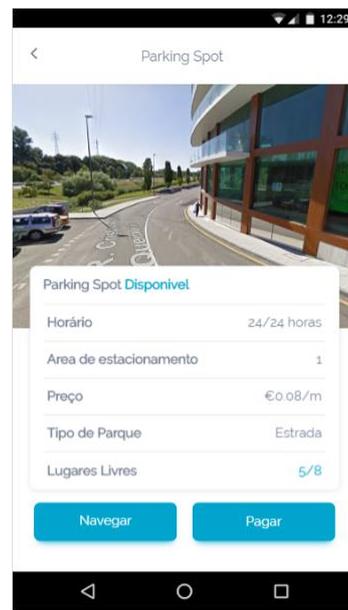


Figura 40: Street View dos Estacionamentos

5.4.3 Integração de Geofence

Esta biblioteca é muito útil no que diz respeito à localização. Olhando para Figura 40, quando um utilizador seleciona a área onde pretende estacionar vai-lhe ser dada a opção de navegar para esse local. Ao navegar será feito um pedido para ser lançada uma aplicação de forma a ler as coordenadas do estacionamento. O utilizador vai sair da aplicação e entrar noutra, nesse momento será necessário ir vigiando a localização do utilizador.



Figura 41: Geofence [64]

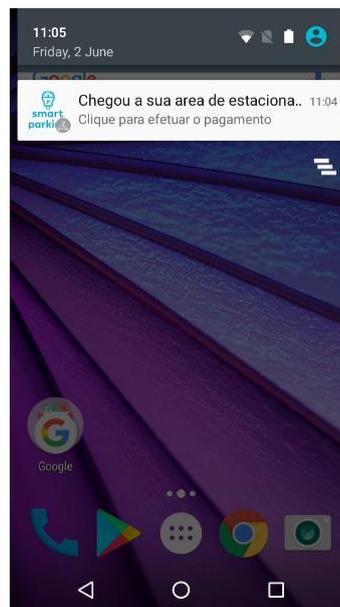


Figura 42: Notificação usando o Geofence

O Geofence [71] serve para isso mesmo, monitorizar a localização do smartphone. Usando a Figura 41 para ajudar a explicar, é definida uma área para a qual se está à espera que o utilizador se dirija. Ao entrar nessa mesma área é disparado um evento que notifica o utilizador desse acontecimento. Neste caso é lançada uma notificação que informa o utilizador que já pode efetuar o pagamento do seu estacionamento. Ao clicar na notificação, representada na Figura 42, o utilizador é logo dirigido para a página de pagamento de estacionamento, evitando assim que tenha de voltar a abrir a aplicação, selecionar o estacionamento e entrar no pagamento do mesmo.

5.4.4 Integração de marker-clustering

Quando um agente de fiscalização abrir aplicação vão-lhe ser mostrados todos os lugares que ele deve verificar, ou seja, todos os lugares que pertencem a todas as áreas das quais ele é responsável pela fiscalização. Para não sobrecarregar o mapa de ícones de várias cores foi introduzido o marker-clustering [72]. Esta ferramenta consegue agrupar os lugares e consoante é feito um zoom in os ícones vão-se separando, tal como pode ser visto na Figura 43. Do lado esquerdo consegue-se perceber que se tem dois conjuntos de estacionamentos, um na Forca composto por 8 lugares e outro mais a norte composto por 2 lugares. Quando o agente se encontra na Forca e ao fazer zoom in, vai ver o estado de todos os lugares pertencentes a esse conjunto individualmente.

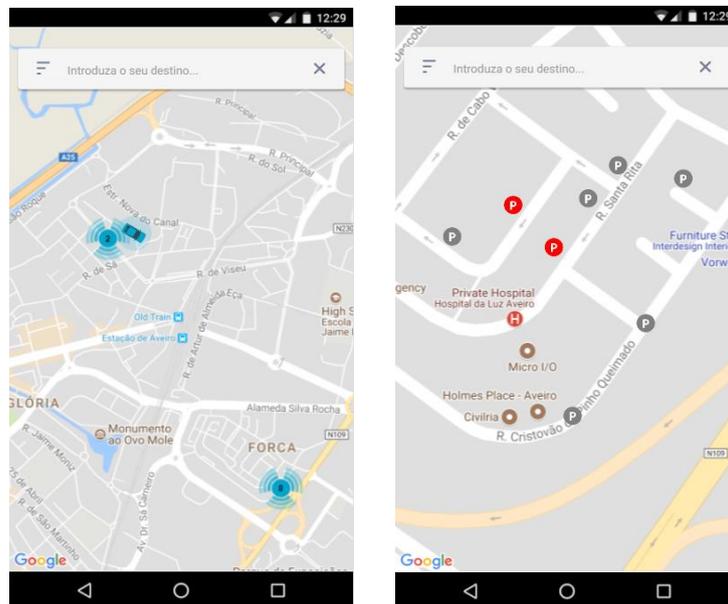


Figura 43: Marker-clustering

Da parte dos técnicos também é usado marker-clustering para agrupar os sensores. Desta forma o utilizador não vê todos os sensores espalhados pelo mapa, mas sim grupos de sensores que se encontram juntos. No que diz respeito ao condutor serão agrupadas as áreas de estacionamento.

5.4.5 Integração de angularjs-slider

Para facilitar a tarefa ao utilizador no que diz respeito à pesquisa de lugares de estacionamento, é dada a opção de filtrar as suas pesquisas. Como nem todos os utilizadores estão disponíveis a pagar o mesmo por um lugar, foram introduzidos os filtros. Nos filtros, o utilizador consegue definir um intervalo de preço dentro do qual pretende obter os resultados da pesquisa de estacionamento. A melhor forma é usar um controlo deslizante duplo, o angularjs-slider [73]. Tendo em conta que um intervalo tem um mínimo e um máximo, o nosso deslizante tem de ser duplo, permitindo ao utilizador deslizar as extremidades até aos limites que pretende definir. Após integrar o angularjs-slider, foi mudado o estilo do mesmo de forma a ficar de acordo com a restante aplicação em termos de cores e formato.

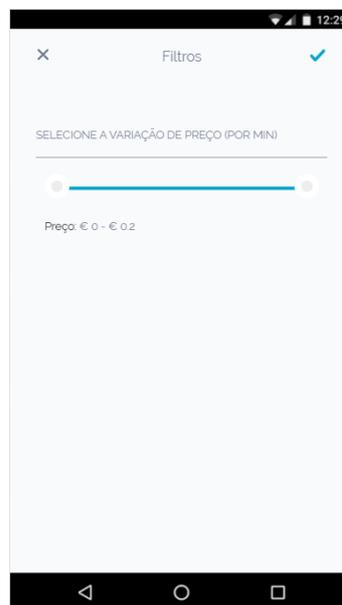


Figura 44: Integração de angularjs-slider

5.4.6 Integração de BarcodeScanner

Com esta biblioteca [74] o utilizador pode ler códigos QR [75] e obter os dados associados ao mesmo, não necessitando assim de introduzir os dados manualmente. Como nem sempre as condições meteorológicas são as melhores, graças aos botões físicos de aumentar e diminuir o volume do dispositivo, os utilizadores conseguem ativar e desativar respetivamente a luz da câmara. Assim sendo, a falta de claridade não será nenhum obstáculo para os nossos utilizadores.

Os agentes de fiscalização vão ter de introduzir os dados dos bilhetes em papel de forma a garantir a gestão completa dos estacionamento. Com a ajuda desta ferramenta conseguem ler o código fornecido no bilhete e assim obter o registo do pagamento sem ser necessário estar a introduzir todos os dados relacionados com o bilhete, tal como ilustrado na Figura 45.

Por seu lado, os técnicos podem usar o BarcodeScanner de forma a ler os códigos QR dos sensores para evitar introduzir o identificador do sensor e assim facilitar a instalação, tal como explicado na secção 5.3.11.

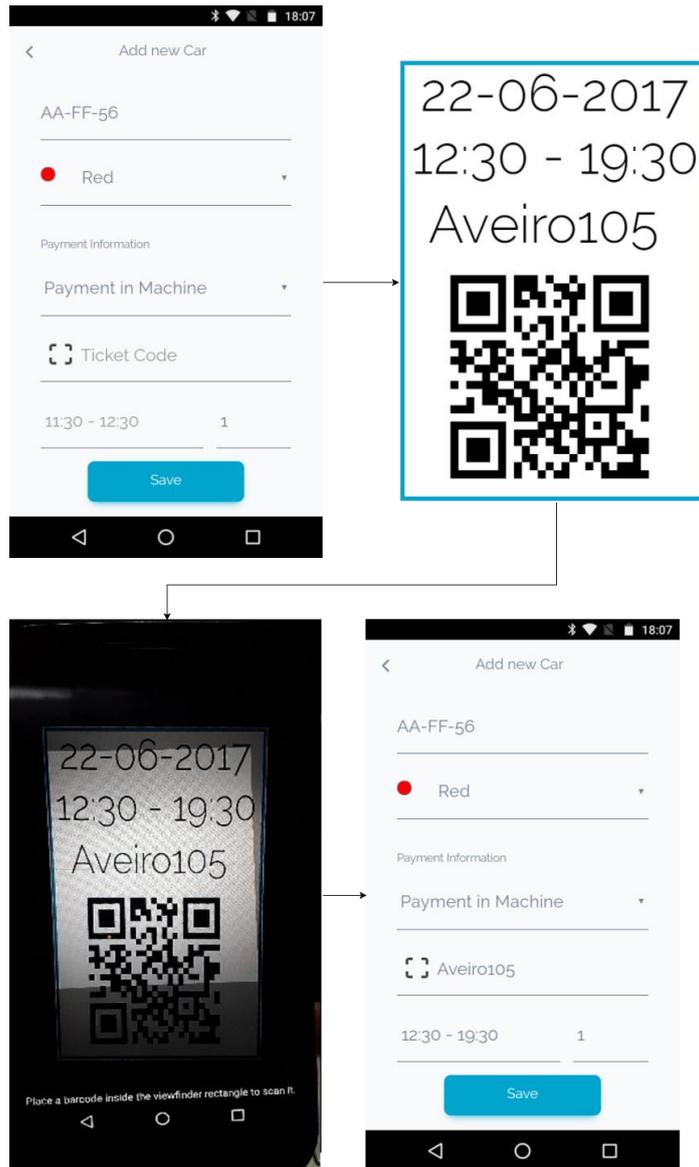


Figura 45: Leitura de bilhete

5.5 Testes

A seguir vão ser apresentados os dois tipos de testes que foram efetuados. Para começar os testes unitários que forma efetuados usando frameworks e depois os testes feitos por utilizadores e os respetivos resultados.

5.5.1 Testes unitários

Os testes unitários foram usados de forma a verificar dados válidos e inválidos na entrada e saída do sistema. Foram criados vários testes com a entrada de dados válidos, de forma a obter resultados corretos e incorretos tal como eram esperados, também foram introduzidos dados inválidos para obter resultados errados. Com estes testes deu para validar pedidos à API, funções, etc.

De forma a efetuar esses testes foram instalados vários frameworks via npm:

- Karma [76] – framework usada para criar testes
- Karma-Jasmine [77] – framework para escrever testes
- Karma-CLI [78] – framework para correr comandos karma na linha de comandos
- Angular-mocks [79]– framework para simular controladores nos testes

Após isso foi criado um ficheiro “my.config.js” com o karma de forma a indicar os caminhos para as bibliotecas, controladores, pedidos à API. Para correr os testes foi usado o browser Google Chrome [80].

```
28 06 2017 14:15:58.584:INFO [karma]: Karma v1.7.0 server started at http://0.0.0.0:9876/
28 06 2017 14:15:58.584:INFO [launcher]: Launching browser Chrome with unlimited concurrency
28 06 2017 14:15:58.601:INFO [launcher]: Starting browser Chrome
28 06 2017 14:16:00.118:INFO [Chrome 60.0.3112 (Windows 10 0.0.0)]: Connected on socket _nt2RxnU2h1DH3kIAAAA with id 32921331
Chrome 60.0.3112 (Windows 10 0.0.0) LoginController: login test2 FAILED
  Expected 'parking.Street Officer' to equal 'parking.Driver'.
    at Object.<anonymous> (Controllers/controllers.tests.js:58:40)
Chrome 60.0.3112 (Windows 10 0.0.0) LoginController: login test3 FAILED
  Expected undefined to equal 'parking.Driver'.
    at Object.<anonymous> (Controllers/controllers.tests.js:71:30)
Chrome 60.0.3112 (Windows 10 0.0.0): Executed 3 of 3 (2 FAILED) (0.011 secs / 0.108 secs)
[14:16:00] Finished 'test' after 2.34 s
```

Figura 46: Teste unitário de login

Após fazer vários testes sobre as várias funcionalidades e os vários controladores da aplicação, e de ter obtido os resultados esperados, vai ser usada a Figura 46 para exemplificar um desses testes. No caso representado na figura mencionada, foram feitos três testes para o caso do login. No primeiro foram introduzidas credenciais de um utilizador com perfil de condutor, no segundo credenciais de um utilizador com perfil de agente de fiscalização e no último forma introduzidas credenciais erradas. O resultado esperado em cada um dos testes era um utilizador com perfil de condutor. Como se pode verificar na figura acima, dois dos três testes não tiveram sucesso.

5.5.2 Testes de utilizadores

De forma a validar as funcionalidades e o bom funcionamento da aplicação, a mesma foi testada em vários cenários. Após partilhar a aplicação com um grupo de 12 utilizadores de várias faixas etárias, foi fornecido a cada um deles um documento com vários casos de uso que os futuros utilizadores da aplicação vão ter no dia-a-dia. Esses cenários foram uma simples alteração de dados pessoais, pesquisas de estacionamento e pagamentos entre outros. Os cenários foram criados de forma a corresponder às ações que os utilizadores da aplicação vão ter no dia-a-dia. Nenhum dos utilizadores conhecia a aplicação nem o propósito da mesma. Não foi explicado a nenhum deles como usar a aplicação, nem indicadas instruções de como fazer as tarefas.

Para cada cenário criado foi pedido aos utilizadores para classificarem a tarefa numa escala de zero a cinco, em que zero corresponde a uma aplicação muito difícil de usar e cinco uma aplicação muito fácil de usar. Em caso de a resposta do utilizador ser inferior a metade da escala foi pedido ao utilizador de deixar um comentário sobre qual a dificuldade encontrada. Após analisar as respostas dos utilizadores que testaram a aplicação e fazendo uma média dos resultados, o parecer foi bastante positivo, sendo o resultado de 4,75/5. No final de cada questionário também foi pedido aos utilizadores de indicarem se já tinham usado uma aplicação com o mesmo propósito e se sim qual. Dos doze utilizadores, apenas um utilizador utilizou uma aplicação deste tipo, a “Via Verde Estacionar”, reconhecendo que a forma de pagar um estacionamento na aplicação de Smart Parking é bastante mais simples e fácil de usar.

Para concluir o questionário foi pedido aos utilizadores para indicarem a sua preferência entre o tradicional parquímetro e a aplicação desenvolvida. As respostas foram unânimes na preferência da Smart Parking Mobile Application.

Olhando para todo o desenvolvimento da aplicação, conclui-se que os objetivos e todos os prazos definidos no início da dissertação foram cumpridos. Olhando para os resultados dos testes, nenhum utilizador teve dificuldades em perceber o funcionamento da aplicação, nem a usar a aplicação e pelo grau de satisfação dos mesmos conclui-se que a aplicação pode conquistar o público facilmente sem criar grandes entraves.

5.6 Dificuldades

As primeiras dificuldades surgiram logo no momento de começar a desenvolver a aplicação, tendo em conta que desconhecia o WebStorm, foi preciso primeiro instalar e depois perceber o funcionamento do mesmo. Depois como nunca tinha programado em Javascript, HTML nem CSS, foi necessário ver todo o funcionamento do mesmo. A seguir foi preciso ver o que era feito em cada módulo e como ligar tudo. Foi necessário seguir alguns tutoriais fornecidos no site do Ionic de forma a perceber como começar um projeto novo e como é escrito todo o código. Assim sendo foram surgindo dúvidas iniciais e algumas dificuldades até mesmo na integração de bibliotecas externas, mas com a ajuda de toda a equipa de desenvolvimento da Ubiwhere consegui rapidamente ambientar-me às ferramentas.

Foi criado um Splashscreen para quando é iniciada a aplicação, o objetivo deste ecrã é mostrar uma imagem ao utilizador quando abrir a aplicação, enquanto a mesma carrega. Após iniciar a aplicação e analisar o comportamento em Android e iOS, conclui-se que ao lançar a aplicação surgia primeiro um ecrã em branco seguido do Splashscreen. A solução encontrada foi a introdução de um ecrã de fundo com a cor de fundo da aplicação que ficará por baixo de todas as views. O Splashscreen é composto por uma imagem animada, com o fundo da mesma cor que o ecrã descrito anteriormente. Esta imagem vai aparecendo linha a linha como se estivesse a ser desenhada por cima do primeiro ecrã até aparecer o logotipo da aplicação.

A introdução de uma biblioteca de calendário e de selecionador de tempo de forma a conseguir introduzir datas e tempos de estacionamento na aplicação, não foi tarefa fácil. Com as opções que existem não fazia sentido criar um selecionador de raiz, assim foram testados vários tipos de métodos de introdução, entre o date picker do Cordova e o ionic-timepicker. Um dos problemas que surgiu nos métodos mencionados foi a personalização e o facto de ter de usar bibliotecas distintas para o caso de selecionar uma data e para o caso de selecionar uma hora.

Além das bibliotecas mencionadas foi analisado o `ion-datetime-picker`, com esta ferramenta foi possível definir intervalos de datas, integrar um selecionador de datas e de horas e até personalizar o mesmo. Assim sendo, criei modelos com os diversos idiomas da aplicação, para cada caso foi introduzido um título de forma a facilitar a tarefa ao utilizador ao ter sempre visibilidade sobre o que está a seleccionar. Não foi óbvio perceber a estrutura dos dados e como alterar os mesmos.

Outra dificuldade que surgiu foi o uso do leitor de códigos QR. Inicialmente foi instalada uma biblioteca [81] com a qual foi possível compilar o código sem erros e usar o leitor. Alguns meses depois, com a instalação de uma atualização da biblioteca surgiu um erro no código, que ao compilar gerava duas entradas duplicadas no Android Manifest provocando assim um erro de compilação. Com uma simples alteração ao Android Manifest é possível corrigir o erro, ficando o problema temporariamente corrigido até o código voltar a ser compilado e voltar a haver a mesma entrada duplicada no Android Manifest. De forma a evitar este erro e não ser necessário alterar o Android Manifest a cada compilação do código a biblioteca foi trocada por uma nova [82].

6. Conclusão

As tecnologias hoje em dia estão sempre a evoluir e as pessoas começam a gerir tudo que as rodeia no dia-a-dia à base de simples cliques. Como os smartphones também continuam a evoluir e a acompanhar as novas tecnologias já é possível usar um único dispositivo para isso tudo. As Smart Cities estão a ganhar força na nossa sociedade e a tornarem-se o futuro, tornando necessário usar as novas tecnologias para conseguir gerir as mesmas.

O grande objetivo desta dissertação era criar uma aplicação que se enquadrava no vertical das Smart Cities e ser usada pela Ubiwhere para vender aos seus clientes. Existindo já um sistema de estacionamento foi necessário redefinir algumas coisas e criar uma aplicação móvel para ser usada de forma simples e cómoda. Com os smartphones pessoais as pessoas vão conseguir procurar lugares de estacionamento, efetuar os pagamentos e gerir os tempos de ocupação. De forma a rentabilizar o investimento feito por parte dos clientes junto da Ubiwhere, as empresas vão conseguir monitorizar muito facilmente os estacionamento e saber onde é que está a haver movimento. Desta forma os fiscais conseguem identificar rapidamente a localização certa dos lugares onde poderão estar a ser cometidas infrações. Com este sistema implementado, o incentivo aos condutores a pagar os seus estacionamento também será crescente, tendo em conta que no preciso momento que estacionarem os fiscais vão obter essa informação. Assim sendo, se não efetuarem o devido pagamento a probabilidade de serem autuados vai crescer exponencialmente. A aplicação vai oferecer autonomia total aos investidores, podendo gerir os próprios lugares, introduzir novos lugares, remover ou até mesmo atualizar os preços, sem necessitar de intervenção da parte da Ubiwhere.

Antes de começar o desenvolvimento da aplicação foi-me dada liberdade de optar por criar uma aplicação nativa, seja Android ou iOS, ou então híbrida, ficando ao meu critério a escolha. Tendo em conta que o grande objetivo desta dissertação era criar uma aplicação para ser usada da parte dos clientes da Ubiwhere, não quis restringir os utilizadores a um único sistema operativo. O facto de nunca ter desenvolvido uma aplicação híbrida, uma vez que já tinha criado uma aplicação Android e outra iOS numa unidade curricular de opção de mestrado também pesou na

decisão, e assim decidi optar por uma aplicação híbrida de forma a ampliar horizontes. Durante o desenvolvimento sempre que surgiu uma dúvida ou dificuldade em relação ao código houve sempre apoio por parte de toda a equipa da Ubiwhere. Houve sempre disponibilidade para ouvir as minhas ideias e deixar-me implementar as mesmas.

O facto de a dissertação ter sido desenvolvida no âmbito empresarial foi uma mais-valia, foi possível enquadrar a dissertação num ambiente real e manter o foco nas necessidades dos clientes da Ubiwhere. Com esta aplicação a Ubiwhere pretende trazer uma mais-valia ao Smart Parking e apresentar o lado mais rápido e cómodo de todo o processo. Havendo já um principal objetivo definido, foi-me dada liberdade por parte da Ubiwhere em definir a aplicação, os restantes objetivos e o fluxo da mesma, sendo tudo isso depois discutidos em conjunto com os colaboradores da empresa de forma a limar todas as arestas e a deixar tudo bem definido. Após isso, foram desenhados os esboços correspondentes por um designer da Ubiwhere de forma a criar ecrãs user-friendly que respeitam as normas da empresa.

7. Trabalho Futuro

O objetivo desta dissertação era criar uma aplicação que no futuro venha a substituir os parquímetros. Atualmente não é possível pelo simples facto que ainda existe uma minoria de pessoas que não tem smartphones.

Num futuro próximo terá de ser criada uma API de recuperação de password ou de redefinição da mesma. Atualmente no backend só é permitido alterar a senha de um utilizador depois de um administrador ou o mesmo aceder à sua própria conta. Este mecanismo obriga um utilizador a ter de contactar o administrador do sistema para obter uma senha provisória e depois ser adicionada uma opção no perfil do utilizador de alterar a senha. Quando todo este mecanismo estiver criado será adicionado um novo botão no ecrã de login, de forma a introduzir o email para redefinir a mesma.

Neste momento também não existe nenhuma forma de efetuar o pagamento dos estacionamento. No futuro quando o vertical dos pagamentos estiver criado e a aplicação do mesmo concluída, então terá de ser integrado no Smart Parking. Nas definições já foi criada a opção de gerir os métodos de pagamento, sendo que o mesmo não tem nenhuma funcionalidade atualmente. O objetivo é introduzir os vários métodos de pagamento, sendo que depois a disponibilização ou não dos mesmos será definida pelas entidades que optarem pelo Smart Parking. Poderá haver métodos diferentes para cada uma delas, entre cartões de crédito, integração de PayPal, ou outras.

No lançamento de uma nova versão do Smart Parking também poderá ser integrada uma nova forma de adicionar veículos. Nesta versão o utilizador poderá tirar uma foto de frente ao seu veículo de forma a processar a imagem e obter a matrícula e a cor do carro ou até mesmo a marca. Com esta função pretende-se automatizar o processo de registo de veículos por parte dos utilizadores. A foto teria de ser processada de forma a extrair todos os dados necessários. Os campos relativos aos mesmos seriam preenchidos de forma automática, mas deixando a

possibilidade de edição por parte do utilizador assumido um possível erro durante o processo de reconhecimento de imagem.

Para tentar chegar a mais utilizadores no futuro e ajudar os mesmos a conhecer melhor a aplicação, poderá ser integrada uma forma de registo de bilhete em papel pelos próprios condutores. Ao efetuar o pagamento num parquímetro, o utilizador vai obter um tradicional bilhete em papel. Esse bilhete poderá ser registado pelo utilizador na aplicação com uma simples leitura do código QR e associar a um veículo que terá de estar registado. Assim o condutor consegue visualizar o tempo restante do bilhete e ativar um alarme. Tendo a possibilidade de a qualquer momento então optar por prolongar a sessão usando a aplicação e não necessitar de se deslocar ao seu veículo.

8. Referências

- [1] A. Garric, “Un milliard de voitures dans le monde, la Chine fait la course en tête.” [Online]. Available: <http://ecologie.blog.lemonde.fr/2011/08/26/un-milliard-de-voitures-dans-le-monde-la-chine-fait-la-course-en-tete/>. [Accessed: 12-Oct-2016].
- [2] “Micro I/O.” [Online]. Available: <http://www.microio.pt/>. [Accessed: 14-Jun-2017].
- [3] “Wavecom.” [Online]. Available: <https://www.wavecom.pt/>. [Accessed: 14-Jun-2017].
- [4] “Citibrain.” [Online]. Available: <http://www.citibrain.com/pt/>. [Accessed: 14-Jun-2017].
- [5] “Smart Waste.” [Online]. Available: <http://www.citibrain.com/pt/solutions/smart-waste-pt/>. [Accessed: 14-Jun-2017].
- [6] “Smart Air Quality.” [Online]. Available: <http://www.citibrain.com/pt/solutions/smart-air-quality-pt/>. [Accessed: 14-Jun-2017].
- [7] “Smart Traffic.” [Online]. Available: <http://www.citibrain.com/pt/solutions/smart-traffic-pt/>. [Accessed: 14-Jun-2017].
- [8] “Ubiwhere | Research and Innovation | Idea to Product | User-centered Solutions.” [Online]. Available: <http://www.ubiwhere.com/en>. [Accessed: 28-Sep-2016].
- [9] R. E. Knack, “Pay as you park,” *Planning*, vol. 71, no. 5, pp. 4–8, 2005.
- [10] D. C. Shoup, “Cruising for parking,” *Transp. Policy*, vol. 13, no. 6, pp. 479–486, 2006.
- [11] Citroën Portugal, “Citroën Inspired By You!,” 2016. [Online]. Available: <https://www.youtube.com/watch?v=9-pfStwUjpl>. [Accessed: 19-Oct-2016].
- [12] E. Gantelet and A. Lefauconnier, “The time looking for a parking space: Strategies, associated nuisances and stakes of parking management in France,” *Proc. ETC*, pp. 1–7, 2006.
- [13] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, “iParker—A New Smart Car-Parking System Based on Dynamic Resource Allocation and Pricing,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2637–2647, 2016.

- [14] "Wellness Telecom." [Online]. Available: <http://www.wtelecom.es>. [Accessed: 12-Oct-2016].
- [15] "Parkwhiz." [Online]. Available: <http://www.parkwhiz.com/>. [Accessed: 28-Sep-2016].
- [16] "Parkme." [Online]. Available: <https://www.parkme.com/pt/>. [Accessed: 28-Sep-2016].
- [17] "Google Play." [Online]. Available: <https://play.google.com/store>. [Accessed: 12-Oct-2016].
- [18] "Smart Parking Limited (ASX:SPZ)." [Online]. Available: <http://www.smartparking.com>. [Accessed: 28-Sep-2016].
- [19] "iTunes." [Online]. Available: <https://www.apple.com/itunes/>. [Accessed: 13-Oct-2016].
- [20] S. R. Basavaraju, "Automatic Smart Parking System using Internet of Things (IOT)," *Int. J. Sci. Res. Publ.*, vol. 5, no. 12, pp. 629–632, 2015.
- [21] A. Khanna and R. Anand, "IoT based smart parking system," 2016 *Int. Conf. Internet Things Appl.*, pp. 266–270, 2016.
- [22] "Mobile City." [Online]. Available: <https://www.mobilecity.de/fuer-privatkunden/handyparken.html>. [Accessed: 16-Oct-2016].
- [23] "Micropayment." [Online]. Available: <https://www.w3.org/TR/Micropayment-Markup/>. [Accessed: 20-Oct-2016].
- [24] "Micropaiement." [Online]. Available: <https://fr.wikipedia.org/wiki/Micropaiement>. [Accessed: 19-May-2017].
- [25] "Plus de 5 milliards de possesseurs d'un téléphone portable en 2017." [Online]. Available: http://www.lepoint.fr/monde/plus-de-5-milliards-de-possesseurs-d-un-telephone-portable-en-2017--27-02-2017-2107993_24.php. [Accessed: 24-May-2017].
- [26] "Bitcoin." [Online]. Available: <https://bitcoin.org/en/>. [Accessed: 20-Oct-2016].
- [27] "PayPal." [Online]. Available: <https://www.paypal.com/pt/home>. [Accessed: 20-Oct-2016].
- [28] "Netcaixa Online." [Online]. Available: <https://www.netcaixa.pt/>. [Accessed: 20-Oct-2016].
- [29] "CGD 3D Secure." [Online]. Available: <https://www.cgd.pt/Particulares/Em-Campanha/Pages/3D-Secure-Service.aspx>. [Accessed: 20-Oct-2016].
- [30] "Google Inc." [Online]. Available: <https://www.google.com/about/company/>. [Accessed: 20-Oct-2016].
- [31] "Apple Inc." [Online]. Available: <http://www.apple.com/pt/>. [Accessed: 20-Oct-2016].
- [32] "Android Pay." [Online]. Available: <https://www.android.com/pay/>. [Accessed: 20-Oct-2016].
- [33] "Apple Pay." [Online]. Available: <http://www.apple.com/apple-pay/>. [Accessed: 20-Oct-2016].
- [34] "Google Wallet." [Online]. Available: <https://www.google.com/wallet/>. [Accessed: 20-Oct-

- 2016].
- [35] "Google Wallet (Wikipedia)." [Online]. Available: https://en.wikipedia.org/wiki/Google_Wallet. [Accessed: 20-Oct-2016].
- [36] S. Ember, "Blockchain Is Latest Bitcoin Start-Up to Lure Big Investment," The New York Times, 2014. [Online]. Available: http://dealbook.nytimes.com/2014/10/06/blockchain-is-latest-bitcoin-start-up-to-lure-big-investment/?_r=0. [Accessed: 19-Mar-2017].
- [37] "The New York Times." [Online]. Available: <https://www.nytimes.com/>. [Accessed: 31-May-2017].
- [38] "Blockchain." [Online]. Available: <http://www.coindesk.com/information/what-is-blockchain-technology/>. [Accessed: 20-Jun-2016].
- [39] "Blockchain." [Online]. Available: <http://www.investopedia.com/terms/b/blockchain.asp>. [Accessed: 19-May-2017].
- [40] L. Siciliano, "Bring on Bitcoin: why Richard Branson and Bill Gates support the currency," The Telegraph, 2014. [Online]. Available: <http://www.telegraph.co.uk/finance/currency/11148160/Bring-on-Bitcoin-why-Richard-Branson-and-Bill-Gates-support-the-currency.html>. [Accessed: 17-Mar-2017].
- [41] Bloomberg, "Bring on Bitcoin: Why Branson, Gates Support the Currency," 2014. [Online]. Available: <http://www.bloomberg.com/news/videos/2014-10-07/why-richard-branson-and-bill-gates-support-bitcoin>. [Accessed: 17-Mar-2017].
- [42] M. J. Casey, "Blockchain Lands Biggest Ever Venture Funding Round in Bitcoin Industry," The Wall Street Journal, 2014. [Online]. Available: <http://www.wsj.com/articles/blockchain-lands-biggest-ever-venture-funding-round-in-bitcoin-industry-1412687156?tesla=y>. [Accessed: 19-Mar-2017].
- [43] "Blockchain: o que é e como funciona," 2016. [Online]. Available: <http://computerworld.com.br/blockchain-o-que-e-e-como-funciona>. [Accessed: 26-Oct-2016].
- [44] "Netbill." [Online]. Available: <http://www.netbill.com/>. [Accessed: 02-Apr-2017].
- [45] M. F. Wu, "Micropayment Protocol."
- [46] "Kerberos." [Online]. Available: http://web.mit.edu/kerberos/#what_is. [Accessed: 31-May-2017].
- [47] "Smartphone Market Share." [Online]. Available: <http://www.idc.com/promo/smartphone-market-share/os>. [Accessed: 05-Apr-2017].
- [48] "Xamarin." [Online]. Available: <https://www.xamarin.com/>. [Accessed: 17-Nov-2016].
- [49] "React-Native Github." [Online]. Available: <https://github.com/facebook/react-native>.

[Accessed: 15-Jun-2017].

- [50] "Ionic." [Online]. Available: <https://ionic.io/>. [Accessed: 17-Nov-2016].
- [51] "Meshblu." [Online]. Available: <https://meshblu.readme.io/>. [Accessed: 16-Nov-2016].
- [52] "RabbitMQ." [Online]. Available: <https://www.rabbitmq.com/>. [Accessed: 16-Nov-2016].
- [53] "Celeryproject." [Online]. Available: <http://www.celeryproject.org/>. [Accessed: 16-Nov-2016].
- [54] "Python." [Online]. Available: <https://www.python.org/>. [Accessed: 16-Nov-2016].
- [55] "OAuth2.0." [Online]. Available: <https://oauth.net/2/>. [Accessed: 16-Nov-2016].
- [56] "Swagger." [Online]. Available: <http://swagger.io/>. [Accessed: 31-May-2017].
- [57] "Node.js." [Online]. Available: <https://nodejs.org/en/>. [Accessed: 19-May-2017].
- [58] "Cordova." [Online]. Available: <https://cordova.apache.org/>. [Accessed: 19-Jun-2017].
- [59] "Ionicframework Getting-started." [Online]. Available: <https://ionicframework.com/getting-started/>. [Accessed: 02-May-2017].
- [60] "Jetbrains." [Online]. Available: <https://www.jetbrains.com/>. [Accessed: 31-May-2017].
- [61] "WebStorm." [Online]. Available: <https://www.jetbrains.com/webstorm/>. [Accessed: 31-May-2017].
- [62] "Bower." [Online]. Available: <https://www.npmjs.com/package/bower>. [Accessed: 02-Jun-2017].
- [63] "Gitlab.ubiwhere.com." [Online]. Available: <https://gitlab.ubiwhere.com/>. [Accessed: 31-May-2017].
- [64] "Lista de Indicativos." [Online]. Available: <https://gist.githubusercontent.com/Goles/3196253/raw/9ca4e7e62ea5ad935bb3580dc0a07d9df033b451/CountryCodes.json>. [Accessed: 20-Mar-2017].
- [65] "Google Maps Polygon." [Online]. Available: <https://developers.google.com/maps/documentation/javascript/examples/polygon-simple>. [Accessed: 22-May-2017].
- [66] "Google Maps Polyline." [Online]. Available: <https://developers.google.com/maps/documentation/javascript/examples/polyline-simple>. [Accessed: 22-May-2017].
- [67] "Ionic Diagram." [Online]. Available: <https://pt.slideshare.net/gursesl/crossplatform-mobile-development-with-ionic-framework-and-angular/14?smtNoRedir=1>. [Accessed: 22-Jun-2017].
- [68] "ion-datetime-picker." [Online]. Available: <https://github.com/katemihalikova/ion-datetime->

- picker. [Accessed: 02-Jun-2017].
- [69] “Google Maps API.” [Online]. Available: <https://developers.google.com/maps/?hl=pt>. [Accessed: 02-Jun-2017].
- [70] “Geocoder.” [Online]. Available: <https://developers.google.com/maps/documentation/geocoding/intro?hl=pt>. [Accessed: 02-Jun-2017].
- [71] “Geo-fence.” [Online]. Available: https://applidium.com/en/news/whats_geofencing/. [Accessed: 02-Jun-2017].
- [72] “Marker-clustering.” [Online]. Available: <https://developers.google.com/maps/documentation/javascript/marker-clustering>. [Accessed: 15-May-2017].
- [73] “angularjs-slider.” [Online]. Available: <https://github.com/angular-slider/angularjs-slider>. [Accessed: 08-May-2017].
- [74] “Leitor QR.” [Online]. Available: <https://github.com/wildabeast/BarcodeScanner.git%0A>. [Accessed: 22-Jun-2017].
- [75] “Código QR.” [Online]. Available: <http://g1.globo.com/tecnologia/noticia/2011/05/entenda-o-que-sao-os-qr-codes-codigos-lidos-pelos-celulares.html>. [Accessed: 02-Jun-2017].
- [76] “Karma.” [Online]. Available: <https://www.npmjs.com/package/karma>. [Accessed: 20-Jun-2017].
- [77] “Karma-Jasmine.” [Online]. Available: <https://www.npmjs.com/package/karma-jasmine>. [Accessed: 20-Jun-2017].
- [78] “Karma-CLI.” [Online]. Available: <https://www.npmjs.com/package/karma-cli>. [Accessed: 20-Jun-2017].
- [79] “Angular-Mocks.” [Online]. Available: <https://www.npmjs.com/package/angular-mocks>. [Accessed: 20-Jun-2017].
- [80] “Google Chrome.” [Online]. Available: <https://www.google.com/chrome/index.html>. [Accessed: 20-Jun-2017].
- [81] “\$cordovaBarcodeScanner.” [Online]. Available: <http://ngcordova.com/docs/plugins/barcodeScanner/>. [Accessed: 22-Jun-2017].
- [82] “BarcodeScanner.” [Online]. Available: <https://github.com/wildabeast/BarcodeScanner.git>. [Accessed: 22-Jun-2017].