



**EDGAR RAFAEL  
BOTTO DOS  
SANTOS**

**PLATAFORMA DE  
CARPOOLING PARA A  
MOBILIDADE URBANA  
SUSTENTÁVEL NO IPS**

Relatório de projeto final de curso de  
Licenciatura em Engenharia Informática Ramo  
de Engenharia de Software

**ORIENTADORES**

Prof. Doutor Rui Madeira

Prof.<sup>a</sup> Doutora Patrícia Macedo

Setúbal, novembro de 2021

# Relatório de Projeto Final de Curso

**EDGAR RAFAEL  
BOTTO DOS  
SANTOS**

**PLATAFORMA DE  
CARPOOLING PARA A  
MOBILIDADE URBANA  
SUSTENTÁVEL NO IPS**

## JÚRI

Presidente: Prof. Doutor Bruno Miguel Nunes da Silva - ESTSetúbal

Argente: Prof. Doutor Cláudio Miguel Garcia Loureiro dos Santos Sapateiro - ESTSetúbal

Orientador: Prof. Doutor Rui Miguel Neves Gonçalves Madeira - ESTSetúbal

Setúbal, novembro de 2021

*“If you are not willing to be a fool,  
you can’t become a master.”*

*- Jordan B. Peterson*

## Agradecimentos

Ao longo deste percurso académico conheci muitas pessoas que me marcaram como pessoa e que me levaram ao que sou hoje, e que de forma direta ou indireta ajudaram na concretização deste Projeto Final de Curso.

Primeiramente, gostaria de agradecer a toda a minha família pelo continuo apoio, reconhecimento e admiração que demonstraram ter nas minhas conquistas e objetivos ao longo desta jornada.

Agradeço ao Instituto Politécnico de Setúbal por possibilitar continuar a minha aprendizagem no ensino superior através de uma Licenciatura e, consequentemente, a todos os docentes, que apesar da situação pandémica que perdurou ao longo de pelo menos metade do meu percurso neste curso, fizeram um esforço para conseguirem continuar a transmitir o seu conhecimento da melhor forma possível.

Quero agradecer ao centro de investigação Sustain.RD, do Instituto Politécnico de Setúbal, nomeadamente ao Prof. Rui Madeira e à Prof.<sup>a</sup> Patrícia Macedo por proporcionarem-me a oportunidade de realização deste projeto, pela disponibilidade e paciência para orientarem o mesmo apesar da larga quantidade de trabalho que geralmente têm, e por terem sido excelentes pessoas ao motivarem-me e aconselharem-me ao longo deste processo.

Deixo um agradecimento a todos os meus amigos e colegas que sempre estiveram presentes para me ajudar e que tornaram este percurso especial com todos os momentos que passámos juntos.

Por último, deixo um agradecimento especial aos meus amigos André Pereira, Gonçalo Pereira, Henoch Vitureira e Nicole Fernandes, por me terem apoiado e inspirado quando mais precisei ao longo deste ciclo de estudos.

O meu, Muito Obrigado!

## Resumo

Respeitando os objetivos de sustentabilidade do Instituto Politécnico de Setúbal, um dos seus centros de investigação, o Sustain.RD, propôs o desenvolvimento de uma plataforma para melhorar a mobilidade dos membros da sua comunidade através do carpooling, reduzindo deste modo o número de veículos pessoais utilizados como principal meio de transporte, diminuindo as emissões de dióxido de carbono emitidas pelos mesmos e promovendo uma maior consciencialização ambiental. O carpooling é um meio de um motorista partilhar o seu veículo pessoal de forma informal com mais pessoas, possibilitando a eventual redução de custos.

O foco do projeto aqui descrito foi o desenvolvimento de uma primeira versão desta plataforma, tendo especial atenção no desenvolvimento das funcionalidades de maior importância, tais como, a oferta de boleias, respetiva procura e eventual pedido de participação nas mesmas. O estudante, em conjunto com o seu colega da equipa de desenvolvimento e os orientadores do projeto, prepararam o planeamento de todas as funcionalidades consideradas relevantes que a plataforma deveria ter para ser considerada um produto totalmente desenvolvido, das quais foram selecionadas um conjunto de funcionalidades prioritárias para compor a primeira versão funcional com o propósito de se realizarem testes de usabilidade com uma amostra da comunidade do Instituto. Esta primeira versão, tendo as funcionalidades principais implementadas, servirá como uma base para a adição de novos módulos que poderão enriquecer a interação e motivar os utilizadores.

Tendo em vista a necessidade de disponibilizar esta plataforma aos membros da comunidade independentemente do dispositivo móvel que possuam, decidiu-se utilizar o Flutter, uma *framework* multiplataforma que fornece um kit de desenvolvimento integrado tanto para o sistema operativo móvel Android como para o iOS. Tratando-se de uma aplicação móvel existiu a necessidade de se ter um serviço para alimentar a mesma, serviço este que terá sido desenvolvido com recurso ao Node.js, sendo esta a principal responsabilidade do estudante. Com o propósito de melhorar a organização e visualização do trabalho a ser realizado, converteu-se os requisitos funcionais levantados em histórias de utilizadores e subsequentes tarefas que foram afixadas num quadro inerente à *framework* ágil Kanban.

O resultado deste projeto foi uma plataforma que permite a criação e oferta de boleias para que os utilizadores possam pesquisar e candidatar-se às mesmas. A aplicação móvel facilita a visualização deste processo com recurso a mapas interativos que apresentam a localização geográfica dos pontos de encontro das boleias criadas e respetivas rotas de viagem. Através do sistema Azure Active Directory, utilizado pela Instituto Politécnico de Setúbal, a plataforma permite apenas o acesso aos membros do Instituto, oferecendo uma segurança adicional.

**Palavras-chave:** Carpool, Mobilidade, Sustentabilidade, Partilha de Boleias, Flutter, Node.JS

## Abstract

Meeting the sustainability objectives of the Polytechnic Institute of Setúbal, one of its research centres, Sustain.RD, proposed the development of a platform to improve the mobility of members of its community through carpooling, thus reducing the number of personal vehicles used as the main means of transport, reducing carbon dioxide emissions emitted by them and promoting greater environmental awareness. Carpooling is a way for a driver to share his personal vehicle informally with more people, allowing for possible cost savings.

The focus of the project described here was the development of a first version of this platform, with special attention to the development of the most important features, such as the rides offering, respective searching and possible requests to participate in them. The student, together with his colleague from the development team and the project supervisors, prepared the planning of all the functionalities considered relevant that the platform should have in order to be considered a fully developed product, from which a set of priority functionalities were selected to compose the first functional version with the purpose of conducting usability tests with a sample of the Institute's community. This first version, having the main functionalities implemented, will serve as a basis for the addition of new modules that may enrich interaction and motivate users.

In view of the need to make this platform available to members of the community regardless of the mobile device they have, it was decided to use Flutter, a multiplatform framework that provides an integrated development kit for both the Android and iOS mobile operating systems. Since this is a mobile application there was the need to have a service to feed it, a service that will have been developed using Node.js, being this the student's main responsibility. With the purpose of improving the organization and visualization of the work to be done, the raised functional requirements were converted into user stories and subsequent tasks that were posted on a board inherent to the agile framework Kanban.

The result of this project was a platform that allows the creation and offer of rides so that users can search and apply for them. The mobile application facilitates the visualisation of this process using interactive maps that display the geographical location of the meeting points of the rides created and associated travel routes. Through the Azure Active Directory system used by the Polytechnic Institute of Setúbal, the platform only allows access to members of the Institute. Providing additional security.

**Keywords:** Carpool, Mobility, Sustainability, Ride Sharing, Flutter, Node.JS

## Índice

1.	Introdução .....	1
1.1.	Motivação .....	1
1.2.	Objetivos do Projeto .....	2
1.3.	Estrutura do Relatório .....	2
2.	Enquadramento do Projeto .....	4
2.1.	Apresentação da Entidade Acolhedora.....	4
2.1.1.	IPS – Instituto Politécnico de Setúbal .....	4
2.1.2.	Sustain.RD - Research Center for Engineering a Sustainable Development .....	6
2.2.	Estudo do Interesse Comunitário.....	7
2.3.	Soluções Comerciais Concorrentes .....	8
2.3.1.	Singu Carpooling.....	8
2.3.2.	Comovee .....	9
2.3.3.	Poola .....	10
2.4.	Metodologia de Organização do Projeto.....	11
2.4.1.	Agile.....	11
2.4.2.	Kanban .....	13
3.	Fundamentos Tecnológicos .....	16
3.1.	Tecnologias de Desenvolvimento .....	16
3.1.1.	API RESTful (back-end) .....	16
3.1.1.1.	JavaScript.....	16
3.1.1.2.	Node.js .....	16
3.1.2.	Aplicação Móvel (front-end) .....	17
3.1.2.1.	Dart.....	17
3.1.2.2.	Flutter .....	17
3.1.3.	Persistência de Dados .....	18
3.1.3.1.	MongoDB.....	18
3.1.4.	Notificações .....	18
3.1.4.1.	APNs – Apple Push Notification Service .....	18

3.1.4.2. FCM – Firebase Cloud Messaging.....	19
3.1.5. Gestão de Autenticação .....	19
3.1.5.1. Azure Active Directory .....	19
3.1.5.2. Microsoft Graph API .....	19
3.1.5.3. JSON Web Token .....	20
3.2. Padrões de Software.....	20
3.2.1. Back-end .....	20
3.2.1.1. MVC ( <i>Model-View-Controller</i> ) .....	20
3.2.1.2. REST ( <i>Representational State Transfer</i> ) .....	21
3.2.2. Front-end .....	23
4. Análise e Especificação do Projeto.....	25
4.1. Equipa e Responsabilidades.....	25
4.2. Planeamento das Atividades do Projeto .....	26
4.2.1. Entregas Previstas .....	27
4.2.2. Cronograma Inicial .....	27
4.2.3. Cronograma Real .....	28
4.3. Análise de Âmbito e Especificação de Módulos .....	29
4.3.1. <i>Brainstorming</i> .....	29
4.3.2. Módulos .....	31
4.4. Análise e Especificação de Requisitos .....	32
4.4.1. Requisitos Funcionais .....	32
4.4.1.1. Módulo de Perfil .....	32
4.4.1.2. Módulo de Boleias .....	33
4.4.1.3. Módulo de Gestão de Notificações .....	34
4.4.2. Requisitos Não Funcionais.....	34
4.4.2.1. Requisitos de Qualidade .....	34
4.4.2.2. Requisitos Ambientais .....	37
4.5. Processos de Negócio .....	38
4.5.1. Pesquisa e Pedido de Boleia pelo Passageiro .....	38
4.6. User Stories.....	40

4.7.	Riscos.....	42
4.8.	Interface com o Utilizador.....	43
4.8.1.	Mockups .....	43
4.8.2.	Normas .....	45
4.9.	Persistência de Dados .....	46
4.9.1.	Modelo de Dados Planeado .....	46
4.9.2.	Modelo de Dados Implementado .....	48
4.10.	Arquitetura Geral .....	49
4.11.	Arquitetura Física .....	50
4.11.1.	Diagrama de Componentes .....	51
4.11.2.	Diagrama de Instalação .....	52
4.12.	Diagramas de Estados .....	54
4.13.	Normas de Codificação do Back-end.....	54
4.14.	Custos .....	55
5.	Descrição do Trabalho Realizado.....	57
5.1.	Configuração do Jira Software .....	57
5.2.	API RESTful .....	64
5.2.1.	Preparação da Instância .....	65
5.2.2.	Base de Dados .....	68
5.2.3.	Autenticação.....	70
5.2.4.	Notificações .....	71
5.2.5.	Perfil de Utilizador .....	73
5.2.6.	Bloqueio de Utilizadores.....	74
5.2.7.	Grupos de Boleia.....	74
5.3.	Aplicação Móvel .....	79
5.3.1.	Notificações .....	79
5.3.2.	Mapa Principal.....	80
5.3.3.	Gestão de Veículos .....	82
6.	Discussão do Trabalho Realizado .....	84
6.1.	Validação da Solução.....	84

6.1.1. Testes de Integração.....	84
6.1.2. Testes de Sistema.....	85
6.1.3. Testes de Usabilidade.....	85
6.2. Avaliação do Projeto .....	87
7. Conclusão e Trabalho Futuro .....	89
8. Referências Bibliográficas .....	91
9. Anexos .....	95
9.1. Anexo 1 – Apuração dos Dados Referentes ao Inquérito de Perceção Ambiental em Relação à Mobilidade da Comunidade IPS.....	95
9.2. Anexo 2 – Tarefas Extraídas das User Stories .....	101
9.3. Anexo 3 – Restantes Requisitos Funcionais Levantados.....	103
9.4. Anexo 4 – Questionário de Testes de Usabilidade .....	110
9.5. Anexo 5 – Manual de Utilizador da Aplicação Móvel.....	116

## Índice de Figuras

Figura 2.1 - Organigrama dos órgãos do IPS (retirado de <a href="https://www.ips.pt/ips_si/web_base.gera_pagina?P_pagina=31508">https://www.ips.pt/ips_si/web_base.gera_pagina?P_pagina=31508</a> ) .....	5
Figura 2.2 - Ecrãs da aplicação móvel Singu Carpooling para iPhone (retirado de <a href="https://apps.apple.com/us/app/id922439656">https://apps.apple.com/us/app/id922439656</a> ) .....	9
Figura 2.3 - Exemplo de um ecrã da aplicação Comovee (retirada de <a href="https://www.comovee.com/rideshare-software">https://www.comovee.com/rideshare-software</a> ) .....	10
Figura 2.4 - Ecrãs da aplicação móvel Poola (retirado de <a href="https://poola.app">https://poola.app</a> ) .....	11
Figura 2.5 - Exemplo de um quadro Kanban (retirado de <a href="https://www.scrumportugal.pt/kanban-board">https://www.scrumportugal.pt/kanban-board</a> ) .....	15
Figura 3.1 - Diagrama do padrão MVC (retirado de <a href="https://techterms.com/definition/mvc">https://techterms.com/definition/mvc</a> ) .....	21
Figura 3.2 - Diagrama simplificado do padrão BLoC (retirado de <a href="https://bloclibrary.dev/#/architecture">https://bloclibrary.dev/#/architecture</a> ) .....	24
Figura 3.3 - Diagrama simplificado do padrão Cubit (retirado de <a href="https://bloclibrary.dev/#/coreconcepts?id=cubit">https://bloclibrary.dev/#/coreconcepts?id=cubit</a> ) .....	24
Figura 4.1 - Cronograma Real das Atividades .....	28
Figura 4.2 - Processo de negócio de procura de boleia .....	39
Figura 4.3 - Subprocesso de negócio de configuração de parâmetros de pesquisa de uma boleia .....	40
Figura 4.4 - Principais <i>mockups</i> elaboradas .....	45
Figura 4.5 - Diagrama do modelo de dados planeado .....	47
Figura 4.6 - Diagrama do modelo de dados implementado .....	49
Figura 4.7 - Diagrama de arquitetura geral do sistema .....	50
Figura 4.8 - Diagrama de componentes do <i>back-end</i> .....	52
Figura 4.9 - Diagrama de instalação do sistema .....	53
Figura 4.10 - Diagrama de estados de um agendamento de boleia .....	54
Figura 5.1 - Captura de ecrã dos entregáveis registados no Jira .....	58
Figura 5.2 - Captura de ecrã dos componentes registados no Jira .....	58
Figura 5.3 - Gráfico circular dos módulos, <i>user stories</i> , tarefas e tarefas independentes definidas no Jira Software .....	59
Figura 5.4 - Captura de ecrã dos detalhes de uma tarefa no Jira .....	60

Figura 5.5 - Captura de ecrã do quadro Kanban no Jira .....	61
Figura 5.6 - Diagrama de fluxo de trabalho definido para o projeto no Jira.....	62
Figura 5.7 - Captura de ecrã do relatório dinâmico de horas de trabalho registadas no Jira.....	63
Figura 5.8 - Captura de ecrã da integração do Jira Cloud com o Microsoft Teams .....	64
Figura 5.9 - Captura de ecrã da estrutura de ficheiros da API .....	67
Figura 5.10 - Captura de ecrã de um exemplo do ficheiro access.log da API.....	67
Figura 5.11 - Captura de ecrã do ficheiro .env.example da API.....	68
Figura 5.12 - Diagrama de fluxo de trabalho das notificação <i>push</i> com recurso ao FCM (retirado de <a href="https://firebase.googleblog.com/2018/09/handle-fcm-messages-on-android.html">https://firebase.googleblog.com/2018/09/handle-fcm-messages-on-android.html</a> ) .....	73
Figura 5.13 - Capturas de ecrã da janela principal da aplicação móvel no modo de motorista .	81
Figura 5.14 - Capturas de ecrã da janela principal da aplicação móvel no modo de passageiro .....	82
Figura 5.15 - Capturas de ecrã da janela de gestão de veículos da aplicação móvel .....	83
Figura 6.1 - Escala de aceitação da pontuação SUS (retirado de <a href="https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/">https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/</a> ) .....	87
Figura 6.2 - Diagrama de fluxo acumulado da transição de estado das tarefas atribuídas ao estudante.....	88

## Índice de Tabelas

Tabela 2.1 - Dados Demográficos da Comunidade IPS (extraído dos dados pertencentes ao Sustain.RD) .....	5
Tabela 2.2 – Resumo dos Dados Apurados Referentes à Primeira Questão do Inquérito de Interesse da Comunitário (extraído dos dados pertencentes ao Sustain.RD).....	7
Tabela 2.3 - Princípios ágeis e respetivas ênfases .....	12
Tabela 4.1 - Membros da equipa e respetivas responsabilidades .....	25
Tabela 4.2 - Cronograma Inicial das Atividades .....	28
Tabela 4.3 - Resumo das funcionalidades levantadas através dos vários <i>brainstorms</i> .....	29
Tabela 4.4 - Módulos levantados .....	31
Tabela 4.5 - Requisitos Funcionais Módulo de Perfil.....	32
Tabela 4.6 - Requisitos Funcionais Módulo de Boleias .....	33
Tabela 4.7 - Requisitos Funcionais Módulo de Notificações .....	34
Tabela 4.8 - Requisito de Qualidade RQ1 .....	35
Tabela 4.9 - Requisito de Qualidade RQ2 .....	35
Tabela 4.10 - Requisito de Qualidade RQ3 .....	36
Tabela 4.11 - Requisito de Qualidade RQ4 .....	36
Tabela 4.12 - Requisito de Qualidade RQ5 .....	36
Tabela 4.13 - Requisito de Qualidade RQ6 .....	37
Tabela 4.14 - Requisitos ambientais do sistema .....	37
Tabela 4.15 - <i>User Stories</i> atribuídas ao estudante .....	41
Tabela 4.16 - Riscos .....	42
Tabela 4.17 - Custo mensal do serviço Google Directions API .....	56
Tabela 6.1 - Resultado da implementação dos requisitos de qualidade .....	85

## Índice de Excertos de Código

Excerto de Código 4.1 - Exemplo de documentação do <i>Back-end</i> .....	55
Excerto de Código 5.1 - Teste unitário do modelo de notificações da API.....	69

## Listas de Siglas e Acrónimos

<b>AAD</b>	<i>Azure Active Directory</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>APNs</b>	<i>Apple Push Notification Service</i>
<b>CRUD</b>	<i>Create, Read, Update, Delete</i>
<b>CTeSP</b>	Curso Técnico Superior Profissional
<b>ECTS</b>	<i>European Credit Transfer and Accumulation System</i>
<b>FCM</b>	<i>Firebase Cloud Messaging</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>IPS</b>	Instituto Politécnico de Setúbal
<b>JWT</b>	<i>JSON Web Token</i>
<b>MVC</b>	<i>Model-View-Controller</i>
<b>NPM</b>	<i>Node Package Manager</i>
<b>ODM</b>	<i>Object Document Mapping</i>
<b>ORM</b>	<i>Object Relation Mapping</i>
<b>RAM</b>	<i>Random Access Memory</i>
<b>SaaS</b>	<i>Software as a service</i>
<b>SDK</b>	<i>Software Development Kit</i>
<b>URI</b>	<i>Uniform Resource Identifier</i>
<b>URL</b>	<i>Uniform Resource Locator</i>
<b>VPN</b>	<i>Virtual Private Network</i>
<b>VPS</b>	<i>Virtual Private Server</i>

**WWW**

*World Wide Web*

## 1. Introdução

O presente documento constitui o relatório de projeto final elaborado no âmbito da unidade curricular de Estágio/Projeto em Engenharia de Software do curso de Engenharia Informática, mais especificamente no ramo de Engenharia de Software, pelo estudante Edgar Rafael Botto dos Santos da Escola Superior de Tecnologia de Setúbal, do Instituto Politécnico de Setúbal durante o ano de 2020 e 2021.

Esta unidade curricular tem um valor de 18 créditos ECTS, o que equivale a 486 horas de trabalho previstas para concluir todas as atividades associadas com o desenvolvimento do projeto.

O referido desenvolvimento decorreu entre 29 de julho de 2020 e 31 de janeiro de 2021, tendo sido a maioria do trabalho realizado presencialmente nas instalações do centro de investigação Sustain.RD (*Research Center for Engineering a Sustainable Development*), do IPS, coordenado pelo Prof. Doutor Rui Madeira, sob sua orientação e da Prof.<sup>a</sup> Doutora Patrícia Macedo, ambos docentes no Instituto. A duração deste projeto excede os esperados 3 a 4 meses devido à situação pandémica que se viveu durante a realização do mesmo, não havendo prazo de entrega estritamente definido, ficando este último à mercê da disponibilidade de todos os envolvidos e dos recursos a que se tinha acesso para colmatar os contratemplos implícitos de uma situação deste tipo.

Este relatório tem como objetivo clarificar e descrever em detalhe os processos de desenvolvimento associados ao projeto de investigação designado de “*Carpooling Platform towards a Sustainable Urban Mobility for IPS*” proposto pela presidência do IPS.

### 1.1. MOTIVAÇÃO

Uma vez que o estudante já tinha realizado três estágios curriculares na área da programação de aplicações *desktop*, aplicações web e serviços de *back-end* no âmbito de um curso profissional e de um CTeSP, optou por realizar um Projeto Final de Curso que integrasse uma componente de investigação, com o intuito de aplicar os conhecimentos de engenharia de software adquiridos ao longo da sua formação dentro do ambiente do ensino superior.

O colega de curso do estudante, Henoch Mendes Vitureira, sob indicação do Prof. Rui Madeira, apresentou ao estudante a possibilidade de ser integrado na equipa de desenvolvimento do projeto Carpooling4IPS, o que seria a futura plataforma de partilha de boleias dentro da comunidade do IPS. Após a consulta dos documentos de ideias e estudo de interesse do projeto, previamente elaborados pelos membros do centro de investigação, o estudante mostrou-se interessado no desenvolvimento de uma plataforma que visa reduzir o impacto ambiental e paisagista, promovendo a aproximação dos membros e a sua consciencialização de sustentabilidade.

Após perceber que o projeto tem potencial de investigação, no que se refere à implementação de um algoritmo de correspondência de rotas tendo por base os horários escolares dos utilizadores e de uma abordagem de gamificação com o propósito de motivar e fidelizar os utilizadores, o estudante decidiu aceitar a proposta do projeto como seu Projeto Final de Curso.

## 1.2. OBJETIVOS DO PROJETO

Sendo este um projeto proposto pela presidência do IPS e sendo o IPS atualmente distinguido com o galardão Eco-Escolas desde 2019 pelo conjunto de boas práticas ambientais [1], este projeto tem como propósito contribuir positivamente para este fim.

Como política de responsabilidade social e de sustentabilidade, o IPS pretende formar cidadãos socialmente responsáveis, dotados de competências técnicas e transversais adequadas através do envolvimento da sua comunidade académica na educação, sensibilização e difusão de conhecimento [2]. Dentro das suas preocupações no contexto ambiental e sustentável, a instituição pretende reduzir o número de veículos parados nos seus parques de estacionamento, o que irá reduzir a poluição visual na paisagem do seu campus, o que por sua vez irá reduzir as emissões de dióxido de carbono por parte dos mesmos.

Com a plataforma CarPooling4IPS, o Sustain.RD pretende ir ao encontro destas preocupações ao fornecer uma aplicação móvel que permita que qualquer membro da comunidade IPS possa partilhar lugares disponíveis no seu veículo de forma fácil, amigável e segura, facilitando a descoberta de ofertas de boleia e a partilha de despesas numa viagem entre os vários passageiros.

## 1.3. ESTRUTURA DO RELATÓRIO

Este documento é constituído por sete capítulos. Começando pela introdução do projeto realizado, seguido pelo capítulo de apresentação do enquadramento do projeto, onde é caracterizada a entidade acolhedora, realizando uma breve apresentação do IPS e do Sustain.RD, seguido pelos fundamentos teóricos do trabalho, onde são referidos os pilares de suporte ao desenvolvimento do projeto, nomeadamente, o estudo de interesse comunitário realizado com uma amostra da comunidade IPS, alguns serviços e aplicações de *carpooling* atualmente presentes no mercado e a metodologia de organização de trabalho utilizada. Este último é seguido pelo capítulo de apresentação dos fundamentos tecnológicos do trabalho, onde são descritas as principais tecnologias de desenvolvimento de software em uso e respetivas ferramentas de suporte, seguido pela descrição dos padrões de software utilizados.

Posteriormente, é apresentado o capítulo de análise e especificação do projeto, que detalha as responsabilidades dos membros da equipa de desenvolvimento, apresenta o

planeamento das atividades do projeto, expõem a análise de âmbito, especificação de módulos e requisitos realizada, os respetivos processos de negócio, as *user stories* planeadas, os riscos associados ao desenvolvimento do projeto, as interfaces com o utilizador planeadas, o modelo de persistência de dados, a arquitetura geral e física, diagramas de estados, as normas de codificação aplicadas e levantamento de custos associados. Este é seguido pelo capítulo de descrição do trabalho realizado, que detalha todas as atividades realizadas durante o desenvolvimento do projeto, distinguindo-as por fases e subfases com um subcapítulo atribuído a cada uma.

Após a descrição do trabalho, é apresentada uma validação do sistema desenvolvido, onde é realizada uma discussão e análise do que foi desenvolvido em contraste com o planeamento e a descrição dos testes realizados. Por último apresenta-se o capítulo de conclusões e trabalho futuro, onde é realizada uma análise e balanço do trabalho realizado e o seu impacto, expondo-se as possíveis melhorias e adições de funcionalidades à plataforma como seguimento de trabalho.

## 2. Enquadramento do Projeto

O segundo capítulo destina-se ao enquadramento do projeto realizado, começando-se pela apresentação da entidade acolhedora, seguido pelo estudo dos dados apurados de um questionário de interesse por uma aplicação de partilha de boleias realizado a uma amostra da comunidade IPS, por sua vez seguido da análise à tecnologia que se encontra atualmente disponível no mercado de *carpooling* e por último a apresentação da metodologia de organização de trabalho utilizada.

### 2.1. APRESENTAÇÃO DA ENTIDADE ACOLHEDORA

Este subcapítulo pretende apresentar e caraterizar o Instituto Politécnico de Setúbal, a entidade acolhedora do centro de investigação Sustain.RD, dando a conhecer a história, missão e valores de ambos.

#### 2.1.1. IPS – Instituto Politécnico de Setúbal

O IPS ou *Instituto Politécnico de Setúbal*, com sede em Estefanilha, Setúbal, é uma instituição pública de ensino superior que se insere no subsistema politécnico e surge no âmbito da criação em Portugal da nova rede de ensino superior politécnica em 14 de outubro de 1979, integrando inicialmente as duas Escolas localizadas em Setúbal, nomeadamente:

- Escola Superior de Tecnologia (ESTSetúbal);
- Escola Superior de Educação (ESE).

Atualmente, o IPS engloba, para além destas, mais três Escolas:

- Escola Superior de Ciências Empresariais (ESCE);
- Escola Superior de Tecnologia do Barreiro (ESTBarreiro);
- Escola Superior de Saúde (ESS).

As atividades das cinco Escolas Superiores são coordenadas pelos Serviços Centrais. Existe ainda uma sexta unidade organizacional, orientada para o apoio aos estudantes: os Serviços de Ação Social (SAS), responsável pelo alojamento, alimentação, apoio psicológico e atividades desportivas e recreativas [3].

A Figura 2.1 ilustra a estrutura formal (níveis hierárquicos e as principais relações formais) dos órgãos do IPS.

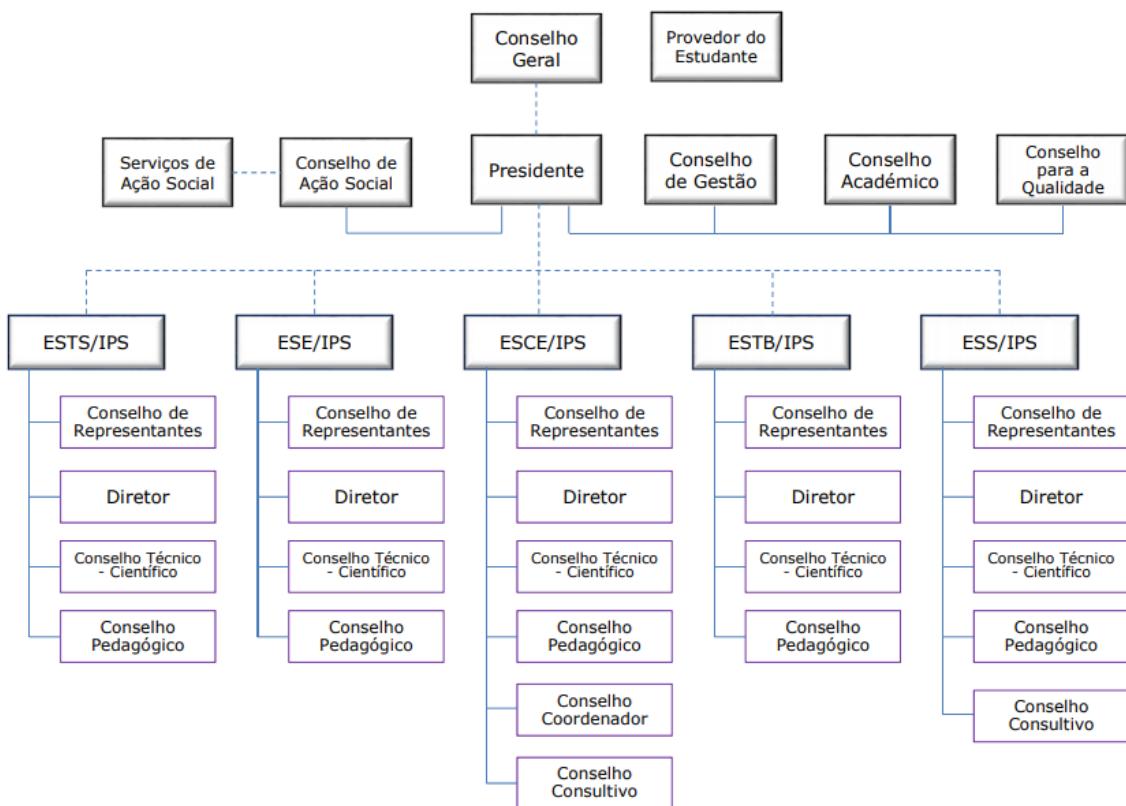


Figura 2.1 - Organograma dos orgãos do IPS (retirado de [https://www.ips.pt/ips\\_si/web\\_base.gera\\_pagina?P\\_pagina=31508](https://www.ips.pt/ips_si/web_base.gera_pagina?P_pagina=31508))

O IPS, como instituição do ensino superior, é um centro de criação, transmissão e difusão da ciência, tecnologia e cultura, intervindo diretamente no desenvolvimento da sociedade e na valorização dos recursos humanos. Como entidade, cabe-lhe naturalmente, legítimas intenções e aspirações para o futuro, cuja explicitação pode ser um instrumento de motivação e inspiração para a sua comunidade. Visa contribuir para o desenvolvimento sustentado na sociedade, em articulação com os parceiros sociais, através do desenvolvimento de atividades de formação, de investigação e desenvolvimento, de transferência de conhecimento e tecnologia, na prestação de serviços e ainda de divulgação e promoção da ciência e da cultura.

No âmbito da sua atividade, o IPS interage através dos seus recursos e competências, quer internamente, quer com a comunidade envolvente. Como forma delineadora de uma cultura, comportamentos e relações, bem como de afirmação das suas convicções, assumem-se como valores fundamentais: Responsabilidade; Excelência e Inovação [4].

Em termos demográficos, à data de 31 de dezembro de 2018, a comunidade IPS apresentava os valores presentes na Tabela 2.1.

Tabela 2.1 - Dados Demográficos da Comunidade IPS (extraído dos dados pertencentes ao Sustain.RD)

Estudantes	Docentes	Não Docentes	Total
6 576	637	188	7 401

De notar que no ano letivo de 2020/2021 entraram 3.000 novos estudantes no instituto, totalizando 7.500 estudantes, sendo que 91% das vagas colocadas a concurso foram preenchidas como 1<sup>a</sup> opção. Do total de estudantes no instituto mais de 550 são alunos estrangeiros.

Vale ressaltar que o IPS tem a 2.<sup>a</sup> maior taxa de empregabilidade dos politécnicos em Portugal e que está entre os primeiros 25 lugares no *ranking*, a nível europeu, no envolvimento com os estudantes.

## 2.1.2. Sustain.RD - Research Center for Engineering a Sustainable Development

O Sustain.RD é um dos três centros de investigação pertencentes à Escola Superior de Tecnologia de Setúbal do Instituto Politécnico de Setúbal, constituído a 23 de janeiro de 2019 e coordenado pelo Prof. Doutor Rui Neves Madeira, que me proporcionou a oportunidade de realizar este projeto sob a sua orientação.

Este centro de investigação tem como missão:

- Promover a execução de atividades de investigação aplicada em áreas chave para o desenvolvimento sustentável;
- Desenvolver a cooperação científica e tecnológica nacional e internacional;
- Promover e apoiar a formação avançada;
- Partilhar o conhecimento adquirido, resultados e inovação com a sociedade.

Na essência, o Centro pretende promover, catalisar e suportar as atividades referidas com vista a obter avanços estratégicos, em tecnologias e sistemas de engenharia complexos, com aplicação em áreas essenciais para a (re)criação de um futuro sustentável para Todos.

Seguindo, para esse efeito, as seguintes linhas de investigação gerais de base:

- Energia Elétrica para a sustentabilidade (geração, transporte, distribuição e utilização);
- Mobilidade Elétrica;
- Ciências da Computação aplicadas ao Desenvolvimento Sustentável (saúde, educação, indústria, economia, etc.);
- Smart Cities;
- Comportamento Organizacional.

Atualmente, o Centro é constituído por 22 docentes vinculados ao IPS e dois membros externos, que entre si e estudantes do instituto participam em projetos dentro do seu âmbito de missão [5].

## 2.2. ESTUDO DO INTERESSE COMUNITÁRIO

Com o propósito de perceber o interesse da comunidade IPS para a existência de uma plataforma de gestão de partilha de boleias entre os membros da comunidade, o Sustain.RD, com o apoio da presidência do IPS, realizou um inquérito de percepção ambiental sobre a mobilidade.

Este inquérito foi realizado individualmente por unidade orgânica do IPS, colocando as duas seguintes questões:

- “Gostaria que existisse no IPS uma plataforma de gestão de boleias para minimizar o número de veículos nos parques de estacionamento do IPS?”,
- “Como se desloca habitualmente para o IPS?”.

A Tabela 2.2 demonstra, de forma resumida, as respostas dos estudantes, docentes e não docentes referentes à primeira questão. Poderá consultar a totalidade dos dados apurados neste inquérito no Anexo 1 – Apuração dos Dados Referentes ao Inquérito de Percepção Ambiental em Relação à Mobilidade da Comunidade IPS.

Tabela 2.2 – Resumo dos Dados Apurados Referentes à Primeira Questão do Inquérito de Interesse da Comunitário  
(extraído dos dados pertencentes ao Sustain.RD)

Comunidade IPS	Não	Sim	Total
<b>Estudantes</b>	130	664	<b>794</b>
	16,4%	83,6%	
<b>Docentes</b>	29	164	<b>193</b>
	15,0%	85,0%	
<b>Não Docentes</b>	12	52	<b>64</b>
	18,8%	81,3%	
<b>Total</b>	<b>171</b>	<b>880</b>	<b>1051</b>
	<b>16,3%</b>	<b>83,7%</b>	

Com uma amostra de 1051 inquiridos, obtiveram-se cerca de 83.7% de respostas positivas à ideia da existência de uma plataforma dedicada ao *carpooling* dentro da comunidade IPS. Aplicando este resultado à totalidade de membros da comunidade IPS, espera-se que existam mais de 7000 potenciais utilizadores.

## 2.3. SOLUÇÕES COMERCIAIS CONCORRENTES

Neste subcapítulo apresentam-se as três soluções comerciais de plataformas de gestão e partilha de boleias entre utilizadores, que se consideraram de maior relevância.

Com o propósito de desenvolver uma aplicação de *carpooling* para a comunidade IPS, decidiu-se investigar sobre o tema de modo a identificar quais as necessidades, vantagens, funcionalidades recorrentes, entre outros fatores relevantes para o desenvolvimento deste projeto. Para além de terem sido identificados alguns artigos científicos de interesse e algumas soluções comerciais no setor, foram também identificadas soluções comerciais mais genéricas tais como a BlaBlaCar<sup>1</sup> e a boleia.net<sup>2</sup>.

### 2.3.1. Singu Carpooling

O Singu Carpooling é uma plataforma dedicada para empresas e centros de negócio, com o objetivo de reunir motoristas e passageiros que realizem deslocações diárias para um local de trabalho comum, desde que pertençam à mesma organização [6].

A Velis Sp. z o.o. Sp.k., a empresa responsável pelo desenvolvimento da plataforma Singu Carpooling, fornece serviços de consultoria com o propósito de disponibilizar uma versão personalizada da plataforma à medida das necessidades de cada um dos seus clientes. Sendo um exemplo de uma versão personalizada desta plataforma, a aplicação de *carpooling* que o Instituto Superior Técnico de Lisboa dispõe aos membros da sua comunidade [7].

A nível de funcionalidade, é uma plataforma bastante completa, fornecendo: suporte para autenticação com base no repositório de utilizadores da organização; criação de rotas possibilitando a definição de um ou mais pontos de encontro; validação da disponibilidade dos passageiros; sistema de classificação e avaliação de cada uma das boleias efetuadas; sistema de mensagens e notificações entre os participantes da viagem. No entanto, considera-se a interface gráfica da aplicação móvel antiquada e pouco apelativa.

De notar, que à data de realização deste relatório a informação em relação a esta plataforma é escassa, tendo a página web oficial da plataforma sido removida.

A Figura 2.2 representa dois ecrãs da aplicação móvel para iPhone da plataforma Singu Carpooling.

<sup>1</sup> <https://www.blablacar.pt/>

<sup>2</sup> <https://www.boleia.net/>

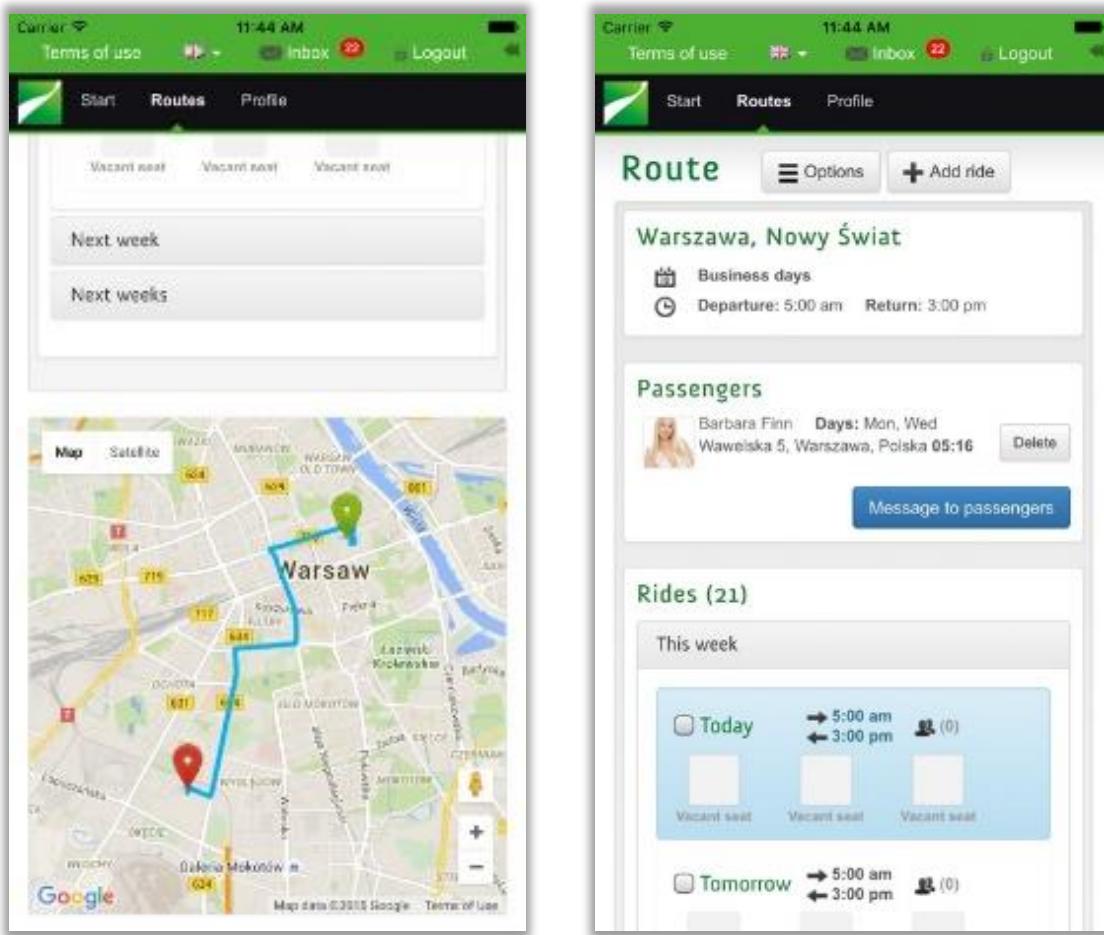


Figura 2.2 - Ecrãs da aplicação móvel Singu Carpooling para iPhone (retirado de <https://apps.apple.com/us/app/id922439656>)

### 2.3.2. Comovee

O Comovee é uma plataforma centralizada de *carpooling* que permite a partilha de boleias por parte de todos os colaboradores de uma organização. Projetado para uma utilização fácil e com um elevado nível de performance em mente, de modo a cumprir com os atuais requisitos corporativos [8].

A nível de funcionamento, é uma plataforma focada na possibilidade de visualizar através de uma interface gráfica moderna e apelativa os percursos programados por colegas de trabalho e realizar pedidos para participar nos mesmos, isto tendo sempre em conta as preferências de privacidade definidas por cada um dos utilizadores. Disponibiliza ainda um sistema de mensagens diretas entre utilizadores e a possibilidade de autenticação utilizando um repositório de utilizadores da organização.

A plataforma também permite a realização de algumas personalizações menores de modo a combinar com a identidade corporativa do cliente. Sendo alguns dos clientes desta

plataforma, empresas globalmente conhecidas, tais como: Volkswagen, Hennessy, Stadler e Lufthansa.

A Figura 2.3 representa um exemplo de um ecrã da aplicação Comovee.

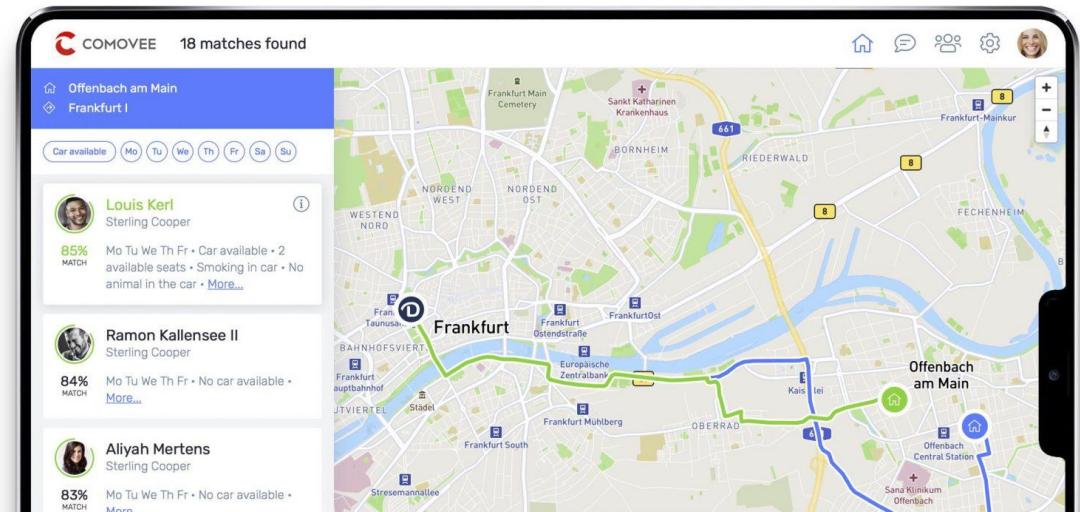


Figura 2.3 - Exemplo de um ecrã da aplicação Comovee (retirada de <https://www.comovee.com/rideshare-software>)

### 2.3.3. Poola

A plataforma Poola, pertencente ao grupo Parkalot, é uma solução completa, personalizável e escalável de *carpooling* direcionada a empresas, que de forma inovadora disponibiliza um simulador onde as empresas poderão testar a viabilidade e os custos associados à implementação e manutenção desta solução na sua comunidade [9]. Sendo utilizada por empresas como a Bonduelle, Dyson e Logitech.

A nível de funcionalidade, é uma solução bastante vasta, que para além das funcionalidades comuns já mencionadas nas plataformas concorrentes, disponibiliza ainda um painel administrativo robusto com um conjunto de dados analíticos relevantes e um avançado algoritmo de correspondência baseado na localização e no horário de trabalho de cada um dos colaboradores de modo a encontrar as melhores oportunidades de partilha de boleia.

A Figura 2.4 representa dois ecrãs da aplicação móvel da plataforma Poola.

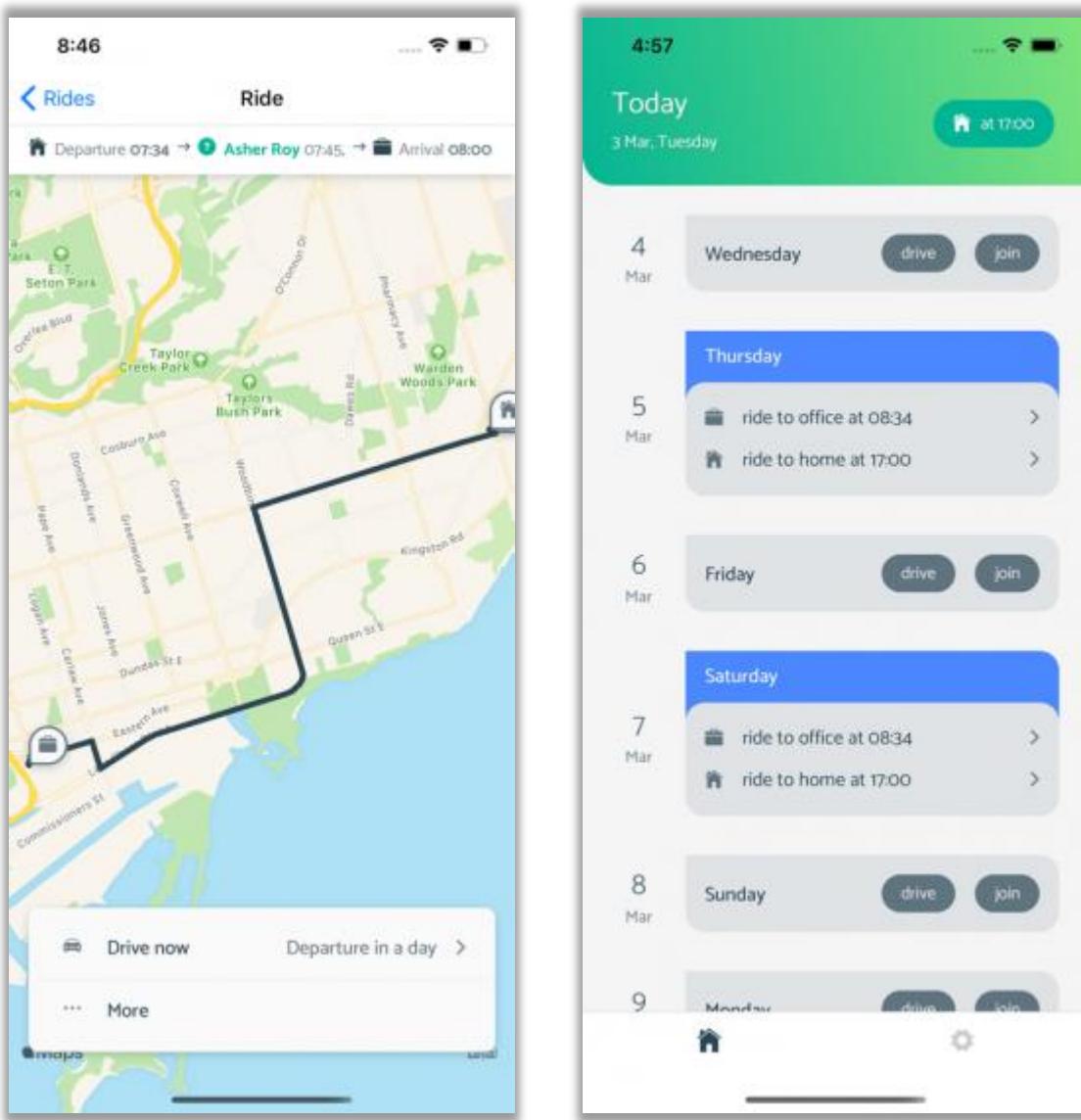


Figura 2.4 - Ecrãs da aplicação móvel Poola (retirado de <https://poola.app>)

## 2.4. METODOLOGIA DE ORGANIZAÇÃO DO PROJETO

No início do projeto não havia planeamento prévio sobre os requisitos funcionais e não funcionais ou quaisquer tarefas associadas aos mesmos. Com isto, devido à necessidade de pronta adaptabilidade às mudanças que iriam certamente ocorrer durante o desenvolvimento da plataforma e à eventual necessidade de ter contacto com o cliente, no caso o IPS, os orientadores aconselharam a utilização de uma *framework* ágil de desenvolvimento de software. Este subcapítulo tem por objetivo especificar o que é a metodologia ágil e a *framework* Kanban seguida pela equipa de desenvolvimento.

### 2.4.1. Agile

As metodologias *Agile* (ágeis) tomam uma abordagem interativa em relação à gestão de projetos e desenvolvimento de *software* que ajuda as equipas a fornecer valor aos seus clientes mais rapidamente e com menos complicações. Em vez de apostar tudo num único lançamento, uma equipa *Agile* entrega trabalho em pequenos, mas consumíveis, incrementos, promovendo a participação cativa das partes interessadas com fornecimento mútuo de *feedback*. Requisitos, planos e resultados são avaliados continuamente de modo que as equipas tenham um mecanismo natural para responder rapidamente à mudança [10].

A Tabela 2.3 contém os doze princípios por detrás das metodologias *Agile* e as ênfases de cada um.

Tabela 2.3 - Princípios ágeis e respetivas ênfases

Princípio Agile	Ênfase
A nossa maior prioridade é satisfazer o cliente através da entrega antecipada e continua de <i>software</i> valioso.	Satisfação do cliente, entrega antecipada e continua, valor.
Permitir abertamente a mudança nos requisitos, mesmo nos momentos mais tardios do desenvolvimento. Os processos ágeis aproveitam a mudança para oferecer vantagem competitiva ao cliente.	Adaptabilidade, competitividade, beneficiar o cliente.
Entregar <i>software</i> funcional frequentemente, desde algumas semanas a alguns meses, com preferência pelo prazo mais curto.	Entregas frequentes.
A gestão de topo e os programadores devem trabalhar diariamente em conjunto ao longo de todo o projeto.	Colaboração.
Desenvolver projetos em torno de indivíduos motivados. Fornecendo-lhes o ambiente e o apoio de que necessitam, confiando neles para que o trabalho seja feito.	Indivíduos motivados, bom ambiente, apoio, confiança.
O método mais eficiente e eficaz de transmitir informação para e dentro de uma equipa de desenvolvimento é a conversa cara-a-cara.	Eficiência, comunicação.
O <i>software</i> funcional é a principal medida de progresso.	Medição de progresso através de entregáveis ( <i>deliverables</i> ).
Os processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores, e utilizadores devem ser capazes de manter um ritmo constante indefinidamente.	Desenvolvimento sustentável, ritmo contante.

A atenção continua à excelência técnica e ao bom <i>design</i> aumenta a agilidade.	Foco na excelência.
Simplicidade, a arte de maximizar a quantidade de trabalho que não se faz, é essencial.	Simplicidade, otimizar o trabalho.
As melhores arquiteturas, requisitos, e desenhos emergem de equipas auto-organizadas.	Auto-organização.
Em intervalos regulares, a equipa reflete sobre como se tornar mais eficaz, depois afina e ajusta o seu comportamento em conformidade.	Melhoria de eficiência e de comportamento.

## 2.4.2. Kanban

Kanban é um método de gestão de trabalho que aplica princípios em conformidade com os princípios *Agile*, com o propósito de melhorar o trabalho em sistemas humanos. É um sistema que visa minimizar as atividades, sem sacrificar a produtividade, tendo como principal objetivo criar mais valor para o cliente, sem gerar custos extra.

Na área do desenvolvimento de *software* o objetivo é fornecer um sistema de gestão, com recursos a processos visuais, que ajude a tomar decisões sobre o quê, quando e quanto produzir. O método Kanban subjacente teve origem num método de produção conhecido por *lean manufacturing* [11], derivado do modelo de operação “*The Toyota Way*” de 1930 da Toyota. No final da década de 1940, a Toyota introduziu a fabricação “*just in time*” em sua produção, tomando uma abordagem em que a produção é baseada na demanda do cliente, em vez da prática padrão de produzir certas quantidades de mercadorias e inseri-las no mercado.

Do japonês, Kanban é literalmente traduzido como “quadro informativo”. O tipo de quadro Kanban, ou em inglês *Kanban Board*, mais simples possui três colunas: “Por Fazer”, “Em Progresso” e “Concluído”. Quando construído e gerido adequadamente, serve como um centro de informações em tempo real, destacando os pontos de congestionamento dentro do sistema e qualquer outra situação que possa interromper a fluidez do trabalho.

No início do século 21, David J. Anderson, engenheiro da Microsoft, e outros participantes relevantes da indústria de software, perceberam como este método concebido pela Toyota poderia mudar positivamente o formato de entrega de produtos e serviços, tornando este processo aplicável a qualquer tipo de empresa que necessite de organizar o seu trabalho.

O método Kanban foca-se na conclusão de tarefas e para tal foram definidos quatro princípios básicos e seis práticas [12]. Os princípios básicos são os seguintes:

- 1º Princípio - Começar com o que é atualmente feito: A flexibilidade do Kanban permite que funcione com os processos, sistemas e fluxos de trabalho existentes, sem interromper o que já está a ser feito com sucesso;
- 2º Princípio – Aceitar a procura por uma mudança evolutiva e incremental: O método Kanban foi projetado para criar mínima resistência, e assim, encorajar a realização de pequenas mudanças incrementais e evolutivas ao processo atual;
- 3º Princípio – Respeitar os processos, as funções e responsabilidades atuais: O Kanban reconhece que processos, funções, responsabilidade e títulos existentes possuem valor e, geralmente, valem a pena ser preservados;
- 4º Princípio – Encorajar atos de liderança em todos os níveis: Este é o princípio Kanban mais recente e relembrar que algumas das melhores lideranças vêm dos atos diários de pessoas na linha frente das suas respetivas equipas. É importante que todos adotem a mentalidade de melhoria constante, de modo a atingir um desempenho ótimo a nível da equipa/departamento/empresa.

De acordo com David Anderson existem seis práticas importantes que precisam de estar presentes para uma implementação bem-sucedida do Kanban, práticas estas que são:

- Visualizar o fluxo de trabalho: Para visualizar o processo com este sistema é necessário um quadro Kanban com cartões e colunas, tal como o apresentado na Figura 2.5. Cada coluna representa um passo no fluxo de trabalho, e cada cartão Kanban, ou em inglês *Kanban Card*, representa uma tarefa. Os cartões geralmente contêm a descrição do trabalho a realizar no formato de *user stories* e/ou tarefas e são colocados no *backlog* do projeto, ou numa das colunas do quadro;
- Limitar o trabalho em progresso: De modo a garantir que não existe mais trabalho em progresso do que aquele que a equipa consegue dar vazão é importante definir um número máximo de cartões que podem estar em simultâneo na coluna “em progresso”;
- Gerir o fluxo: A gestão de fluxo está relacionada com a gestão de trabalho, mas não das pessoas. Em vez de gerir as pessoas e tentar mantê-las ocupadas na totalidade do tempo, deve-se focar na gestão dos processos e entender como se pode acelerar o trabalho;
- Construir políticas de processo explícitas: O processo deve ser definido, divulgado e explicado claramente. Quando todos estão familiarizados com um objetivo comum, são capazes de trabalhar e tomar decisões em relação a mudanças que terão um impacto positivo.

- *Feedback loops:* As realizações de reuniões regulares são de extrema importância para a transferência de conhecimento. Uma boa prática é a realização diária de reuniões em equipa de 10-15 minutos onde se revê o quadro Kanban e se discute aquilo que foi feito no dia anterior e o que se espera fazer no dia corrente;
- Melhorar a colaboração: O modo de atingir uma melhoria continua e mudança sustentável dentro de uma organização é através de uma visão compartilhada de um futuro melhor e a compreensão coletiva dos obstáculos que precisam ser superados.

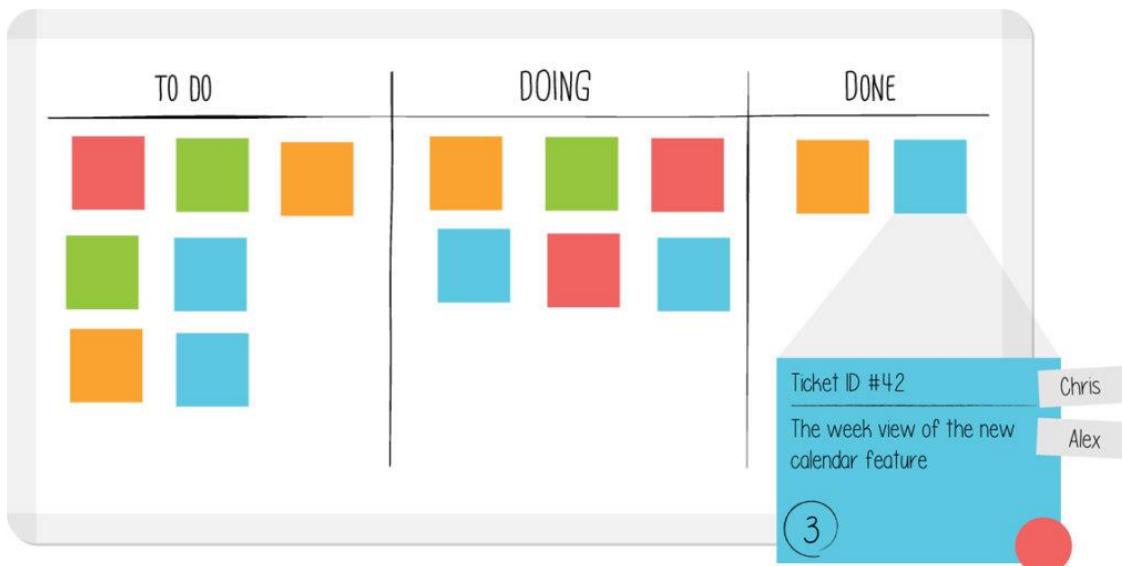


Figura 2.5 - Exemplo de um quadro Kanban (retirado de <https://www.scrumportugal.pt/kanban-board>)

Devido a no início do projeto não existir um planeamento previamente delineado com os requisitos ou entregas previstas, o estudante em conjunto com o seu colega de equipa e orientadores consideraram essencial existir grande flexibilidade quanto ao planeamento do trabalho a ser feito, assim como a sua prioridade, necessidades que porventura possam aparecer e calendarização associada. Ao comparar as metodologias de trabalho Kanban e Scrum, ambas amplamente conhecidas por ambos os estudantes, considerou-se que a primeira seria a que melhor ia ao encontro das necessidades deste projeto, pelo que foi adotada pela equipa.

## 3. Fundamentos Tecnológicos

Neste capítulo são apresentadas as tecnologias utilizadas na elaboração dos vários componentes do projeto desenvolvido. Será realizada uma análise das mesmas e das alternativas consideradas, de modo a efetuar-se um estudo crítico e comparativo entre as possibilidades no mercado. Além das tecnologias, também se faz a exposição das ferramentas de suporte ao desenvolvimento utilizadas. Por último, são abordados os padrões de *software* seguidos durante o desenvolvimento da plataforma.

### 3.1. TECNOLOGIAS DE DESENVOLVIMENTO

#### 3.1.1. API RESTful (back-end)

A tecnologia escolhida para o desenvolvimento do back-end foi o Node.js, que por sua vez utiliza a linguagem de programação JavaScript. Terão sido consideradas outras duas tecnologias a utilizar, nomeadamente, o Python e o Go, no entanto, devido à falta de domínio por parte da equipa em ambas e, no caso particular do Go, a falta de bibliotecas de terceiros que cobrissem as necessidades levantadas, acabou-se por descartar ambas.

---

##### 3.1.1.1. JavaScript

JavaScript é uma linguagem de programação, baseada em scripts, inicialmente implementada como parte dos navegadores web e padronizada pela ECMA International<sup>3</sup> em junho de 1997. Esta linguagem permite criar conteúdos dinâmicos e interativos em tempo real, sem a necessidade de recarregar a página cada vez que interagimos com a aplicação [13].

Atualmente, é das linguagens mais utilizada para programação do lado do cliente, no entanto através do uso de ferramentas como o Node.js é possível utilizar esta linguagem para o desenvolvimento de funcionalidades do lado do servidor.

Utiliza uma sintaxe similar à linguagem de programação C, estando dentro do paradigma de orientação a objetos. Importante referir que apesar da palavra Java estar presente no nome, esta linguagem é totalmente distinta da linguagem de programação Java e apresenta recursos diferentes, não disponibilizados em Java ou C++.

---

##### 3.1.1.2. Node.js

---

<sup>3</sup> European Computer Manufacturers Association: É uma associação dedicada à padronização de sistemas de informação.

O Node.js é um interpretador de JavaScript assíncrono, de código fonte aberto e multiplataforma, orientado a eventos, criado sobre o motor JavaScript V8 do Chrome.

Foi criado em 2009 por Ryan Dahl, com o objetivo em migrar a programação JavaScript do cliente para o servidor, possibilitando a criação de aplicações de alta escalabilidade capazes de manipular milhares de conexões/eventos simultâneos em tempo real numa única máquina física [14].

### 3.1.2. Aplicação Móvel (front-end)

A tecnologia escolhida para o desenvolvimento do front-end foi o Flutter, que por sua vez utiliza a linguagem de programação Dart. Terão sido consideradas outras possibilidades, tais como o desenvolvimento nativo em Java ou Kotlin para Android e Objective-C ou Swift para iOS, foram consideradas ainda algumas possibilidades multiplataformas tais como Apache Cordova, React Native e NativeScript, no entanto, nenhuma destas tecnologias ia ao encontro dos objetivos da equipa de desenvolvimento que pretendia obter uma aplicação pronta a utilizar para Android e iOS com o mínimo de adaptações possível a nível de codificação.

---

#### 3.1.2.1. Dart

Dart é uma linguagem orientada a objetos com estilo semelhante ao C desenvolvida pela Google. Esta linguagem é otimizada para a construção de interfaces de utilizador, podendo ser compilada para código nativo no caso de aplicações móveis, ou para JavaScript no caso de aplicação Web. Dart tem uma sintaxe familiar aos programadores por ser semelhante a outras linguagens predominantes no mercado [15].

A linguagem suporta estruturas como Mixins, Classes Abstratas, Tipos Genéricos e Inferência de Tipos. Mixins são estruturas semelhantes a classes que permitem adicionar um conjunto de características a qualquer outra classe sem o uso de hierarquias pai-filho [16].

---

#### 3.1.2.2. Flutter

Flutter é um SDK que utiliza como linguagem de programação Dart, criado pelo Google e permite o desenvolvimento de aplicações multiplataforma, não estando limitado apenas a plataformas móveis.

O Flutter foi concebido como um sistema extensível dividido em camadas constituídas por uma série de bibliotecas independentes. Nenhuma camada tem acesso privilegiado à camada abaixo, e cada parte do nível da estrutura está concebida para ser opcional e substituível.

### 3.1.3. Persistência de Dados

A tecnologia escolhida para a persistência de dados foi o MongoDB, uma base de dados não-relacional, devido à experiência positiva que os envolvidos tinham com a tecnologia em conjugação com Node.js e bibliotecas apropriadas. Terão sido consideradas outras possibilidades, tais como Microsoft SQL Server ou PostgreSQL que também possui a possibilidade de criar uma secção da base de dados não-relacional, no entanto, considerou-se que seria injustificável a curva de aprendizagem em relação ao PostgreSQL, sendo que dada a natureza do projeto que implica a persistência de um grande conjunto de dados brutos de rotas de viagens e históricos de boleias considerou-se pertinente avançar-se com uma base de dados de natureza não-relacional. Vale mencionar que existem bibliotecas de MongoDB para Node.js que permitem a gestão de relações entre entidades no lado do cliente.

#### 3.1.3.1. MongoDB

MongoDB é um sistema de base de dados não relacional orientado a documentos, também conhecido por base de dados NoSQL, desenvolvido em C++ pela MongoDB Inc. em 2009, possuindo suporte a múltiplas plataformas [17].

As características deste sistema permitem que a informação seja modelada de modo mais natural, utilizando um formato chamado BSON, este formato possui uma grande semelhança à formatação em JSON, pois os dados podem ser agrupados em hierarquias complexas mantendo a sua indexação e a facilidade de acesso aos mesmos, no entanto ao contrário dos tradicionais sistemas de base dados SQL este sistema não suporta relações entre os dados.

### 3.1.4. Notificações

A nível de notificação do tipo *push* recorreu-se ao serviço FCM da Google dado a sua compatibilidade com Android e iOS, não apresentando quaisquer custos de utilização. Apesar de suportar iOS, requer a integração com o APNs da Apple. Neste âmbito não terão sido consideradas outras tecnologias, apesar de ter sido realizada uma rápida pesquisa a nível de opções no mercado, existindo algumas outras possibilidades, no entanto, todas elas pagas.

#### 3.1.4.1. APNs – Apple Push Notification Service

O Apple Push Notification Service é um serviço na nuvem que permite aplicações de terceiros aprovadas instaladas em dispositivos Apple receber notificações *push* através de uma conexão segura a partir de um servidor remoto.

Para receber notificações num dispositivo Apple, a aplicação deverá estar configurada e devidamente registada no APNs. Este serviço realiza a entrega de notificações através de uma

API interna que está presente no iOS e Mac OS. Este sistema terá sido introduzido pela primeira vez em junho de 2009 com o lançamento do iOS 3 para iPhone [18].

### 3.1.4.2. FCM – Firebase Cloud Messaging

O Firebase Cloud Messaging é uma solução para envio de mensagens entre plataformas que permite a entrega confiável de mensagens sem custo [19]. Anteriormente conhecido por Google Cloud Messaging, atualmente permite desenvolvedores de aplicações de terceiros enviar notificações e mensagens para Android, iOS e aplicações web. Este serviço é fornecido pela Firebase, uma subsidiária da Google desde 21 de outubro de 2014 [20].

## 3.1.5. Gestão de Autenticação

Não existindo margem de manobra em relação à tecnologia de gestão de autenticação devido a estar-se dependente da tecnologia utilizada internamente pela instituição, seguiu-se com a implementação da autenticação com recurso ao Azure Active Directory. Terá sido levantado pelo estudante a possibilidade de se utilizar um serviço externo como o Firebase Auth que agilizasse todo o processo de autenticação, no entanto, tais serviços apresentam custos de utilização que seriam injustificados para o projeto dada a possibilidade de implementar tais funcionalidades apesar do esforço extra.

### 3.1.5.1. Azure Active Directory

O Azure Active Directory é um serviço de gestão de identidade e acesso baseado na nuvem desenvolvido pela Microsoft, que ajuda os utilizadores a iniciar sessão e aceder a recursos externos tais como o Microsoft 365, o portal Azure e milhares de outras aplicações SaaS, permitindo ainda o acesso a recursos internos como aplicações numa rede corporativa juntamente com quaisquer aplicações em nuvem desenvolvidas pela própria organização [21].

### 3.1.5.2. Microsoft Graph API

O Microsoft Graph é um ponto de entrada para os dados e inteligência do Microsoft 365. O serviço fornece um modelo unificado de programação que pode ser utilizado para aceder à enorme quantidade de dados no Microsoft 365, Windows 10, Enterprise Mobility + Security. Possibilitando a utilização da riqueza de dados fornecidos pela Microsoft Graph API para construir aplicações para organizações e consumidores que interagem com milhares de utilizadores [22].

### 3.1.5.3. JSON Web Token

JSON Web Token é um padrão aberto (RFC 7519) que define uma forma compacta e autónoma de transmitir informações de forma segura entre partes através de um objeto JSON. Esta informação pode ser verificada e confiada porque é assinada digitalmente. O JWT pode ser assinado usando um segredo (com algoritmo HMAC) ou um par de chaves publicas/privadas usando RSA ou ECDSA.

## 3.2. PADRÕES DE SOFTWARE

### 3.2.1. Back-end

Para o desenvolvimento da RESTful API em Node.js foram utilizados dois padrões de software, nomeadamente, uma variante do padrão de desenho de software comumente conhecido por MVC, ou *Model-View-Controller*, e o estilo de arquitetura de software conhecido por REST, ou *Representational State Transfer*.

#### 3.2.1.1. MVC (*Model-View-Controller*)

O padrão MVC foi introduzido pela primeira vez por Trygve Reenskaug nos anos 70 durante uma visita à empresa de investigação e desenvolvimento *Xerox Palo Alto Research Center* [23]. No entanto, apenas em 1988 é que este padrão foi referido num artigo no *The Journal of Object Technology* como sendo um conceito geral, tendo ao longo dos anos evoluído [24], originando variantes, tais como: *Hierarchical Model-View-Controller* (HMVC), *Model-View-Adapter* (MVA), *Model-View-Presenter* (MVP), *Model-View-Viewmodel* (MVVM), entre muitas outras.

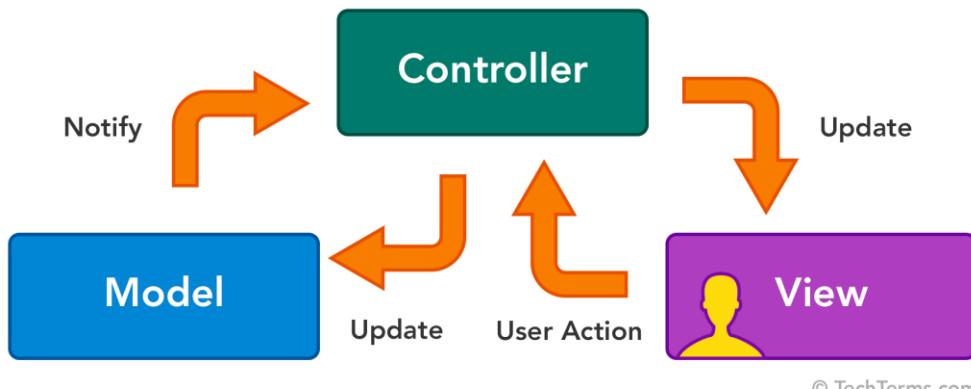
O referido padrão MVC é comumente utilizado para o desenvolvimento de interfaces de utilizador modernas, fornecendo partes fundamentais para o desenho de programas *desktop* ou *mobile*, bem como para aplicações *web* [25]. Este padrão adapta-se bem a linguagens orientadas a objetos, uma vez que os diferentes modelos, interfaces de utilizador e controladores podem ser tratados como objetos e reutilizados.

Os modelos (*models*) representam os dados utilizados por um programa. Poderão ser uma base de dados, um ficheiro, ou um simples objeto. Comumente integra-se a lógica de manipulação, conversão e validação de dados dentro dos modelos, tornando a camada de dados e respetiva lógica independente do resto do programa.

As interfaces de utilizador (*views*) são o meio pelo qual os componentes gráficos são apresentados num programa. Estes componentes poderão ser janelas, gráficos, botões, texto ou qualquer outro elemento visível pelo utilizador.

Os controladores (*controllers*) integram a lógica de negócio do programa, tendo a responsabilidade de manipular os dados e respetivas interfaces de utilizador. Recebem as interações do utilizador com a interface e realizam as atualizações necessários nos modelos e interfaces correspondentes.

Tal como visível no diagrama da Figura 3.1, as três partes do padrão MVC estão interconectadas. A interface de utilizador (*view*) apresenta o modelo ao utilizador. O controlador (*controller*) aceita as interações do utilizador e atualiza o modelo e interfaces de acordo. O modelo (*model*) notifica o controlador de mudanças efetuadas aos dados. Embora não tenha necessariamente de existir um controlador por interface de utilizador, ou um modelo por controlador, é comum criar-se para cada interface um controlador, aumentando deste modo a separação de responsabilidades sem prejudicar a legibilidade do código-fonte [26].



© TechTerms.com

Figura 3.1 - Diagrama do padrão MVC (retirado de <https://techterms.com/definition/mvc>)

A utilização deste padrão não é um requisito no desenho de um programa, no entanto, diversas linguagens de programação e IDE suportam este padrão, isto, mais o facto de aumentar significativamente a facilidade de reutilização e manutenção do código-fonte, torna o padrão MVC uma escolha comum no desenvolvimento de aplicações.

Para o desenvolvimento deste projeto em particular, apesar de se tratar de uma API e não possuir interfaces gráficas, o estudante decidiu utilizar o padrão MVC, mais especificamente uma variante não oficial que determina que as *views* sejam substituídas por *routes*, isto deve-se ao facto de que apesar dos *endpoints* da API funcionarem como um tipo de interface de utilizador que aceitam entrada de dados, não possuem quaisquer elementos gráficos. O principal motivo para a escolha deste padrão deve-se ao facto de ser um projeto de investigação que terá uma grande probabilidade de continuar em aberto após o trabalho realizado no âmbito desta unidade curricular pelo estudante e o seu colega de equipa, sendo que provavelmente virá a passar por diversas equipas ao longo do seu desenvolvimento, deste modo a divisão de responsabilidades como um meio de aumentar a modularidade da plataforma será pertinente.

---

### 3.2.1.2. REST (Representational State Transfer)

REST, ou *Representational State Transfer* é um estilo de arquitetura de software que define um conjunto de restrições a serem utilizadas para a criação de web services. Uma RESTful API é uma interface de programação de aplicações (API) que utiliza pedidos HTTP para aceder a dados e que está em conformidade com as restrições definidas pela arquitetura REST. Esta arquitetura foi introduzida e definida em 2000 por Roy Fielding em sua tese de doutoramento [27].

Num serviço *RESTful Web* os pedidos são feitos ao URI de um recurso, tal pedido origina uma resposta formatada em HTML, XML, JSON, ou outro formato. O protocolo mais frequentemente utilizado para tais pedidos e respetivas respostas é o HTTP, fornecendo operações (métodos HTTP) tais como a consulta de recursos (GET), criação de recursos (POST), atualização de recursos (PUT) e eliminação de recursos (DELETE) [28]. Ao utilizar um protocolo sem estado (*stateless*) e operações padrão, os sistemas RESTful visam um desempenho rápido, fiabilidade e capacidade de crescer através da reutilização de componentes que podem ser manipulados e atualizados sem afetar o sistema como um todo, mesmo durante o seu funcionamento.

Existe um conjunto de seis princípios que restringem as formas como o servidor pode processar e responder aos pedidos do cliente [29], de modo que, ao operar de acordo com tais restrições, o sistema ganhe propriedades não funcionais desejáveis, tais como performance, escalabilidade, simplicidade, modificabilidade, visibilidade, portabilidade e fiabilidade [27]. Se um sistema violar qualquer uma destas restrições, o mesmo não poderá ser considerado RESTful.

As restrições formais do REST são:

- Ter uma arquitetura cliente/servidor formada por clientes, servidores e recursos, com solicitações realizadas através do protocolo HTTP;
- Estabelecer uma comunicação *stateless* entre cliente e servidor. Isso significa que nenhuma informação do cliente é armazenada entre pedidos GET e todos os pedidos são separados e não relacionados;
- Armazenar dados em cache para otimizar as interações entre cliente e servidor;
- Ter uma interface uniforme entre os componentes para que as informações sejam transferidas em formato padronizado. Para tal, é necessário que os recursos sejam identificáveis e estejam separados das representações enviadas ao cliente; os recursos possam ser manipulados pelo cliente por meio da representação recebida com informações suficiente para tais ações; as mensagens auto descritivas retornadas ao cliente contenham informações suficientes para descrever como processá-las; hipertexto e hipermédia como meio de gestão de estados da aplicação;
- Ter um sistema em camadas que organiza os tipos de servidores (responsáveis pela segurança, pela gestão de carga, entre outros) envolvidos na recuperação das informações solicitadas em hierarquias;

- Possibilitar código sob demanda (opcional): a capacidade de enviar um código executável do servidor para o cliente quando solicitado, de modo a estender ou customizar temporariamente a funcionalidade de um dado cliente.

Embora uma RESTful API precise de estar em conformidade com as restrições suprarreferidas, não deixa de ser de mais fácil implementação e utilização do que um protocolo prescrito, como o SOAP (*Simple Object Access Protocol*), que tem requisitos específicos, como a utilização forçada de XML para a troca de mensagens e exigências de segurança que devem ser cumpridas, estes requisitos, e outros, tornam o protocolo mais lento e pesado, para além de aumentar a complexidade de implementação do mesmo.

Em comparação, a arquitetura REST é composta de um conjunto de diretrizes que podem ser implementadas conforme necessário, tornando as RESTful API mais rápidas, leves e escaláveis, o que é ideal para a Internet das Coisas (IoT).

Tendo em conta os benefícios supramencionados da utilização de REST em contraste com outras alternativas e da limitação de tempo existente para o desenvolvimento deste projeto, o estudante, em conjunto com o seu colega de equipa, considerou adequado a utilização desta arquitetura para o desenvolvimento da API (servidor) responsável por alimentar a aplicação móvel (cliente).

### 3.2.2. Front-end

Foram utilizados três padrões de software para o desenvolvimento da aplicação móvel em Flutter, nomeadamente: Repository; BLoC e Cubit.

O padrão de arquitetura Repository consiste no desacoplamento e na agregação da lógica de acesso e manipulação de dados provenientes de fontes externas, promovendo a orientação a objetos na interação com a camada de dados [30].

O BLoC ou *Business Logic Component* é um padrão projetado por Paolo Soares e Gong Hui, da Google e apresentado pela primeira vez durante a DartConf 2018. Este padrão foi inicialmente concebido para permitir a reutilização de código fonte da lógica de negócio independentemente da plataforma, tendo como principal objetivo simplificar a gestão de estados [31].

O padrão BLoC foi utilizado por meio da biblioteca ‘flutter\_bloc’, recomendada pela documentação oficial do Flutter e desenvolvida por Felix Angelov. Esta biblioteca tem como objetivo abstrair o padrão em questão, de modo a simplificar a experiência do programador. A Figura 3.2 representa o funcionamento do padrão com recurso à biblioteca, onde a interface (UI) transmite eventos ao BLoC, que por sua vez processa a lógica de negócios, incluindo a comunicação com o exterior para obter dados (quando aplicável) e devolve estados que afetam a parte gráfica envolvida [32].

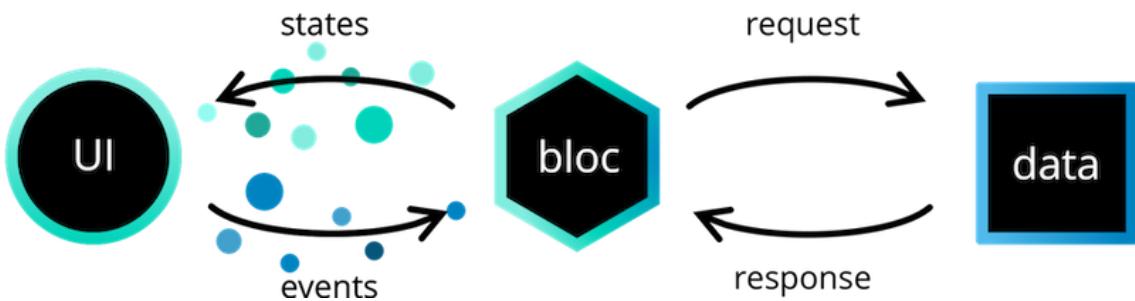


Figura 3.2 - Diagrama simplificado do padrão BLoC (retirado de <https://bloclibrary.dev/#/architecture>)

O Cubit é um subconjunto da implementação do padrão BLoC da biblioteca de Felix Angelov, que substitui os eventos do padrão BLoC tradicional por métodos. Isto significa que em vez de envirmos eventos, apenas invocamos um método público presente na implementação de um dado Cubit, simplificando significativamente a utilização do padrão BLoC. A Figura 3.3 ilustra o funcionamento de um Cubit [32].

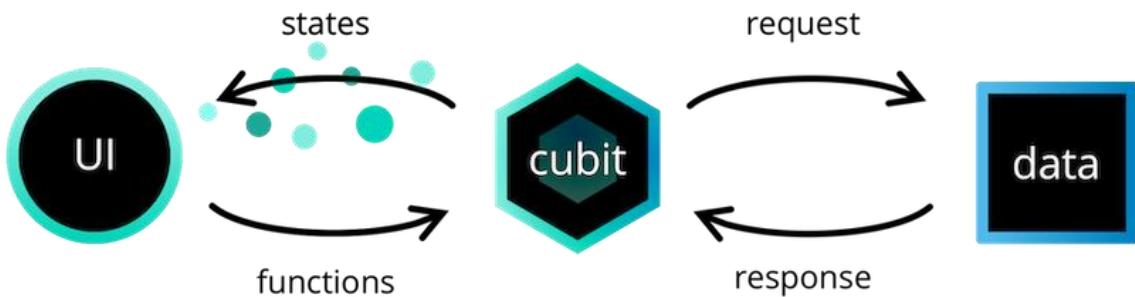


Figura 3.3 - Diagrama simplificado do padrão Cubit (retirado de <https://bloclibrary.dev/#coreconcepts?id=cubit>)

## 4. Análise e Especificação do Projeto

O projeto teve início com o levantamento de ideias, em conjunto com os orientadores, em relação às funcionalidades que uma plataforma de gestão e partilha de boleias deveria integrar. Este conjunto de ideias levantadas foram refinadas e posteriormente transformadas em módulos, requisitos não funcionais e requisitos funcionais. Estes últimos originaram *user stories* com a descrição simplificada das respetivas funcionalidades da plataforma na perspetiva do utilizador. Tendo definido os requisitos, e as tecnologias a utilizar, efetuou-se o devido levantamento de potenciais riscos que pudessem atrasar, limitar ou impedir de todo o desenvolvimento desta plataforma.

Com uma sólida visão das funcionalidades que a plataforma iria integrar, passou-se à retificação do funcionamento dos principais requisitos através da elaboração de diagramas de processos de negócio. Elaboraram-se *mockups* da interface de utilizador da aplicação móvel e um protótipo interativo com o objetivo de permitir que potenciais utilizadores pudessem experimentar e fornecer *feedback* sobre as funcionalidades planeadas e respetiva *user experience* da proposta de interface produzida. Em seguida, elaborou-se o modelo de persistência de dados de acordo com o paradigma não-relacional do MongoDB. A análise e especificação da plataforma terminou com o estabelecimento da arquitetura geral, lógica e física do sistema, tal como dos respetivos diagramas de estados, normas de codificação do servidor e custos dos serviços necessários.

### 4.1. EQUIPA E RESPONSABILIDADES

A equipa de desenvolvimento deste projeto foi composta pelo estudante Edgar Rafael Botto dos Santos e o seu colega de curso Henoch Mendes Vitureira. Com o propósito de se clarificar o trabalho realizado por cada membro da equipa, os orientadores sugeriram a divisão de responsabilidades, tendo o estudante ficado responsável por desenvolver maioritariamente o *back-end*, isto é, a API responsável por alimentar os dados da aplicação móvel.

A Tabela 4.1 contém os nomes dos membros da equipa de desenvolvimento e respetivas responsabilidades.

Tabela 4.1 - Membros da equipa e respetivas responsabilidades

Nome	Responsabilidades
<b>Edgar Santos</b>	Realizar pesquisas por aplicações concorrentes e outras informações úteis para o levantamento de requisitos do projeto.
	Desenvolver as <i>mockups</i> referentes à aplicação móvel, com o apoio dos colegas de equipa na tomada de decisões e realização de tarefas menores;

	<ul style="list-style-type: none"> <li>Preparar o Jira de acordo com a metodologia de organização do projeto a utilizar;</li> <li>Definir e preparar o ambiente de desenvolvimento em Node.js;</li> <li>Estabelecer a estrutura de diretórias do projeto em Node.js, as normas de codificação a seguir e os padrões de software a utilizar;</li> <li>Desenvolver a maioria das funcionalidades da API do servidor em Node.js;</li> <li>Participar nas discussões e tomadas de decisão do desenvolvimento da interface gráfica da aplicação móvel;</li> <li>Desenvolver parte das funcionalidades da aplicação móvel em Flutter.</li> </ul>
<b>Henoch Vitureira</b>	<ul style="list-style-type: none"> <li>Realizar pesquisas por artigos académicos/científicos relacionados com o tema Car Pooling, e outras informações úteis para o levantamento de requisitos do projeto.</li> <li>Planejar e desenvolver o diagrama de modelo de dados inicial, com o apoio dos colegas de equipa na tomada de decisões;</li> <li>Preparar o ambiente de desenvolvimento em Flutter;</li> <li>Estabelecer a estrutura de diretórias do projeto em Flutter, as normas de codificação a seguir e os padrões de software a utilizar;</li> <li>Desenvolver a maioria das funcionalidades da aplicação móvel em Flutter;</li> <li>Participar nas discussões e tomadas de decisão do desenvolvimento da API;</li> <li>Desenvolver parte das funcionalidades da API.</li> </ul>

## 4.2. PLANEAMENTO DAS ATIVIDADES DO PROJETO

Tal como referido anteriormente, a equipa de desenvolvimento seguiu de modo geral os princípios das metodologias ágeis, realizando a implementação da plataforma em várias iterações após uma especificação inicial do mesmo. Deste modo prevê-se que como resultado de cada iteração seja produzida uma versão funcional da plataforma, permitindo ainda que os requisitos especificados possam sofrer alterações em função da avaliação realizada.

Dada as limitações temporais mencionadas anteriormente, no âmbito do projeto realizado ao abrigo da unidade curricular de Estágio/Projeto em Engenharia de Software foi produzida apenas a primeira versão da plataforma, que incluiu a realização das tarefas

associadas à implementação do módulo de Perfil e de Boleias, cuja especificação está presente no capítulo 4.3.

## 4.2.1. Entregas Previstas

O projeto foi planeado para ter um total de quatro entregas/versões previstas.

A primeira versão contemplaria a implementação de funcionalidades essenciais referentes ao Perfil e Gestão de Boleias, que conteria a autenticação do utilizador, a gestão de informação individual e as operações de oferta, procura e solicitação de boleias.

Na segunda versão seria a integração de um sistema de comunicação direta (*chat*), possibilidade de realizar avaliações de boleias (*feedback*) e visualizar informações referentes ao valor monetário associado às mesmas, tal como custo de combustível e despesas com portagens.

Já na terceira versão planeou-se a adição de um sistema de gamificação motivador baseado em progressão por níveis, conquistas e recompensas virtuais e possivelmente físicas (proporcionadas pela instituição de ensino).

Por último, a quarta versão iria complementar as anteriores ao introduzir a possibilidade de os utilizadores visualizarem o impacto ambiental causado pela utilização desta plataforma, através da apresentação de estatísticas relevantes, como gramas de dióxido de carbono evitadas, litros de combustível fóssil poupanados, número de lugares de estacionamento não ocupados, número de viagens realizadas, entre outros dados.

## 4.2.2. Cronograma Inicial

Inicialmente, tal como demonstrado na Tabela 4.2, terão sido planeadas quatro atividades principais, nomeadamente três entregas que contemplariam o planeamento do projeto, respetivos módulos inicialmente levantados e a respetiva configuração das ferramentas utilizadas e da base de dados.

Tomando como referência as entregas previstas no subcapítulo 4.2.1, a entrega 2 presente na Tabela 4.2 corresponderia à primeira versão da plataforma, versão esta que foi efetivamente desenvolvida e apresentada neste documento. Já a entrega 3 terá sido um planeamento ambicioso que em relação às entregas previstas referenciadas corresponderia à segunda, terceira e quarta versão da plataforma, deste modo, à data deste documento, é possível concluir que a previsão destas três versões ter lugar num espaço de pouco mais de um mês terá sido de todo infundamentado. De notar ainda que este cronograma foi realizado ainda antes do levantamento das *user stories* e numa altura em que se esperaria a existência de uma equipa de desenvolvimento constituída por pelo menos quatro a cinco membros, ao contrário de apenas dois.

Em suma, o planeamento inicial seria o projeto ter início no mês de agosto de 2020, e um final previsto para outubro de 2020, previsão esta que após a equipa se deparar com o real nível de complexidade de um sistema deste género, ainda durante o desenvolvimento da primeira versão, terá sido adiada para meados de 2022.

Tabela 4.2 - Cronograma Inicial das Atividades

	Ago.	Set.	Out.
<b>Entrega 1 – Planeamento</b>	█		
<b>Configuração de ferramentas necessárias e base de dados</b>	█	█	
<b>Entrega 2 – Autenticação e Perfil, Gestão de Boleias</b>	█	█	█
<b>Entrega 3 – Chat, Gamificação, Estatísticas, Alguns Complementos para Boleias</b>		█	█

### 4.2.3. Cronograma Real

A Figura 4.1 representa o cronograma real do projeto e respetivas *milestones* à data deste documento. Este projeto regeu-se pelo desenvolvimento da primeira versão, com começo no mês de agosto, e entrega a meio do mês de janeiro, não tendo sido realizado mais desenvolvimento até ao momento. Deste modo todas as tarefas posteriores à data corrente são uma previsão com base no desenvolvimento já realizado.

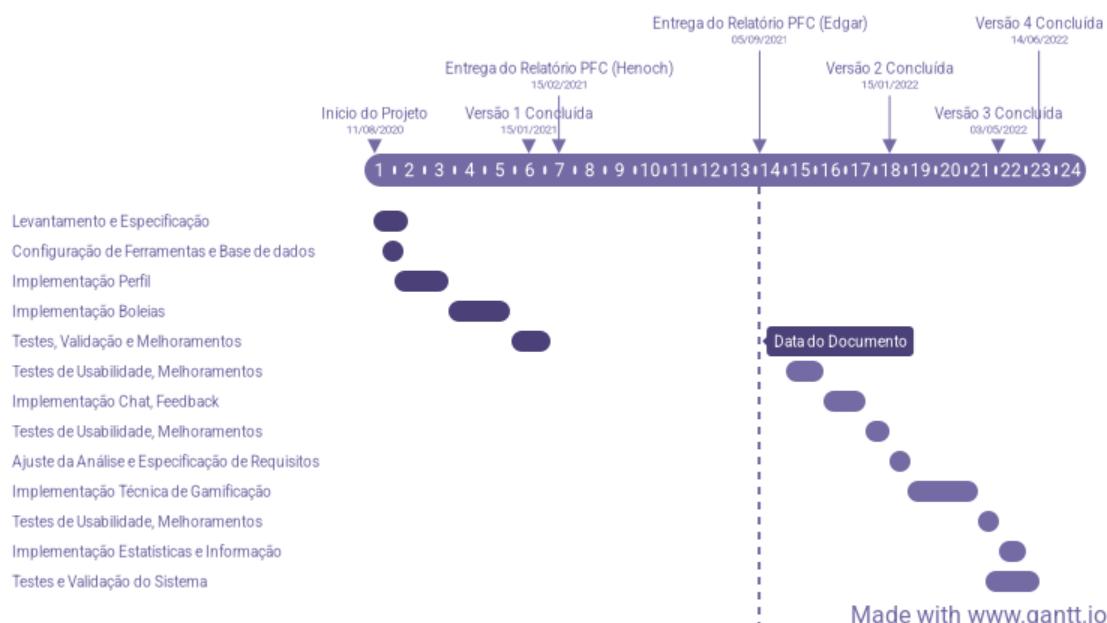


Figura 4.1 - Cronograma Real das Atividades

## 4.3. ANÁLISE DE ÂMBITO E ESPECIFICAÇÃO DE MÓDULOS

O objetivo principal para o desenvolvimento deste projeto é a obtenção de uma plataforma base de gestão e partilha de boleias que permita de forma acessível a continuação do trabalho sobre a mesma por outras equipas.

A pedido direto da equipa de investigadores do Sustain.RD considerou-se a adição de um sistema de gamificação para motivar o uso da plataforma pela comunidade IPS, além da adição de um conjunto de estatísticas de cariz ecológica, como emissões de dióxido de carbono evitadas e o combustível fóssil poupança. Estes requisitos pretendem ir ao encontro dos objetivos do centro de investigação em atingir um futuro com uma maior consciência ambiental.

### 4.3.1. Brainstorming

O levantamento dos requisitos envolveu a realização de vários *brainstorms* entre os membros da equipa de desenvolvimento. Adicionalmente, também foi essencial reunir com os orientadores e com o cliente final, assim como a realização de um estudo de soluções semelhantes presentes no mercado.

A sistematização das ideias resultados de trabalho de levantamento realizado deu origem a um conjunto de funcionalidades consideradas chave, resumidas na Tabela 4.3.

Tabela 4.3 - Resumo das funcionalidades levantadas através dos vários *brainstorms*

Âmbito	Funcionalidades
<b>Autenticação</b>	Autenticação com o Microsoft Active Directory do IPS.
<b>Perfil</b>	<p>Nome;</p> <p>Localidade;</p> <p><b>Como motorista:</b></p> <ul style="list-style-type: none"> <li>• Gestão de veículos (inclui informação de lugares disponíveis no veículo, tipo de combustível e consumo em L/100KM);</li> <li>• Definir pontos de origem e rotas tomadas para o IPS;</li> <li>• Permitir definir se uma boleia tem ida e volta ou apenas num sentido;</li> <li>• Definir horas e saída do ponto de origem e de saída do IPS;</li> <li>• Possibilidade de fazer um horário de boleias, consultável por outros utilizadores ou apenas por amigo, a definir pelo motorista;</li> <li>• Definir se está disponível para dar boleias, notificar passageiros de boleias regulares;</li> </ul>

	<p><b>Como passageiro:</b></p> <ul style="list-style-type: none"> <li>Definir possíveis pontos de recolha com base no GPS.</li> </ul>
<b>Mapa de Boleias</b>	<p>Requisitar e oferecer boleia;</p> <p><i>Matching</i> de boleia;</p> <p>Procurar condutores por localidade ou raio baseado num ponto específico;</p> <p>Boleia imediata: inserir morada/localidade &gt; verificar boleias que passam na localização indicada naquele momento &gt; requisitar boleia;</p> <p>Agendar boleia: mesmo processo que boleia imediata, mas com data e hora posteriores, sujeito a processo de aprovação;</p> <p><b>Como motorista:</b></p> <ul style="list-style-type: none"> <li>Boleia imediata: pode aceitar ou recusar pedido de boleia;</li> <li>Pedido de confirmação de realização de viagem agendada 1:30 e 30m antes da hora definida, os passageiros deverão ser notificados da decisão do motorista;</li> <li>Visualizar trajeto até ao ponto de encontro do passageiro;</li> <li>Pode aceitar ou recusar pedidos de boleia agendados;</li> </ul> <p><b>Como passageiro:</b></p> <ul style="list-style-type: none"> <li>Pode solicitar a participação regular numa boleia imediata, o motorista em questão será notificado para validação.</li> </ul>
<b>Social</b>	<p><b>Amigos:</b></p> <ul style="list-style-type: none"> <li>Deverá ser possível adicionar amigos e visualizar uma listagem dos mesmos;</li> <li>Deverá ser possível visualizar os pontos e avaliações de amigos;</li> <li>Deverá ser possível conversar-se via <i>chat</i> com os amigos;</li> </ul> <p><b>Chat:</b></p> <ul style="list-style-type: none"> <li>Deverá ser possível enviar mensagens privadas a um motorista, após este ter aceitado o pedido de contacto;</li> <li>Deverá ser automaticamente aberto um chat de grupo entre motorista e passageiros de uma boleia agendada.</li> </ul>
<b>Tratamento de Custos</b>	No final de cada viagem deve ser apresentado o valor que cada passageiro terá de pagar, baseado nos quilómetros e rota tomada pelo motorista;

	A divisão de custos aplica-se no ponto de entrada e de saída de cada passageiro, por isso é esperado que diferentes passageiros tenham valores diferentes a pagar.
<b>Elementos de Gamificação</b>	<p>Deverá ser possível avaliar cada boleia;</p> <p>Os condutores receberão algum tipo de ranking baseado nas avaliações positivas ou negativas;</p> <p>Os condutores poderão adquirir ícones de perfil, utilizando os pontos obtidos através da avaliação.</p>

### 4.3.2. Módulos

Com o levantamento das funcionalidades chave realizado e após discussão entre a equipa e orientadores, foi possível definir os módulos que iriam constituir a plataforma. A Tabela 4.4 contém os módulos levantados, respetivas descrições e prioridades.

Os módulos de Perfil, Boleias e Gestão de Notificações foram definidos como prioritários, pois contemplavam as funcionalidades correspondentes à gestão de informação de utilizadores, da partilha de boleias e das respetivas notificações, funcionalidades estas que foram consideradas determinantes para uma primeira versão funcional da plataforma.

Tabela 4.4 - Módulos levantados

ID	Nome	Descrição	Prioridade
M1	Módulo de Perfil	Componente que agrupa as funcionalidades relacionados com a autenticação, perfil e viaturas dos utilizadores	Must Have
M2	Módulo de Boleias	Componente que agrupa as funcionalidades relacionadas com os pedidos e oferta de boleias, histórico de boleias, gestão de grupos de boleia e agendamentos, classificações de utilizadores e algoritmos relacionados	Must Have
M3	Módulo de Feedback	Componente que agrupa as funcionalidades complementares ao módulo de boleias	Should Have
M4	Módulo de Comunicação	Componente que agrupa as funcionalidades relacionadas com o chat entre grupos de boleia e utilizadores para requisições de boleias	Should Have
M5	Módulo de Gamificação	Componente que agrupa as funcionalidades relacionadas com o sistema de níveis e conquistas	Should Have

<b>M6</b>	Módulo de Gestão de Notificações	Componente que agrupa as funcionalidades relacionadas com a gestão de notificações	Should Have
<b>M7</b>	Módulo de Estatísticas	Componente que agrupa as funcionalidades relacionadas com as estatísticas sobre os recursos/emissões poupadados com a utilização da aplicação	Could Have
<b>M8</b>	Módulo de Informação	Componente que agrupa as funcionalidades relacionadas com a apresentação de notícias/novidades e histórico de notificações	Could Have

## 4.4. ANÁLISE E ESPECIFICAÇÃO DE REQUISITOS

Neste subcapítulo apresentam-se a especificação dos requisitos levantados para a plataforma.

### 4.4.1. Requisitos Funcionais

Com base no *brainstorming* efetuado foi possível levantar mais de cem requisitos funcionais, o que demonstrou o grau de complexidade que uma plataforma deste género poderá ter.

A Tabela 4.5, Tabela 4.6 e Tabela 4.7 contêm os requisitos funcionais referentes aos módulos de Perfil, de Boleias e de Gestão de Notificações, respetivamente. Mesmo com o foco apenas nestes três módulos, houve um elevado número de requisitos especificados. Deste modo apenas os requisitos planeados de implementar no âmbito da primeira versão da plataforma estão presentes nesta secção do documento, os restantes requisitos poderão ser consultados no Anexo 3 – Restantes Requisitos Funcionais Levantados.

#### 4.4.1.1. Módulo de Perfil

Tabela 4.5 - Requisitos Funcionais Módulo de Perfil

ID	Descrição	Prioridade
RF1	O sistema deverá permitir a autenticação de utilizadores através do sistema de autenticação do IPS.	Must Have
RF2	O sistema deverá ser capaz de obter os dados do utilizador através do sistema de autenticação do IPS.	Must Have
RF3	O sistema deverá permitir ao utilizador visualizar o seu perfil.	Must Have

RF4	O sistema deverá permitir ao utilizador editar os dados, dentro do âmbito desta plataforma, do seu perfil.	Must Have
RF5	O sistema deverá permitir ao utilizador gerir as suas viaturas.	Must Have
RF6	O sistema deverá permitir ao utilizador bloquear outro utilizador.	Must Have
RF7	O sistema deverá permitir ao utilizador visualizar e gerir os utilizadores bloqueados	Must Have
RF8	O sistema deverá permitir ao utilizador aceder ao sistema de informação para recuperar a sua password.	Should Have
RF9	O sistema deverá permitir ao utilizador visualizar o perfil público de outros utilizadores.	Should Have

#### 4.4.1.2. Módulo de Boleias

Tabela 4.6 - Requisitos Funcionais Módulo de Boleias

ID	Descrição	Prioridade
RF13	O sistema deverá permitir ao utilizador alterar a vista do mapa/agendamentos entre Passageiro e Motorista, com as devidas funcionalidades associadas.	Must Have
RF14	O sistema deverá permitir ao utilizador visualizar a sua localização atual no mapa.	Must Have
RF15	O sistema deverá permitir ao utilizador, como motorista, gerir grupos de viagem.	Must Have
RF16	O sistema deverá permitir ao utilizador, como passageiro, pesquisar por motoristas com grupos de boleia criados dentro do raio de alcance e dos critérios de filtragem definidos.	Must Have
RF17	O sistema deverá permitir ao utilizador, como passageiro, visualizar os detalhes de um grupo de viagem.	Must Have
RF18	O sistema deverá permitir ao utilizador, como passageiro, realizar um pedido de boleia a um motorista que tenha sido sugerido na pesquisa.	Must Have
RF19	O sistema deverá notificar o motorista ao receber um pedido de boleia de um passageiro.	Must Have
RF20	O sistema deverá notificar o passageiro quando um dos seus pedidos de viagens é aceite ou recusado.	Must Have

RF21	O sistema deverá permitir ao utilizador, como motorista, inicializar a aplicação de navegação do dispositivo móvel com o seu ponto de partida (localização atual) e destino, tendo como ponto de passagem o ponto de encontro	Must Have
RF22	O sistema deverá permitir ao utilizador, como passageiro, inicializar a aplicação de navegação do dispositivo móvel com o seu ponto de partida (localização atual) e destino (ponto de encontro)	Must Have
RF23	O sistema deverá permitir ao utilizador, como motorista, visualizar, aceitar e recusar pedidos de boleias realizados por passageiros	Must Have
RF24	O sistema deverá permitir ao utilizador, como motorista, indicar quando chegou ao destino e estacionou	Must Have
RF97	O sistema deverá permitir ao utilizador renovar um grupo de viagem recorrente.	Must Have
RF98	O sistema deverá automaticamente suspender um grupo de viagem recorrente, que não tenha sido renovado, após a sua data de expiração.	Must Have
RF99	O sistema deverá notificar o motorista da necessidade de renovar o grupo de viagem caso continuo valido com alguns dias de antecedencia à data de expiração	Should Have

#### 4.4.1.3. Módulo de Gestão de Notificações

Tabela 4.7 - Requisitos Funcionais Módulo de Notificações

ID	Descrição	Prioridade
RF75	O sistema deverá permitir ao utilizador receber notificações push	Must Have

#### 4.4.2. Requisitos Não Funcionais

Tratando-se de uma primeira versão da plataforma em que se pretenderia apenas criar uma base de trabalho que pudesse ser usada e expandida posteriormente, os requisitos não funcionais não foram considerados prioridade, pelo que foram estabelecidos apenas os essenciais para o projeto.

#### 4.4.2.1. Requisitos de Qualidade

A nível de requisitos de qualidade, considerou-se a questão de segurança em relação à autenticação, atratividade, compreensão da interface, internacionalização do sistema e manutenibilidade do mesmo. Estes encontram-se descritos de acordo com a ISO/IEC 9126 [33] na Tabela 4.8, Tabela 4.9, Tabela 4.10, Tabela 4.11, Tabela 4.12 e Tabela 4.13.

Tabela 4.8 - Requisito de Qualidade RQ1

<b>RQ1: O sistema deverá prevenir o acesso de utilizadores que não façam parte da comunidade IPS</b>	
<b>Categoria</b>	Funcionalidade
<b>Classificação</b>	Segurança
<b>Teste</b>	O acesso é testado inserindo credenciais inexistentes no repositório de utilizadores do IPS no ecrã de autenticação da aplicação movel.
<b>Planeado</b>	O sistema deverá prevenir o acesso de utilizadores não pertencentes não repositórios de utilizadores do IPS.
<b>Pior Caso</b>	O sistema deverá prevenir o acesso de utilizadores que não tenham efetuado o devido registo das suas credenciais.

Tabela 4.9 - Requisito de Qualidade RQ2

<b>RQ2: A aplicação móvel deverá ter um aspeto visual apelativo o suficiente de modo que a maioria dos utilizadores achem que a mesma é visualmente apelativa.</b>	
<b>Categoria</b>	Usabilidade
<b>Classificação</b>	Atratividade
<b>Teste</b>	A apreciação estética da aplicação móvel será testada com recurso a um questionário com perguntas que tem como objetivo avaliar o aspeto visual da mesma. Estas questões têm de permitir analisar o nível de satisfação dos utilizadores, de modo a determinar se um está satisfeito com a estética da aplicação.
<b>Planeado</b>	Pelo menos 80% dos utilizadores consideram o aspeto visual da aplicação movel apelativo.
<b>Pior Caso</b>	Pelo menos 50% dos utilizadores consideram o aspeto visual da aplicação movel apelativo.

Tabela 4.10 - Requisito de Qualidade RQ3

<b>RQ3: A aplicação móvel deverá ser intuitiva o suficiente de modo que a maioria dos utilizadores usufruam das funcionalidades disponibilizadas sem dificuldades.</b>	
<b>Categoria</b>	Usabilidade
<b>Classificação</b>	Aprendizagem
<b>Teste</b>	Este requisito será testado com recurso a um questionário com perguntas referentes à usabilidade por parte dos utilizadores. Cada utilizador que identifique algum entrave em usufruir de qualquer das funcionalidades deverá identificar, no inquérito, a dificuldade em questão. Este utilizador será contabilizado como parte dos utilizadores que tiveram dificuldades na utilização da aplicação.
<b>Planeado</b>	Pelo menos 70% dos utilizadores consideram o a aplicação móvel intuitiva.
<b>Pior Caso</b>	Pelo menos 50% dos utilizadores consideram o a aplicação móvel intuitiva.

Tabela 4.11 - Requisito de Qualidade RQ4

<b>RQ4: O sistema deverá estar internacionalizado e/ou localizado para o idioma português e inglês.</b>	
<b>Categoria</b>	Usabilidade
<b>Classificação</b>	Compreensão
<b>Teste</b>	Alterar o idioma de apresentação a partir da aplicação móvel e verificar se os conteúdos estão devidamente traduzidos.
<b>Planeado</b>	O sistema suporta os idiomas português e inglês.
<b>Pior Caso</b>	O sistema suporta apenas o idioma português.

Tabela 4.12 - Requisito de Qualidade RQ5

<b>RQ5: O código-fonte do sistema deverá ser devidamente documentado.</b>	
<b>Categoria</b>	Manutenibilidade

<b>Classificação</b>	Analisabilidade
<b>Teste</b>	Este requisito poderá ser testado com a validação manual dos ficheiros de código-fonte, de modo a desenvolver uma lista ou obter um consenso da percentagem de código que está em devidamente documentado.
<b>Planeado</b>	Pelo menos 90% do código fonte do sistema deverá estar devidamente documentado devidamente documentado.
<b>Pior Caso</b>	Pelo menos 30% do código fonte do sistema deverá estar devidamente documentado devidamente documentado.

Tabela 4.13 - Requisito de Qualidade RQ6

<b>RQ6: O sistema deverá ser desenvolvido de forma a evitar um código “sujo”.</b>	
<b>Categoria</b>	Manutenibilidade
<b>Classificação</b>	Analisabilidade
<b>Teste</b>	Este requisito poderá ser testado com a validação manual dos ficheiros de código-fonte, de modo a desenvolver uma lista ou obter um consenso da percentagem de ficheiros que contem código “sujo”.
<b>Planeado</b>	Pelo menos 90% do código fonte do sistema deverá ser desenvolvido de modo a evitar código “sujo”.
<b>Pior Caso</b>	Pelo menos 40% do código fonte do sistema deverá ser desenvolvido de modo a evitar código “sujo”.

#### 4.4.2.2. Requisitos Ambientais

A Tabela 4.14 apresenta os principais requisitos ambientais levantados para o sistema a desenvolver.

Tabela 4.14 - Requisitos ambientais do sistema

ID	Categoria	Requisito
RA1	Hardware	O dispositivo móvel do utilizador deverá estar equipado com um sensor GPS

RA2	Software	A aplicação móvel deverá ser compatível com dispositivos Android 6+ e iOS 12+.
RA3	Cliente	O dispositivo móvel deverá ter uma ligação estável à Internet.
RA4	Software	A aplicação móvel deverá ser capaz de comunicar com o servidor do sistema através de uma RESTful API.
RA5	Software	O servidor deverá ser compatível com Node.js 12+.
RA6	Software	Os dados do sistema deverão ser armazenados no MongoDB 4.4+.

## 4.5. PROCESSOS DE NEGÓCIO

Os processos de oferecer boleias como motorista de um veículo e procurar pelas mesmas como passageiro possuem um elevado grau de complexidade, daí a necessidade de serem analisadas detalhadamente de modo a evitar incoerências lógicas no seu fluxo de trabalho, que poderiam comprometer o desenvolvimento do projeto. Para a equipa se certificar de que os processos foram adequadamente planeados, elaboraram-se diagramas de processo de negócios para as funcionalidades de maior relevância e complexidade, tendo ficado da responsabilidade do estudante o processo referente à pesquisa e pedido de boleia pelo passageiro.

### 4.5.1. Pesquisa e Pedido de Boleia pelo Passageiro

A Figura 4.2 e a Figura 4.3 representam os diagramas de processo de negócio “Procurar Boleia” e do respetivo subprocesso “Configura Parâmetros de Pesquisa”.

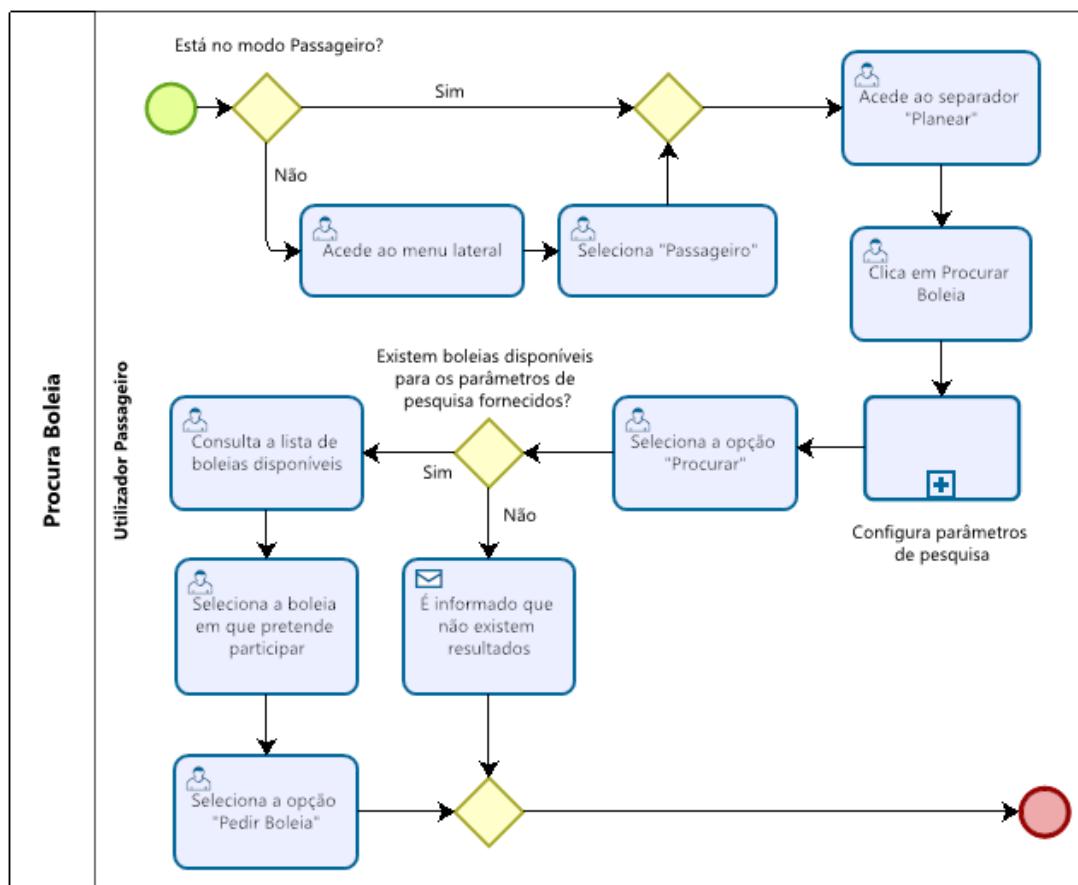


Figura 4.2 - Processo de negócio de procura de boleia

O processo de procurar boleia como passageiro começa com a transição para o modo de Passageiro na aplicação, caso este não seja o modo corrente, passando ao acesso do utilizador ao separador “Planear” da aplicação, onde este configura os parâmetros de pesquisa que se enquadram nas suas necessidades. Ao selecionar a opção “Procurar” serão apresentadas as boleias que correspondem aos parâmetros de pesquisa definidos pelo utilizador, caso existam, em seguida deverá selecionar em qual das boleias resultantes pretende participar. O processo termina após o utilizador selecionar a opção “Pedir Boleia”, podendo também terminar imediatamente caso não existam resultados para os parâmetros de pesquisa fornecidos.

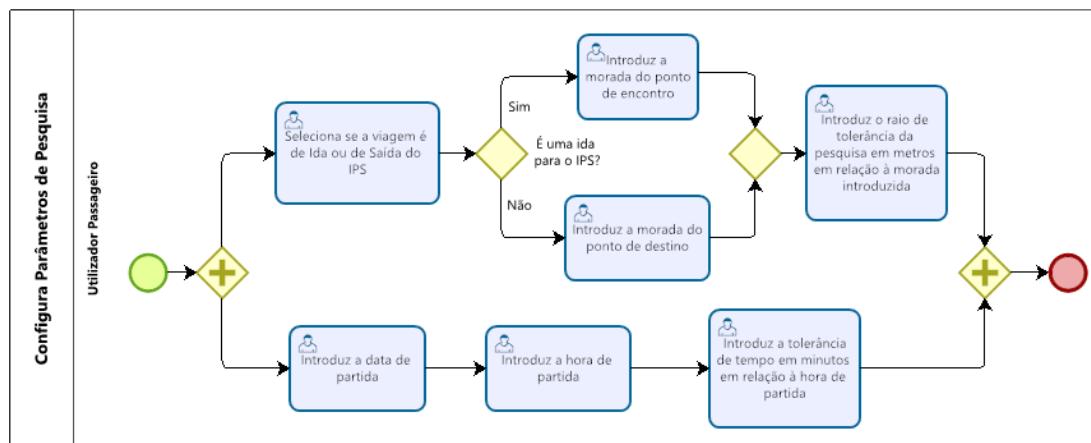


Figura 4.3 - Subprocesso de negócio de configuração de parâmetros de pesquisa de uma boleia

Os parâmetros referentes à pesquisa de boleia que se pretende efetuar têm todos de ser preenchidos antes de avançar para o passo seguinte. O utilizador deve indicar se a boleia será com destino ao IPS ou a um local particular, com partida na instituição de ensino, pelo que terá de indicar num mapa um ponto de partida no primeiro caso, ou um ponto de destino no segundo, devendo em seguida indicar o raio de tolerância em metros em relação ao ponto indicado de modo a obter um maior número de resultados. Adicionalmente o utilizador deve indicar a data e hora de partida da boleia, devendo ainda indicar a tolerância de tempo em minutos em relação à hora de partida indicada, possibilitando deste modo a apresentação de resultados num intervalo válido às necessidades do utilizador.

## 4.6. USER STORIES

Posteriormente ao processo de análise e levantamento de requisitos a equipa, juntamente com os orientadores, decidiu elaborar *user stories* com base nos requisitos funcionais selecionados para a primeira versão da plataforma, com a descrição simplificada das respetivas funcionalidades na perspetiva do utilizador.

O motivo pelo qual se começou por realizar a especificação de requisitos funcionais em vez de se ter apenas realizado a especificação de *user stories* deve-se ao facto de não existir um plano das funcionalidades que a plataforma deveria contemplar, tendo sido necessário realizar uma revisão detalhadas das funcionalidades entre a equipa e os orientadores.

A utilização de *user stories* para a elaboração dos cartões do quadro Kanban permitiu a visualização clara do trabalho a ser realizado para a equipa, permitindo ainda a avaliação destas funcionalidades por terceiros.

A Tabela 4.15 contém as *user stories* atribuídas ao estudante e a especificação dos requisitos funcionais que lhes deram origem, tal como a sua respetiva prioridade.

Tabela 4.15 - User Stories atribuídas ao estudante

Módulo	Requisitos Funcionais	User Storie	Prioridade
<b>Perfil</b>	RF1; RF2	Como um Utilizador, quero utilizar as minhas credenciais do IPS, de modo a poder-me autenticar na aplicação.	Muito Alto
	RF3; RF4	Como um Utilizador, quero ser capaz de visualizar e editar o meu perfil, de modo a poder atualizar os meus dados pessoais.	Muito Alto
	RF5	Como um Utilizador, quero gerir os meus veículos, com o objetivo de definir propriedades relevantes tais como tipo de combustível e consumo.	Muito Alto
	RF6; RF7	Como um Utilizador, quero poder gerir e bloquear outros utilizadores, de modo a poder ter controlo total das pessoas com quem eu posso interagir.	Baixo
	RF9	Como um Utilizador, quero poder ver o perfil de outros utilizadores, de modo a poder consultar a informação de possíveis motoristas e passageiro de um mesmo grupo de boleia.	Médio
<b>Boleias</b>	RF14	Como um Utilizador, quero ver a minha localização no mapa principal, de modo a poder ver onde é que estou localizado.	Alto
	RF15	Como um Motorista, quero gerir os meus grupos de boleia, de modo a poder defini-los e editá-los conforme necessário.	Muito Alto
	RF16	Como um Passageiro, quero procurar por grupos de boleia disponíveis dentro de um determinado raio à minha volta ou nas proximidades, tendo em consideração as minhas restrições horárias, de modo a poder encontrar uma boleia adequada.	Muito Alto
	RF17; RF18	Como um Passageiro, quero ser capaz de visualizar os detalhes de grupos de boleia pesquisados, de modo a poder fazer um pedido de boleia que se adequo a mim.	Alto

	RF19	Como um Motorista, quero ser notificado quando receber um pedido de boleia de um passageiro.	Baixo
	RF21	Como um Motorista, quero ser capaz de iniciar um grupo de boleia, de modo que a aplicação de mapas do meu dispositivo possa abrir com o meu destino e pontos de encontro do grupo.	Alto
	RF22	Como um Passageiro, quero ser capaz de visualizar a rota até ao ponto de encontro, do grupo de boleia ativo.	Médio
	RF24	Como um Motorista, ao estacionar quero ser capaz de definir a boleia atual como estando terminada.	Alto
<b>Gestão de Notificações</b>	RF75	Como um Utilizador, quero ser capaz de receber notificações PUSH no meu dispositivo.	Médio

## 4.7. RISCOS

A nível de riscos consideraram-se questões como a falta de *know-how* em relação ao tema *Car Pooling*, utilização de tecnologia e linguagens de programação desconhecidas pela equipa de desenvolvimento, questões relacionadas com a *user experience* e ainda as limitações a nível de realização de diversos tipos de teste devido ao distanciamento social obrigatório existente à data deste documento. Na Tabela 4.16 é possível visualizar em detalhe os principais riscos levantados.

Tabela 4.16 - Riscos

ID	Risco	Impacto
R1	A inexistência de <i>know how</i> por parte da equipa de desenvolvimento em relação ao tema <i>Car Pooling</i> é um potencial risco, pois o levantamento de requisitos faz parte das tarefas desta equipa e a falta de um <i>producto owner</i> na equipa poderá colocar em causa a pertinência dos requisitos levantados e selecionados para cada versão da plataforma.	Elevado
R2	A falta de experiência por parte da equipa de desenvolvimento em relação às tecnologias a utilizar neste projeto, apesar que pertinentes para os resultados esperados, poderá ter um impacto negativo na eficiência e qualidade do desenvolvimento. No entanto, tendo em consideração o alto nível de adaptabilidade e capacidade de	Reduzido

	autoaprendizagem dos membros desta equipa, considerou-se este um risco aceitável e que poderia facilmente ser mitigado através da utilização de tecnologias alternativas.	
R3	A falta de um membro na equipa de desenvolvimento especializado na criação de interfaces gráficas, junto com a inexperiência da equipa em relação à <i>user experience</i> esperada em aplicações moveis, levanta um risco a nível da usabilidade e qualidade gráfica da interface produzida.	Moderado
R4	Dada a situação pandémica vivida a nível mundial à data de redação deste documento e devido às limitações de distanciamento social existentes, considera-se que poderá existir dificuldade em encontrar utilizadores que nos permitam realizar os devidos testes de modo a obter-se o <i>feedback</i> necessário para garantir a qualidade do produto desenvolvido. No entanto, tratando-se de um projeto desenvolvido a nível universitário e direcionado para a comunidade da instituição em questão é possível procurar métodos alternativos de modo a reduzir o impacto deste risco.	Reduzido

## 4.8. INTERFACE COM O UTILIZADOR

A nível de interface com o utilizador existiu a preocupação de fornecer resultados de qualidade que fossem ao encontro das necessidades deste projeto, para tal a equipa decidiu realizar algumas *mockups* e definir algumas normas a seguir.

### 4.8.1. Mockups

Com o propósito de explorar as decisões visuais da interface com o utilizador e respetiva experiência de utilização, a equipa desenvolveu *mockups* das principais funcionalidades planeadas da aplicação móvel e posteriormente um protótipo interativo com recurso à ferramenta Adobe XD. Este processo permitiu planear e reconsiderar, juntamente com os orientadores, as hipóteses a serem tomadas quanto ao desenvolvimento da interface com o utilizador.

A Figura 4.4 representa os principais *mockups* elaboradas, correspondentes ao perfil, gestão de boleias como motorista e como passageiro, estatísticas, gamificação e chat.



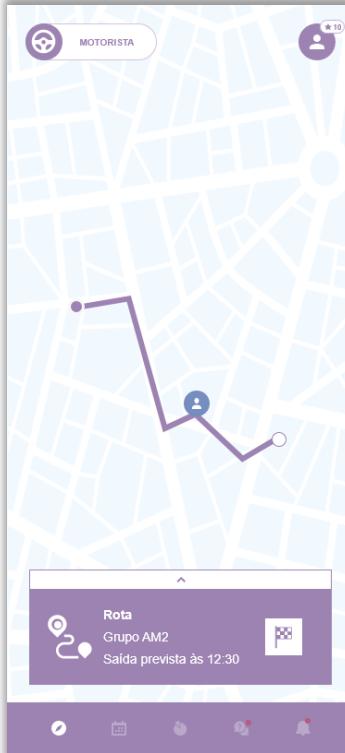
**PERFIL**

**Edgar Santos**  
Manteigadas, Setúbal

★★★★★  
22 Boleias Pedidas | 140 Boleias Dadas | 6 CONQUISTAS

**COMENTÁRIOS**  
"Sempre pontual"  
Há 6 dias

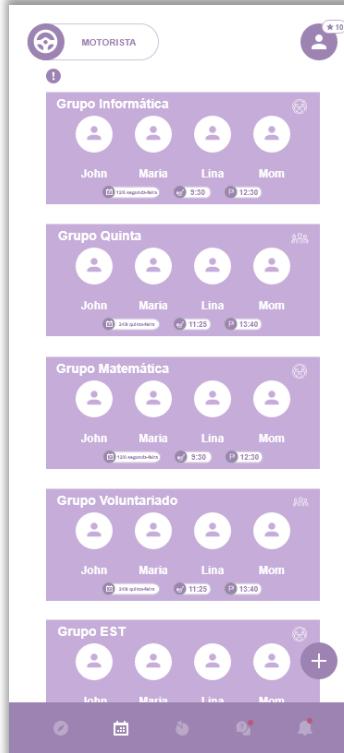
**HISTÓRICO**  
20 KM | 120 DETALHES



**MOTORISTA**

Map showing a route with a purple line and a blue dot indicating the current location.

**Rota**  
Grupo AM2  
Saída prevista às 12:30



**MOTORISTA**

- Grupo Informática**: John, Maria, Lina, Mom (120 passageiros)
- Grupo Quinta**: John, Maria, Lina, Mom (210 passageiros)
- Grupo Matemática**: John, Maria, Lina, Mom (130 passageiros)
- Grupo Voluntariado**: John, Maria, Lina, Mom (310 passageiros)
- Grupo EST**: John, Maria, Lina, Mom (110 passageiros)



**CRIAR GRUPO DE BOLEIA**

Map showing a route with a purple line and a blue dot indicating the current location.

497 Evergreen Rd. Roseville, CA 95673

IDADE SAÍDA

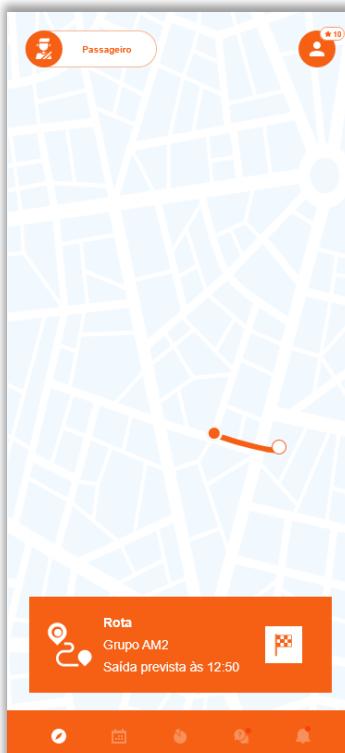
Ponto de Partida/Chegada

Horas de Chegada/Saída

Nome do Grupo

John, Maria, Lina, Mom

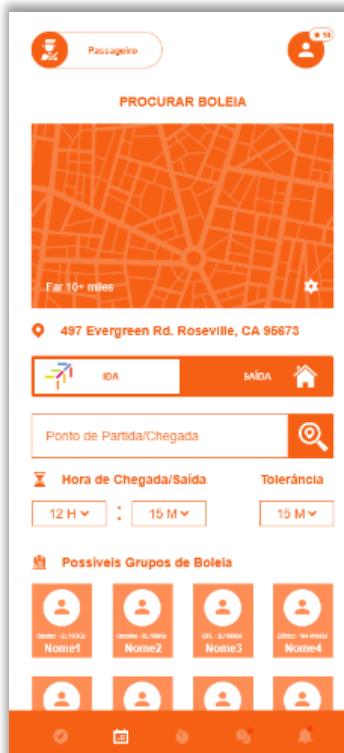
**CREAR**



**Passageiro**

Map showing a route with an orange line and a red dot indicating the current location.

**Rota**  
Grupo AM2  
Saída prevista às 12:50



**PROCURAR BOLEIA**

Map showing a route with an orange line and a red dot indicating the current location.

497 Evergreen Rd. Roseville, CA 95673

IDADE SAÍDA

Ponto de Partida/Chegada

Horas de Chegada/Saída

Tolerância

Possíveis Grupos de Boleia

- Nome1
- Nome2
- Nome3
- Nome4

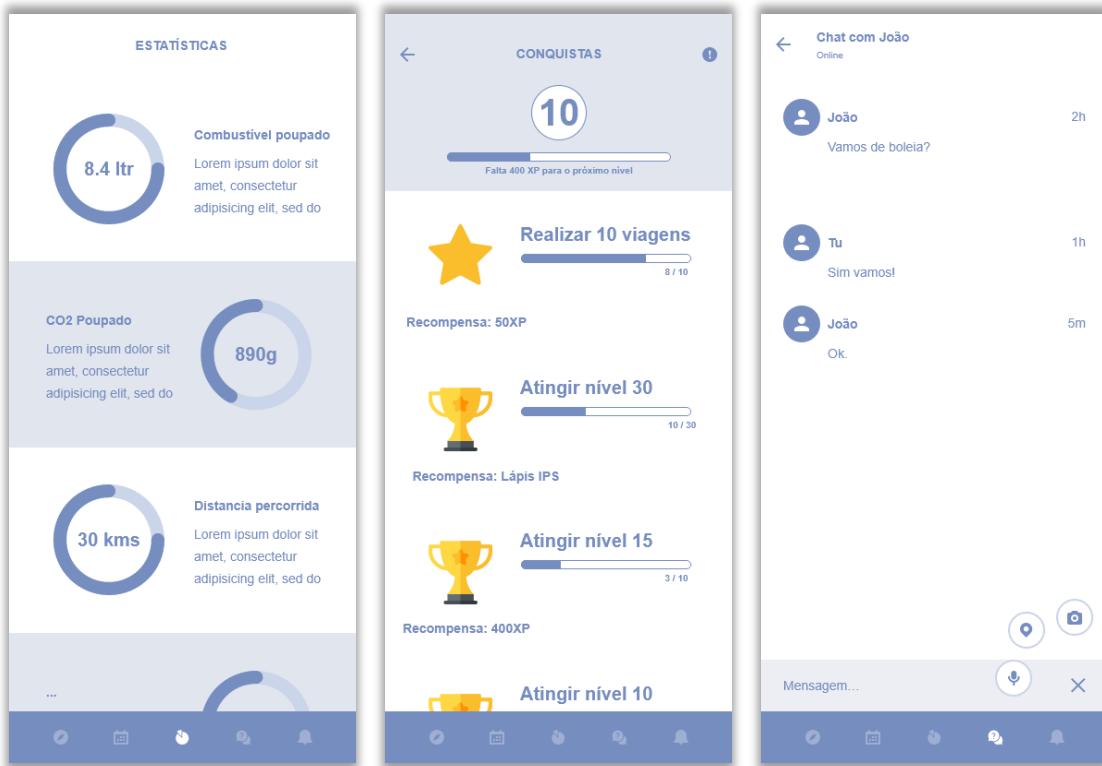


Figura 4.4 - Principais *mockups* elaboradas

#### 4.8.2. Normas

Devido à equipa não ter experiência com Flutter, nem conhecimento das limitações desta tecnologia, não foram definidas normas rigorosas em relação à interface com o utilizador. Tendo as únicas normas estabelecidas sido:

- A interface com o utilizador da aplicação deverá seguir as normas do Material Design [34], criado pela Google com o objetivo de ajudar equipas a construir experiências digitais de alta qualidade para aplicações móveis e web;
- Os ícones utilizados na aplicação deverão ser provenientes do projeto Material Design Icons [35] mantido por Austin Andrews;
- Os ecrãs relacionados com as funcionalidades do motorista deverão utilizar como cor principal o roxo;
- Os ecrãs relacionados com as funcionalidades do passageiro deverão utilizar como cor principal o cor-de-laranja;
- Os ecrãs de funcionalidades gerais aplicáveis a ambos os modos deverão utilizar como cor principal o azul.

## 4.9. PERSISTÊNCIA DE DADOS

Neste capítulo é feito, de forma sucinta, a descrição das coleções da base de dados.

Começasse por apresentar-se o modelo inicialmente planeado pelo estudante e em seguida apresenta-se o modelo efetivamente implementado, enumerado e justificado as principais diferenças entre eles.

Apesar da natureza não-relacional do MongoDB, pretende-se que os modelos de dados aqui apresentados demonstrem o tipo de relação existente entre as várias coleções, sendo tal indicação meramente informativa.

### 4.9.1. Modelo de Dados Planeado

Na Figura 4.5 pode visualizar-se o diagrama do modelo de dados inicialmente planeado. Este levantamento terá sido realizado pelo estudante na fase inicial do projeto, após o levantamento de módulos e requisitos. Não possuindo ainda uma visão global do projeto e respetivos dados o estudante focou-se nas principais coleções e atributos.

A nível de coleções tem-se a de notificações que possui os *templates* das diversas notificações possíveis do sistema, a de histórico de notificações que corresponde às notificações já enviadas ao utilizador, a coleção de utilizadores com os principais atributos levantados pelo estudante tendo em conta a existência do módulo de gamificação, a coleção de grupos de boleia que possui todos os dados necessários ao funcionamento de um grupo, a coleção histórico de boleias que possui uma referência à coleção de grupos de boleia adicionando a informação da data e respetivos custos a uma viagem já realizada. Foram ainda planeadas as coleções direcionadas ao módulo de *chat*, nomeadamente a de salas de chat e a de mensagens de chat pertencentes a uma dada sala.

Por fim, foi considerada a necessidade de uma ou mais coleções para o módulo de gamificação e conquistas. No entanto, dada a complexidade de tal módulo, e não fazendo o mesmo parte da primeira versão da plataforma, o estudante achou por bem deixar este planeamento para uma fase futura em que o mesmo seja justificável devido à provável necessidade de realização de investigação sobre o tema.

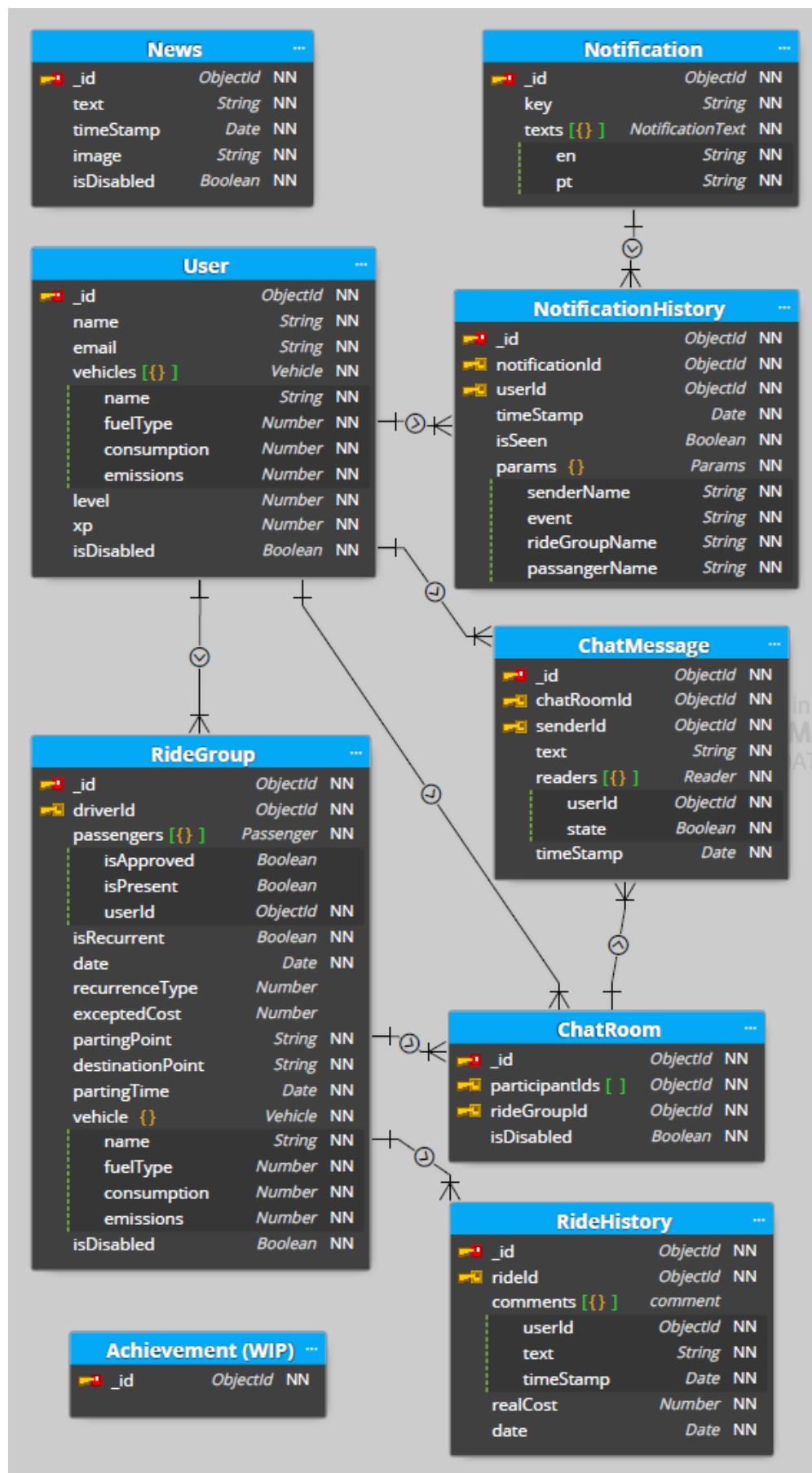


Figura 4.5 - Diagrama do modelo de dados planeado

## 4.9.2. Modelo de Dados Implementado

Durante o desenvolvimento da primeira versão da plataforma foram surgindo várias necessidades que não teriam sido contempladas no planeamento inicial, dando origem a um novo modelo de dados. A Figura 4.6 representa o modelo de dados efetivamente implementado.

Apesar deste modelo ser essencialmente igual ao planeado, as principais diferenças encontram-se nos atributos de cada coleção, que de forma geral aumentaram de quantidade significativamente dada a necessidade de guardar diversos outros dados que não teriam sido inicialmente contemplados.

A coleção de histórico de notificações terá sofrido uma alteração no nome passando-se a chamar de notificações do utilizador. A nível de atributos possui um maior detalhe nos dados persistidos.

Revendo os requisitos funcionais e módulos levantados deu-se pela falta da coleção de notícias. Nesta versão também decidiu-se retirar a coleção de conquistas pois não se conhecia qual a estrutura que esta deveria de tomar.

As coleções que sofreram maiores alterações foram as relacionadas ao utilizador e às boleias, a nível do utilizador terão sido adicionados um considerável número de atributos, sendo que o documento de veículos que antes pertencia à coleção de utilizadores agora possui a sua própria coleção chamada de veículos, sendo que na coleção de utilizadores apenas é guardada uma referência para o mesmo. A nível das boleias terão sido criadas duas novas coleções, a coleção de rotas de viagem onde se pretende guardar como um tipo de *cache* as rotas calculadas pelo Google Maps Platform Directions API e a coleção de agendamentos de boleias, esta última permitiu flexibilizar o funcionamento do sistema de boleias dando a possibilidade de um grupo de boleias possuir a qualquer momento zero, um ou mais agendamentos de boleias facilitando a gestão das boleias recorrentes.

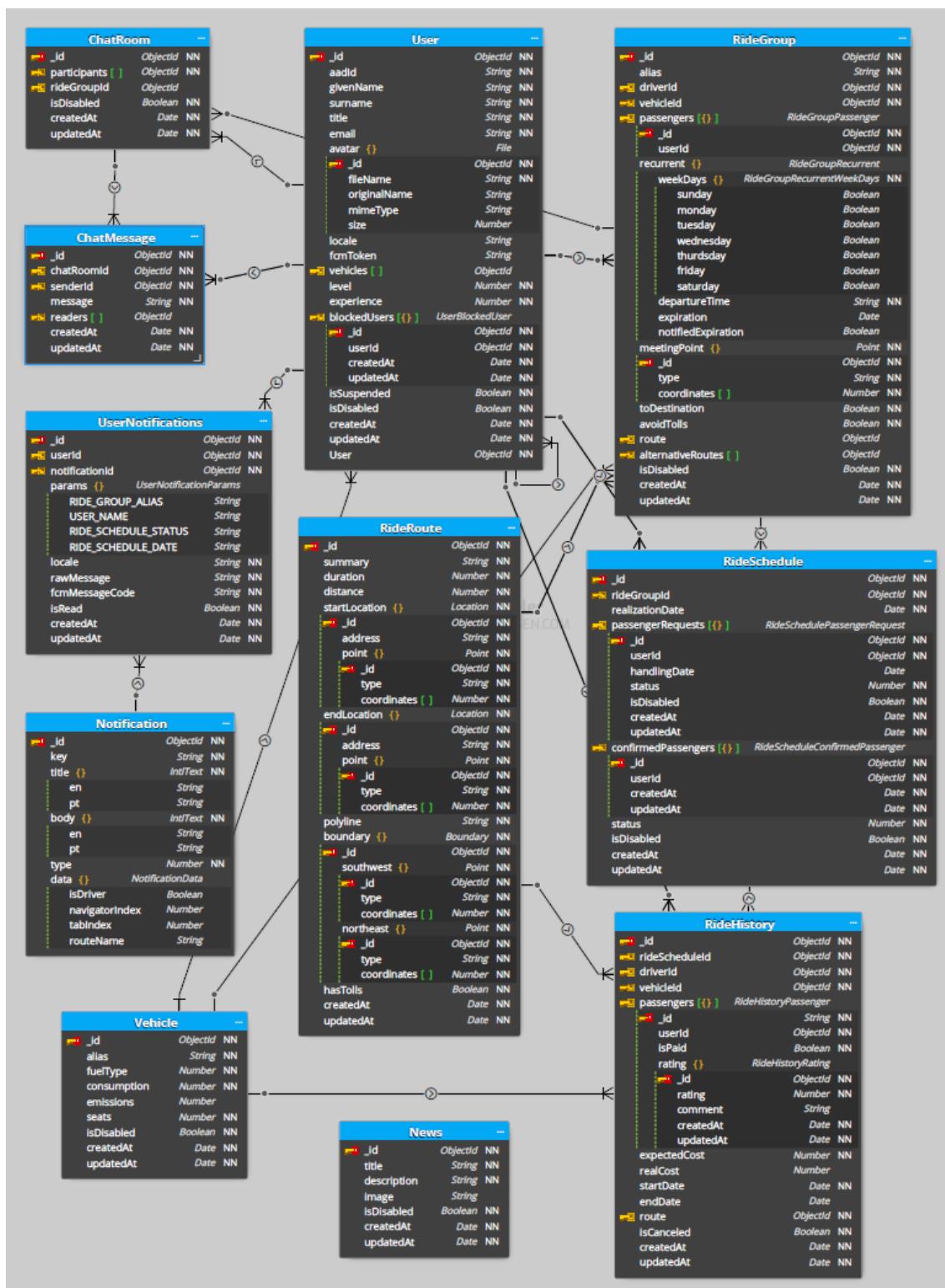


Figura 4.6 - Diagrama do modelo de dados implementado

## 4.10. ARQUITETURA GERAL

Tratando-se de uma aplicação móvel existe a necessidade da mesma ter acesso a uma API que irá processar todos os pedidos realizados pelo cliente, possibilitando a existência de

conteúdo dinâmico. Devido a ser uma plataforma destinada ao uso pela comunidade IPS considerou-se que os serviços necessários ficassem hospedados nas infraestruturas da instituição.

O Google Maps Platform é responsável por fornecer à aplicação móvel mapas dinâmicos e interativos através do serviço Maps SKD, é também responsável pelo cálculo de rotas no *backend* das viagens agendadas com recurso ao serviço Directions API de modo a estas serem persistidas e posteriormente servidas quando solicitado.

Ainda dentro do ecossistema da Google utilizou-se a plataforma Firebase, mais concretamente o serviço Firebase Cloud Messaging, responsável pelo envio de notificações do tipo PUSH para dispositivos Android e iOS.

Por último, tendo em conta que um dos pontos fulcrais de uma plataforma que pretende servir uma comunidade fechada encorajando uma utilização segura da mesma, recorreu-se ao serviço Azure Active Directory da Microsoft para a integração do sistema de autenticação da instituição com a plataforma a desenvolver.

A Figura 4.7 ilustra a arquitetura geral do sistema implementado.

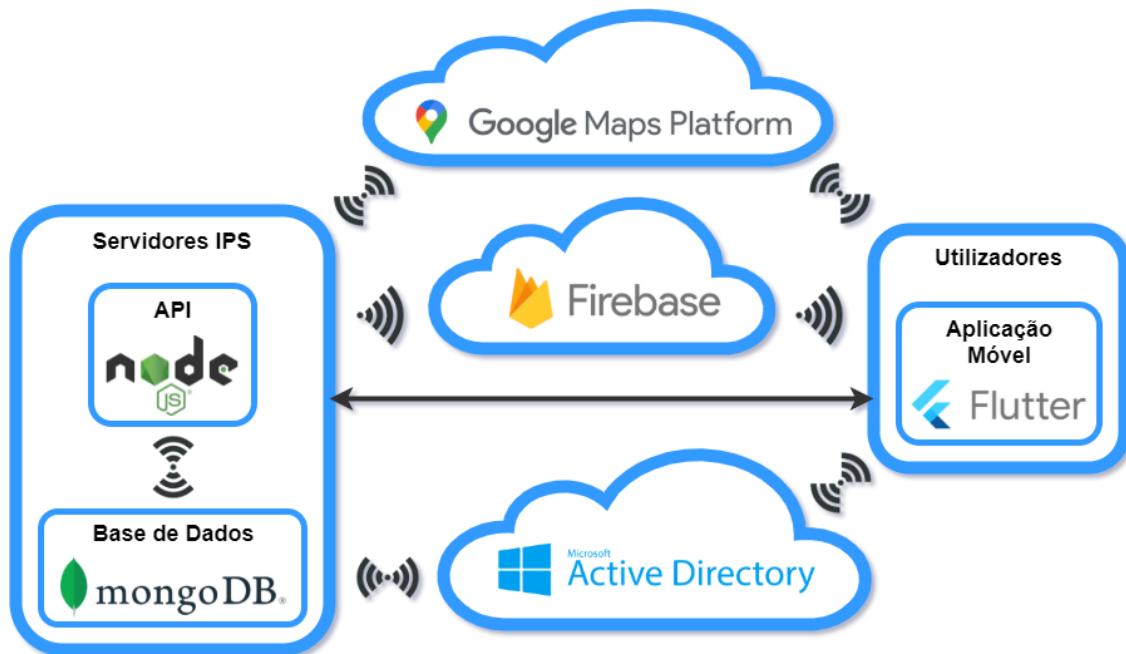


Figura 4.7 - Diagrama de arquitetura geral do sistema

## 4.11. ARQUITETURA FÍSICA

Neste subcapítulo é apresentado o diagrama de componentes onde mostra a estrutura do sistema de software e o diagrama de instalação que descreve os componentes de hardware e software e a sua interação.

## 4.11.1. Diagrama de Componentes

Na Figura 4.8 é apresentado o diagrama de componentes referente ao *back-end* da plataforma. Neste diagrama é possível visualizar que o sistema se encontra dividido em três partes menores totalmente independentes entre elas.

Das três partes que constituem o sistema, têm-se o subsistema utilitário (*utils*) que agrupa os componentes de gestão de boleias e de notificações, componentes estes que fornecem apenas recursos para uso interno da aplicação e constituem os principais elementos do sistema. O subsistema de autenticação (*authentication*) disponibiliza apenas um componente que é responsável pela verificação da autenticação e pelo login, sendo este último acessível por sistemas exteriores.

Por último, o subsistema de utilizadores (*users*) que constitui a maioria do sistema implementado e é constituído por sete componentes, que englobam funcionalidades como a disponibilização e gestão do perfil de utilizador, a gestão de grupos de boleia e respetivos agendamentos, definições da conta de utilizador, gestão de utilizadores bloqueados e gestão de veículos. Destes sete componentes, todos eles disponibilizam recursos para uso de sistemas exteriores. De modo geral, pode-se considerar este subsistema o menos independente entre os três, pois o seu correto funcionamento requer a presença do subsistema de autenticação e de utilitários, sendo, no entanto, o componente de gestão de boleias o único realmente essencial de estar presente no sistema.

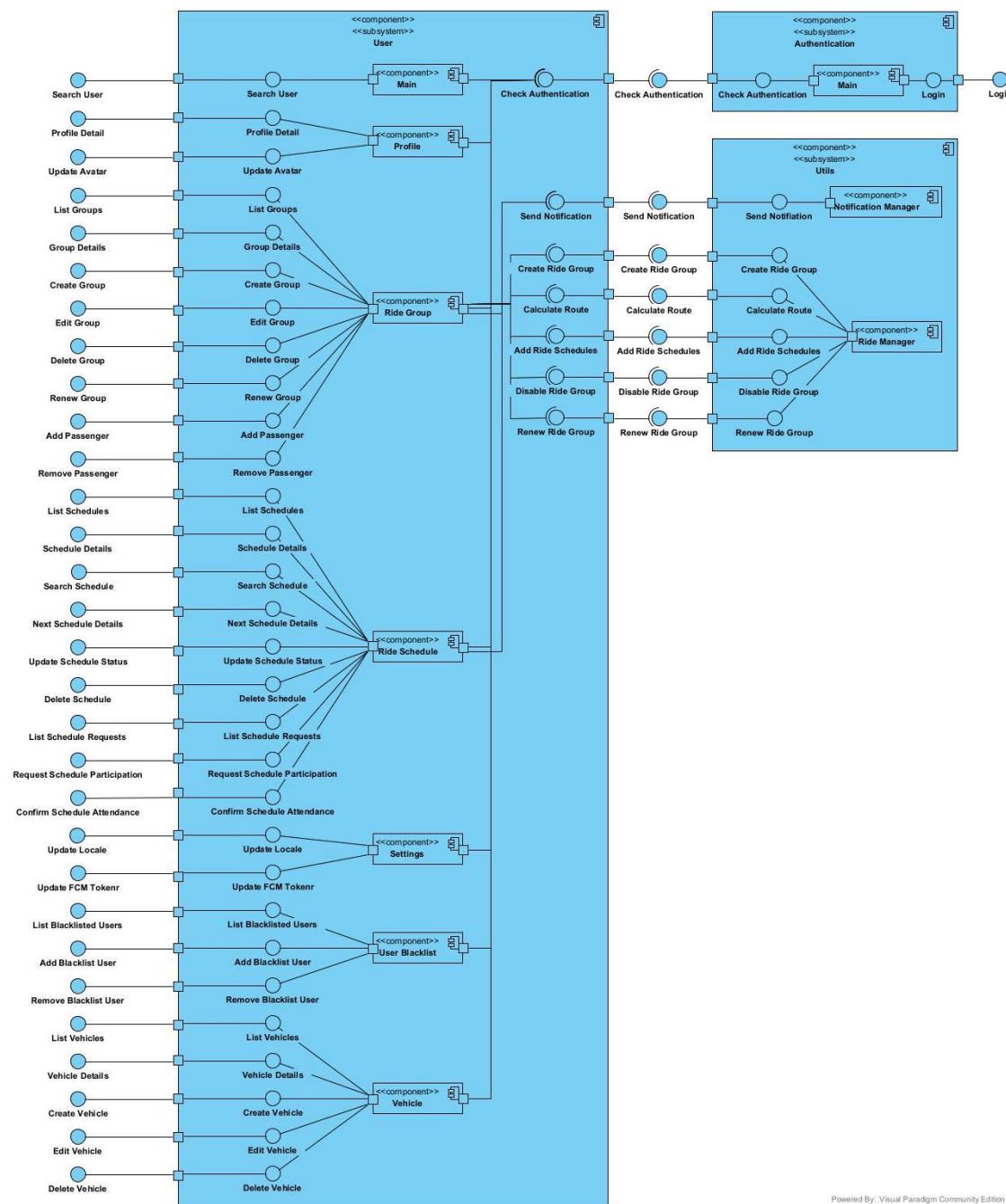


Figura 4.8 - Diagrama de componentes do back-end

#### 4.11.2. Diagrama de Instalação

Na Figura 4.9 é apresentado o diagrama de instalação do sistema. De modo geral o sistema encontra-se dividido em cinco pacotes independentes, que correspondem a diferentes prestadores de serviços, nomeadamente a Google, Apple e Microsoft, os restantes dois correspondem ao utilizador da aplicação móvel e à infraestrutura do IPS.

A nível da infraestrutura do IPS, foi cedido ao Sustain.RD o acesso a uma VPS com o sistema operativo Windows. Nesta VPS encontra-se a API Server desenvolvida no âmbito deste

projeto, que por sua vez disponibiliza a RESTful API para utilização externa. Ainda na VPS, tem-se como instância do servidor de base de dados MongoDB, sendo este último acedido pelo primeiro. A API Server depende de dois serviços externos fornecidos pela Google, nomeadamente o Google Maps Platform Directions API e o Firebase Cloud Messaging, ambos os serviços acessíveis através do protocolo TCP/IP. De notar que a VPS se encontra dentro de uma rede privada sendo necessário a utilização de uma VPN para a gestão da mesma.

A nível do utilizador existem duas possibilidades. Poderá ser utilizado um smartphone com Android ou com iOS, dependente do sistema em questão os serviços externos utilizados variam. Tanto para Android como para iOS, ambos utilizam a RESTful API da API Server através do protocolo TCP/IP, outro serviço comum a ambos é o Azure Active Directory da Microsoft, acessível através do protocolo OAuth 2.0.

Dependendo da plataforma, no caso do Android é utilizado o serviço Maps SDK for Android da Google Maps Platform e o serviço Firebase Cloud Messaging, já no caso do iOS é ligeiramente diferente pois é utilizado o serviço Maps SDK for iOS da Google Maps Platform e em vez de se utilizar o FCM utiliza-se o Apple Push Notification Service, todos estes serviços são acessíveis através de TCP/IP.

Por fim, o serviço Firebase Cloud Messaging comunica de forma autónoma com o serviço Apple Push Notification Service através de TCP/IP.

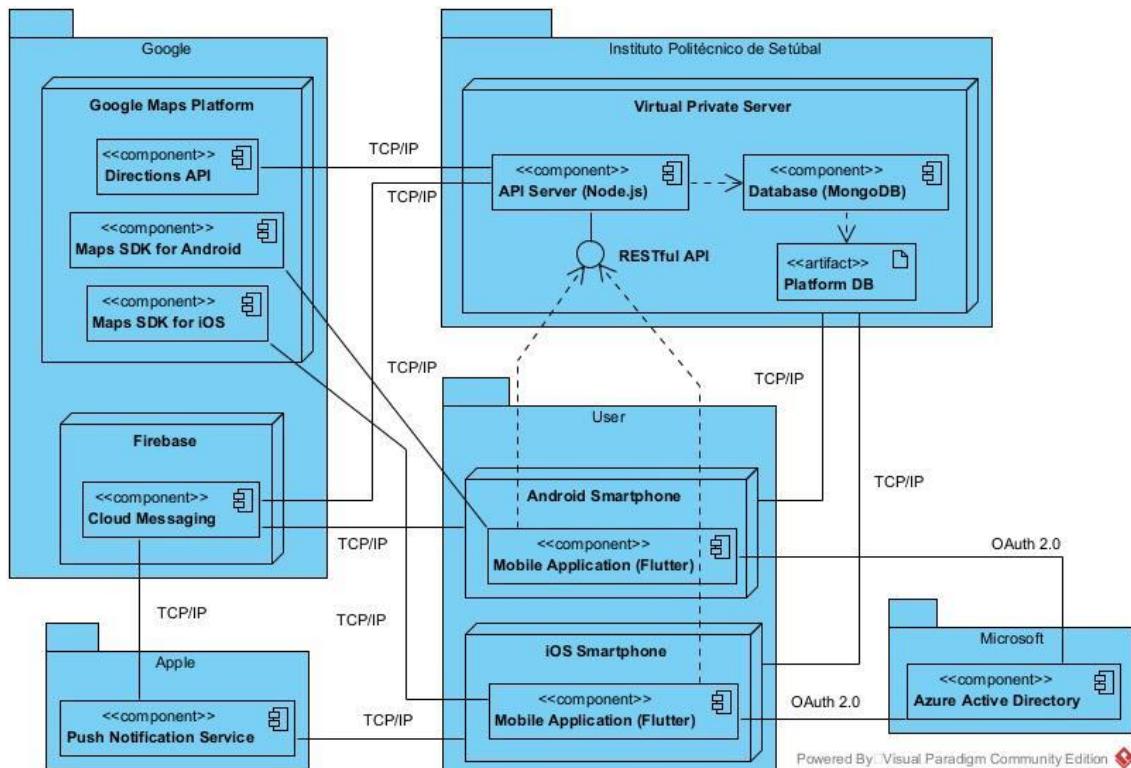


Figura 4.9 - Diagrama de instalação do sistema

## 4.12. DIAGRAMAS DE ESTADOS

Dada a complexidade da funcionalidade de agendamento de boleias, considerou-se pertinente realizar um levantamento de quais os estados que um agendamento poderia tomar, tal como as suas relações.

Para este fim, o estudante decidiu realizar um diagrama de estados, representado na Figura 4.10, onde é possível verificar que uma boleia começa sempre com o estado de agendada (*scheduled*), em seguida passa para o estado de esperando por passageiros (*waiting passengers*). A partir deste estado há duas hipóteses, a boleia pode ser cancelada (*canceled*) ou pode ser iniciada (*started*), sendo que uma boleia iniciada pode ser terminada de modo normal sendo marcada como finalizada (*finished*), podendo também ser cancelada (*canceled*). Uma boleia agendada que ainda não tenha sido iniciada não poderá ser marcada como cancelada, pois o estado de cancelado tem como propósito indicar que algo correu mal e a mesma não irá avançar. Neste caso, a boleia será apenas identificada como eliminada, tornando-se o seu estado irrelevante.

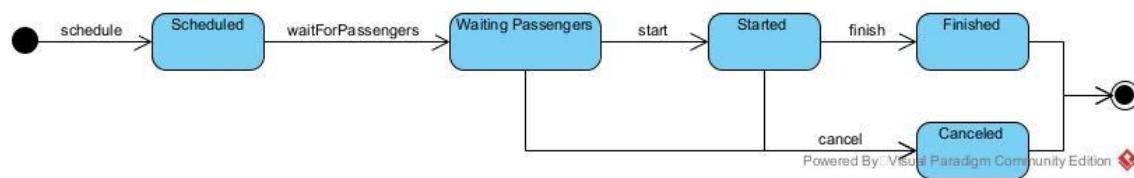


Figura 4.10 - Diagrama de estados de um agendamento de boleia

## 4.13. NORMAS DE CODIFICAÇÃO DO BACK-END

O JavaScript Standard Style, também conhecido por StandardJS<sup>4</sup>, terá sido utilizado como conjunto de regras de codificação em JavaScript a seguir neste projeto para o *back-end*. O motivo para esta decisão deve-se ao facto de o mesmo ser utilizado por um vasto conjunto de aplicações conhecidas, tais como Electron, Atom e GoDaddy, sendo cada vez mais utilizado como *standard* em aplicações de código-aberto. Como ponto adicional a favor, a biblioteca *eslint* para Node.js foi utilizada, sendo responsável pela validação e correção automática do código escrito, que possui suporte nativo a este estilo de codificação.

No seguinte endereço <https://github.com/standard/standard/blob/master/docs/RULES-ptbr.md> encontra-se uma lista das regras aplicadas pelo StandardJS. A especificação de JavaScript utilizada neste projeto terá sido o ECMAScript2020, padronizada pela Ecma International na especificação ECMA-262 e ISSO/IEC 16262.

A documentação do código terá sido realizada seguindo as normas do JSDoc<sup>5</sup>, utilizando as respetivas *tags* quando devido, conforme mostrado no Excerto de Código 4.1.

<sup>4</sup> <https://standardjs.com/>

<sup>5</sup> <https://jsdoc.app/>

```
/**
 * @module UserSettingRoute
 * @category Routes
 */

'use strict'

/**
 * Update the user's settings firebase cloud messaging registration token.
 *
 * @name User Settings Set FCM Token
 * @path {PUT} /user/:user/settings/fcmtoken
 * @auth This route requires a valid user's token. If authentication fails it will return a 401 http status code.
 * @header {string} Authorization The authentication user's token.
 * @params {string} :user Use 'me' to reference the logged-in user or specify a user's id to act on the behalf of that user.
 * @body {string} fcmToken The new user's fcm token.
 * @code {204} The user fcm token update succeeded.
 * @code {500} An internal server error.
 * @example <caption>Possible input:</caption>
 * {
 *   "fcmToken": "ABC123"
 * }
 */
router.put('/fcmtoken', UserSettingsController.editSettingsFcmToken)

/**
 * The router-layer of the user's setting.
 */
export default router
```

Excerto de Código 4.1 - Exemplo de documentação do *Back-end*

## 4.14. CUSTOS

A nível de custos não foi realizado um estudo detalhado devido a tratar-se de um projeto desenvolvido a nível interno de e para a própria instituição de ensino.

Deste modo, a nível de desenvolvimento e ferramentas utilizadas existiu o cuidado por parte da equipa de preferir opções gratuitas ou, que de algum modo, fossem disponibilizadas sem custos adicionais, através da própria instituição como, por exemplo, através do programa GitHub Student Developer Pack [36].

Em relação aos custos de hospedagem, considerou-se que os mesmos não seriam relevantes pois a instituição em questão possui a sua própria infraestrutura interna de servidores dedicados para este fim.

Os únicos custos que não foram possíveis evitar, de modo a garantir a qualidade do produto desenvolvido, foram os do serviço Web Directions API da Google. O serviço em questão é cobrado com base na quota de utilização. Na Tabela 4.17 - Custo mensal do serviço Google Directions API é possível visualizar os respetivos custos do mesmo em dólares americanos à data de redação deste documento. Durante o desenvolvimento foi cedida uma conta de testes por parte do centro de investigação.

Tabela 4.17 - Custo mensal do serviço Google Directions API

0 – 100,000 pedidos	100,001 – 500,000 pedidos	500,000+ pedidos
0.005 USD por pedido (5.00 USD por 1000)	0.004 USD por pedido (4.00 USD por 1000)	Sob consulta

## 5. Descrição do Trabalho Realizado

Este capítulo dedica-se à exposição das funcionalidades implementadas pelo estudante, tanto na RESTful API como na aplicação móvel, tal como a descrição de todo o trabalho inicial de preparação envolvente.

### 5.1. CONFIGURAÇÃO DO JIRA SOFTWARE

O estudante ficou responsável pela preparação do Jira Software, com o propósito de se tirar um melhor proveito desta ferramenta para a gestão do trabalho associado ao projeto. Começou-se por registar-se uma conta na página web da Atlassian, empresa responsável pelo Jira, tendo-se em seguida avançado com a criação de uma organização no Jira Software Cloud e de um projeto dentro dessa organização. Após a criação do projeto no Jira, avançou-se com a configuração do mesmo, tendo-se definido alguns detalhes, como a categoria do projeto a desenvolver e uma descrição do mesmo, adicionalmente o estudante também adicionou um dos orientadores do projeto, o professor Rui Madeira, como líder do projeto no Jira.

Com o propósito de facilitar a percepção do progresso do projeto, o estudante decidiu registar no Jira os dois primeiros entregáveis, permitindo assim a fácil visualização da progressão dos mesmos. Na Figura 5.1 é possível visualizar uma captura de ecrã do registo dos entregáveis no Jira. De notar que as datas de início e de lançamento apresentadas na figura são previsões, sendo que a realidade, tal como referido no subcapítulo 4.2.3, foi consideravelmente diferente.

Já a nível da separação do contexto das tarefas a realizar, o estudante decidiu criar três componentes no Jira, de modo a agregar as tarefas relativas ao *Back-end*, ao *Front-end* e à instalação e/ou configuração de serviços de terceiros, permitindo deste modo a rápida visualização da quantidade de trabalho pendente em relação a cada um destes componentes, sendo que na Figura 5.2 é possível visualizar uma captura de ecrã da configuração aplicada no Jira a este nível.

Projects / CarPooling4IPS

## Releases

[Create version](#)

Version	Status	Progress	Start date	Release date	Description	...
Delivery 2	UN...	<div style="width: 20%; background-color: #ccc; height: 10px;"></div>	Sep 28, 2020	Oct 27, 2020	Second delivery with the implementation of relevant but not mandatory features	...
Delivery 1	UN...	<div style="width: 80%; background-color: #007bff; height: 10px;"></div>	Aug 17, 2020	Sep 25, 2020	First delivery with the implementation of mandatory features	...

Figura 5.1 - Captura de ecrã dos entregáveis registados no Jira

Projects / CarPooling4IPS / Project settings

## Components

[Create component](#)

Component	Description	Component lead	Default assignee	Issues	...
Services	Component that aggregates all tasks related to the installation and/or configuration of required services	 Edgar Santos	Project default	1 issue	...
Front-end Mobile	Component that aggregates all the tasks related to the front-end, specifically the mobile application	 HenochV	Project default	26 issues	...
Back-end	Component that aggregates all tasks related to the back-end	 Edgar Santos	Project default	25 issues	...

Figura 5.2 - Captura de ecrã dos componentes registados no Jira

O Jira permite a criação de quadros de tarefas em *Scrum* ou *Kanban*, tendo-se pelos motivos já enumerados no subcapítulo 2.4 gerado um quadro do tipo *Kanban*, no qual terão sido adicionadas as *user stories* e respetivas tarefas associadas a cada membro da equipa. Existiu o cuidado de adicionar uma descrição e anexos com a informação considerada relevante à realização das tarefas, tal como a indicação do requisito funcional, da prioridade, do componente, do entregável e do responsável atribuído às mesmas. Tendo ainda existido o cuidado extra de se associar as tarefas relacionadas entre si, deixando a indicação de quais poderiam impedir ou limitar o desenvolvimento de outras. Para além da criação das *user stories* e tarefas supramencionadas, foram ainda criados *Epics* com o propósito de agregar as *user stories* criadas, deste modo decidiu-se que os *Epics* iriam representar os módulos levantados para o desenvolvimento da plataforma. De notar que nem todas as tarefas foram associadas a *user stories*, isto deve-se ao facto destas tarefas não estarem diretamente associadas a um ou mais requisitos funcionais da plataforma, tais como tarefas de preparação de instâncias de projeto ou configuração de serviços de terceiros.

No âmbito da realização do primeiro entregável foram criados no Jira: 3 módulos (*Epics*), 18 *user stories*, 43 tarefas e 6 tarefas independentes não associadas a qualquer *user story*. A Figura 5.3 representa um gráfico circular dos dados supramencionados. A especificação das *user stories* que deram origem às tarefas podem ser consultadas no subcapítulo 4.6. No Anexo 2 – Tarefas Extraídas das User Stories encontram-se as tarefas planeadas e adicionadas ao Jira Software de acordo com as *user stories* descobertas.

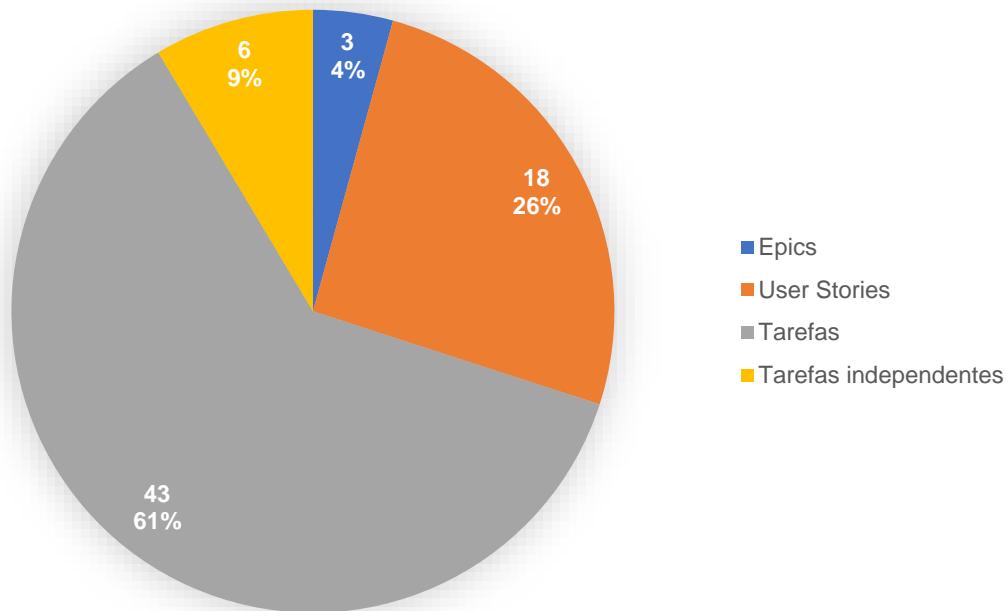
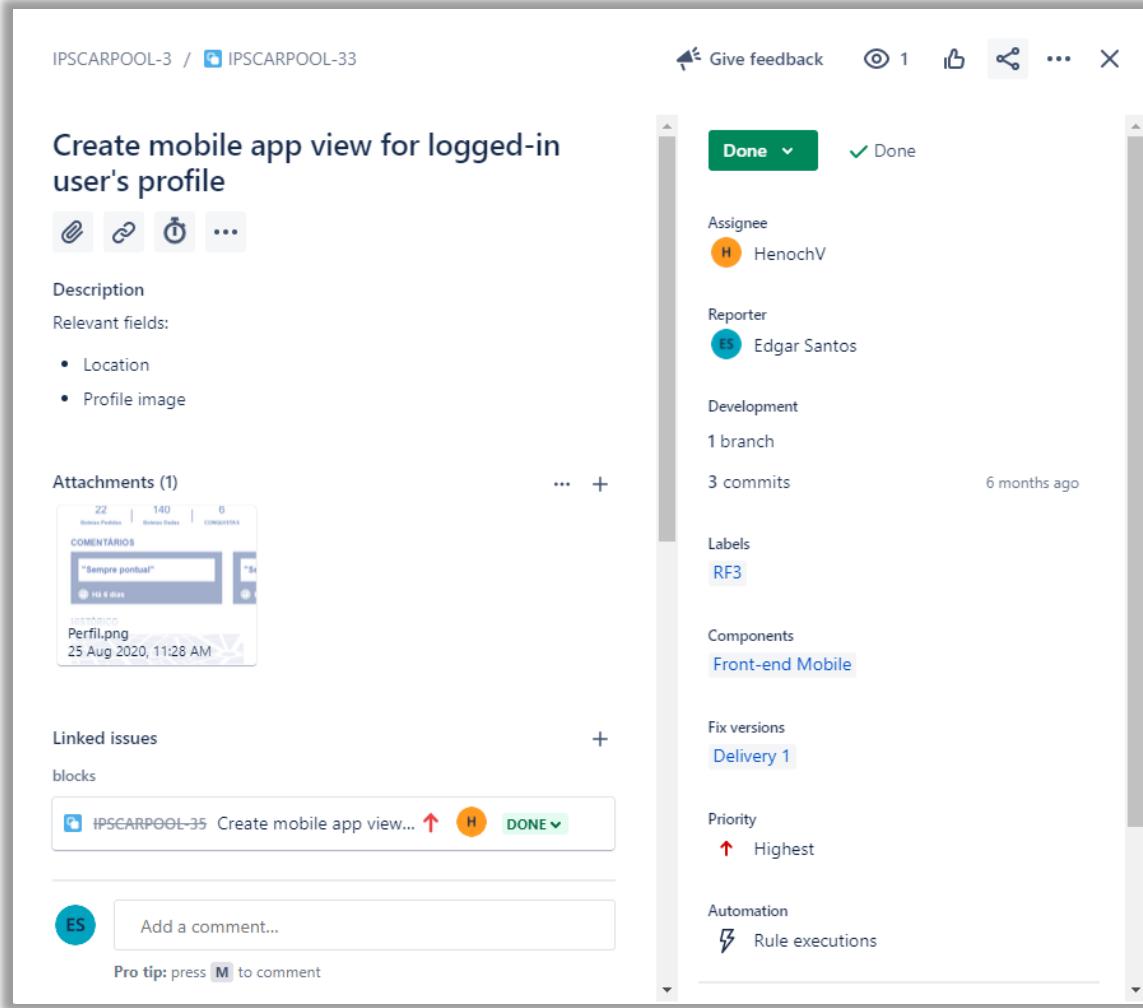


Figura 5.3 - Gráfico circular dos módulos, *user stories*, tarefas e tarefas independentes definidas no Jira Software

Na Figura 5.4 é possível visualizar uma captura de ecrã da janela de detalhes de uma das tarefas registada no Jira, de notar que os campos apresentados e a organização dos mesmos foram da preferência do estudante, que realizou esta configuração a nível do projeto.



IPSCARPOOL-3 / IPSCARPOOL-33

**Create mobile app view for logged-in user's profile**

**Description**

Relevant fields:

- Location
- Profile image

**Attachments (1)**

Perfil.png (25 Aug 2020, 11:28 AM)

**Linked issues**

blocks

IPSCARPOOL-35 Create mobile app view... **DONE**

Add a comment... ES

Pro tip: press M to comment

**Done** ✓ Done

**Assignee** HenochV

**Reporter** Edgar Santos

**Development**  
1 branch  
3 commits 6 months ago

**Labels** RF3

**Components** Front-end Mobile

**Fix versions** Delivery 1

**Priority** Highest

**Automation** Rule executions

Figura 5.4 - Captura de ecrã dos detalhes de uma tarefa no Jira

Ainda a nível do quadro *Kanban* gerado, o estudante realizou algumas personalizações de modo a facilitar a visualização das tarefas por parte da equipa de desenvolvimento, tendo selecionado cuidadosamente os campos de maior importância a serem apresentados na visualização rápida dos cartões no quadro, atribuindo cores aos cartões de acordo com a sua prioridade e adicionado filtros rápidos que permitissem a visualização das tarefas pertencentes a cada componente. Em relação à estruturação do quadro *Kanban*, o estudante decidiu agrupar as tarefas por *user story* e criar quatro colunas para organizar as tarefas de acordo com o seu progresso, nomeadamente: pendente, em desenvolvimento, em fase de testes e concluído. Na Figura 5.5 é possível visualizar uma captura de ecrã do Jira, referente ao quadro *Kanban* do projeto durante o seu desenvolvimento.

Projects / CarPooling4IPS / CarPooling4IPS Board

## Kanban board

TO DO DEVELOPING TESTING DONE

IPSCARPOOL-1 UNIT TESTING 3 sub-tasks As a User, I want to use my IPS credentials, so I can authenticate myself in the app

- Create mobile app view for User login  
4 days, 5 hours, 42 minutes  
Front-end Mobile  
RF1  
IPSCARPOOL-31
- Create API route to authenticate the User against the IPS authentication system  
1 week, 2 hours, 57 minutes  
Back-end  
RF1  
IPSCARPOOL-32
- API should retrieve User data from IPS authentication system  
1 day  
Back-end  
RF2  
IPSCARPOOL-2

IPSCARPOOL-3 UNIT TESTING 4 sub-tasks As a User, I want to be able to view and edit my profile, so I can update my personal data

- Create mobile app view for logged-in user's profile  
1 week, 56 minutes  
Front-end Mobile  
RF3  
IPSCARPOOL-33
- Create API route to retrieve logged-in user profile data  
3 days, 2 hours, 40 minutes  
Back-end  
RF3  
IPSCARPOOL-34
- Create API route to edit logged-in user's profile data  
4 hours  
Back-end  
RF4  
IPSCARPOOL-36
- Create mobile app view for edit logged-in user's profile  
2 days, 1 hour, 40 minutes  
Front-end Mobile  
RF4  
IPSCARPOOL-35

IPSCARPOOL-4 UNIT TESTING 2 sub-tasks As a User, I want to manage my vehicles, in order to define relevant properties such as fuel type and consumption

**Quickstart**

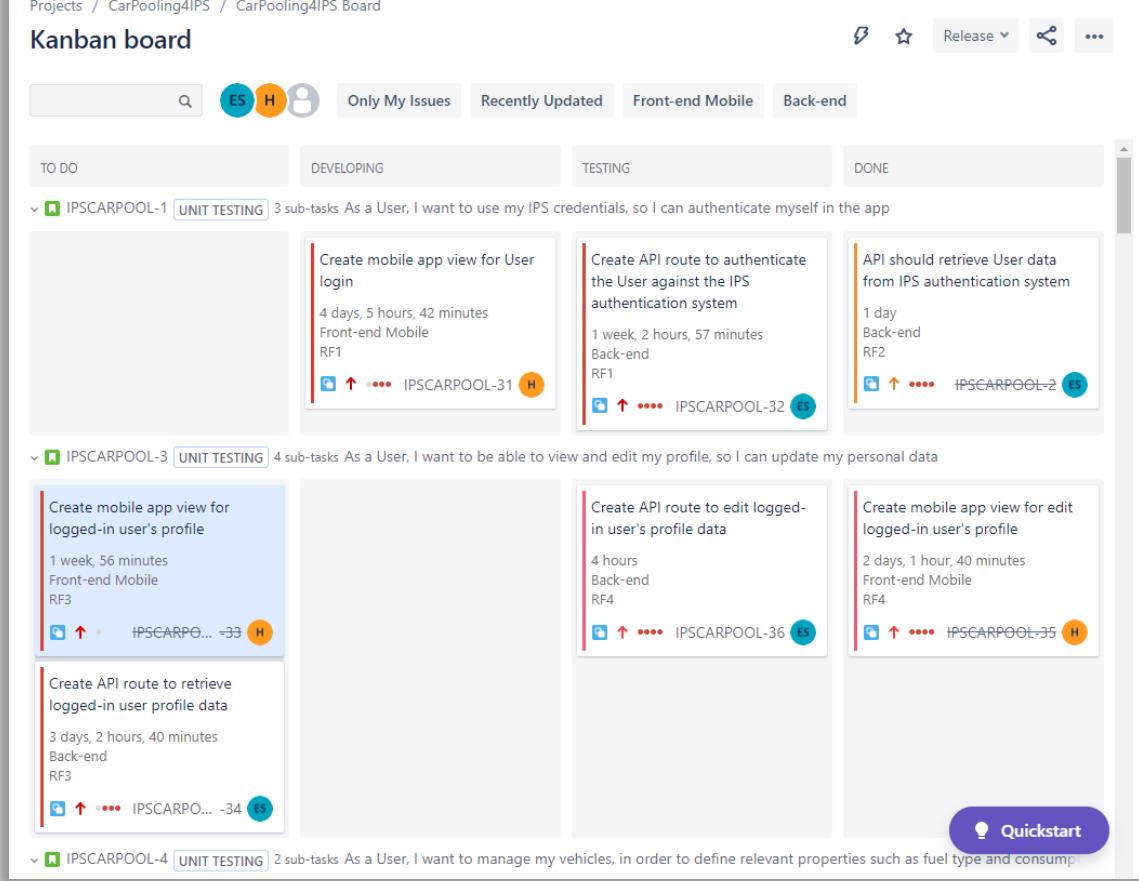


Figura 5.5 - Captura de ecrã do quadro Kanban no Jira

A nível de fluxo de trabalho, o estudante decidiu deixá-lo o mais simples possível, retirando as limitações das transições entre os vários estados de uma tarefa e adicionado dois novos estados possíveis para uma tarefa, nomeadamente a realização de testes unitários e de testes de integração. Na Figura 5.6 é possível visualizar o diagrama resultante do fluxo de trabalho definido pelo estudante no Jira.

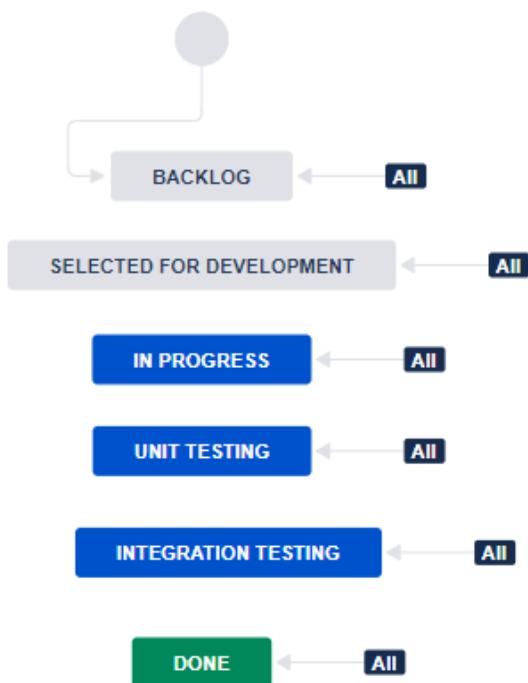


Figura 5.6 - Diagrama de fluxo de trabalho definido para o projeto no Jira

Quanto ao registo de horas de trabalho, o estudante considerou limitadora a funcionalidade de controlo de horas oferecida por padrão pelo Jira. Deste modo, e após a investigação de diversas alternativas gratuitas disponíveis, acabou-se por se decidir a utilização do Clockwork Automated Timesheets Free desenvolvido por HeroCoders. Este *addon*, apesar de limitado na versão gratuita, fornece uma vasta gama de funcionalidades úteis, tais como a possibilidade de utilização de um cronómetro que permite o registo de horas de trabalho numa dada tarefa de forma automatizada; a possibilidade de visualização das horas de trabalho efetuadas por membro da equipa na janela de detalhes de uma determinada tarefa; o registo automático das horas de trabalho com base nas transições de estado de uma tarefa; a disponibilização de um relatório dinâmico, personalizável e extensível do registo das horas de trabalho efetuadas. Na Figura 5.7 é possível visualizar uma captura de ecrã, retirada do Jira durante o desenvolvimento do projeto, de um relatório dinâmico de horas de trabalho semanais agrupadas por membro da equipa e em seguida por componente.

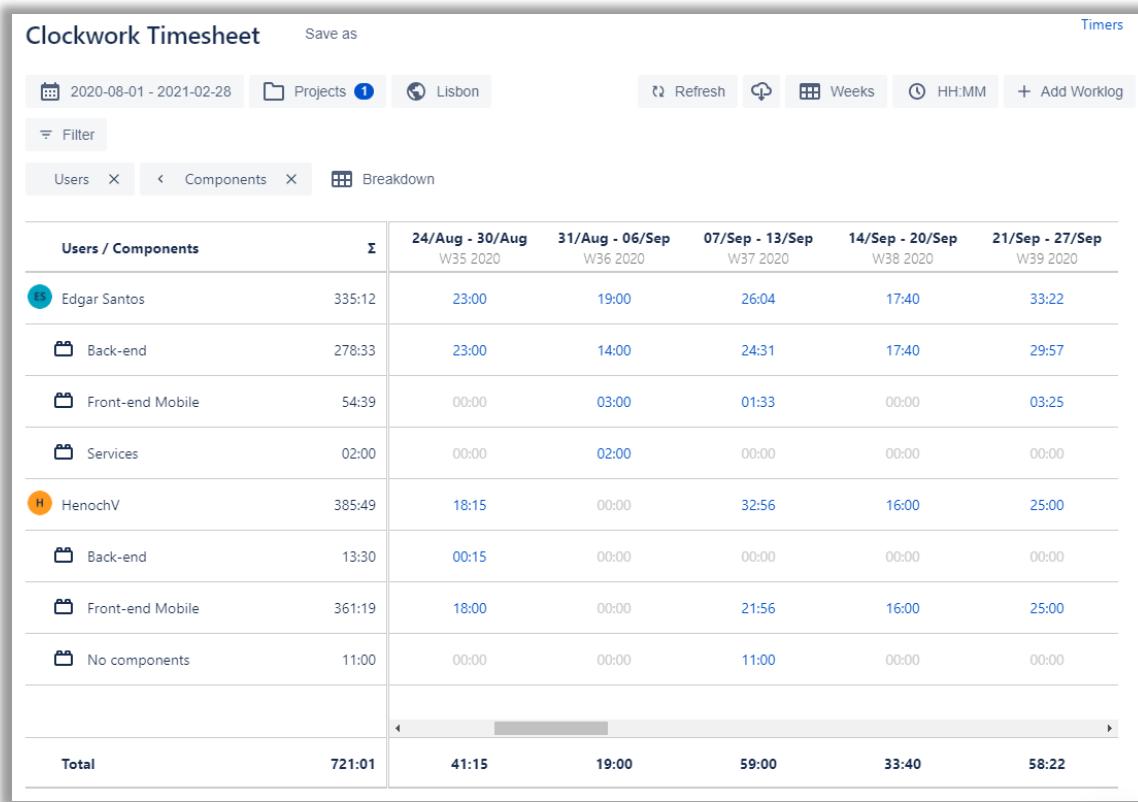


Figura 5.7 - Captura de ecrã do relatório dinâmico de horas de trabalho registadas no Jira

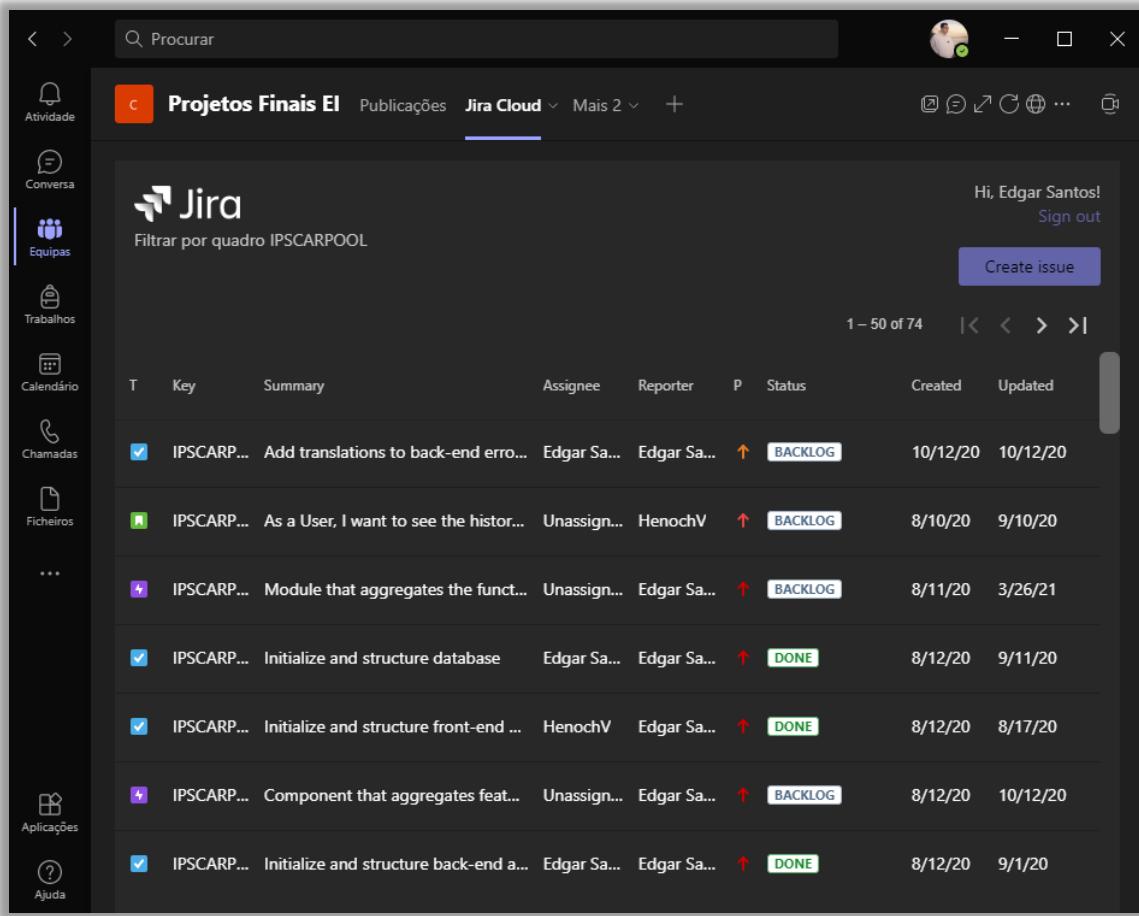
Com o propósito de tirar proveito das capacidades de automatização e integração disponibilizadas pelo Jira, o estudante decidiu realizar a integração com o GitHub, o que permitiu a visualização das *branches* e *commits* associados a cada tarefa através do Jira, tal como visível na parte central direita da Figura 5.4. Permitindo ainda a integração do Jira com o GitKraken, facilitando deste modo a associação das *branches* com as tarefas.

A nível de automatização, o estudante ativou algumas das regras disponibilizadas por padrão pelo Jira e criou outras que terá considerado relevantes, sendo as regras ativas as seguintes:

- Ao criar uma *branch* no GitHub, caso seja possível relacioná-la com uma tarefa registada no sistema, o estado desta será automaticamente atualizado como estando em progresso.
- Ao adicionar numa dada tarefa a indicação que esta duplica uma outra tarefa, a mesma será automaticamente marcada como concluída.
- Ao terminar todas as tarefas associadas a uma dada tarefa principal, o sistema irá automaticamente finalizar essa tarefa principal.
- Ao atualizar o estado de uma tarefa como estando finalizada, o sistema irá automaticamente gerar um comentário com esta indicação nas tarefas relacionadas.

- Ao adicionar um comentário numa tarefa, caso esse comentário possua uma ou mais referencias a outras tarefas registadas no sistema, as mesmas serão automaticamente transformadas em hiperligações para as respetivas tarefas.

Por fim, ainda a nível das integrações realizadas, o estudante considerou pertinente disponibilizar uma listagem das tarefas do Jira e respetivo progresso numa aba apropriada para o efeito na equipa do *Microsoft Teams*, equipa esta que foi criada pelos orientadores como meio de comunicação e partilha de informações prioritárias acerca do projeto. Na Figura 5.8 é possível visualizar uma captura de ecrã da janela do Microsoft Teams com a listagem das tarefas registadas no Jira durante o desenvolvimento do projeto.



T	Key	Summary	Assignee	Reporter	P	Status	Created	Updated
<input checked="" type="checkbox"/>	IPSCARP...	Add translations to back-end erro...	Edgar Sa...	Edgar Sa...	↑	BACKLOG	10/12/20	10/12/20
<input checked="" type="checkbox"/>	IPSCARP...	As a User, I want to see the histor...	Unassign...	HenochV	↑	BACKLOG	8/10/20	9/10/20
<input checked="" type="checkbox"/>	IPSCARP...	Module that aggregates the funct...	Unassign...	Edgar Sa...	↑	BACKLOG	8/11/20	3/26/21
<input checked="" type="checkbox"/>	IPSCARP...	Initialize and structure database	Edgar Sa...	Edgar Sa...	↑	DONE	8/12/20	9/11/20
<input checked="" type="checkbox"/>	IPSCARP...	Initialize and structure front-end ...	HenochV	Edgar Sa...	↑	DONE	8/12/20	8/17/20
<input checked="" type="checkbox"/>	IPSCARP...	Component that aggregates feat...	Unassign...	Edgar Sa...	↑	BACKLOG	8/12/20	10/12/20
<input checked="" type="checkbox"/>	IPSCARP...	Initialize and structure back-end a...	Edgar Sa...	Edgar Sa...	↑	DONE	8/12/20	9/1/20

Figura 5.8 - Captura de ecrã da integração do Jira Cloud com o Microsoft Teams

## 5.2. API RESTFUL

Neste subcapítulo é descrito em detalhe o trabalho realizado pelo estudante a nível da API RESTful, que essencialmente corresponde à maioria das suas tarefas.

Como resultado final da implementação da primeira versão da aplicação, o estudante procedeu com a geração da documentação da API, estando a mesma disponível através do seguinte endereço <https://carpooling4ips.github.io/documentation-back-end-api/>.

## 5.2.1. Preparação da Instância

A primeira responsabilidade do estudante em relação à API foi investigar e implementar uma estrutura de ficheiros adequada ao projeto a desenvolver e decidir os padrões de software a utilizar, tal como a definição dos ficheiros de arranque do Node.js e outros relevantes. Em relação ao padrão de software MVC escolhido, os motivos para a utilização do mesmo já foram enumerados no capítulo 3.2.1.

O estudante começou por realizar a inicialização de um repositório GIT e a criação do respetivo ficheiro *.gitignore*, adequado ao ambiente Node.js e aos IDE Visual Studio Code e WebStorm, tendo ainda em conta outros ficheiros que deveriam ser ignorados, tais como a pasta de *uploads* e de documentação gerada localmente. Em seguida procedeu-se à inicialização de um projeto em Node.js, tendo-se criado o ficheiro *package.json* onde terão sido realizadas algumas configurações iniciais, tal como a definição de scripts para inicializar a API, validar e aplicar automaticamente parte das normas de codificação definidas, gerar a documentação da API, iniciar o servidor de depuração do Node.js, executar testes unitários e gerar um relatório de *Coverage Tests* em relação aos testes unitários.

A nível de dependências do Node.js utilizadas, o estudante teve o cuidado de separar as dependências gerais das dependências utilizadas apenas durante o desenvolvimento, reduzindo deste modo a existência de código não utilizado no ambiente de produção. Por predefinição, o Node.js utiliza como gestor de dependências o NPM, no entanto, para este projeto o estudante preferiu a utilização do gestor de dependências YARN, devido à sua velocidade, capacidade de trabalhar sem uma conexão ativa à *internet* e principalmente devido à sua capacidade de garantir o funcionamento da instância desenvolvida entre múltiplos sistemas com as mesmas características.

Na Figura 5.9 é possível visualizar uma captura de ecrã da estrutura de ficheiros da API do projeto, sendo os ficheiros e diretórios de maior relevância os seguintes:

- /docs – Diretório onde são guardados os ficheiros HTML da documentação gerada de modo automático pela dependência *jsdoc*;
- /logs – Diretório onde são guardados os ficheiros de registo da API, sendo que na primeira fase do projeto apenas existe o ficheiro *access.log* que mantém um registo detalhado dos acessos à API por parte da aplicação cliente, tal como visível no exemplo da Figura 5.10;
- /src/models – Diretório que contém os ficheiros de código dos modelos de dados da API.
- /src/controllers – Diretório que contém os ficheiros de código da lógica de negócio da API, que por sua vez fazem uso dos ficheiros do diretório */src/models*;

- /src/routes – Diretório que contém os ficheiros de código das rotas da API, que por sua vez fazem uso dos ficheiros do diretório /src/controllers;
- /src/locales – Diretório que contém os ficheiros com os diversos textos do sistema traduzidos em diferentes idiomas, no entanto, apesar do projeto ter sido preparado para suportar múltiplos idiomas, esta funcionalidade não pertence à primeira fase de entrega do projeto, pelo qual as traduções não terão sido efetivamente realizadas;
- /src/utils – Diretório que contém ficheiros de código que agreguem um conjunto de funcionalidades úteis que podem ser reutilizadas noutras partes do sistema;
- /src/main.js – Ficheiro de código responsável por inicializar o servidor da API e respetivos serviços e dependências necessárias;
- /test – Diretório que contém os ficheiros referentes aos testes unitários;
- /uploads – Diretório onde são guardados os ficheiros carregados pela aplicação cliente, no caso, os ficheiros das imagens de perfil carregadas pelos utilizadores;
- .env – Ficheiro de variáveis de ambiente das configuração utilizadas pelo projeto, tais como as credenciais de acesso à base de dados e ao sistema de autenticação do IPS;
- .env.example – Ficheiro de exemplo das variáveis de ambiente das configurações utilizadas pelo projeto. Ao contrário do ficheiro .env, este ficheiro é incluído no repositório Git e não contém informações confidenciais, sendo que na Figura 5.11 é possível visualizar uma captura de ecrã deste ficheiro de exemplo;
- .eslintrc.json – Ficheiro de configurações da dependência *eslint*, responsável pela validação e formatação do código JS escrito nos restantes ficheiros;
- .mocharc.json – Ficheiro de configurações da dependência *mocha*, responsável pela execução dos testes unitários;
- jsdoc.config.json – Ficheiro de configurações da dependência *jsdoc*, responsável pela geração da documentação do projeto;
- index.js – Ficheiro de código principal do projeto, responsável por inicializar a conexão com a base de dados e executar o ficheiro /src/index.js.

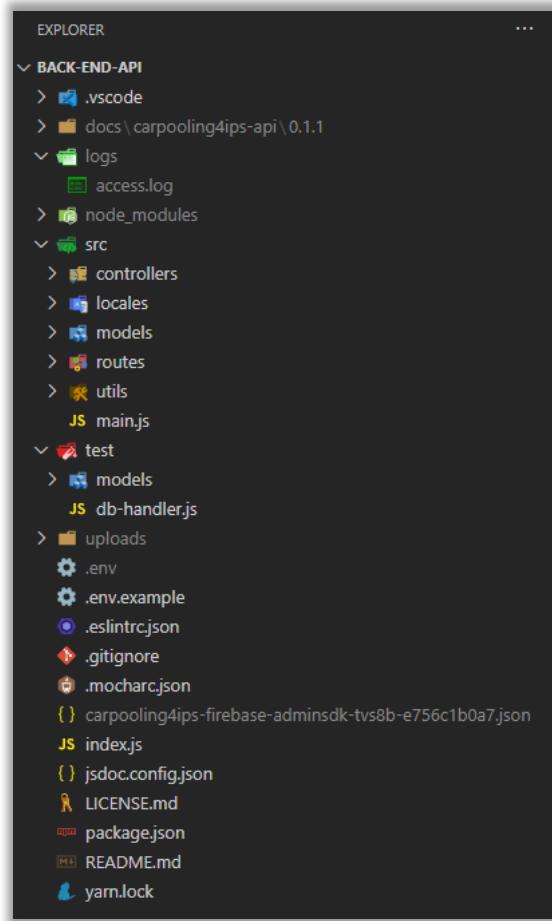


Figura 5.9 - Captura de ecrã da estrutura de ficheiros da API

```
logs > access.log
1 ::ffff:127.0.0.1 - - [09/Feb/2021:17:44:00 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 401 - "-" "Dart/2.10 (dart:io)"
2 ::ffff:127.0.0.1 - - [09/Feb/2021:17:44:00 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 401 - "-" "Dart/2.10 (dart:io)"
3 ::ffff:127.0.0.1 - - [09/Feb/2021:17:44:47 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 401 - "-" "Dart/2.10 (dart:io)"
4 ::ffff:127.0.0.1 - - [09/Feb/2021:17:44:59 +0000] "POST /api/auth/login HTTP/1.1" 200 282 "-" "Dart/2.10 (dart:io)"
5 ::ffff:127.0.0.1 - - [09/Feb/2021:17:45:00 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 204 - "-" "Dart/2.10 (dart:io)"
6 ::ffff:127.0.0.1 - - [09/Feb/2021:17:45:01 +0000] "GET /api/user/me/rideSchedule/next?mode=driver HTTP/1.1" 200 1983 "-" "Dart/2.10 (dart:io)"
7 ::ffff:127.0.0.1 - - [09/Feb/2021:17:45:43 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 204 - "-" "Dart/2.10 (dart:io)"
8 ::ffff:127.0.0.1 - - [09/Feb/2021:17:45:43 +0000] "POST /api/auth/login HTTP/1.1" 200 282 "-" "Dart/2.10 (dart:io)"
9 ::ffff:127.0.0.1 - - [09/Feb/2021:17:45:44 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 204 - "-" "Dart/2.10 (dart:io)"
10 ::ffff:127.0.0.1 - - [09/Feb/2021:17:45:44 +0000] "GET /api/user/me/rideSchedule/next?mode=driver HTTP/1.1" 200 1983 "-" "Dart/2.10 (dart:io)"
11 ::ffff:127.0.0.1 - - [09/Feb/2021:17:58:59 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 204 - "-" "Dart/2.10 (dart:io)"
12 ::ffff:127.0.0.1 - - [09/Feb/2021:17:58:59 +0000] "POST /api/auth/login HTTP/1.1" 200 282 "-" "Dart/2.10 (dart:io)"
13 ::ffff:127.0.0.1 - - [09/Feb/2021:17:59:00 +0000] "PUT /api/user/me/settings/fcmtoken HTTP/1.1" 204 - "-" "Dart/2.10 (dart:io)"
14 ::ffff:127.0.0.1 - - [09/Feb/2021:17:59:00 +0000] "GET /api/user/me/rideSchedule/next?mode=driver HTTP/1.1" 200 1983 "-" "Dart/2.10 (dart:io)"
15
```

Figura 5.10 - Captura de ecrã de um exemplo do ficheiro access.log da API

```

.env.example
  You, 3 months ago | 1 author (You)
  1 # .env.example, commit to repo
  2
  3 # Environment
  4 NODE_ENV=production
  5 DEBUG=carpooling4ips,carpooling4ips:database,carpooling4ips:api,i18n:warn,i18n:error
  6
  7 # Internationalization
  8 SUPPORTED_LOCALES=en,pt
  9 DEFAULT_LOCALE=en
 10
 11 # API
 12 API_HOSTNAME=127.0.0.1
 13 API_PORT=2450
 14
 15 # Database
 16 MONGODB_URI=
 17
 18 # Logs
 19 LOGS_FOLDER_PATH=./logs
 20
 21 # Uploads
 22 UPLOADS_FOLDER_PATH=./uploads
 23
 24 # Authentication
 25 AUTH_SECRET=
 26 AUTH_ALGORITHM=HS256
 27 AUTH_ISSUER=api.carpooling4ips.ips.pt
 28 AUTH_EXPIRATION=4h
 29 AUTH_MICROSOFT_GRAPH_API_URI=https://graph.microsoft.com/v1.0
 30
 31 # Ride Group PReferences
 32 # Notes:
 33 #   RIDE_GROUP_FIXED_POINT=latitude,longitude
 34 RIDE_GROUP_MAX_DISTANCE="P1W"
 35 RIDE_GROUP_OVERLAP_THRESHOLD="P4H"
 36 RIDE_GROUP_RECURRENT_EXPIRATION="P2W"
 37 RIDE_GROUP_RECURRENT_EXPIRATION_NOTIFY="P2D"
 38 RIDE_GROUP_FIXED_POINT=0,0
 39 RIDE_ROUTE_MATCH_MAX_DISTANCE_METERS=50
 40 RIDE_SCHEDULE_NEXT_MAX_DATE_VARIATION="PT15M"
 41
 42 # Google Maps API
 43 GOOGLE_MAPS_API_KEY=
 44
 45 # Google Firebase
 46 # Notes:
 47 #   Environment variable GOOGLE_APPLICATION_CREDENTIALS name can not be changed
 48 GOOGLE_APPLICATION_CREDENTIALS=

```

Figura 5.11 - Captura de ecrã do ficheiro `.env.example` da API

## 5.2.2. Base de Dados

A nível de base de dados, pelos motivos já referidos no subcapítulo 3.1.3, optou-se por uma solução não relacional, mais especificamente pelo MongoDB. Existe um vasto número de possibilidades a nível de interligação entre o MongoDB e o Node.js, desde simples bibliotecas que permitem o acesso nativo aos dados, a bibliotecas mais complexas que permitem a abstração do sistema de base de dados utilizado e a simplicidade da integração com a estrutura de dados nativa da linguagem de programação em uso, também conhecidas por ODM, ou ORM no caso de bases de dados relacionais. No caso deste projeto, optou-se pelo uso de um ODM para Node.js dedicado ao MongoDB, a biblioteca *mongoose*, que para além dos benefícios

supramencionados, permite, entre outros, a validação de dados, a adição de lógica de negócio diretamente relacionada com os dados, a possibilidade de adicionar relações aos dados entre coleções e a capacidade de efetuar transações mantendo a integridade dos dados. De modo geral, fornece um vasto conjunto de funcionalidades comumente presentes apenas em bases de dados relacionais [37].

Com base no modelo de dados apresentado no subcapítulo 4.9.1, o estudante ficou encarregue de desenvolver os ficheiros de código dos esquemas e dos modelos de dados de acordo com as normas do *mongoose*, tendo o cuidado de definir as validações, índices e relações adequadas.

Sendo o modelo de dados o foco numa API baseada em operações CRUD, o estudante considerou de extrema importância a realização de documentação detalhada e, principalmente, a realização de testes unitários às validações e estrutura do mesmo, de modo a garantir a correta persistência dos diversos dados. No Excerto de Código 5.1 é possível visualizar um exemplo de implementação de um dos testes unitários realizados.

```
describe('models/notification', function () {
  it('should be invalid if missing required fields', function (done) {
    const document = new NotificationModel()

    document.validate(function (err) {
      // Required fields
      expect(err.errors.key).to.exist
      expect(err.errors.title).to.exist
      expect(err.errors.body).to.exist

      done()
    })
  })

  it('should be invalid if incorrect data types', function (done) {
    const document = new NotificationModel({
      key: 'NEW_TEST',
      title: 'Test notification',
      body: 'This is a new notification test.'
    })

    document.validate(function (err) {
      expect(err).to.be.null

      done()
    })
  })
})
```

Excerto de Código 5.1 - Teste unitário do modelo de notificações da API

No entanto, devido a limitações técnicas, alguns destes testes não foram possíveis de ser completados, pois por predefinição o *mongoose* não é capaz de realizar a validação de dados que possuam relações entre modelos sem possuir uma conexão ativa a uma base de dados MongoDB. Dentro das diversas soluções disponíveis para esta questão, o estudante considerou a utilização de uma base de dados simulada em memória RAM como sendo a mais adequada devido à sua rapidez e persistência dos dados apenas durante a execução dos testes. No entanto, à data de realização desta tarefa as bibliotecas disponíveis para este efeito não eram compatíveis com a versão de Node.js em uso. Por recomendação dos orientadores, preferiu-se avançar com a realização das restantes tarefas e deixar esta questão para outra fase de desenvolvimento.

### 5.2.3. Autenticação

O estudante ficou responsável pela integração do sistema de autenticação do IPS com a API da plataforma a desenvolver, garantindo deste modo que os utilizadores desta plataforma serão membros da comunidade IPS, aumentando assim a confiança dos utilizadores em relação à sua segurança. As tarefas inerentes a esta responsabilidade abrangem os requisitos funcionais 1 e 2.

O IPS utiliza como sistema de autenticação para os membros da sua comunidade a plataforma de identidade universal *Azure Active Directory*, da Microsoft. Com o propósito de realizar a integração supramencionada e devido à falta de informação útil transmitida por parte da Divisão Informática do IPS, o estudante viu-se na necessidade de investigar sobre o funcionamento e integração do AAD com aplicações de terceiros. Esta investigação foi dispendiosa a nível de recursos temporais, devido ao vasto número de artigos relacionados com o tema presentes na plataforma de documentação da Microsoft [38], sendo que parte dos artigos encontram-se desatualizados, existindo incoerências entre eles.

O AAD fornece uma vasta gama de serviços de identidade distintos, entre eles o *Azure AD* e o *Azure AD B2C*. Após a realização de múltiplos testes por parte do estudante e do seu colega de equipa, chegou-se à conclusão de que para o propósito deste projeto dever-se-ia utilizar o serviço *Azure AD* com recurso ao diretório de utilizador do IPS. De modo a deixar claro o motivo pelo qual o serviço *Azure AD B2C* não seria aplicável neste projeto, é devido ao propósito deste serviço disponibilizar um novo diretório de utilizadores para uma organização, permitindo o registo de novos utilizadores com endereços de correio eletrónico de diferentes domínios [39].

Deste modo, o estudante começou por realizar o registo da aplicação no portal do Azure, utilizando para este fim as suas credenciais de acesso ao sistema de informação do IPS. Para o registo da aplicação foi necessário indicar o nome da mesma e o tipo de contas suportadas. No caso seria uma aplicação *single-tenant*, ou seja, uma aplicação que permite exclusivamente a autenticação de utilizadores pertencentes ao diretório de utilizadores do IPS. Após o registo da

aplicação foi necessário proceder com as devidas configurações do serviço, nomeadamente conceder as permissões necessárias para aceder aos dados básicos dos utilizadores e adicionar a plataforma Android, realizando o devido registo do seu endereço de redireccionamento. Devido, à data deste relatório, os recursos necessários para o desenvolvimento de aplicações moveis para o sistema operativo iOS não terem chegado a ser disponibilizados, não foi possível realizar o registo da plataforma iOS no serviço *Azure AD*.

Do lado da API, o estudante começou por realizar a instalação e configuração da biblioteca *passport* para Node.js, sendo esta biblioteca um *middleware* de autenticação compatível com a biblioteca Express.js utilizada, ou seja, permite a integração de um ou mais sistemas de autenticação com uma aplicação baseada em Express.js, de forma simples, flexível e modular. Cada sistema de autenticação suportado por esta biblioteca é chamado de estratégia, sendo que à data deste relatório, existem cerca de quinhentas estratégias públicas disponíveis para utilização. No âmbito deste projeto foi utilizada apenas uma estratégia, a *passport-jwt*, que permite a autenticação de utilizadores utilizando *JSON Web Token*.

Com recurso à biblioteca *passport* e *passport-jwt* foi possível controlar quais as rotas da API que necessitariam da autenticação do utilizador para serem acedidas, recorrendo simplesmente à adição do seguinte trecho de código à declaração de cada rota que se pretenda proteger: “`passport.authenticate('jwt',{session:false})`”.

Por fim, o estudante criou o endpoint `POST /auth/login` na API, com o propósito de autenticar os utilizadores da plataforma. Esta rota espera receber como parâmetro uma *access token* para o *Microsoft Graph API*, a obtenção desta *token* é da responsabilidade da aplicação cliente. Utilizando esta *token* a API realiza um pedido à *Microsoft Graph API*, caso a *access token* não seja valida a autenticação falha, caso contrário o servidor recebe a resposta da *Microsoft Graph API* contendo os dados pessoais do utilizador a autenticar, provenientes do diretório de utilizadores do IPS. Com recurso a estes dados, a API verifica se o utilizador existe na base de dados local e, caso não exista, será adicionado um novo registo à base de dados local com os dados pessoais de maior relevância do utilizador, tais como o seu nome e email organizacional. Por fim, a API gera um JWT assinado contendo a identificação do utilizador, retornando-o para a aplicação cliente.

Passando o JWT recebido no cabeçalho *Authorization* do pedido HTTP realizado à API, a aplicação cliente poderá realizar pedidos autenticados e aceder a dados protegidos. De notar que cada JWT gerado tem uma validade máxima de quatro horas e após este período será necessário a aplicação cliente realizar novamente o processo de autenticação.

## 5.2.4. Notificações

O estudante foi também o responsável pela implementação do envio de notificações *push* através da API para os dispositivos móveis. Referente aos requisitos funcionais 19 e 75.

Para este fim, começou por se registrar uma conta Google para o projeto, de modo a poder-se utilizar o serviço Firebase fornecido pela empresa e posteriormente o serviço Google Maps. Já na consola do Firebase, nas configurações do projeto, gerou-se uma chave privada de administração para o SDK do Firebase, chave esta que será utilizada pela API para comunicação entre os serviços.

A nível da API foi necessário instalar a dependência *firebase-admin* de modo a ter-se acesso aos serviços administrativos do Firebase. Dado que as notificações podem ser disparadas em diversas situações e que geralmente dependem da chamada de certos *endpoints* da API por parte dos utilizadores, o estudante decidiu criar uma classe responsável pela gestão das notificações, disponibilizando um método público para o envio de uma determinada notificação para um dado utilizador. Analisando cuidadosamente o modelo de dados disponibilizado no subcapítulo 4.9.2 é possível verificar a existência de duas coleções relacionadas com as notificações, sendo o propósito da coleção *Notification* manter uma listagem imutável das diversas notificações existentes na plataforma, enquanto a coleção *UserNotification* serve como um histórico de notificações enviadas para cada utilizador.

De modo a simplificar a implementação das notificações do lado da aplicação móvel e agilizar o desenvolvimento, o estudante teve preferência por deixar a responsabilidade das traduções das notificações do lado da API, para este fim e devido ainda à necessidade de persistir na base de dados a *token* única do FCM dos dispositivos moveis clientes associados a cada utilizador, foram desenvolvidos os seguintes *endpoints*:

- *PUT /user/me /settings/locale* – Permite definir a linguagem de preferência do utilizador autenticado.
- *PUT /user/me /settings/fcmtoken* – Permite definir na base de dados a *token* única do FCM do dispositivo movel cliente do utilizador autenticado. No caso do *token* passado em argumento já estar associado a outro utilizador, o mesmo será desassociado desse utilizador e reassociado ao novo utilizador.

O método público de envio de notificações pertencente à classe criada, começa por verificar se o identificador único do utilizador passado existe na base de dados e se o mesmo possui uma *token* do FCM associada, em seguida com recurso à listagem de notificações existentes na base de dados e através da chave única de notificação passada em argumento, obtém o tipo de notificação e o texto da notificação no idioma de preferência do utilizador. Após este processo, o sistema realiza a substituição dos *placeholders* existentes no texto da notificação pelos parâmetros passados em argumento, contextualizando deste modo a notificação a ser enviada. Por fim, o sistema envia a notificação para o serviço FCM e guarda em histórico na base de dados a notificação enviada e respetivos dados, tal como o identificador único gerado pelo serviço para referência futura.

O diagrama presente na Figura 5.12 demonstra o fluxo de uma notificação ao ser enviada pela API (App Server) para o serviço FCM, que em seguida envia a notificação para o dispositivo móvel de destino. No caso de o dispositivo de destino utilizar o sistema operativo iOS em vez de Android é necessário a notificação ser enviada pelo APNs em vez do FCM, no entanto, é possível configurar o FCM para utilizar automaticamente o APNs quando necessário, abstraindo deste modo o desenvolvedor desta questão e evitando a necessidade de implementação do suporte ao APNs na API.

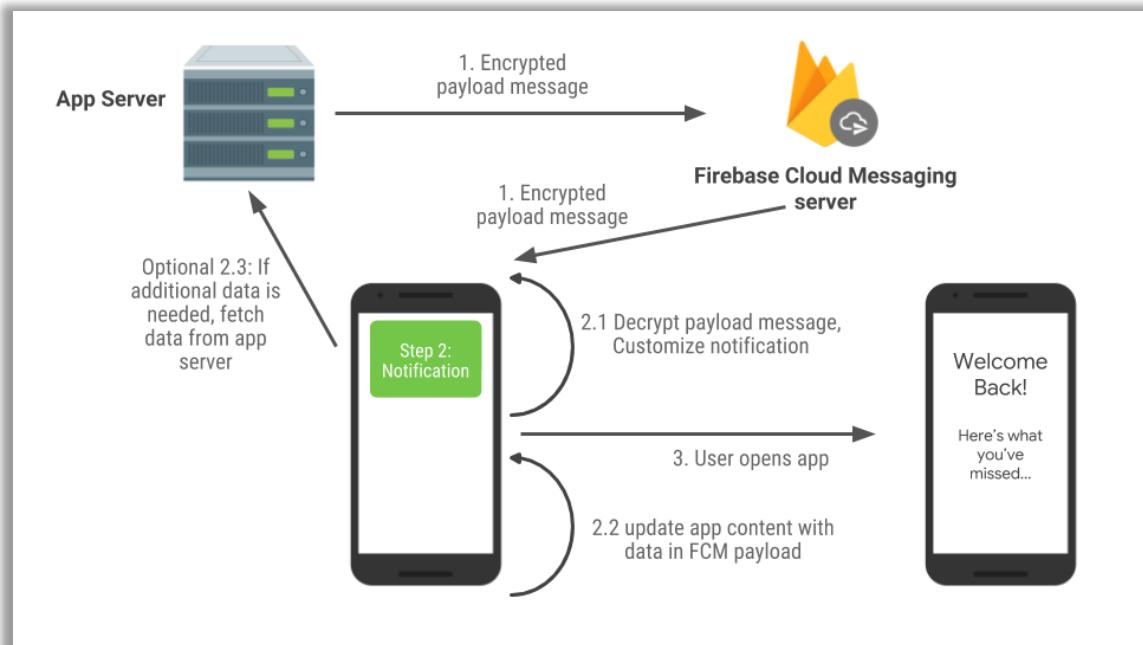


Figura 5.12 - Diagrama de fluxo de trabalho das notificações push com recurso ao FCM (retirado de <https://firebase.googleblog.com/2018/09/handle-fcm-messages-on-android.html>)

## 5.2.5. Perfil de Utilizador

O estudante ficou responsável pelo desenvolvimento das rotas da API relacionados com o perfil de utilizador, referentes aos requisitos funcionais 3, 4 e 9.

Para este fim foram desenvolvidos os seguintes *endpoints*:

- *GET /user/:userId/profile* – Permite a consulta dos dados de perfil do utilizador passado no argumento *:userId*, que poderá ser o ID de um utilizador em específico ou a palavra “me” que referencia o utilizador autenticado.

Recorrendo ao cruzamento de múltiplos dados presentes na base de dados local e a um conjunto de cálculos simplistas, esta rota retorna os seguintes dados: primeiro e último nome; título dentro da organização; imagem de perfil; classificação de 1 a 5; número de avaliações recebidas; número de boleias oferecidas; número de boleias recebidas; número total de conquistas obtidas; comentários deixados no perfil.

- *PUT /user/me/profile/avatar* – Permite ao utilizador autenticado realizar o carregamento de uma nova imagem para o seu perfil de utilizador. De notar que ao carregar uma nova imagem, a imagem anterior será automaticamente removida de modo a reduzir a cota de armazenamento utilizada pela API.

Por predefinição, a biblioteca Express.js, utilizada pela API para a receção de pedidos HTTP, não suporta a receção de formulários codificados em *multipart/form-data*, comumente utilizado para o carregamento de ficheiros através do protocolo HTTP. Deste modo, recorreu-se à utilização da biblioteca *multer*, um *middleware* compatível com Express.js que possibilita, de forma simplista, a implementação da capacidade de receber ficheiros através de um pedido HTTP e guardá-los localmente, sendo possível manter o nome e outros meta-dados pertencentes ao ficheiro de origem.

## 5.2.6. Bloqueio de Utilizadores

O estudante ficou responsável pelo desenvolvimento das rotas da API relacionados com o bloqueio de utilizadores, ou tal como foi chamado durante o desenvolvimento, a lista negra de utilizadores. As tarefas inerentes a esta responsabilidade são referentes aos requisitos funcionais 6 e 7.

Para este fim foram desenvolvidos os seguintes *endpoints*:

- *GET /user/me/userBlacklist* – Permite ao utilizador autenticado consultar os utilizadores adicionados à sua lista negra e a respetiva data de bloqueio.
- *POST /user/me/userBlacklist* – Permite ao utilizador autenticado adicionar um novo utilizador, através do seu ID, à sua lista negra de utilizadores, impedindo deste modo contactos futuros provenientes deste utilizador e removendo o mesmo de listagens e pesquisas futuras.
- *DELETE /user/me/userBlacklist* – Permite ao utilizador autenticado remover um utilizador, através do seu ID, da sua lista negra de utilizadores, restaurando assim a possibilidade de contacto com este utilizador e permitindo que o mesmo volte a ser apresentado em listagens e pesquisas futuras.

## 5.2.7. Grupos de Boleia

O estudante ficou responsável pelo desenvolvimento das rotas da API relacionadas com a gestão dos grupos de boleia e dos agendamentos de boleias. As tarefas inerentes a esta responsabilidade são referentes aos requisitos funcionais 15, 16, 17, 18 e 29.

Para este fim foram desenvolvidos os seguintes *endpoints* para gestão dos grupos de boleia e respetivos dados associados:

- *GET /user/me/rideGroup* – Permite ao utilizador autenticado consultar os grupos de boleia a que pertence e respetivos detalhes, tais como veículo, passageiros e rota recomendada. Retorna os grupos de boleia criados pelo utilizador ou aqueles nos quais este é passageiro, dependendo do modo passado, motorista ou passageiro, respetivamente.
- *GET /user/me/rideGroup/:groupId* – Permite ao utilizador autenticado consultar os detalhes do grupo de boleia passado no argumento *:groupId* e respetivos agendamentos. Retorna ainda o número de boleias oferecidas pelo motorista até ao momento e a média da classificação, numa escala de 0 a 5, do mesmo.
- *POST /user/me/rideGroup* – Permite ao utilizador autenticado criar um novo grupo de boleia como motorista. Este *endpoint* recebe como parâmetros de entrada o nome do grupo, o veículo a utilizar, a lista de passageiros permanentes a serem associados às boleias agendadas do grupo, as coordenadas do ponto de encontro, se é uma viagem de ida ou de volta para o IPS, se deverão ser evitadas portagens, a data e hora de partida ou, caso seja um grupo recorrente, a hora de partida e os respetivos dias da semana da boleia.

O processo de criação de um grupo de boleia é complexo e consiste em vários passos, por este motivo o estudante considerou pertinente a utilização da capacidade de transações da biblioteca *mongoose*, garantindo deste modo a consistência dos dados. O sistema começa por criar o grupo de boleia tendo em conta os parâmetros recebidos, em seguida avança com o cálculo da rota da viagem e por fim gera os respetivos agendamentos das boleias, ou agendamento no caso de não ser recorrente. De notar que o sistema não permite a criação de um grupo de boleia não recorrente com uma data posterior a uma semana da data atual, esta limitação foi proposta pelos orientadores do projeto com o propósito de evitar erros de introdução por parte dos utilizadores e reduzir a existência de previsões de boleias que poderiam acabar por não se realizar.

A nível do cálculo da rota da viagem utilizou-se o serviço *Google Directions API* devido à alta qualidade dos seus dados, pelo menos relativamente a Portugal. Terão sido consideradas outras opções, como o *HERE* ou até mesmo dentro das opções gratuitas o *Openrouteservice*, no entanto, seja a nível de preços, a nível de complexidade de implementação ou a nível de qualidade dos dados, acabaram por ser descartadas. Devido aos custos destes serviços serem com base no número de pedidos de cálculo de rotas realizados, tal como indicado no subcapítulo **Error! Reference source not found.**, e devido a esta plataforma necessitar de calcular as mesmas rotas mais do que uma vez, o estudante decidiu, respeitando os termos de serviço da Google, guardar localmente na

base de dados as rotas calculadas e ao receber um novo pedido de cálculo de uma rota verificar se a mesma foi calculada anteriormente e caso afirmativo retornar os dados já guardados.

- *PUT /user/me/rideGroup/:groupId* – Permite ao utilizador autenticado modificar os detalhes do grupo de boleia passado no argumento *:groupId*, desde que o utilizador seja o motorista desse grupo. Devido à complexidade dos grupos de boleia, decidiu-se, pelo menos nesta primeira versão da plataforma, permitir apenas a modificação do nome do grupo.
- *DELETE /user/me/rideGroup/:groupId* – Permite ao utilizador autenticado eliminar o grupo de boleia passado no argumento *:groupId* e respetivos agendamentos futuros, desde que o utilizador seja o motorista desse grupo. A nível interno e por uma questão de histórico, os dados não são removidos da base de dados e sim marcados como desativados. Por fim, este *endpoint* notifica, tanto os passageiros permanentes do grupo como os temporários pertencentes às respetivas boleias, que o grupo de boleia foi removido e que qualquer boleia subsequente ao mesmo terá sido cancelada.
- *POST /user/me/rideGroup/:groupId/passenger* – Permite ao motorista do grupo de boleia passado no argumento *:groupId* adicionar um novo utilizador à lista de passageiros permanentes do grupo, desde que haja lugares disponíveis. Por fim, notifica o utilizador que foi adicionado como um passageiro ao respetivo grupo de boleia.
- *DELETE /user/me/rideGroup/:groupId/passenger/:passengerId* – Permite ao motorista do grupo de boleia passado no argumento *:groupId* remover o passageiro passado no argumento *:passengerId* da listagem de passageiros permanentes do grupo, notificando o mesmo que já não pertence a este grupo.
- *POST /user/me/rideGroup/:groupId/renew* – Permite ao motorista do grupo de boleia passado no argumento *:groupId* realizar a renovação do grupo, gerando deste modo os respetivos agendamentos de boleias para os sete dias subsequentes e atualizando a data de expiração do grupo. No entanto, a nível interno são gerados os agendamentos de boleias até duas semanas posteriores à data atual, de modo a evitar que sejam criados novos grupos de boleia nesse período que possam ter datas sobrepostas. Dada a complexidade da manipulação de datas e à preocupação com a mudança de horário de verão, o estudante decidiu utilizar a biblioteca *luxon* de modo a simplificar este processo.

Dois dias antes da data de expiração de um grupo de boleia, o respetivo motorista receberá uma notificação com a indicação para renovar o grupo caso o mesmo continue válido. Este sistema foi proposto pelos orientadores do projeto com o propósito de reduzir a existência de grupos de boleia inválidos, seja por o

utilizador ter-se esquecido de remover o mesmo ou por ter deixado de utilizar a plataforma.

Da parte de gestão dos agendamentos, foram desenvolvidos os seguintes *endpoints*:

- *GET /user/me/rideSchedule* – Permite ao utilizador autenticado consultar os seus agendamentos de boleias para um dado mês ou semana de um dado ano. Dependendo do modo passado, motorista ou passageiro, são retornadas as respetivas boleias formatadas de modo apropriado a possibilitar a visualização num calendário.
- *GET /user/me/rideSchedule/search* – Permite ao utilizador autenticado pesquisar por agendamentos de boleias que vão de acordo com as suas necessidades. Os parâmetros de pesquisa a receber são a data e hora pretendida, a tolerância em minutos em relação ao horário especificado, se procura uma boleia de ida ou de volta do IPS, as coordenadas onde se pretende encontrar com o motorista e o raio em quilómetros de tolerância em relação às coordenadas especificadas. Agendamentos cujo grupo de boleia esteja expirado são automaticamente descartados, assim como boleias sem vagas disponíveis. Por fim, são retornados os agendamentos que vão ao encontro dos parâmetros de pesquisa ordenados por distância crescente, caso a distância seja a mesma a ordenação realiza-se por duração da viagem.
- *GET /user/me/rideSchedule/next* – Permite ao utilizador obter os detalhes do seu próximo agendamento de boleia, tendo em consideração o modo passado ser motorista ou passageiro. O sistema considera como próximo agendamento qualquer boleia cujo seu estado seja “iniciada” ou “à espera de passageiros”, no caso de não haver nenhuma boleia em progresso o sistema retorna o primeiro agendamento que tenha início nos próximos quinze minutos, caso contrário, o sistema retorna à indicação que não há agendamentos futuros dentro dos critérios supramencionados.
- *GET /user/me/rideSchedule/:scheduleId* – Permite ao utilizador autenticado consultar os detalhes do agendamento de boleia passado no argumento *:scheduleId*. Retorna ainda a média da classificação, numa escala de 0 a 5, do motorista.
- *DELETE /user/me/rideSchedule/:scheduleId* – Permite ao utilizador autenticado eliminar o agendamento de boleia passado no argumento *:scheduleId*, desde que o utilizador seja o motorista desse agendamento. A nível interno e por uma questão de histórico, os dados não são removidos da base de dados e sim marcados como desativados. Por fim, este *endpoint* notifica, tanto os

passageiros permanentes do grupo como os temporários pertencentes a este agendamento de boleia, que o agendamento foi cancelado.

- *POST /user/me/rideSchedule/:scheduleId/confirmAttendance* – Permite ao passageiro confirmar a sua presença no agendamento de boleia passado no argumento *:scheduleId*, desde que o mesmo pertença à lista de passageiros deste agendamento ou ao respetivo grupo de boleia. Por fim, este *endpoint* notifica o motorista que o passageiro em questão confirmou a sua presença.

No momento este sistema de confirmação de presenças não é utilizado pela plataforma a nível funcional, sendo apenas uma indicação de caráter informativo para o motorista. No entanto, pretende-se no futuro que a plataforma utilize esta informação para o cálculo das despesas da boleia, pois um passageiro que tenha confirmado a sua presença e acabe por não comparecer terá na mesma de cobrir as respetivas despesas.

- *PUT /user/me/rideSchedule/:scheduleId/status* – Permite ao motorista modificar o estado do agendamento de boleia passado no argumento *:scheduleId* de acordo com o diagrama de estados representado na Figura 4.10. Dependendo do novo estado do agendamento o sistema realiza a devida gestão dos passageiros e respetivo histórico de boleias, adicionalmente o sistema notifica os passageiros do novo estado.

Adicionalmente, com o propósito de possibilitar a pesquisa de utilizadores registados na plataforma de modo a permitir a adição manual de passageiros a um grupo de boleia, ou a um agendamento em particular, foi desenvolvido o *endpoint GET /user/me/search* que retorna uma listagem dos utilizadores que tenham no seu nome, título ou email a palavra a pesquisar. De notar, que esta rota tem em consideração o utilizador autenticado que realizou o pedido e deste modo evita o retorno de utilizadores que, apesar de cumprirem os requisitos da pesquisa, estão presentes na lista negra do utilizador.

Dada a extensividade deste conjunto de tarefas, e o inter-relacionamento entre os *endpoints* enumerados, o estudante considerou pertinente a criação de uma classe responsável pela criação dos grupos de boleia, respetivos cálculos de rotas e gestão dos agendamentos associados aos mesmos. Para além da capacidade, com recurso à dependência *node-cron*, de notificar automaticamente o motorista para renovar os grupos de boleia prestes a expirar. O estudante teve ainda o cuidado de preparar a classe de modo a suportar e tirar partido da funcionalidade de transações da biblioteca *mongoose*.

## 5.3. APLICAÇÃO MÓVEL

Neste subcapítulo é descrito em detalhe o trabalho realizado pelo estudante a nível da aplicação móvel.

Como resultado final da implementação da primeira versão da aplicação, o estudante avançou com a geração de um manual de utilizador da aplicação, estando o mesmo disponível para consulta no Anexo 5 – Manual de Utilizador da Aplicação Móvel.

### 5.3.1. Notificações

O estudante ficou responsável pela preparação da aplicação móvel para receber notificações *push*. As tarefas inerentes a esta responsabilidade são parte do requisito funcional 75.

Esta tarefa implicou que o estudante investigasse a forma mais adequada de implementar notificações *push* originárias do FCM numa aplicação móvel em Flutter. Em adição a este desafio levantou-se ainda a questão se seria pertinente e adequado a implementação desta tarefa recorrendo ao padrão BLoC. O estudante, após realizar uma série de pesquisas em relação a este tema, apesar de no momento de tais pesquisas existir informação limitada sobre este tópico, em conjunto com o seu colega equipa que teria maior experiência com Flutter, tomou a decisão de prosseguir com a implementação das notificações *push* respeitando o padrão BLoC.

O estudante avançou com a criação dos eventos e estados adequados ao sistema de notificações, que essencialmente corresponderiam a um evento de receção de uma notificação, a um evento de erro de notificação, a um estado de arranque do sistema de notificações, a um estado de erro das notificações e por fim a três estados que corresponderiam aos tipos de notificações definidos, tendo cada um destes três estados propriedades distintas.

Com o propósito de auxiliar o funcionamento do BLoC a criar, avançou-se com a definição de um modelo de dados capaz de suportar os parâmetros que comportariam os dados associados a uma notificação, dos principais parâmetros, para além do título e conteúdo da notificação, o de maior importância seria o correspondente ao tipo de notificação, existindo três tipos distintos, tal como referido no ultimo paragrafo, nomeadamente: o tipo “*info*” que tem como único propósito a apresentação da mensagem da notificação; o tipo “*route*” que para além da mensagem a transmitir tem um comportamento especial ao clicar-se sobre a notificação em questão, redirecionando o utilizador para a janela da aplicação móvel correspondente ao conteúdo da notificação; e por fim o tipo “*navigate*” que essencialmente apresenta o mesmo funcionamento que o tipo anterior, no entanto, em termos de implementação em vez de redirecionar o utilizador para uma outra janela, altera o separador no menu navegação inferior para o correspondente ao conteúdo da notificação.

Por fim, com algum *input* do seu colega de equipa, o estudante procedeu à criação do BLoC recorrendo ao modelo, eventos e estados anteriormente criados. O maior desafio desta fase do desenvolvimento terá sido a percepção da necessidade de, para além de preparar a aplicação móvel para receber as notificações originadas pelo FCM, ser ainda necessário implementar notificações locais. Isto deve-se ao funcionamento do sistema de notificações interno do sistema operativo Android e iOS, limitação esta que impede o sistema de apresentar a notificação quando a aplicação está aberta e em execução, ficando à total responsabilidade do programador decidir como reagir e/ou apresentar a informação recebida pela notificação. A solução apresentada pelo estudante foi a criação através de notificações locais de uma réplica da notificação recebida por parte do FCM.

### 5.3.2. Mapa Principal

O estudante ficou responsável pela implementação da interface gráfica, na aplicação móvel, do ecrã principal que é apresentado após a autenticação. As tarefas inerentes a esta responsabilidade são parte dos requisitos funcionais 14, 21 e 22.

O ecrã principal é composto por um mapa dinâmico que preenche o espaço disponível na janela, uma barra de título na parte superior do ecrã e um menu de navegação rápida na parte inferior. No caso de haver um agendamento de boleia para um futuro próximo, o nome, data e estado do mesmo serão apresentados numa janela flutuante na parte inferior do ecrã, tal como a rota da viagem, que é desenhada no mapa.

Na Figura 5.13 é possível visualizar duas capturas de ecrã da janela principal da aplicação no modo de motorista. O modo de motorista é identificável pela cor roxa dos elementos. No caso da imagem da esquerda, o motorista tinha um agendamento ativo no estado “à espera de passageiros”. O motorista pode a qualquer momento avançar para o próximo estado de um agendamento ativo pressionando o botão “próxima etapa”, ou abrir a aplicação de navegação do dispositivo pressionando o botão “direções”.

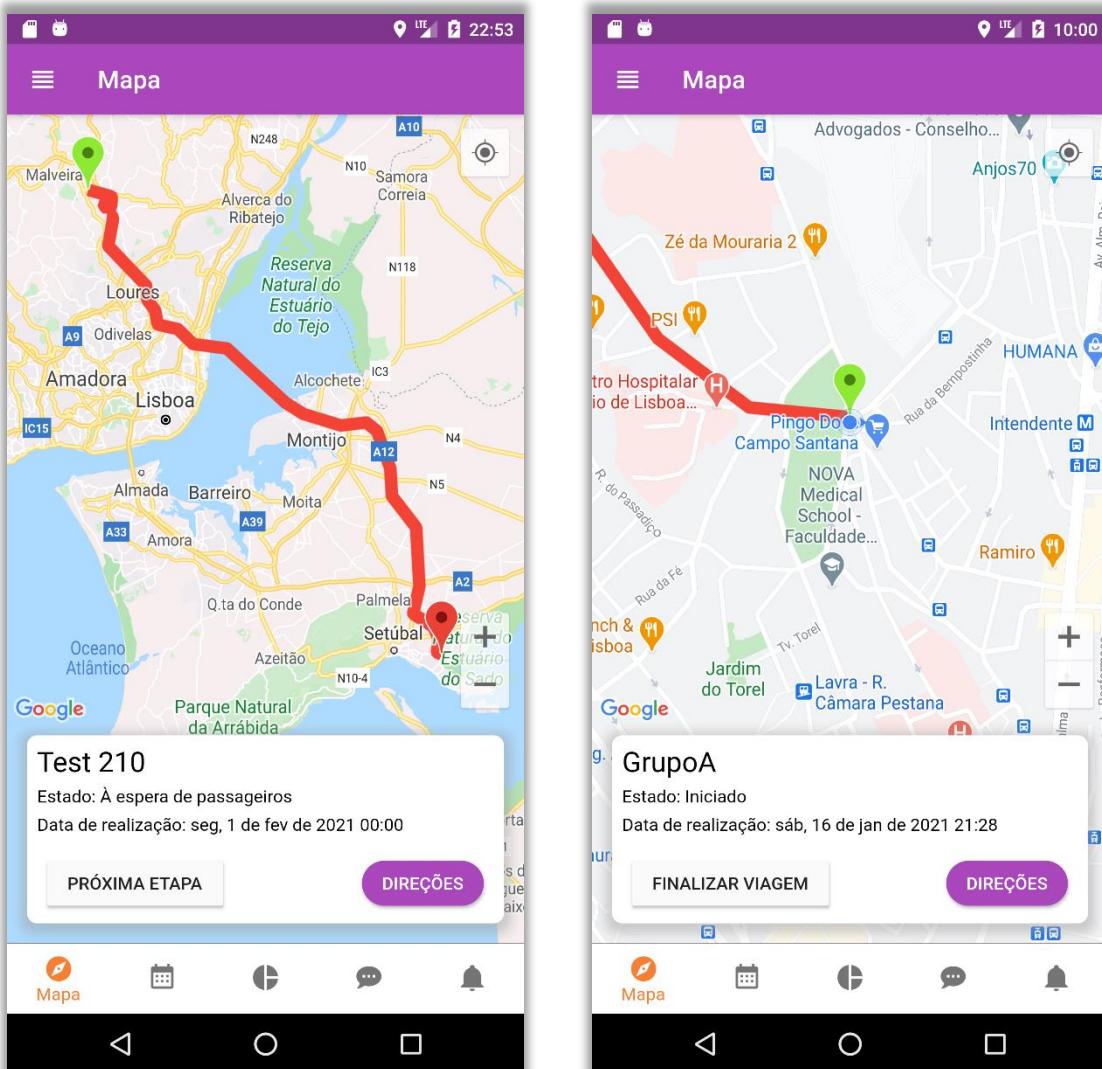


Figura 5.13 - Capturas de ecrã da janela principal da aplicação móvel no modo de motorista

Na Figura 5.14 é possível visualizar do lado esquerdo uma captura de ecrã da janela principal da aplicação no modo de passageiro. O modo de passageiro é identificável pela cor-de-laranja dos elementos. O passageiro pode a qualquer momento abrir a aplicação de navegação do dispositivo móvel pressionando o botão “direções”. As coordenadas de destino serão automaticamente preenchidas na aplicação de navegação permitindo deste modo visualizar as direções e iniciar a navegação para o destino, tal como está visível na captura de ecrã no lado esquerdo da referida figura 5.14.

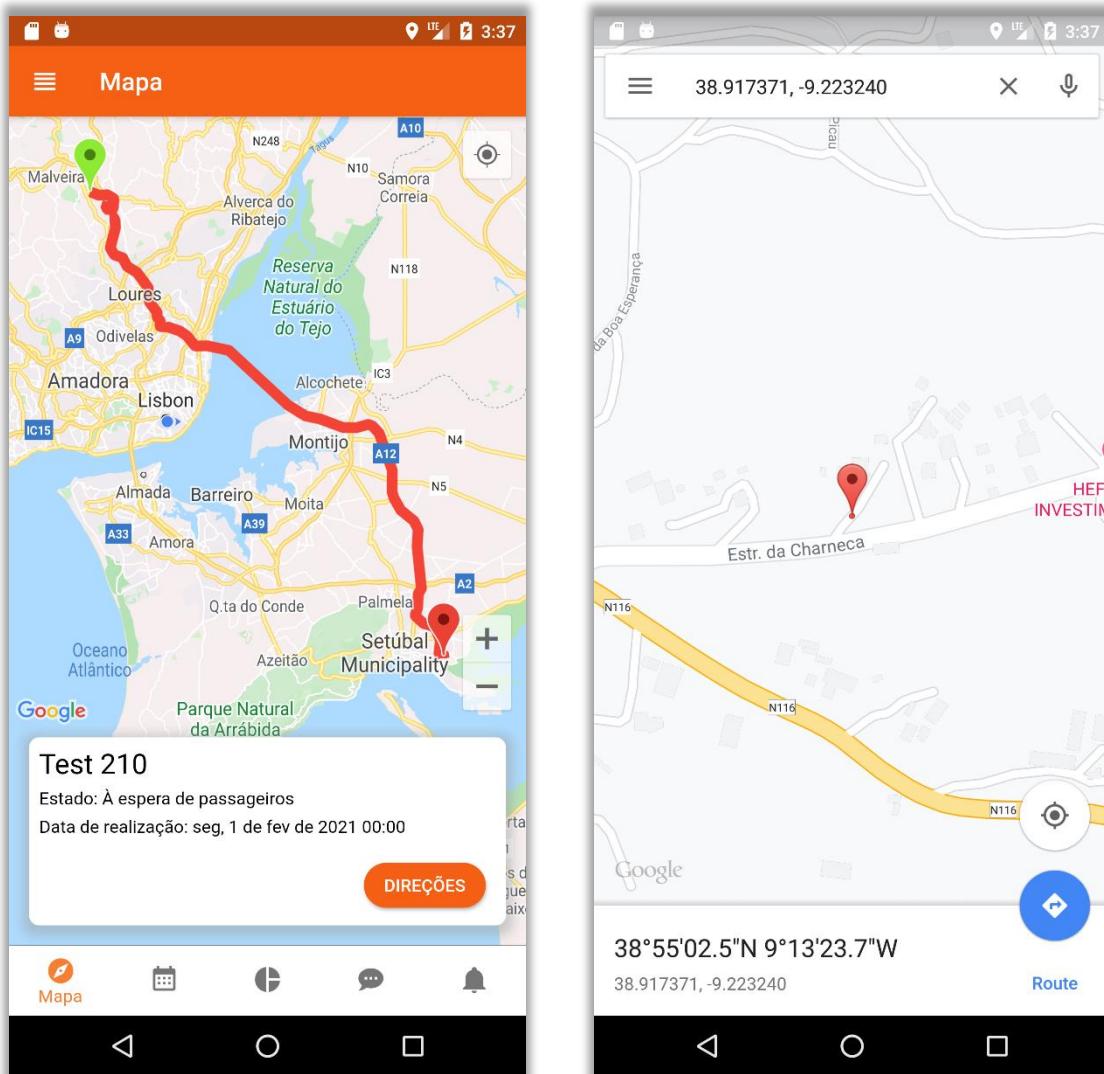


Figura 5.14 - Capturas de ecrã da janela principal da aplicação móvel no modo de passageiro

### 5.3.3. Gestão de Veículos

O estudante ficou responsável pela implementação da interface gráfica, na aplicação móvel, dos ecrãs relacionados com a gestão de veículos. As tarefas inerentes a esta responsabilidade são parte do requisito funcional 5.

A secção de gestão de veículos é acessível através do botão “Veículos” presente no menu lateral da aplicação. Na Figura 5.15 é possível visualizar duas capturas de ecrã das janelas relacionadas com a gestão de veículos.

Na parte esquerda da referida figura temos a janela que é apresentada ao entrar nesta secção da aplicação, onde são listados todos os veículos pertencentes ao utilizador. Para cada veículo listado é possível “Editar” ou “Apagar” o mesmo, no entanto, devido ao modo de como o consumo, emissões de CO<sub>2</sub> e custos associados a uma determinada rota são calculados seria necessária uma implementação mais complexa e demorada de modo a ser possível editar os

detalhes de um veículo, com exceção do nome do veículo que é utilizado apenas como uma referência para o utilizador. Deste modo para alterar qualquer outro campo será necessário apagar o veículo atual e criar um novo. De notar que por uma questão de consistência dos dados não é possível apagar veículos que estejam associados a um grupo de boleia.

Para criar um novo veículo basta pressionar o botão flutuante com o símbolo “+” presente no canto inferior direito da janela de listagem. Na captura de ecrã presente na parte direita da figura supramencionada tem-se a janela de criação de um novo veículo. Nesta janela é possível indicar um nome para o veículo, o tipo de combustível através de uma *dropdown*, o consumo, as emissões de CO<sub>2</sub> e o número de lugares disponíveis. De modo a impedir erros de introdução por parte do utilizador, o estudante decidiu tirar proveito da capacidade nativa do Flutter que permite adicionar instruções de validação a um determinado campo de introdução de dados através da simples adição do parâmetro “validator:”.

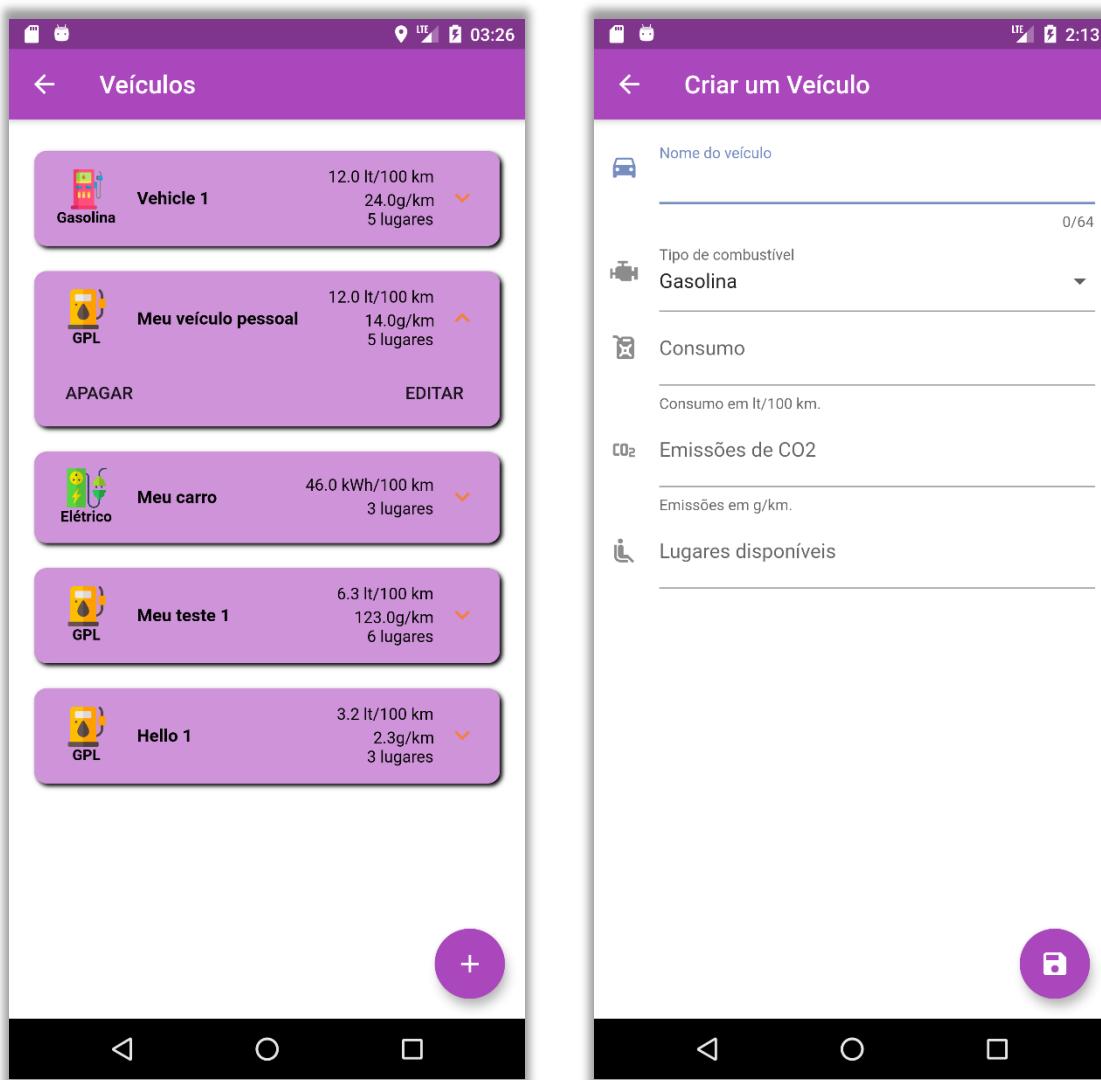


Figura 5.15 - Capturas de ecrã da janela de gestão de veículos da aplicação móvel

## 6. Discussão do Trabalho Realizado

Com o propósito de avaliar o trabalho realizado, começou-se por realizar a validação da solução através da realização de diversos testes, seguindo-se da avaliação do projeto com base num balanço comparativo da calendarização do mesmo.

### 6.1. VALIDAÇÃO DA SOLUÇÃO

A nível de validação da solução realizou-se testes de integração e de sistema. Apresenta-se também a especificação dos testes de usabilidade, embora não tenha sido possível realizar os mesmos (a decorrer em novembro de 2021).

Os testes de integração permitiram validar o funcionamento dos componentes à medida que estes eram desenvolvidos e integrados com o sistema, revelando possíveis problemas subjacentes aos mesmos, viabilizando deste modo a sua antecipada resolução. Com os testes de sistema foi possível realizar um levantamento geral das funcionalidades desenvolvidas e por desenvolver, permitindo ainda verificar os requisitos de qualidade definidos em contraste com o planeamento.

#### 6.1.1. Testes de Integração

Usualmente, a prática de testes de integração implicam a verificação de que os módulos implementados ao longo das iterações de um sistema funcionam entre si como esperado. Dado que no caso particular deste projeto procurou-se implementar apenas partes de módulos, devido às restrições descritas anteriormente, os testes realizados neste âmbito focaram-se na interação entre a API da plataforma com os componentes de interface desenvolvidos da aplicação móvel.

Devido ao desenvolvimento da API progredir rapidamente em relação à aplicação móvel, o estudante considerou pertinente a realização antecipada destes testes manualmente através de ferramentas como o Postman<sup>6</sup>, testando deste modo o correto funcionamento de cada rota desenvolvida, simulando possíveis erros. Com a conclusão do desenvolvimento de um componente e a sua integração com a aplicação móvel, testava-se o funcionamento de toda a plataforma para que se levantassem possíveis falhas que seriam corrigidas até que o resultado fosse satisfatório.

<sup>6</sup> <https://www.postman.com/>

## 6.1.2. Testes de Sistema

A plataforma foi testada como um todo após a realização das devidas integrações, de modo a validar a interação entre os componentes e com a totalidade do sistema. Com isto foi possível verificar quais as *user stories*, tarefas e requisitos de qualidade que terão sido implementados com sucesso.

A nível dos requisitos de qualidade, apresentados no subcapítulo 4.4.2.1, conseguiu-se implementar praticamente todos, com exceção dos RQ2 e RQ3 devido a dependerem dos testes de usabilidade abordados na introdução deste subcapítulo. Os requisitos e os seus respetivos resultados, no final do desenvolvimento do projeto, encontram-se representados na Tabela 6.1.

Tabela 6.1 - Resultado da implementação dos requisitos de qualidade

ID	Requisito	Resultado
RQ1	O sistema deverá prevenir o acesso de utilizadores que não façam parte da comunidade IPS.	O sistema permite apenas acesso de utilizadores pertencentes à comunidade IPS.
RQ2	A aplicação móvel deverá ter um aspetto visual apelativo o suficiente de modo que a maioria dos utilizadores achem que a mesma é visualmente apelativa.	Não pôde ser testado até à data final do projeto.
RQ3	A aplicação móvel deverá ser intuitiva o suficiente de modo que a maioria dos utilizadores usufruam das funcionalidades disponibilizadas sem dificuldades.	Não pôde ser testado até à data final do projeto.
RQ4	O sistema deverá estar internacionalizado e/ou localizado para o idioma português e inglês.	O sistema permite a alteração do idioma entre português e inglês.
RQ5	O código-fonte do sistema deverá ser devidamente documentado.	A documentação do sistema encontra-se gerada no formato HTML.
RQ6	O sistema deverá ser desenvolvido de forma a evitar um código “sujo”.	O código-fonte do sistema terá sido desenvolvido seguindo boas práticas e os padrões adequados.

## 6.1.3. Testes de Usabilidade

Com o objetivo de obter *feedback* da primeira versão desenvolvida da plataforma planeou-se a realização de testes de usabilidade a um conjunto de utilizadores interessados neste sistema e pertencentes à comunidade IPS. Os testes teriam como objetivo avaliar a

pertinências das funcionalidades implementadas, validar a interface gráfica desenvolvida tanto a nível visual como a nível de experiência de utilização e apurar informação relevante ao âmbito da plataforma como um todo através de um questionário, tais como dados demográficos, hábitos de transporte para o IPS, frequência de recurso a boleias e questões acerca das preocupações quanto à sustentabilidade na mobilidade dos utilizadores.

Estes testes seriam realizados através da observação direta da interação dos utilizadores com a interface da aplicação móvel, seguindo um guia de testes com a indicação da sequência de ações a realizar de modo a percorrer todas as funcionalidades implementadas. Após a realização dos testes de observação direta, os utilizadores são submetidos a um questionário anónimo que procura obter informações relevantes no âmbito da plataforma sobre os mesmos, possuindo ainda uma secção para classificar a usabilidade da aplicação móvel, integrando uma parte baseada na metodologia SUS que é uma referência na área.

A metodologia SUS, ou de forma não abreviada *System Usability Scale*, permite avaliar em forma de escala a usabilidade global de um produto ou serviço, incluindo software [40]. O método encontra-se dividido em três características, nomeadamente:

- **Eficácia:** Verificar se os utilizadores conseguem atingir o seu objetivo;
- **Eficiência:** Saber a quantidade de esforço e recursos gastos para atingir esse objetivo;
- **Satisfação:** Saber se a experiência dos utilizadores foi satisfatória.

Após uma sessão de testes guiados com um produto, o utilizador em questão é submetido a um questionário de dez perguntas. Tais perguntas são concebidas de modo a serem respondidas rapidamente e com o mínimo de esforço possível, de modo a obter-se um *feedback* rápido e real.

As opções de resposta são baseadas na Escala Likert de cinco pontos, que permite avaliar as respostas às questões de forma qualitativa ordinal, indo de “discordo totalmente” a “concordo totalmente”. De modo a calcular a métrica de usabilidade geral, atribui-se valores pontuais de 1 a 5, correspondentes à escala de resposta.

Ao observar as respostas de um inquirido e a pontuação correspondente para cada resposta é possível calcular a pontuação SUS geral, numa escala de 0 a 100. Somando o total de pontos das perguntas de número ímpar (pergunta 1, 3, etc.) e subtrair 5 do mesmo, obtém-se o valor de X. Ao somar os pontos das perguntas par e subtrair esse total do número 25, obtém-se o valor de Y. Ao somar X e Y e multiplicando-se por 2.5 chega-se à pontuação SUS geral. Um exemplo deste cálculo seria:

- **Ímpar:**  $(4 + 5 + 3 + 3 + 4) - 5 = 19 - 5 = 14$
- **Par:**  $25 - (2 + 1 + 3 + 1 + 1) = 25 - 8 = 17$
- **Pontuação SUS:**  $(14 + 17) \times 2.5 = 31 \times 2.5 = 77.5$

A pontuação SUS média é de 68, deste modo a obtenção de uma pontuação inferior a este valor significa que existem problemas com o design do produto a ser testado que precisam de ser revistos e corrigidos, enquanto um valor superior a este indica a necessidade de pequenas melhorias [41]. As pontuações abaixo de 51 requerem atenção imediata no que diz respetivo a questões de usabilidade ou pontos que precisem de ser resolvidos ou investigados mais profundamente. O total de pontos pode ser categorizado desde inaceitável a aceitável, a Figura 6.1 representa a escala de aceitação da pontuação SUS.

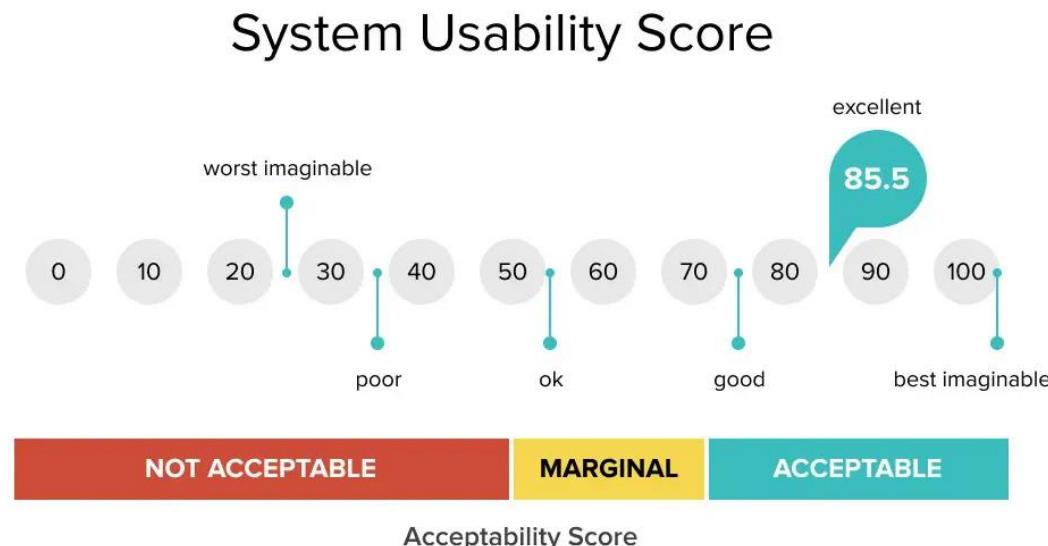


Figura 6.1 - Escala de aceitação da pontuação SUS (retirado de <https://xd.adobe.com/ideas/process/user-testing/system-usability-scale-ux/>)

Embora os testes de usabilidade tenham sido planeados, devido às circunstâncias de confinamento geral existentes durante o período de desenvolvimento do projeto, não foi possível realizá-los com os potenciais utilizadores, pelo que este processo terá ido definido como trabalho futuro. A equipa elaborou o questionário com perguntas de caracterização do indivíduo, perguntas SUS referentes à aplicação móvel e ainda algumas questões de resposta aberta para opinião sobre a plataforma. Questionário este que pode ser consultado no Anexo 4 – Questionário de Testes de Usabilidade.

## 6.2. AVALIAÇÃO DO PROJETO

O desenvolvimento do projeto estendeu-se cronologicamente devido à sua complexidade, que em conjunto com o tempo despendido na aprendizagem das novas tecnologias e serviços utilizados, acabou por pressionar ligeiramente o cumprimento da calendarização planeada. Esta situação foi ainda mais agravada devido às restrições governamentais a que a equipa, os orientadores e o IPS estiveram sujeitos devido à crise sanitária nacional a acontecer durante o desenvolvimento deste projeto. Vale mencionar ainda

que tal como referido anteriormente, o estudante terá ficado responsável pela preparação, integração e planificação do Jira Software. Este processo terá sido de grande utilidade ao longo do desenvolvimento pois facilitou consideravelmente a organização da equipa e dos orientadores mas, no entanto, o tempo despendido pelo estudante para alcançar estes objetivos terá sido demasiado elevado, tempo este que poderia ter sido aproveitado de melhor forma de modo a impedir atrasos na calendarização planeada.

A Figura 6.2 representa o diagrama de fluxo acumulado das transições de estado dos cartões Kanban das tarefas atribuídas ao estudante, retirado do Jira Software. Os estados correspondem às colunas estabelecidas para o quadro Kanban do projeto: *To Do* (por fazer), *Developig* (em desenvolvimento), *Testing* (em testes) e *Done* (finalizado). De notar que no final houve várias tarefas que terão ficado no estado de *Testing*, isto deve-se a uma questão de priorização de alocação de tempo para a conclusão da primeira versão, sendo que terão sido realizados diversos tipos de testes, incluindo testes unitários, para todas as principais funcionalidades do sistema, tendo ficado para trabalho futuro a conclusão dos testes unitários de apenas tarefas de menor impacto no sistema.

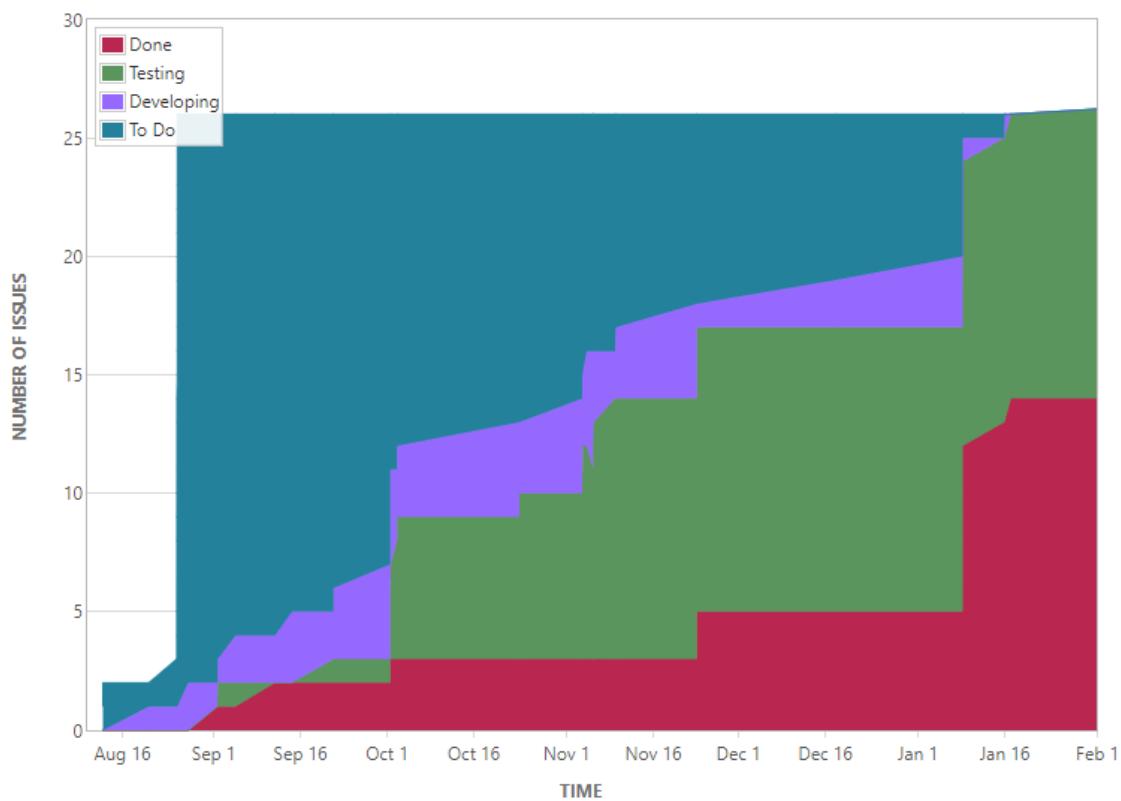


Figura 6.2 - Diagrama de fluxo acumulado da transição de estado das tarefas atribuídas ao estudante

## 7. Conclusão e Trabalho Futuro

Em conclusão, a oportunidade proporcionada pelo Sustain.RD de dar início ao desenvolvimento de uma plataforma de partilha de boleias para uso interno pela comunidade IPS possibilitou ao estudante contribuir positivamente para a sustentabilidade na mobilidade da instituição a que pertence, envolvendo o mesmo num ambiente de investigação.

O trabalho realizado pelo estudante, em conjunto com o seu colega de equipa, forneceu à instituição uma primeira versão funcional da plataforma. Com os testes de usabilidade planeados vai ser possível obter informação relevante para o melhoramento e possível continuação do desenvolvimento. Esta versão permitiu à equipa e aos orientadores avaliar e refinar as suas ideias iniciais quanto ao projeto como um todo, fornecendo dados que possam influenciar o planeamento de futuros módulos a implementar.

O cumprimento da calendarização do projeto revelou-se eventualmente um desafio para o estudante. Tal situação deve-se à grande dimensão do projeto em questão, ponto este que implicou uma grande alocação de horas exclusivamente para o adequado planeamento inicial do projeto. Outros fatores de impacto terão sido o tempo alocado à aprendizagem da utilização de serviços como o Microsoft Azure Active Directory e Google Directions API, a preocupação em documentar ao máximo o código fonte desenvolvido com perspetiva a futuro desenvolvimento por parte de outros indivíduos e por ultimo, mas não menos importante, o facto de todos os envolvidos estarem submetidos à situação de crise sanitária, que terá sido de todo desafiante tanto para o estudante como para o resto da equipa e respetivos orientadores. Deste modo, ao contrário do que o estudante pretendia, a data de finalização do desenvolvimento da primeira versão da plataforma acabou por exceder o prazo limite planeado.

A nível de trabalho futuro, considera-se prioritários os módulos de chat e de gamificação, visto que as comunicações diretas entre utilizadores propõem a flexibilização no agendamento de boleias e porque a existência de elementos relacionados com a progressão em jogos motiva a sua utilização. De notar que a introdução de técnicas de gamificação numa aplicação móvel é um método muito utilizado não só pela indústria de videojogos, mas por outros setores que adotam estas metodologias para aumentar a taxa de adesão e de fidelização. Com este último ponto em mente vale mencionar, que à data deste documento, o módulo de gamificação encontra-se a ser investigado e eventualmente desenvolvido por um estudante de Mestrado em Engenharia Informática da Faculdade de Ciências e Tecnologia (FCT) da NOVA.

Alguns aspetos planeados para serem melhorados em versões futuras passam por calcular o custo de combustível por viagem tendo em consideração o número de pessoas presentes no veículo, o tipo de combustível e os preços médios do mercado de combustíveis fosseis em tempo real. Outro aspeto que, apesar de não passar de uma proposta, terá sido aprovado pelos orientadores seria o desenvolvimento e implementação de algoritmos para realizar a procura de boleias ao longo da totalidade da rota de uma viagem, de modo a permitir a correspondência de boleias em pontos intermédios. Ainda no âmbito do desenvolvimento de

algoritmos relacionados com o sistema de procura de boleias, a equipa sugere a determinação da disponibilidade de boleias com base no horário de um utilizador e da sua respetiva localização, determinado de modo automático boleias disponíveis que correspondam às suas necessidades.

No que consta ao futuro individual, o estudante pretende prosseguir estudos e, dentro dos possíveis, participar em outros projetos de investigação que contribuam para o desenvolvimento pessoal e profissional do mesmo.

## 8. Referências Bibliográficas

- [1] Instituto Politécnico de Setúbal, "IPS - Politécnico de Setúbal renova galardão Eco-Escolas," 2 setembro 2020. [Online]. Available: [https://www.ips.pt/ips\\_si/noticias\\_geral.ver\\_noticia?P\\_NR=7829](https://www.ips.pt/ips_si/noticias_geral.ver_noticia?P_NR=7829). [Acedido em 7 março 2021].
- [2] Instituto Politécnico de Setúbal, "IPS - Responsabilidade Social e Sustentabilidade no IPS," [Online]. Available: [https://www.ips.pt/ips\\_si/web\\_base.gera\\_pagina?P\\_pagina=41982](https://www.ips.pt/ips_si/web_base.gera_pagina?P_pagina=41982). [Acedido em 7 março 2020].
- [3] Instituto Politécnico de Setúbal, "IPS - Apresentação," [Online]. Available: [https://www.ips.pt/ips\\_si/web\\_base.gera\\_pagina?P\\_pagina=30847](https://www.ips.pt/ips_si/web_base.gera_pagina?P_pagina=30847). [Acedido em 24 fevereiro 2021].
- [4] Instituto Politécnico de Setúbal, "IPS - Visão, Missão & Valores," [Online]. Available: [https://www.ips.pt/ips\\_si/web\\_base.gera\\_pagina?P\\_pagina=30310](https://www.ips.pt/ips_si/web_base.gera_pagina?P_pagina=30310). [Acedido em 24 fevereiro 2021].
- [5] Instituto Politécnico de Setúbal, "Sustain.RD - Research Center for Engineering and Sustainable Development - ESTSetúbal/IPS," [Online]. Available: <https://www.estsetubal.ips.pt/id-e-empresas/centros-investigacao/SustainRD>. [Acedido em 24 fevereiro 2021].
- [6] Velis Velis, "Singu Carpooling Demo for Android - APK Download," [Online]. Available: <https://apkpure.com/singu-carpooling-demo/com.velisit.democarpooling>. [Acedido em 26 fevereiro 2021].
- [7] Técnico Lisboa, "Plataforma de boleias (Carpooling) exclusiva do Técnico," 15 setembro 2017. [Online]. Available: <https://tecnico.ulisboa.pt/pt/noticias/campus-e-comunidade/plataforma-de-boleias-carpooling-exclusiva-do-tecnico/>. [Acedido em 26 fevereiro 2021].
- [8] Comovee, "Rideshares for Employees - Comovee," [Online]. Available: <https://www.comovee.com/rideshare-software/>. [Acedido em 26 fevereiro 2021].
- [9] Poola, "Corporate Carpooling - Poola.app," [Online]. Available: <https://poola.app/>. [Acedido em 26 fevereiro 2021].

- [10] Atlassian, "What is Agile? | Atlassian," [Online]. Available: <https://www.atlassian.com/agile>. [Acedido em 21 abril 2021].
- [11] J. P. Womack, D. T. Jones e D. Roos, *The Machine That Changed the World*, Simon and Schuster, 2007.
- [12] Kanbanize, "What Is Kanban? Explained in 10 Minutes | Kanbanize," [Online]. Available: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>. [Acedido em 19 maio 2021].
- [13] MDN Web Docs, "JavaScript | MDN," 20 julho 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Acedido em 28 julho 2021].
- [14] Node.js, "About | Node.js," [Online]. Available: <https://nodejs.org/en/about/>. [Acedido em 28 julho 2021].
- [15] Google, "Language tour | Dart," [Online]. Available: <https://dart.dev/guides/language/language-tour>. [Acedido em 16 setembro 2021].
- [16] Google, "Flutter | Flutter architectural overview," [Online]. Available: <https://flutter.dev/docs/resources/architectural-overview>. [Acedido em 16 setembro 2021].
- [17] MongoDB Inc., "What Is MongoDB | MongoDB," [Online]. Available: <https://www.mongodb.com/what-is-mongodb>. [Acedido em 17 setembro 2021].
- [18] K. Herbert, "What is Apple Push Notification service (APNs)? - Definition from WhatIs.com," [Online]. Available: <https://whatis.techtarget.com/definition/Apple-Push-Notification-service-APNs>. [Acedido em 17 setembro 2021].
- [19] Firebase, "Firebase Cloud Messaging," [Online]. Available: <https://firebase.google.com/docs/cloud-messaging>. [Acedido em 17 setembro 2021].
- [20] J. Tamplin, "The Firebase Blog: Firebase is Joining Google!," 21 outubro 2014. [Online]. Available: <https://firebase.googleblog.com/2014/10/firebase-is-joining-google.html>. [Acedido em 17 setembro 2021].
- [21] Microsoft, "What is Azure Active Directory? - Azure Active Directory | Microsoft Docs," 6 maio 2020. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis>. [Acedido em 17 setembro 2021].

- [22] Microsoft, “Overview of Microsoft Graph - Microsoft Graph | Microsoft Docs,” 9 março 2021. [Online]. Available: <https://docs.microsoft.com/en-us/graph/overview>. [Acedido em 17 setembro 2021].
- [23] T. Reenskaug, “Notes and Historical documents,” [Online]. Available: <https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html>. [Acedido em 16 maio 2021].
- [24] M. Fowler, “GUI Architectures,” 18 julho 2006. [Online]. Available: <https://martinfowler.com/eaaDev/uiArchs.html>. [Acedido em 2021 maio 2021].
- [25] TechTerms - The Tech Terms Computer Dictionary, “MVC (Model-View-Controller) Definition,” 7 março 2018. [Online]. Available: <https://techterms.com/definition/mvc>. [Acedido em 16 maio 2021].
- [26] C2.com, “Model View Controller History,” 11 junho 2012. [Online]. Available: <https://wiki.c2.com/?ModelViewControllerHistory>. [Acedido em 16 maio 2021].
- [27] R. T. Fielding, “Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST),” 2000. [Online]. Available: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm). [Acedido em 17 maio 2021].
- [28] R. Fielding, “RFC 7231 - Hypertext Transfer Protocol: Semantics and Content, Section 4,” junho 2014. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7231#section-4>. [Acedido em 17 maio 2021].
- [29] L. Richardson e S. Ruby, RESTful Web Services, Sebastopol, Calif.: O'Reilly Media, Inc., 2007.
- [30] R. P. Fostaini, “Entendendo o Repository Pattern,” 30 maio 2018. [Online]. Available: <https://medium.com/@renicius.pagotto/entendendo-o-repository-pattern-fcdd0c36b63b>. [Acedido em 22 fevereiro 2021].
- [31] D. Boelens, “Reactive Programming — Streams — BLoC,” 2018 setembro 2018. [Online]. Available: <https://medium.com/flutter-community/reactive-programming-streams-bloc-6f0d2bd2d248>. [Acedido em 23 fevereiro 2021].
- [32] F. Angelov, “bloc - Dart API docs,” [Online]. Available: <https://pub.dev/documentation/bloc/latest/>. [Acedido em 23 fevereiro 2021].

- [33] I. Padayachee, P. Kotzé e A. V. der Merwe, “ISO 9126 external systems quality characteristics, sub-characteristics and domain specific criteria for evaluating e-Learning systems,” janeiro 2010.
- [34] Google, “Introduction - Material Design,” [Online]. Available: <https://material.io/design/introduction>. [Acedido em 7 março 2021].
- [35] A. Andrews, “Material Design Icons,” [Online]. Available: <https://materialdesignicons.com/>. [Acedido em 7 março 2021].
- [36] GitHub Education, “GitHub Student Developer Pack - GitHub Education,” [Online]. Available: <https://education.github.com/pack>. [Acedido em 26 agosto 2021].
- [37] Fandom, “Mongoose | Node.js Wiki | Fandom,” [Online]. Available: <https://nodejs.fandom.com/wiki/Mongoose>. [Acedido em 01 abril 2021].
- [38] Microsoft, “Documentação do Azure Active Directory | Microsoft Docs,” [Online]. Available: <https://docs.microsoft.com/pt-pt/azure/active-directory/>. [Acedido em 16 março 2021].
- [39] T. Onyszko, “Microsoft Azure Active Directory guide - AAD, B2C, B2B (with diagrams),” 22 dezembro 2016. [Online]. Available: <https://www.predicagroup.com/blog/azure-ad-b2b-b2c-puzzled-out/>. [Acedido em 17 março 2021].
- [40] Usability.gov, “System Usability Scale (SUS) | Usability.gov,” [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. [Acedido em 13 setembro 2021].
- [41] A. Smyk, “The System Usability Scale & How it’s Used in UX | Adobe XD Ideas,” 17 março 2020. [Online]. Available: <https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/>. [Acedido em 13 setembro 2021].

## 9. Anexos

### 9.1. ANEXO 1 – APURAÇÃO DOS DADOS REFERENTES AO INQUÉRITO DE PERCEÇÃO AMBIENTAL EM RELAÇÃO À MOBILIDADE DA COMUNIDADE IPS

Gostaria que existisse no IPS uma plataforma de gestão de boleias para minimizar o número de veículos nos parques de estacionamento do IPS?

Unidade Orgânica do IPS – ESCE			Não	Sim	Total	
<b>Identificação</b>	Docente	N.º Respostas	12	45	<b>57</b>	
		% Respostas	21,1%	78,9%	<b>100,0%</b>	
	Estudante	N.º Respostas	40	218	<b>258</b>	
		% Respostas	15,5%	84,5%	<b>100,0%</b>	
	Não-Docente	N.º Respostas	2	8	<b>10</b>	
		% Respostas	20,0%	80,0%	<b>100,0%</b>	
<b>Total</b>		<b>N.º Respostas</b>	<b>54</b>	<b>271</b>	<b>325</b>	
		<b>% Respostas</b>	<b>16,6%</b>	<b>83,4%</b>	<b>100,0%</b>	

Unidade Orgânica IPS – ESE			Não	Sim	Total	
<b>Identificação</b>	Docente	N.º Respostas	3	29	<b>32</b>	
		% Respostas	9,4%	90,6%	<b>100,0%</b>	
	Estudante	N.º Respostas	20	91	<b>111</b>	
		% Respostas	18,0%	82,0%	<b>100,0%</b>	
	Não-Docente	N.º Respostas	1	3	<b>4</b>	
		% Respostas	25,0%	75,0%	<b>100,0%</b>	
<b>Total</b>		<b>N.º Respostas</b>	<b>24</b>	<b>123</b>	<b>147</b>	
		<b>% Respostas</b>	<b>16,3%</b>	<b>83,7%</b>	<b>100,0%</b>	

Unidade Orgânica do IPS – ESS	Não	Sim	Total
-------------------------------	-----	-----	-------

<b>Identificação</b>	Docente	N.º Respostas	2	29	<b>31</b>
		% Respostas.	6,5%	93,5%	<b>100,0%</b>
	Estudante	N.º Respostas	17	78	<b>95</b>
		% Respostas.	17,9%	82,1%	<b>100,0%</b>
	Não-Docente	N.º Respostas	0	3	<b>3</b>
		% Respostas.	0,0%	100,0%	<b>100,0%</b>
<b>Total</b>		<b>N.º Respostas</b>	<b>19</b>	<b>110</b>	<b>129</b>
		<b>% Respostas.</b>	<b>14,7%</b>	<b>85,3%</b>	<b>100,0%</b>

Unidade Orgânica do IPS – ESTBarreiro			Não	Sim	Total
<b>Identificação</b>	Docente	N.º Respostas	4	14	<b>18</b>
		% Respostas.	22,2%	77,8%	<b>100,0%</b>
	Estudante	N.º Respostas	16	80	<b>96</b>
		% Respostas.	16,7%	83,3%	<b>100,0%</b>
	Não-Docente	N.º Respostas	2	0	<b>2</b>
		% Respostas.	100,0%	0,0%	<b>100,0%</b>
<b>Total</b>		<b>N.º Respostas</b>	<b>22</b>	<b>94</b>	<b>116</b>
		<b>% Respostas.</b>	<b>19,0%</b>	<b>81,0%</b>	<b>100,0%</b>

Unidade Orgânica do IPS – ESTSetúbal			Não	Sim	Total
<b>Identificação</b>	Docente	N.º Respostas	7	44	<b>51</b>
		% Respostas.	13,7%	86,3%	<b>100,0%</b>
	Estudante	N.º Respostas	37	197	<b>234</b>
		% Respostas.	15,8%	84,2%	<b>100,0%</b>
	Não-Docente	N.º Respostas	1	9	<b>10</b>
		% Respostas.	10,0%	90,0%	<b>100,0%</b>
<b>Total</b>		<b>N.º Respostas</b>	<b>45</b>	<b>250</b>	<b>295</b>
		<b>% Respostas.</b>	<b>15,3%</b>	<b>84,7%</b>	<b>100,0%</b>

Unidade Orgânica do IPS – Serviços Centrais			Não	Sim	Total	
<b>Identificação</b>	Docente	N.º Respostas	1	3	4	
		% Respostas	25,0%	75,0%	100,0%	
	Não-Docente	N.º Respostas	6	29	35	
		% Respostas	17,1%	82,9%	100,0%	
<b>Total</b>		<b>N.º Respostas</b>	<b>7</b>	<b>32</b>	<b>39</b>	
		<b>% Respostas</b>	<b>17,9%</b>	<b>82,1%</b>	<b>100,0%</b>	

Como se desloca habitualmente para o IPS?

Cruzamentos de dados com a pergunta: Gostaria que existisse no IPS uma plataforma de gestão de boleias para minimizar o número de veículos nos parques de estacionamento do IPS?

Unidade Orgânica do IPS – ESCE			Não	Sim	Total	
<b>Como se desloca habitualmente para o IPS?</b>	A pé	N.º Respostas	6	32	38	
		% Respostas	15,8%	84,2%	100,0%	
	Bicicleta	N.º Respostas	0	1	1	
		% Respostas	0,0%	100,0%	100,0%	
	Mota	N.º Respostas	1	2	3	
		% Respostas	33,3%	66,7%	100,0%	
	Outro	N.º Respostas	0	3	3	
		% Respostas	0,0%	100,0%	100,0%	
	Transporte Público	N.º Respostas	22	95	117	
		% Respostas	18,8%	81,2%	100,0%	
	Viatura Própria	N.º Respostas	25	138	163	
		% Respostas	15,3%	84,7%	100,0%	
<b>Total</b>		<b>N.º Respostas</b>	<b>54</b>	<b>271</b>	<b>325</b>	
		<b>% Respostas</b>	<b>16,6%</b>	<b>83,4%</b>	<b>100,0%</b>	

Unidade Orgânica do IPS – ESSE			Não	Sim	Total
<b>Como se desloca habitualmente para o IPS?</b>	A pé	N.º Respostas	1	4	<b>5</b>
		% Respostas	20,0%	80,0%	<b>100,0%</b>
	Mota	N.º Respostas	0	1	<b>1</b>
		% Respostas	0,0%	100,0%	<b>100,0%</b>
	Outro	N.º Respostas	0	1	<b>1</b>
		% Respostas	0,0%	100,0%	<b>100,0%</b>
	Transporte Público	N.º Respostas	16	58	<b>74</b>
		% Respostas	21,6%	78,4%	<b>100,0%</b>
	Viatura Própria	N.º Respostas	7	59	<b>66</b>
		% Respostas	10,6%	89,4%	<b>100,0%</b>
<b>Total</b>		<b>N.º Respostas</b>	<b>24</b>	<b>123</b>	<b>147</b>
		<b>% Respostas</b>	<b>16,3%</b>	<b>83,7%</b>	<b>100,0%</b>

Unidade Orgânica do IPS – ESS			Não	Sim	Total
<b>Como se desloca habitualmente para o IPS?</b>	A pé	N.º Respostas	3	25	<b>28</b>
		% Respostas	10,7%	89,3%	<b>100,0%</b>
	Outro	N.º Respostas	0	3	<b>3</b>
		% Respostas	0,0%	100,0%	<b>100,0%</b>
	Transporte Público	N.º Respostas	8	28	<b>36</b>
		% Respostas	22,2%	77,8%	<b>100,0%</b>
	Viatura Própria	N.º Respostas	8	54	<b>62</b>
		% Respostas	12,9%	87,1%	<b>100,0%</b>
<b>Total</b>		<b>N.º Respostas</b>	<b>19</b>	<b>110</b>	<b>129</b>
		<b>% Respostas</b>	<b>14,7%</b>	<b>85,3%</b>	<b>100,0%</b>

Unidade Orgânica do IPS – ESTBarreiro			Não	Sim	Total
	A pé	N.º Respostas	2	5	<b>7</b>

<b>Como se desloca habitualmente para o IPS?</b>		% Respostas	28,6%	71,4%	<b>100,0%</b>
	Transporte Público	N.º Respostas	6	37	<b>43</b>
		% Respostas	14,0%	86,0%	<b>100,0%</b>
	Viatura Própria	N.º Respostas	14	52	<b>66</b>
		% Respostas	21,2%	78,8%	<b>100,0%</b>
	<b>Total</b>		<b>N.º Respostas</b>	<b>22</b>	<b>94</b>
		<b>% Respostas</b>	<b>19,0%</b>	<b>81,0%</b>	<b>100,0%</b>

<b>Unidade Orgânica do IPS – ESTSetúbal</b>			<b>Não</b>	<b>Sim</b>	<b>Total</b>	
<b>Como se desloca habitualmente para o IPS?</b>	A pé	N.º Respostas	2	24	<b>26</b>	
		% Respostas	7,7%	92,3%	<b>100,0%</b>	
	Bicicleta	N.º Respostas	0	6	<b>6</b>	
		% Respostas	0,0%	100,0%	<b>100,0%</b>	
	Mota	N.º Respostas	1	2	<b>3</b>	
		% Respostas	33,3%	66,7%	<b>100,0%</b>	
	Outro	N.º Respostas	1	2	<b>3</b>	
		% Respostas	33,3%	66,7%	<b>100,0%</b>	
	Transporte Público	N.º Respostas	20	82	<b>102</b>	
		% Respostas	19,6%	80,4%	<b>100,0%</b>	
	Viatura Própria	N.º Respostas	21	134	<b>155</b>	
		% Respostas	13,5%	86,5%	<b>100,0%</b>	
<b>Total</b>		<b>N.º Respostas</b>	<b>45</b>	<b>250</b>	<b>295</b>	
		<b>% Respostas</b>	<b>15,3%</b>	<b>84,7%</b>	<b>100,0%</b>	

<b>Unidade Orgânica do IPS – Serviços Centrais</b>			<b>Não</b>	<b>Sim</b>	<b>Total</b>
<b>Como se desloca habitualmente para o IPS?</b>	Transporte Público	N.º Respostas	2	5	<b>7</b>
		% Respostas	28,6%	71,4%	<b>100,0%</b>
	Viatura Própria	N.º Respostas	5	27	<b>32</b>

		% Respostas	15,6%	84,4%	<b>100,0%</b>
<b>Total</b>	<b>N.º Respostas</b>		<b>7</b>	<b>32</b>	<b>39</b>
	<b>% Respostas</b>		<b>17,9%</b>	<b>82,1%</b>	<b>100,0%</b>

## 9.2. ANEXO 2 – TAREFAS EXTRAÍDAS DAS USER STORIES

Módulo	Descrição	Tipo de Tarefa
M1 - Módulo de Perfil	Inicializar e estruturar a API <i>Back-end</i>	<i>Back-end</i>
	Inicializar e estruturar a base de dados	<i>Back-end</i>
	Criar rota na API para autenticar o Utilizador no sistema de autenticação do IPS	<i>Back-end</i>
	A API deve obter os dados do Utilizador do sistema de autenticação IPS	<i>Back-end</i>
	Criar uma rota na API de pesquisa de utilizadores	<i>Back-end</i>
	Criar rota na API obter os dados de perfil de outros utilizadores	<i>Back-end</i>
	Criar rota na API para editar os dados de perfil do utilizador autenticado	<i>Back-end</i>
	Criar rota na API obter os dados de perfil utilizador autenticado	<i>Back-end</i>
	Criar rota na API para gerir (CRUD) os utilizadores bloqueados de um dado utilizador	<i>Back-end</i>
	Criar rota na API para bloquear um utilizador	<i>Back-end</i>
M2 - Módulo de Boleias	Criar ecrã na aplicação móvel para gestão (CRUD) de veículos do utilizador autenticado	<i>Front-end</i>
	Criar rota na API para pedir uma boleia	<i>Back-end</i>
	Criar rota na API para obter os detalhes de um grupo selecionado	<i>Back-end</i>
	Criar rota na API para obter os grupos/boleia disponíveis que correspondam os filtros recebidos por parâmetros de formulário	<i>Back-end</i>
	Criar rota na API para gerir os grupos/boleia do utilizador autenticado	<i>Back-end</i>
	Alterar a rota da API de pedido de boleia de modo a enviar uma notificação ao motorista do grupo	<i>Back-end</i>

	Criar rota na API para definir a boleia atual como terminada	<i>Back-end</i>
	Criar ecrã na aplicação móvel para o mapa principal utilizando o Google Maps SDK	<i>Front-end</i>
	Obter a localização atual através do GPS e apresentar a mesma no ecrã do mapa principal	<i>Front-end</i>
	Abrir a aplicação de mapas predefinida do dispositivo com a rota calculada desde o início da boleia até ao ponto de encontro	<i>Front-end</i>
	Criar botão na aplicação móvel no ecrã do mapa principal para dar início à boleia atual como passageiro	<i>Front-end</i>
	Abrir a aplicação de mapas predefinida do dispositivo com a rota calculada desde o início da boleia até ao destino, com o ponto de encontro como uma paragem pelo caminho	<i>Front-end</i>
	Criar botão na aplicação móvel no ecrã do mapa principal para dar início à boleia atual como motorista	<i>Front-end</i>
	Criar botão na aplicação móvel para finalizar a boleia atual no ecrã de boleias	<i>Front-end</i>
<b>M6 - Módulo de Gestão de Notificações</b>	Adicionar a classe <i>Notification</i> na API com suporte ao FCM	<i>Back-end</i>
	Integrar o FCM na aplicação móvel e configurá-lo para aceitar notificação PUSH	<i>Front-end</i>
	Preparar o serviço Firebase Cloud Messaging (FCM)	<i>Serviços</i>

## 9.3. ANEXO 3 – RESTANTES REQUISITOS FUNCIONAIS LEVANTADOS

Módulo	ID	Descrição	Prioridade
M1 - Módulo de Perfil	RF10	O sistema deverá permitir ao utilizador definir uma viatura como predefinida.	Should Have
	RF11	O sistema deverá permitir ao utilizador alterar a sua imagem de perfil por um avatar pré-definido.	Should Have
	RF95	O sistema deverá permitir ao utilizador selecionar o idioma de apresentação do conteúdo.	Should Have
	RF12	O sistema deverá permitir ao utilizador alterar a imagem de fundo do seu perfil por uma pré-definida.	Could Have
	RF96	O sistema deverá permitir ao utilizar importar e gerir o seu horário escolar do Sistema de Informação do IPS.	Could Have
M2 - Módulo de Boleias	RF25	O sistema deverá permitir ao utilizador visualizar o histórico de viagens com os respetivos detalhes no seu perfil.	Must Have
	RF26	O sistema deverá permitir ao utilizador, como motorista, definir a presença e gerir os passageiros no início e durante a viagem.	Should Have
	RF27	O sistema deverá notificar o passageiro quando removido de um grupo de viagem.	Should Have
	RF100	O sistema deverá notificar o motorista para dar início ao grupo de viagem e marcar os passageiros presentes ao chegar à hora de início do grupo de viagem.	Should Have
	RF29	O sistema deverá notificar os passageiros de um grupo de viagem quando o motorista dá início ao grupo de viagem.	Should Have
	RF30	O sistema deverá permitir ao utilizador, como motorista, dar início a um grupo de viagem.	Should Have
	RF31	O sistema deverá permitir ao utilizador visualizar uma pré-visualização da do grupo de viagem ativo.	Should Have

	RF101	O sistema deverá permitir ao utilizador, como motorista, confirmar a criação de um grupo de viagem que seja no mesmo sentido e que tenha uma proximidade horaria de outro grupo de viagem que esteja a oferecer e ativo.	Should Have
	RF102	O sistema deverá permitir ao utilizador, como passageiro, confirmar a sua presença no grupo de viagem no dia anterior à sua realização.	Should Have
	RF103	O sistema deverá notificar os passageiros de um grupo de viagem no dia anterior à realização do mesmo para confirmarem a sua presença.	Should Have
	RF28	O sistema deverá detetar automaticamente a presença dos membros do grupo de viagem e pedir confirmação ao motorista ao iniciar a viagem.	Could Have
	RF32	O sistema deverá notificar o motorista quando um passageiro partilha a sua localização precisa.	Could Have
	RF33	O sistema deverá permitir ao utilizador, como passageiro, partilhar a sua localização precisa com o motorista para recolha no ponto de encontro.	Could Have
	RF104	O sistema deverá permitir ao utilizador, como passageiro, encontrar um grupo de viagem com uma rota que passe no local definido pelo utilizador.	Could Have
	RF105	O sistema deverá permitir ao utilizador, como motorista, automaticamente criar grupos de viagem com base no seu horário escolar.	Could Have
	RF106	O sistema deverá permitir ao utilizador, como passageiro, encontrar grupos de viagens com base no seu horário escolar.	Could Have
M3 - Módulo de Feedback	RF34	O sistema deverá permitir ao utilizador visualizar a classificação e comentários de motoristas.	Should Have
	RF35	O sistema deverá permitir ao utilizador, como passageiro, visualizar a estimativa de custo máxima e mínima do grupo de viagem.	Should Have

	RF36	O sistema deverá permitir ao utilizador, como motorista, visualizar detalhes de um grupo de viagem ante de dar início ao mesmo.	Should Have
	RF37	O sistema deverá notificar o utilizador que tem um grupo de viagem planeado num futuro próximo.	Should Have
	RF38	O sistema deverá permitir o motorista indicar se foram utilizadas portagens durante uma viagem e o custo das mesmas.	Should Have
	RF39	O sistema deverá permitir ao utilizador classificar outro utilizador (1 a 5 estrelas) após uma viagem.	Should Have
	RF40	O sistema deverá permitir ao utilizador escrever um comentário no perfil de outro utilizador após uma viagem.	Could Have
	RF41	O sistema deverá permitir ao utilizador atribuir elogios de um conjunto pré-definido a outro utilizador após uma viagem.	Could Have
	RF42	O sistema deverá permitir ao utilizador visualizar a classificação e comentários no perfil dos utilizadores.	Could Have
	RF43	O sistema deverá permitir ao passageiro retificar os custos calculados automaticamente pela plataforma e dos extras indicados pelo motorista após uma viagem.	Could Have
	RF44	O sistema deverá permitir ao utilizador, como motorista, marcar como confirmado o pagamento dos passageiros referentes a uma viagem.	Could Have
	RF45	O sistema deverá notificar o passageiro que o motorista confirmou o pagamento referente à viagem realizada.	Could Have
	RF46	O sistema deverá notificar o passageiro de falta de comparência a um grupo de viagem.	Could Have
	RF47	O sistema deverá permitir ao utilizador visualizar os elogios recebidos no seu perfil.	Could Have
M4 - Módulo	RF48	O sistema deverá criar automaticamente um grupo de chat entre os membros de um grupo de viagem.	Must Have

	RF49	O sistema deverá permitir ao utilizador, como motorista, iniciar um chat com um passageiro que tenha realizado um pedido de boleia.	Must Have
	RF50	O sistema deverá permitir ao utilizador dentro de um chat enviar mensagens de texto.	Must Have
	RF51	O sistema deverá permitir ao utilizador visualizar o estado de envio de uma mensagem.	Must Have
	RF52	O sistema deverá notificar o utilizador ao receber uma mensagem de outro utilizador ou grupo de chat, apenas quando a janela de chat não está aberta.	Must Have
	RF53	O sistema deverá notificar o utilizador caso o envio de uma mensagem falhe.	Must Have
	RF54	O sistema deverá permitir ao utilizador remover mensagens enviadas.	Should Have
	RF55	O sistema deverá permitir ao utilizador visualizar a data e hora de quando as mensagens foram processadas e enviadas pelo sistema.	Should Have
	RF56	O sistema deverá permitir ao utilizador, como motorista, enviar uma mensagem privada a um passageiro pertencente ao grupo de chat.	Should Have
	RF57	O sistema deverá permitir ao utilizador, como passageiro, enviar uma mensagem privada ao motorista pertencente ao grupo de chat.	Should Have
	RF58	O sistema deverá permitir ao utilizador visualizar os participantes num grupo de chat, assim como os seus estados (online, última vez online).	Should Have
	RF59	O sistema deverá permitir ao utilizador reenviar uma mensagem em caso de falha.	Could Have
	RF60	O sistema deverá permitir ao utilizador visualizar se e quando uma mensagem foi lida.	Could Have

M5 - Módulo de Gamificação	RF61	O sistema deverá permitir ao utilizador editar mensagens enviadas.	Could Have
	RF62	O sistema deverá permitir ao utilizador visualizar o seu nível, experiência atual e experiência necessária para alcançar o nível seguinte.	Must Have
	RF63	O sistema deverá calcular a experiência necessário para o nível seguinte através de um algoritmo exponencial para o efeito.	Must Have
	RF64	O sistema deverá permitir ao utilizador visualizar todas as conquistas criadas através do BackOffice e respetivo progresso.	Must Have
	RF65	O sistema deverá suportar múltiplos tipos de conquistas, tais como baseadas em nível de utilizador, emissões de CO2 evitadas, número de viagens efetuadas, número de determinados elogios recebidos, etc..	Must Have
	RF66	O sistema deverá recompensar o utilizador com a atribuição de experiência e/ou outros brindes ao alcançar conquistas.	Must Have
	RF67	O sistema deverá atribuir experiência, respeitando às emissões de CO2 e consumos de combustíveis fosseis evitados, ao motorista/passageiro ao concluir uma viagem.	Must Have
	RF68	O sistema deverá dinamizar as recompensas que o utilizador recebe com recurso a <i>loot boxes</i> .	Should Have
	RF69	O sistema deverá permitir ao utilizador visualizar todas as conquistas já concluídas e respetiva(s) recompensa(s) atribuída.	Should Have
	RF70	O sistema deverá permitir ao utilizador visualizar o nível e principais conquistas alcançadas de outros utilizadores ao aceder aos seus perfis.	Should Have
	RF71	O sistema deverá atribuir ao utilizador experiência com base na avaliação (1 a 5 estrelas) recebida após a conclusão de uma viagem.	Should Have

	RF72	O sistema deverá permitir a criação de conquistas sequenciais com recurso a um algoritmo adequado.	Should Have
	RF73	O sistema deverá atribuir um bónus de experiência ao motorista por transportar 3 ou mais passageiros numa viagem.	Should Have
	RF74	O sistema deverá atribuir ao utilizador experiência por cada elogio recebido após a conclusão de uma viagem.	Could Have
M6 - Módulo de Gestão de Notificações	RF76	O sistema deverá permitir ao utilizador aceder à página referente à notificação recebida através do centro de notificações do dispositivo móvel.	Must Have
	RF77	O sistema deverá permitir ao utilizador configurar quais notificações deseja receber e o respetivo método de receção.	Should Have
	RF78	O sistema deverá permitir ao utilizador visualizar o histórico de notificações recebidas.	Should Have
	RF79	O sistema deverá permitir o utilizador definir notificações como silenciosas que serão apenas apresentadas no histórico de notificações.	Could Have
	RF80	O sistema deverá permitir ao utilizador receber notificações via e-mail.	Could Have
M7 - Módulo de Estatísticas	RF81	O sistema deverá permitir ao utilizador visualizar as emissões de CO2 estimadas que foram evitadas com a utilização da plataforma.	Must Have
	RF82	O sistema deverá permitir ao utilizador visualizar o número de quilómetros percorridos evitados com a utilização da aplicação.	Must Have
	RF83	O sistema deverá permitir ao utilizador visualizar os consumos de combustível fosseis estimados que foram evitadas com a utilização da plataforma.	Must Have

M8 - Módulo de Informação	RF84	O sistema deverá permitir ao utilizador visualizar uma estimativa dos gastos monetários dos utilizadores evitados com a utilização da plataforma.	Must Have
	RF85	O sistema deverá permitir ao utilizador visualizar o número de viagens realizadas com recurso à plataforma como motorista e como passageiro.	Must Have
	RF86	O sistema deverá permitir o utilizador visualizar as estatísticas com valores referentes à totalidade do sistema.	Must Have
	RF87	O sistema deverá permitir ao utilizador visualizar o número de pessoas diferentes com quem o utilizador já realizou viagens.	Should Have
	RF88	O sistema deverá permitir ao utilizador visualizar as estatísticas em maior detalhe, pudendo filtrar por ano, mês e semana.	Should Have
	RF89	O sistema deverá permitir ao utilizador visualizar a média de lugares de estacionamento poupadados no campus.	Should Have
	RF90	O sistema deverá permitir o utilizador visualizar as estatísticas com valores referentes ao utilizador em questão.	Should Have
	RF91	O sistema deverá permitir ao utilizador visualizar as notícias publicadas através do BackOffice.	Must Have
	RF92	O sistema deverá permitir ao utilizador visualizar um conjunto de recomendações de segurança em relação ao COVID-19.	Must Have
	RF93	O sistema deverá apresentar ao utilizador um ecrã introdutório acerca da plataforma e quais as vantagens da utilização da mesma.	Should Have
	RF94	O sistema deverá notificar o utilizador ao serem publicadas novas notícias.	Should Have

## 9.4. ANEXO 4 – QUESTIONÁRIO DE TESTES DE USABILIDADE

### CarPoolingByIPS - Questionário de Usabilidade

Este questionário anónimo tem como propósito avaliar a facilidade de utilização, percebida pelo utilizador, da versão de demonstração da aplicação móvel de Carpooling do Instituto Politécnico de Setúbal.

[Seguinte](#)



Página 1 de 4

## CarPoolingByIPS - Questionário de Usabilidade

\*Obrigatório

### Caracterização do Indivíduo

Conjunto de questões relacionadas com a caracterização do utilizador que responder a este questionário.

Qual a sua faixa etária? \*

- 19 ou menos
- 20 - 24
- 25 - 29
- 30 - 34
- 35 - 39
- 40 - 44
- 45 - 49
- 50 - 54
- 55 - 59
- 60 - 64
- 64 - 69
- + 70

Qual o seu estatuto no IPS? \*

- Estudante
- Docente
- Não Docente

A qual unidade orgânica do IPS pertence?

Por favor, ignore esta questão no caso de não ser aplicável ao seu estatuto.

- Escola Superior de Tecnologia de Setúbal (ESTSetúbal/IPS)
- Escola Superior de Educação (ESE/IPS)
- Escola Superior de Ciências Empresariais (ESCE/IPS)
- Escola Superior de Tecnologia do Barreiro (ESTBarreiro/IPS)
- Escola Superior de Saúde (ESS/IPS)

Qual o meio de transporte que costuma utilizar para realizar a deslocação entre o trajeto da sua residência e do Campus do IPS? \*

- Veículo pessoal de 1 ou 2 lugares
- Veículo pessoal de 4 ou mais lugares
- À boleia com um indivíduo pertencente à comunidade IPS
- À boleia com um indivíduo fora da comunidade IPS
- Transportes públicos
- A pé
- Outra: \_\_\_\_\_

No caso de realizar as suas deslocações com veículo pessoal, costuma oferecer boleia a outros membros da comunidade IPS?

Por favor, ignore esta questão no caso de não se deslocar frequentemente utilizando um veículo pessoal.

- Frequentemente
- Às vezes
- Raramente
- Nunca

Qual o seu nível de conhecimento tecnológico? \*

- Baixo
- Médio
- Alto

Conhece aplicações similares à aplicação que se pretende avaliar? \*

Selecionar

[Anterior](#)

[Seguinte](#)

Página 2 de 4

## CarPoolingByIPS - Questionário de Usabilidade

\*Obrigatório

### Avaliação de Usabilidade da Aplicação Móvel

Conjunto de questões que pretendem avaliar a percepção do utilizador em relação à usabilidade da aplicação móvel de Carpooling do IPS.

1 - Penso que gostaria de utilizar esta aplicação com frequência. \*

Opinião	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2 - Considero a aplicação desnecessariamente complexa. \*

Opinião	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3 - Considero a aplicação fácil de utilizar. \*

Opinião	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4 - Penso que necessitaria de auxílio técnico para poder utilizar esta aplicação. \*

Opinião	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5 - Considero que as diversas funcionalidades desta aplicação estão bem integradas. \*

Opinião	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6 - Considero que havia demasiada inconsistência nesta aplicação.\*

	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
Opinião	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7 - Imagino que a maioria das pessoas aprenderia a utilizar esta aplicação muito rapidamente.\*

	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
Opinião	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8 - Achei a aplicação muito complicada de utilizar.\*

	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
Opinião	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9 - Senti-me bastante confiante ao utilizar a aplicação.\*

	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
Opinião	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10 - Precisei de aprender muitas coisas antes de utilizar este sistema.\*

	Discordo Completamente	Discordo	Neutro	Concordo	Concordo Completamente
Opinião	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

[Anterior](#)

[Seguinte](#)

 Página 3 de 4

## CarPoolingByIPS - Questionário de Usabilidade

### Opinião

Gostaríamos de conhecer a sua opinião em maior detalhe em relação à aplicação que acabou de avaliar.

### Aspetos que considerou positivos

A sua resposta

### Aspetos que considerou negativos

A sua resposta

### Observações

A sua resposta

[Anterior](#)

[Submeter](#)

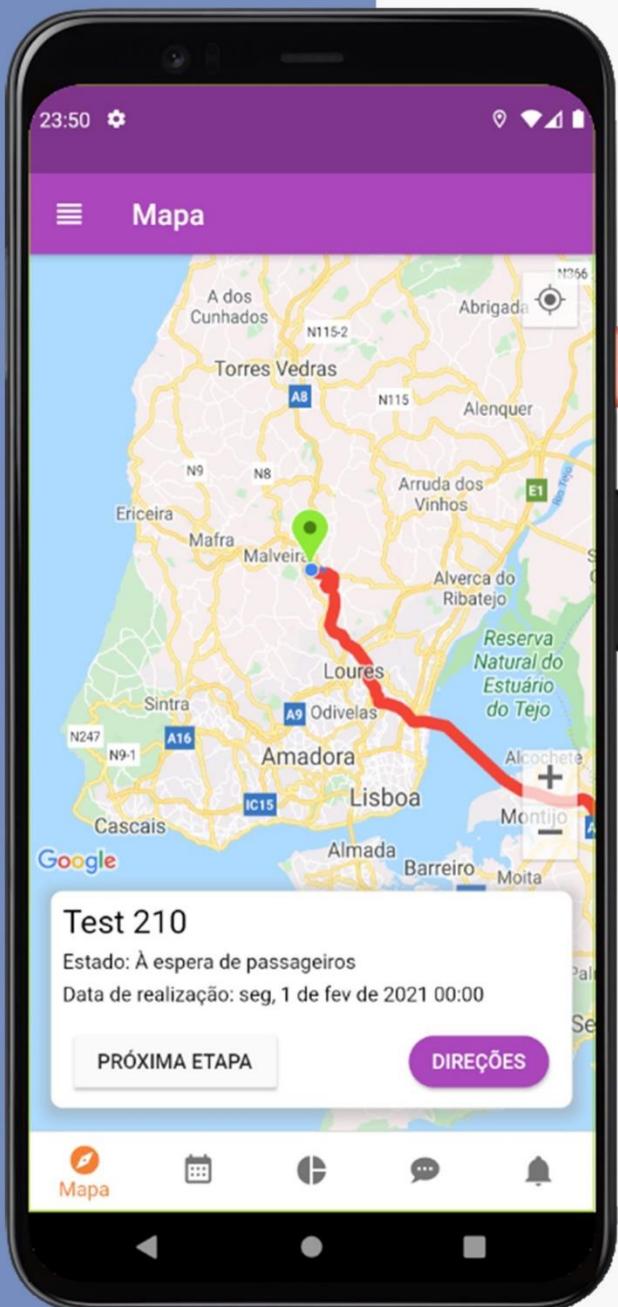


Página 4 de 4

## 9.5. ANEXO 5 – MANUAL DE UTILIZADOR DA APLICAÇÃO MÓVEL

# CarPooling

By



## Manual de Utilizador

Versão 1

# Índice

Introdução.....	3
1. Geral .....	4
1.1. Login.....	5
1.2. Menu Lateral .....	7
1.3. Perfil de Utilizador.....	8
1.4. Definições .....	9
1.5. Gestão de Veículos.....	12
1.6. Criar Veículo .....	14
2. Motorista .....	16
2.1. Mapa .....	17
2.2. Gerir Grupos de Boleia .....	19
2.3. Detalhes de Grupo de Boleia .....	20
2.4. Criar Grupo de Boleia .....	22
2.5. Gerir Pedidos de Boleia .....	28
2.6. Agendamentos .....	30
3. Passageiro .....	31
3.1. Mapa .....	32
3.2. Gerir Grupos de Boleia .....	33
3.3. Gerir Pedidos de Boleia.....	36
3.4. Agendamentos .....	38
3.5. Procurar Boleia .....	40

# Introdução

Este manual tem o objetivo de instruir a comunidade do Instituto Politécnico de Setúbal a utilizar a sua aplicação de gestão e partilha de boleias CarPoolingByIPS.

Descreve as funcionalidades atualmente disponíveis e como as executar de forma simples e ilustrativa, dando a conhecer a aplicação que se espera ter um impacto positivo na sustentabilidade do futuro sustentável do instituto.

# 1. Geral

Funcionalidades  
gerais.

# 1.1. Login

1

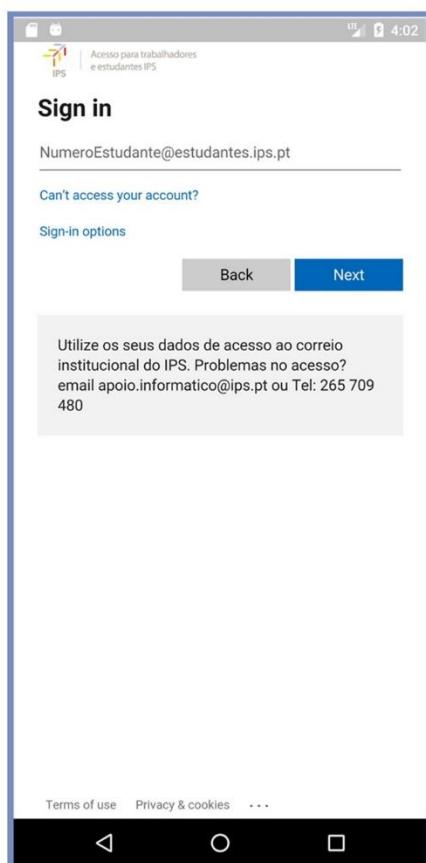
Na primeira execução da aplicação será apresentado um ecrã de autenticação onde será necessário introduzir as suas credenciais de acesso fornecidas pelo Instituto Politécnico de Setúbal. Deverá utilizar as mesmas credenciais de acesso que costuma utilizar para aceder ao Sistema de Informação e à plataforma Moodle.



**Nota:** Poderá ser necessário realizar novamente este processo após um longo período de inatividade.

2

Comece por introduzir o seu email institucional no campo [NumeroEstudante@estudantes.ips.pt](mailto:NumeroEstudante@estudantes.ips.pt) apresentado na imagem ao lado.

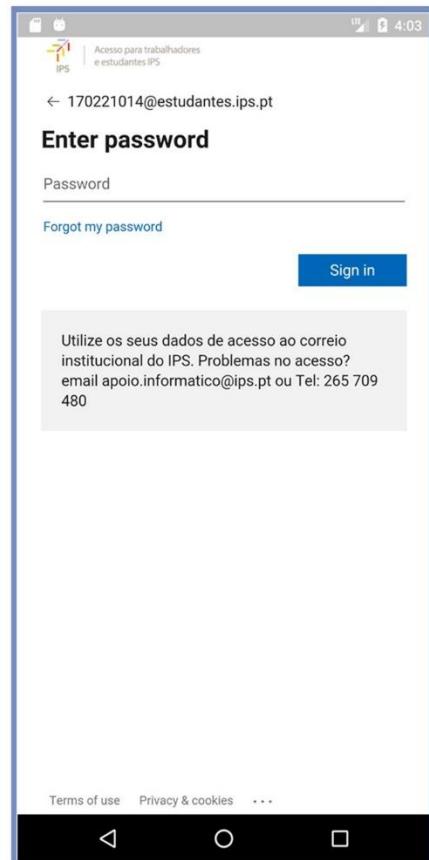


**Nota:** No caso de ser um estudante recomendasse a utilização do endereço de email institucional formado pelo número de estudante em vez de pelo nome.

# 1.1. Login

3

Em seguida introduza a sua palavra-chave no campo correspondente.



4

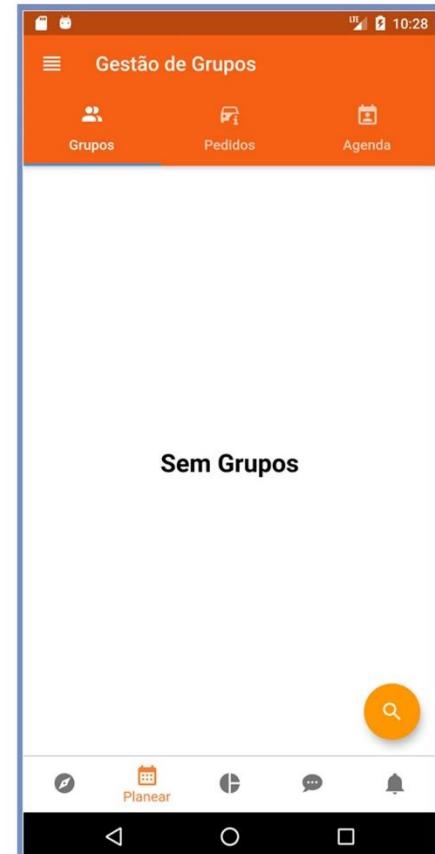
Poderá ser necessário permitir o acesso a um conjunto de permissões requeridas para o correcto funcionamento da aplicação. Ao conceder estas permissões a aplicação terá acesso ao seu endereço de email institucional, ao seu nome e ao seu cargo na instituição. Estes dados serão apenas utilizados para fins de identificação do utilizador perante outros utilizadores pertencentes à mesma instituição.

**Nota:** No caso de encontrar dificuldades no processo de autenticação solicitamos que entre em contacto com a Divisão Informática do Instituto Politécnico de Setúbal.

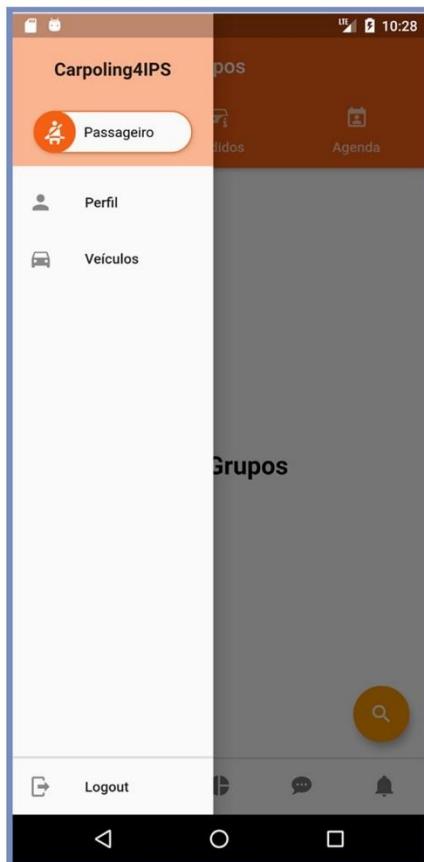
## 1.2. Menu Lateral

1

Para aceder ao menu lateral da aplicação pressione o botão formado por quatro linhas brancas presente no canto superior esquerdo do ecrã.



2



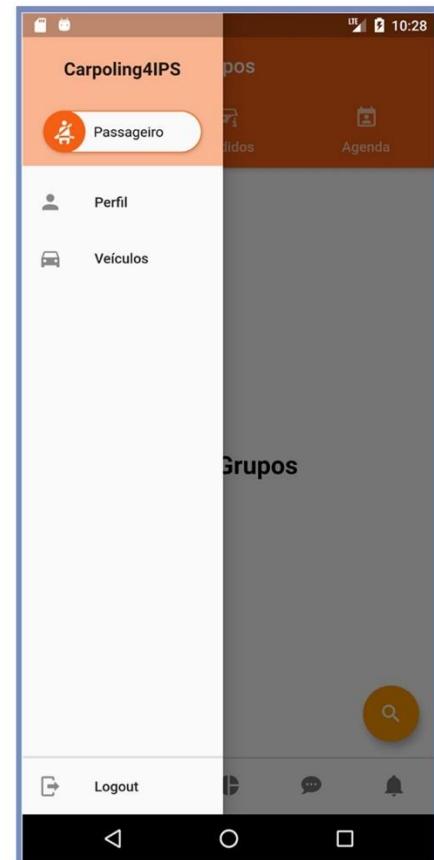
Através do menu lateral é possível aceder ao seu Perfil de utilizador, aos seus Veículos pessoais e desconectar a sua conta de utilizador pressionando o botão "Logout" presente na parte inferior do menu, é ainda possível alterar o modo de funcionamento da aplicação entre Motorista e Passageiro.

**Nota:** A aplicação é apresentada em roxo para o modo de Motorista e em cor-de-laranja para o modo de Passageiro.

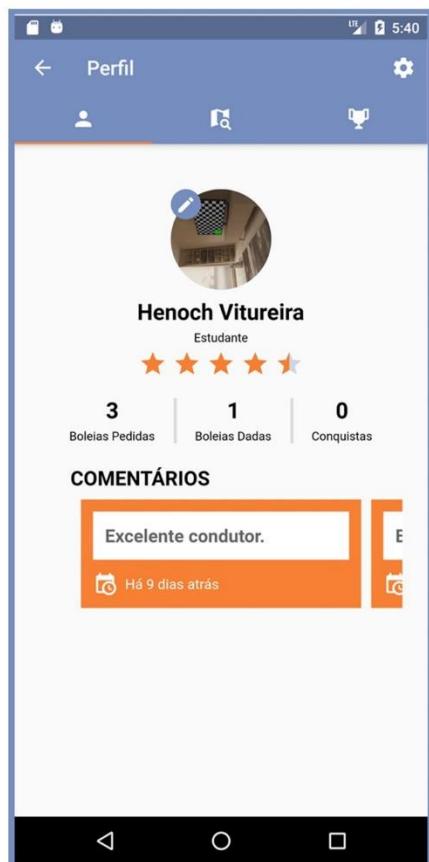
# 1.3. Perfil de Utilizador

1

Para aceder ao seu perfil de utilizador comece por abrir o menu lateral e pressionar o botão "Perfil".



2



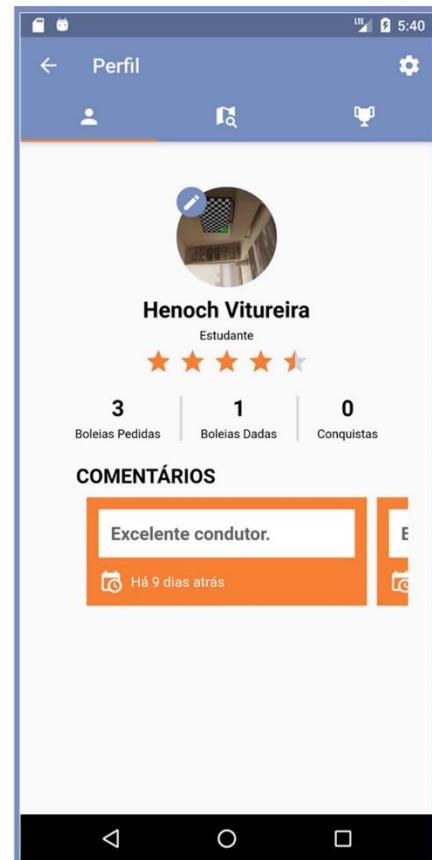
Nesta página são apresentados os seus dados de utilizador carregados automaticamente através do Sistema de Informação do IPS, nomeadamente o seu nome de utilizador e o seu cargo na instituição. A imagem de perfil não é carregado automaticamente por motivos de privacidade, podendo a qualquer momento alterar a mesma clicando sobre a imagem.

**Nota:** Poderá ainda visualizar o número de boleias em que foi passageiro, o número de boleias em que foi motorista, o número de conquistas alcançadas (ainda não disponível) e os comentários deixados por outros utilizadores referentes às boleias dadas.

# 1.4. Definições

1

Para aceder às definições da aplicação comece por abrir o menu lateral e aceder ao seu perfil de utilizador. Em seguida pressione o botão em forma de roda dentada presente no canto superior direito do ecrã de perfil.



2



Na página de definições é possível alterar o idioma da aplicação, ativar o modo escuro (ainda não disponível), gerir os utilizadores bloqueados e visualizar os termos e condições do serviço CarPoolingByIPS.

## 1.4. Definições

3

Através da página de definições ao aceder à opção "Idioma" é apresentada uma nova página com a listagem dos idiomas suportados.

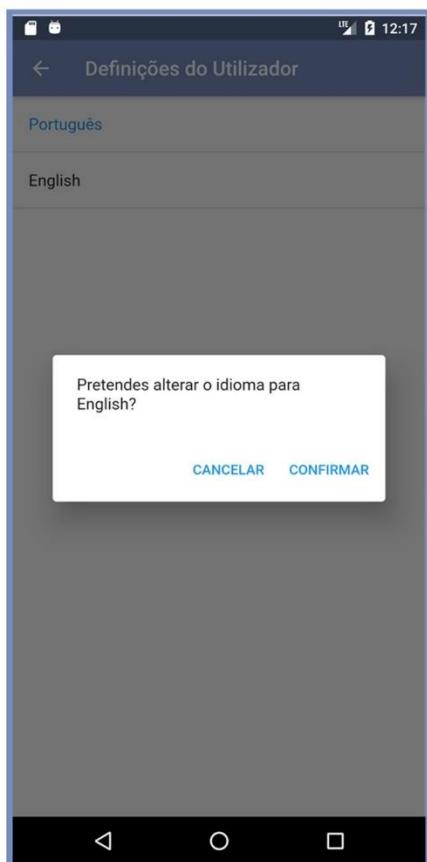


**Nota 1:** Por predefinição a aplicação utiliza o idioma do dispositivo móvel.

**Nota 2:** Neste momento a tradução para Inglês está incompleta.

4

Ao clicar sobre um dos idiomas listados será apresentado um dialogo de confirmação onde será possível cancelar esta ação ou avançar com a alteração do idioma de apresentação da aplicação.



## 1.4. Definições

5

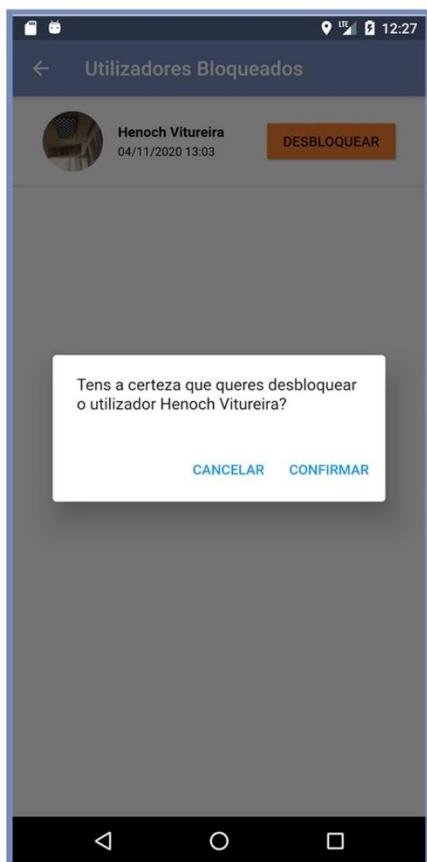
Através da página de definições ao aceder à opção "Utilizadores Bloqueados" é apresentada uma nova página com a listagem dos utilizadores que foram bloqueados pelo utilizador.



**Nota:** Os utilizadores bloqueados não serão apresentados nas listagens de passageiros nem ao procurar por boleias.

6

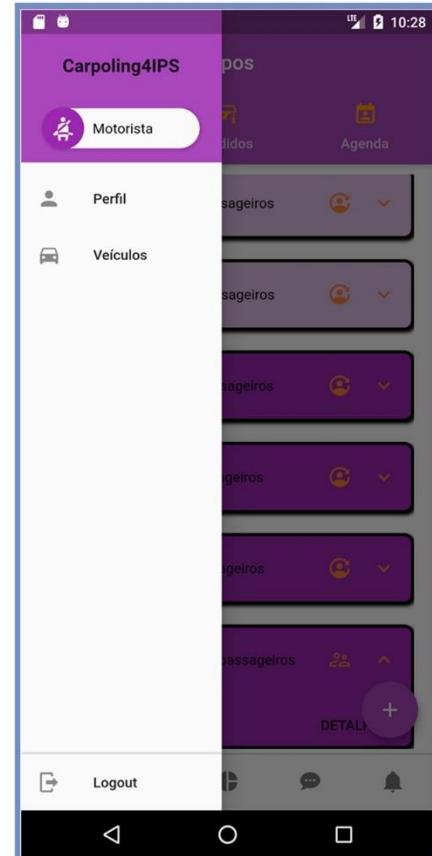
Para desbloquear um utilizador basta clicar no botão cor-de-laranja com o texto "Desbloquear" apresentado a seguir ao nome do utilizador bloqueado. Será necessário confirmar a ação no diálogo de confirmação apresentado.



# 1.5. Gestão de Veículos

1

Para gerir os seus veículos pessoais comece por abrir o menu lateral e pressionar o botão "Veículos".



2



Nesta página são listados os seus veículos pessoais que adicionou anteriormente à aplicação.

Ao clicar sobre a seta cor-de-laranja presente na extremidade direita de cada veículo listado é possível aceder ao botão de edição e remoção do veículo.

# 1.5. Gestão de Veículos

3

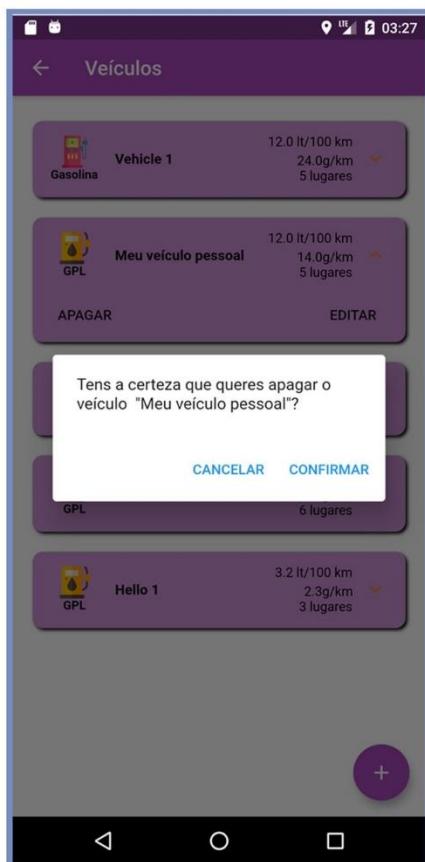
Ao pressionar o botão "Editar" de um veículo será aberto um dialogo um onde será possível introduzir o novo nome do veículo. Após a introdução do novo nome poderá confirmar a alteração do mesmo pressionado o botão "Confirmar" ou cancelar a edição do veículo pressionando o botão "Cancelar".



**Nota:** O nome dado ao veículo é apenas para referencia pessoal e não será apresentado a outros utilizadores.

4

Ao pressionar o botão "Apagar" de um veículo será apresentado um dialogo para confirmar a remoção do veículo pessoal da sua conta de utilizador. Para confirmar esta ação basta pressionar o botão "Confirmar".



**Nota:** Não é possível apagar um veículo que esteja a ser utilizado num grupo de boleia. Neste caso será necessário primeiro apagar o grupo de boleia.

# 1.6. Criar Veículo

1

Para criar um veículo comece por aceder à página de gestão de veículos e em seguida pressione botão "+" presente no canto inferior direito da página.



2



Na página de criação de veículo são apresentados os seguintes campos a serem preenchidos: nome do veículo; tipo de combustível (gasolina, gasóleo, GPL ou elétrico); consumo em litros por 100 km ou, no caso de veículos elétricos, em quilowatt-hora por 100 km; emissões de CO<sub>2</sub> em gramas por km (não aplicável a veículos elétricos); número de lugares disponíveis.

**Nota 1:** O nome dado ao veículo é apenas para referência pessoal e não será apresentado a outros utilizadores.

**Nota 2:** O número de lugares disponíveis representa todos os lugares que podem ser ocupados pelos passageiros de uma boleia dada.

# 1.6. Criar Veículo

3

Para confirmar a criação do veículo pressione o botão com um ícone de uma disquete presente no canto inferior direito da página. No caso de terem sido inseridos dados inválidos será apresentada uma mensagem a vermelho por baixo do campo em questão informando qual o erro.



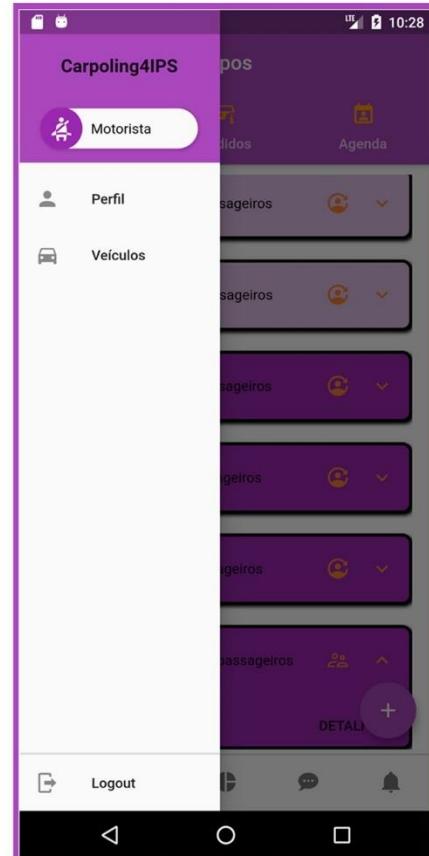
# 2. Motorista

Funcionalidades  
específicas do modo  
de motorista.

## 2.1. Mapa

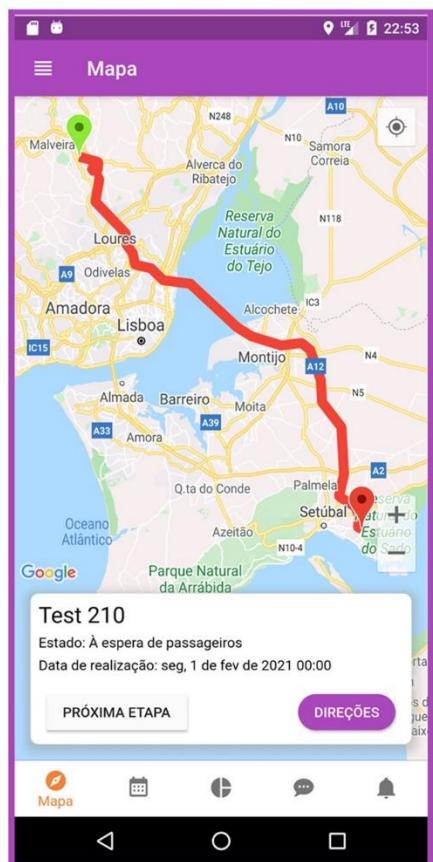
1

Comece por confirmar se a aplicação está no modo de Motorista. No caso de não estar, abra o menu lateral e pressione o botão com o texto "Passageiro" para mudar o modo da aplicação para Motorista.



Nota: Poderá facilmente identificar o modo em que a aplicação está pela cor da mesma. A cor roxa está associada com o modo de Motorista e a cor cor-de-laranja está associada com o modo de Passageiro.

2



Selecione a primeira opção, com o nome de "Mapa", no menu inferior da aplicação. Nesta secção da aplicação é apresentado um mapa com a sua localização atual com um marcador azul e caso tenha uma viagem atualmente em progresso ou para começar num futuro próximo a mesma será apresentada, tal como visível na imagem ao lado.

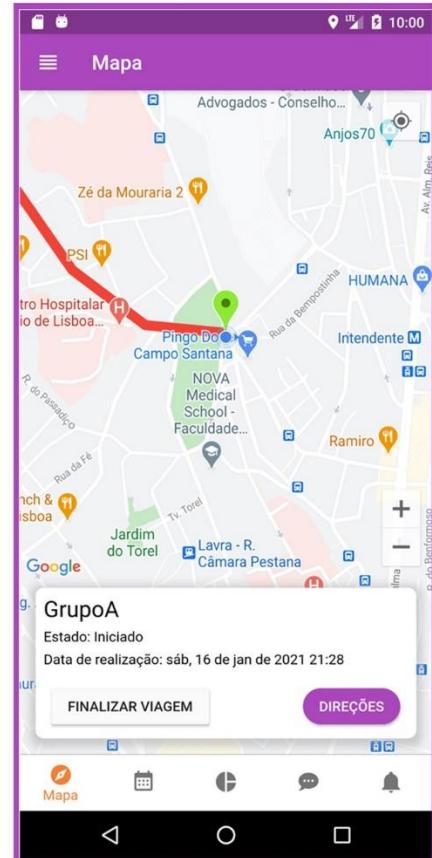
Nota: Existem 3 estados possíveis para uma viagem: agendado, à espera de passageiros e iniciado.

## 2.1. Mapa

3

Ao faltar 15 minutos para a data de realização de uma viagem a mesma será apresentada neste ecrã com o estado de agendado. O motorista poderá avançar para o próximo estado da viagem pressionando o botão "Próxima Etapa" ou "Finalizar Viagem". Ao clicar sobre o botão "Direções" é aberta a aplicação de navegação predefinida do dispositivo móvel com as coordenadas de destino da viagem.

Nota: O motorista deverá apenas avançar do estado "agendado" para o estado "à espera de passageiros" quando chegar ao local definido como ponto de encontro na criação do grupo de boleia.



## 2.2. Gerir Grupos de Boleia

1

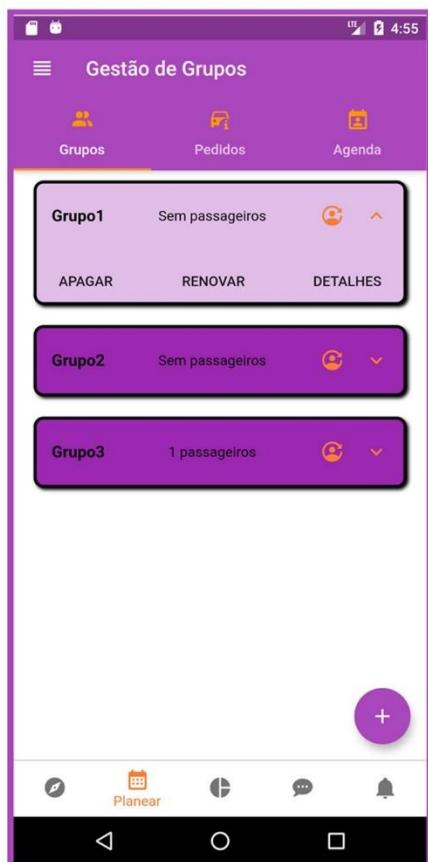
Para aceder ao ecrã de listagem dos grupos de boleia dos quais é motorista comece por mudar o modo da aplicação para Motorista, em seguida selecione a segunda opção, com o nome de "Planear", no menu inferior da aplicação, por fim selecione o separador "Grupos" na parte superior da janela.



Nota: O ícone a cor-de-laranja presente do lado esquerdo da seta de cada grupo de boleia representa a recorrência do grupo.

2

Ao clicar sobre o cartão de um dos grupos de boleia o mesmo será expandido, apresentando a listagem dos passageiros do grupo, caso disponível, e um conjunto de 3 botões, através dos quais é possível "Apagar" o grupo, "Renovar" o grupo quando expirado e é recorrente e aceder aos "Detalhes" do grupo.



Nota 1: Um grupo de boleia recorrente deverá ser renovado semanalmente.

Nota 2: É possível identificar um grupo de boleia expirado por ser listado com uma cor mais clara em relação aos restantes grupos.

## 2.3. Detalhes de Grupo de Boleia

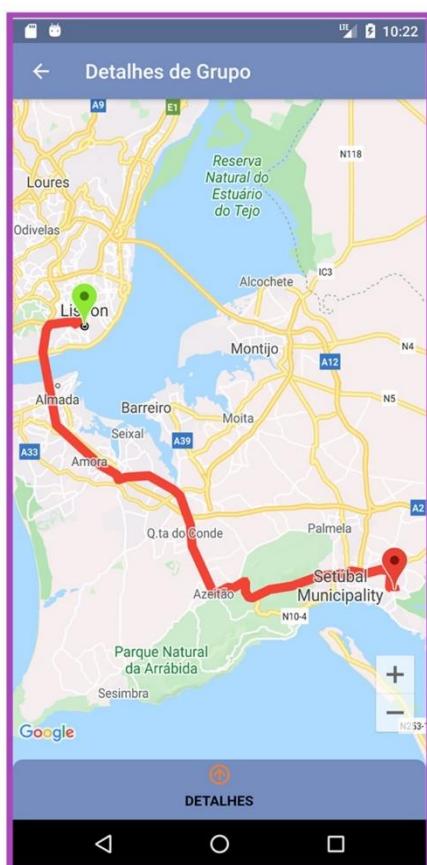
1

Para aceder aos detalhes de um grupo de boleia comece por aceder ao ecrã de listagem de grupos, expanda o cartão do grupo que pretende visualizar os detalhes e pressione o botão "Detalhes".



2

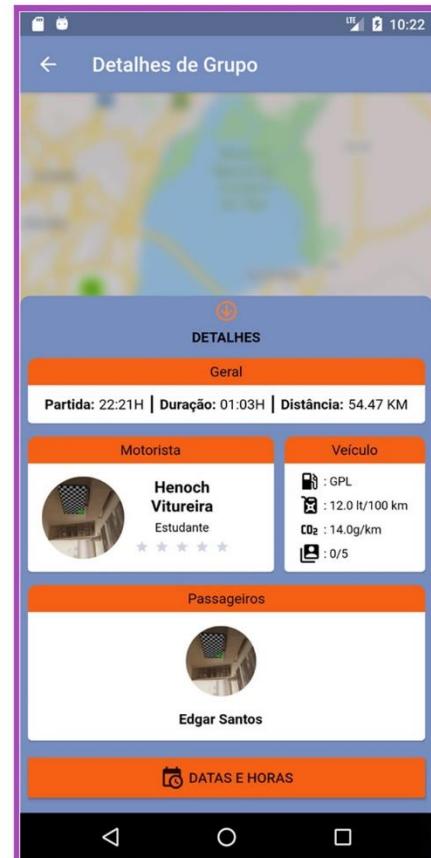
Na janela de detalhes de um grupo de boleia é apresentado um mapa com a rota de menor distância do ponto de encontro (marcador verde) até ao destino (marcador vermelho).



## 2.3. Detalhes de Grupo de Boleia

3

Para visualizar mais detalhes sobre o grupo de boleia pressione o botão "Detalhes" na parte inferior da janela. Nesta aba são apresentados os dados do grupo, tal como a hora de partida, a duração e a distancia do melhor trajeto calculo, o motorista, os dados do veiculo e a listagem de passageiros.



4

Sendo que um grupo de boleia por si só não corresponde a uma viagem, o mesmo tem agendamentos associados que representam uma viagem para um determinado dia e hora. Para visualizar os agendamentos associados ao grupo pressione o botão "Datas e horas" presente na parte inferior da janela de detalhes. Neste dialogo poderá visualizar a que dias da semana as viagens deste grupo se realizam e quais os próximos agendamentos do grupo.



Nota: Os dias da semana em que o grupo realiza viagens são apresentados com fundo azul, enquanto que os restantes são apresentados a cinzento.

## 2.4. Criar Grupo de Boleia

1

Para criar um novo grupo de boleia comece por aceder ao ecrã de listagem de grupos e pressione o botão "+" no canto inferior direito.



2

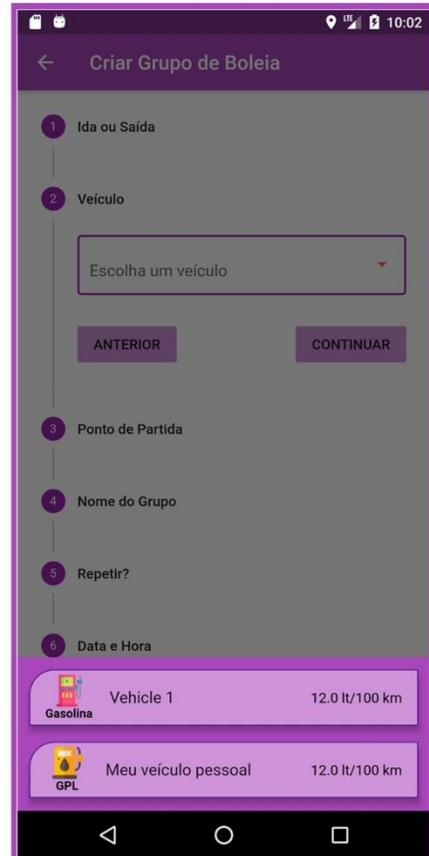


Nesta janela é apresentado um formulário separado em 7 partes para a criação de um grupo de boleia. No primeiro ponto deverá indicar se este grupo de boleia será de "Ida" para o IPS ou de "Saída" do IPS, ou seja, de regresso para o ponto de encontro. Neste ponto deverá ainda indicar se o trajeto calculado pela aplicação deverá evitar a passagem por portagens quando possível.

## 2.4. Criar Grupo de Boleia

3

No segundo ponto deverá clicar sobre o campo "Escolha um veículo" e selecionar o veículo que será utilizado para as viagens deste grupo de boleia na listagem apresentada.

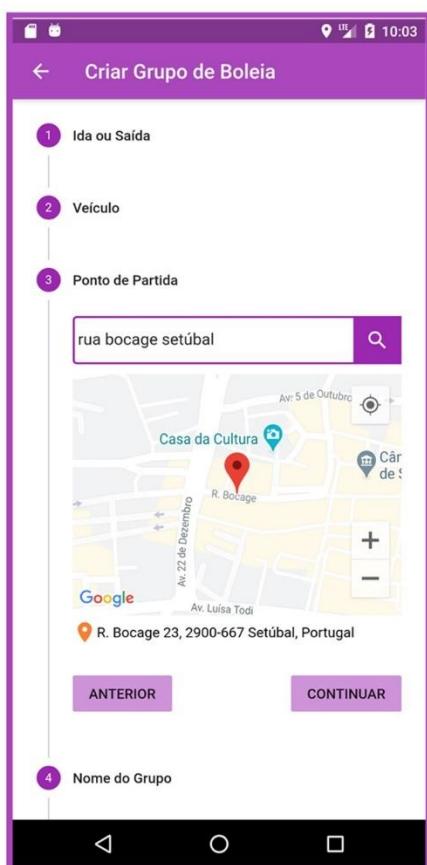


Nota 1: Não será possível mudar o veículo associado ao grupo de boleia posteriormente.

Nota 2: Na listagem são apresentados apenas 2 veículos por vez, no entanto, é possível arrastar para cima a listagem de modo a visualizar os restantes veículos.

4

No terceiro ponto deverá indicar o ponto de encontro pretendido para o grupo de boleia. Por predefinição é selecionada a localização atual do utilizador, mas é possível selecionar um novo ponto de encontro interagindo com o mapa ou utilizando a caixa de pesquisa presente no ecrã.



Nota: Considera-se como ponto de encontro a localização onde o motorista apanha ou larga todos os passageiros.

## 2.4. Criar Grupo de Boleia

5

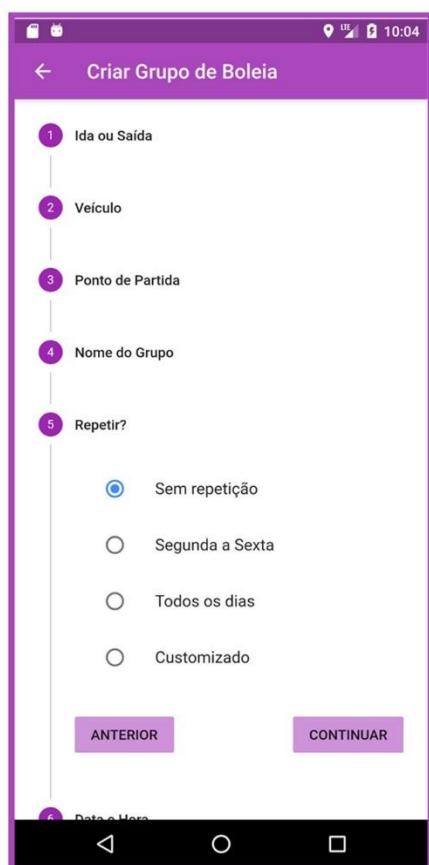
No quarto ponto deverá indicar o nome do grupo de boleia pelo qual será identificado pelos restantes utilizadores da aplicação.



Nota: Um motorista não pode ter grupos de boleia com nomes duplicados.

6

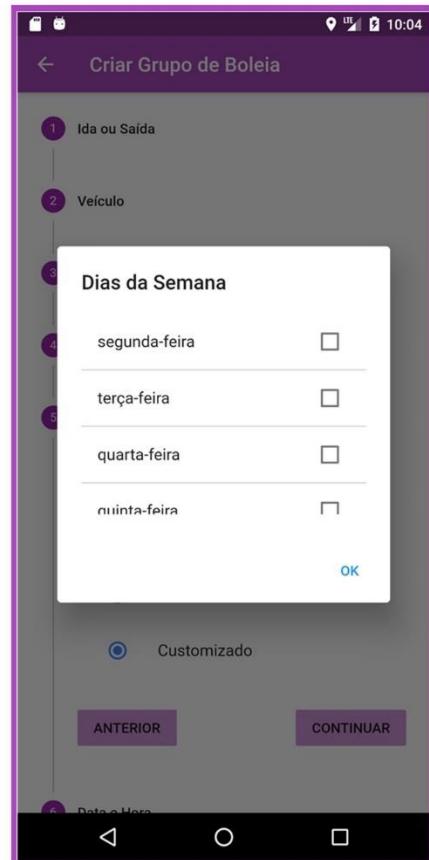
No quinto ponto deverá indicar se é um grupo de boleia recorrente ou de realização única. A opção "Sem repetição" permite criar um grupo de boleia que corresponde apenas a uma viagem e após o agendamento dessa viagem é automaticamente desativo, enquanto que as restantes opções permitem criar um grupo de boleia que irá se repetir todas as semanas de acordo com a opção selecionada.



## 2.4. Criar Grupo de Boleia

7

Ainda no quinto ponto, no caso de selecionar a opção "Customizado" será apresentado um dialogo onde poderá selecionar especificamente em quais dias da semana este grupo de boleia deverá gerar agendamentos de viagens.



8

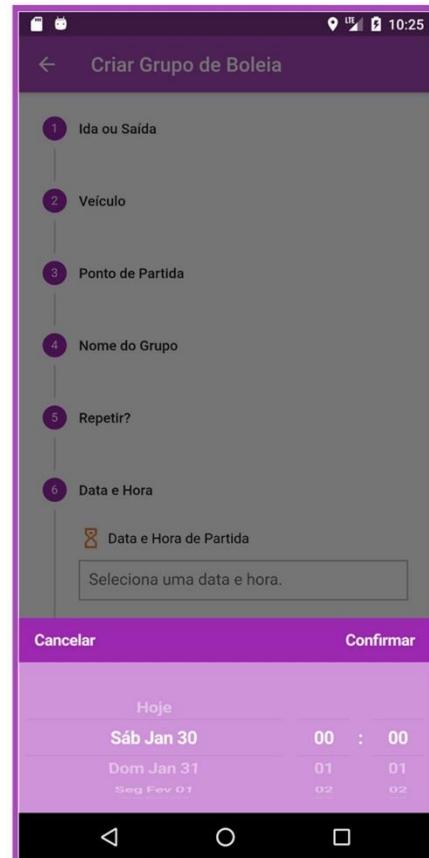
No sexto ponto deverá indicar a hora de partida para as viagens agendadas para este grupo de boleia.



## 2.4. Criar Grupo de Boleia

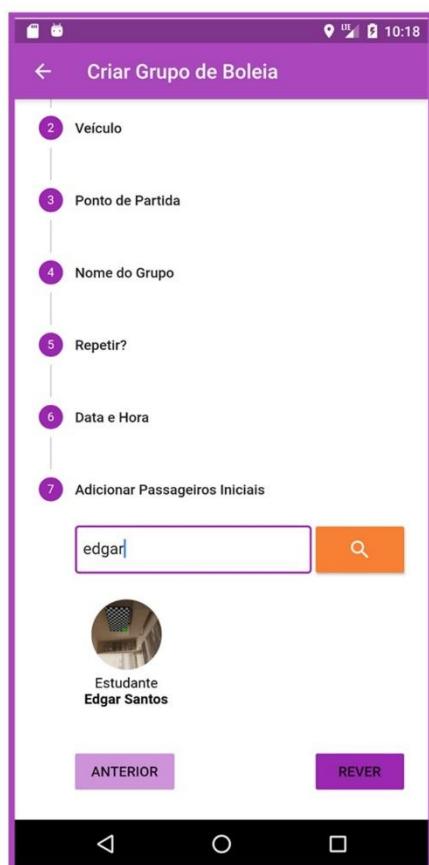
9

Ainda no sexto ponto, no caso de ter selecionado a opção "Sem repetição" será pedido para indicar, para além da hora, o dia em que esta viagem em particular se irá realizar.



10

No sétimo ponto deverá indicar quais os passageiros permanentes que irão fazer parte deste grupo de boleia. Deverá utilizar o campo de pesquisa para procurar pelo primeiro nome, último nome, cargo na instituição ou endereço de e-mail institucional do utilizador.

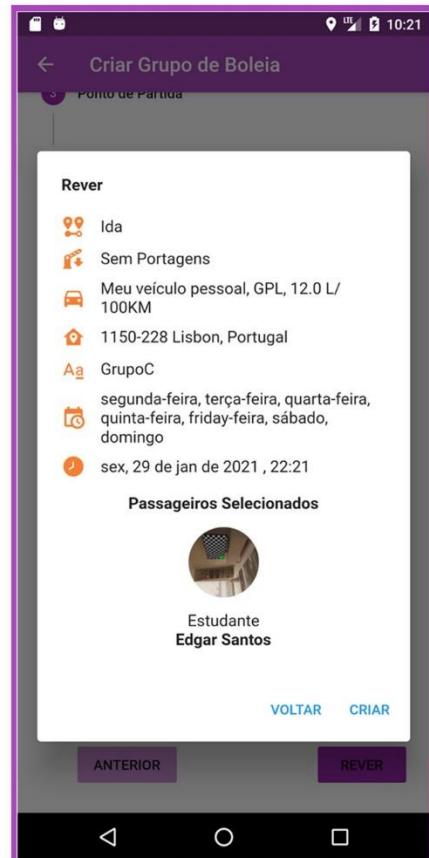


Nota 1: Os passageiros adicionados diretamente pelo motorista a um grupo de boleia irão automaticamente participar em todos os agendamentos do mesmo.  
Nota 2: Neste momento ainda não é possível editar os passageiros permanentes posteriormente à criação do

## 2.4. Criar Grupo de Boleia

11

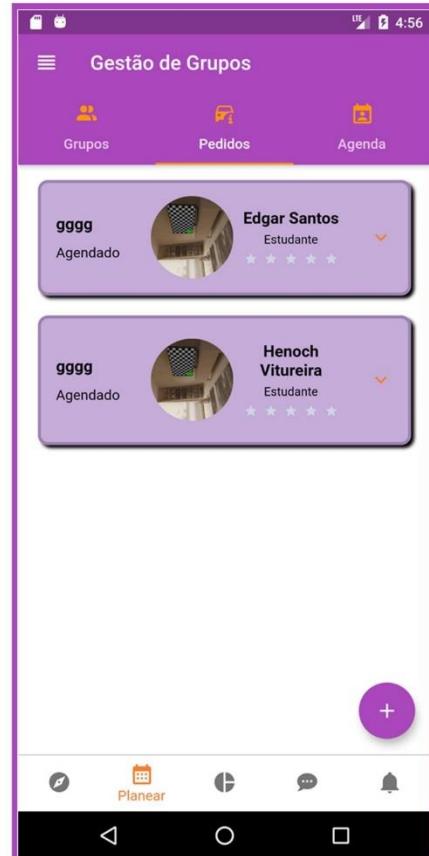
Após o correto preenchimento de todos os campos do formulário poderá pressionado o botão "Rever" apresentado no sétimo ponto do formulário. Ao pressionar este botão será apresentado um dialogo com um resumo dos dados preenchidos no formulário para revisão, no caso de todos os dados estarem corretos poderá avançar com a criação do grupo de boleia pressionando o botão "Criar".



## 2.5. Gerir Pedidos de Boleia

1

Para aceder ao ecrã da pedidos de boleia realizados por passageiros comece por mudar o modo da aplicação para Motorista, em seguida selecione a segunda opção, com o nome de "Planear", no menu inferior da aplicação, por fim selecione o separador "Pedidos" na parte superior da janela.



2

Nesta ecrã poderá visualizar uma listagem de cartões com todos os pedidos de boleia realizados por passageiros. Por cada pedido de boleia é possível visualizar o nome do grupo de boleia correspondente, o estado da viagem, a identificação do passageiro que realizou o pedido e a classificação de 0 a 5 estrelas do passageiro.

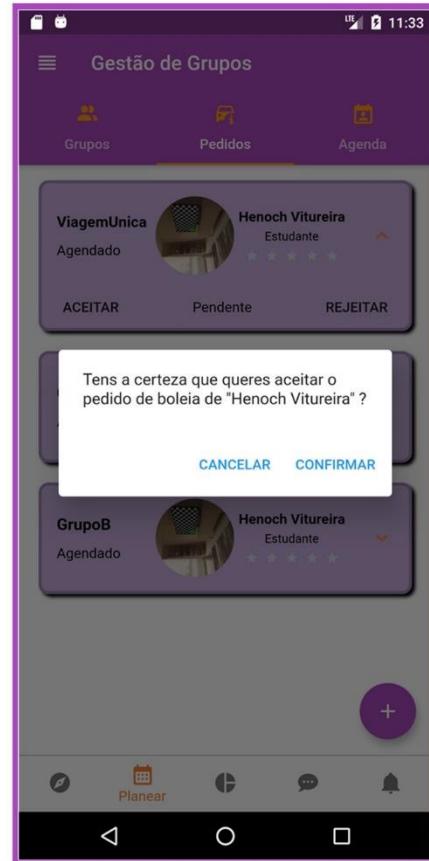


## 2.5. Gerir Pedidos de Boleia

3

Ao clicar sobre o cartão de um dos pedidos de boleia o mesmo será expandido, apresentando o estado atual do pedido e, caso o pedido esteja pendente, um conjunto de 2 botões, através dos quais é possível "Aceitar" ou "Rejeitar" o pedido de viagem realizado pelo passageiro.

Nota: Um pedido de viagem pode ter os seguintes estados:  
pendente, aceite ou rejeitado.



## 2.6. Agendamentos

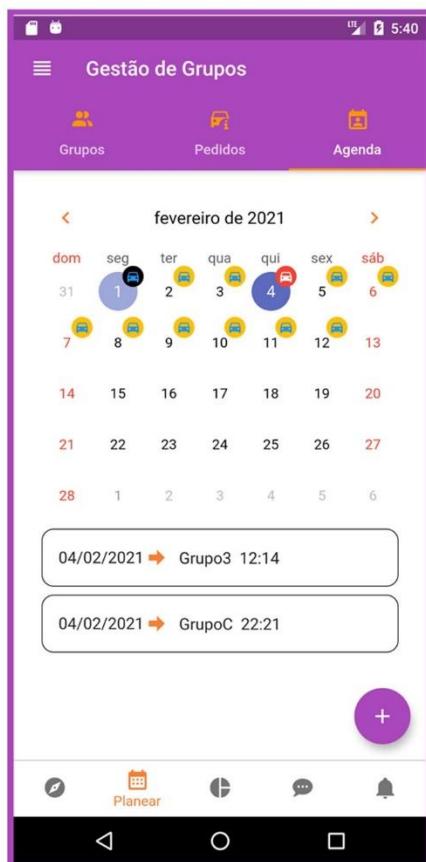
1

Para aceder ao calendário dos próximos agendamentos de viagens em que é motorista comece por mudar o modo da aplicação para Motorista, em seguida selecione a segunda opção, com o nome de "Planear", no menu inferior da aplicação, por fim selecione o separador "Agenda" na parte superior da janela.



2

Neste ecrã poderá visualizar um calendário com os dias dos próximos agendamentos de viagem como motorista marcados a amarelo. Ao pressionar sobre um dia dos calendários será apresentado por baixo uma listagem dos agendamentos para esse dia com a identificação do nome do grupo de boleia e a hora da viagem.



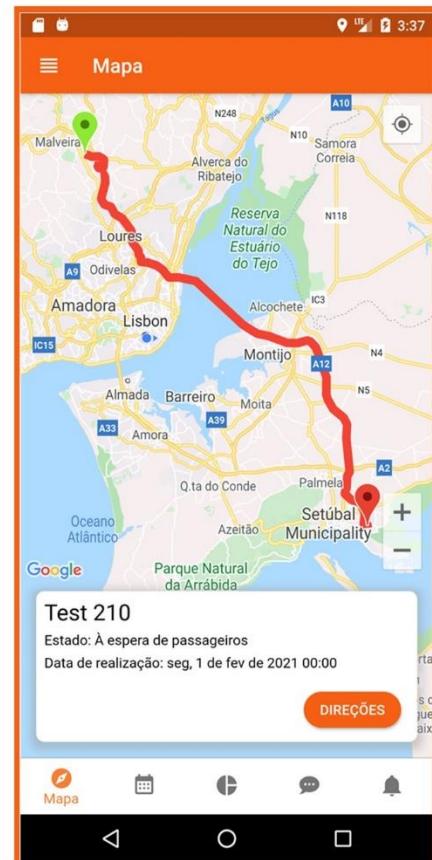
# 3. Passageiro

Funcionalidades  
específicas do modo  
de passageiro.

### 3.1. Mapa

1

Certifique-se de que se encontra no modo Passageiro e clique no primeiro ícone da barra de navegação inferior caso já não tenha. Caso tenha uma viagem como Passageiro 15 minutos após a data e hora atual, nesta página consegue visualizar um mapa interativo com a rota da sua boleia. Abaixo consegue visualizar um quadro em branco que irá conter o nome do grupo de boleia que realiza a viagem em que se encontra, o seu estado e data e hora de partida. Os estados poderão ser "Agendado", "À espera de passageiros" e "Iniciado".



2

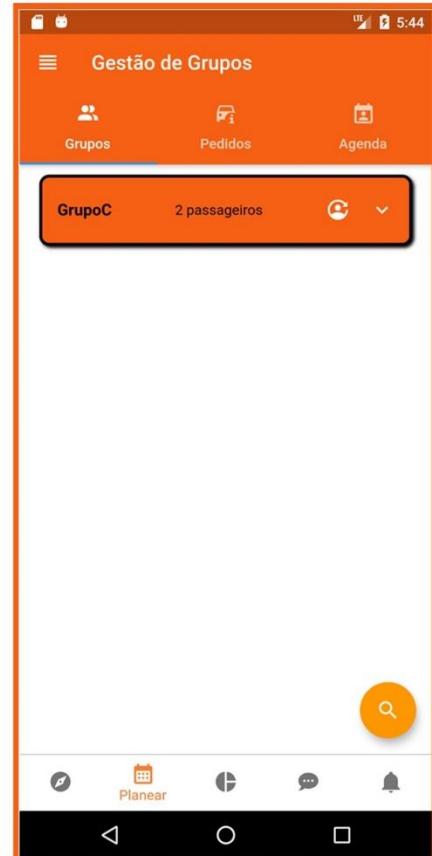


Caso não tenha uma viagem ativa, aparecerá uma mensagem com essa informação e o mapa não terá rota desenhada nem quadro de estado. Se clicar no botão cor-de-laranja "DIREÇÕES" o seu dispositivo irá abrir a sua aplicação de mapas definida por defeito com o seu ponto de destino, no qual pode acionar a opção para seguir as direções até mesmo.

## 3.2. Gerir Grupos de Boleia

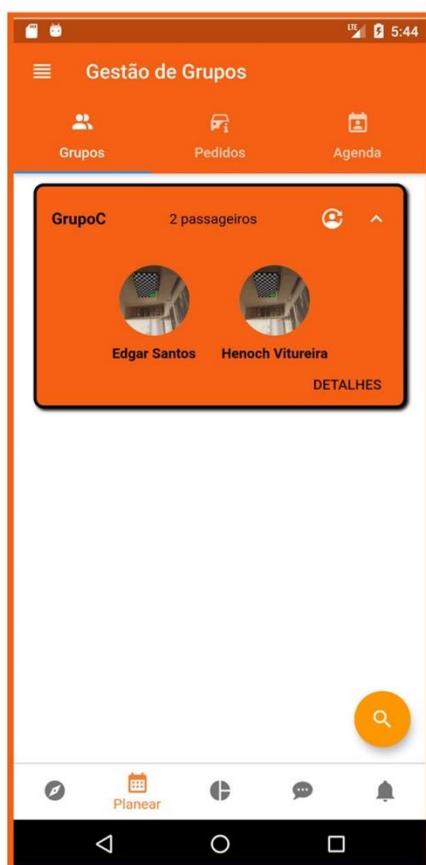
1

Certifique-se de que se encontra no modo Passageiro e clique no segundo ícone da barra de navegação inferior. Agora pode visualizar os grupos de boleia de que é passageiro, representados como cartões expansíveis. Pode visualizar os seus nomes, número de passageiros e tipo de recorrência através do ícone à esquerda do botão de expansão (o que se encontra à direta dos cartões).



2

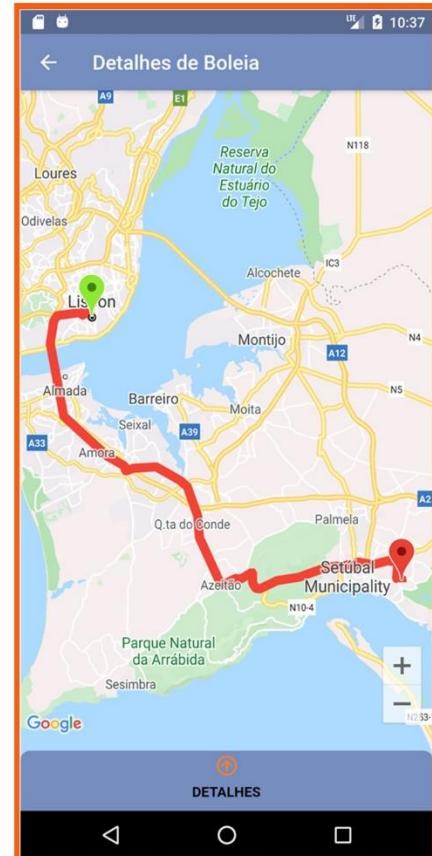
Clique no ícone mais à direta de um cartão para o expandir. Após expandir um cartão, pode visualizar os passageiros que fazem parte do grupo e o botão de acesso aos detalhes do mesmo.



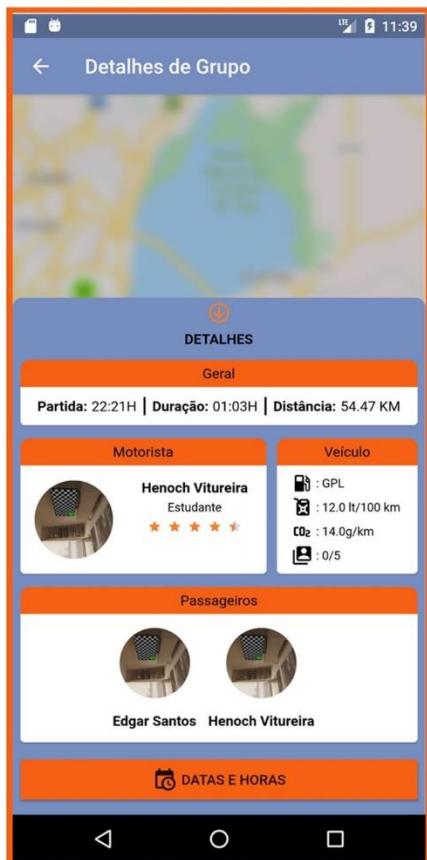
## 3.2. Gerir Grupos de Boleia

3

Se clicar no botão "DETALHES", irá ter acesso à janela de detalhes do grupo de boleia correspondente. No ecrã de detalhes é possível visualizar um mapa com a rota da viagem que esse grupo realiza, com o seu ponto de partida a verde, e o ponto de chegada a vermelho. Este mapa é interativo, o que significa que pode ser navegado e ampliado. Abaixo é possível ver uma aba com um ícone de uma seta a apontar para cima, se clicar no mesmo, pode visualizar informação mais detalhada acerca do grupo.



4



Se clicar ou arrastar para cima a área do ícone da aba de detalhes, a aba irá abrir e revelar os detalhes acerca da viagem feita pelo grupo, nomeadamente, a hora de partida, duração e distância percorrida. Também pode visualizar a informação e foto do motorista e do veículo a ser utilizado, como o tipo de combustível, consumo, emissões e lugares disponíveis. Em baixo são visíveis os passageiros atuais do grupo e um botão que abre informação sobre as datas das próximas viagens.

## 3.2. Gerir Grupos de Boleia

5

Após clicar no botão mais abaixo irá conseguir visualizar um diálogo. Na parte superior constam círculos que representam os dias da semana, caso estejam a azul, é porque a viagem do grupo repete-se nesse dia. Em baixo pode ver um calendário interativo que apresenta um ícone amarelo nos dias das próximas viagens do grupo. O calendário suporta gestos como clicar no seu cabeçalho ou arrastamento para consultar viagens outros meses. Pode fechar o calendário no botão "FECHAR" ou a clicar no exterior.



### 3.3. Gerir Pedidos de Boleia

1

Para aceder ao painel de gestão dos seus pedidos de viagem/boleia, certifique-se de que se encontra no modo Passageiro, clique no segundo ícone da barra de navegação inferior e na segunda opção da barra superior, "Pedidos". Aqui pode visualizar o estado dos seus pedidos através de cartões expansíveis. À esquerda pode ver o nome do grupo de boleia de que pediu viagem e o estado em que a viagem se encontra, como por exemplo "Agendado". À direita é possível visualizar a foto, nome, título e classificação do motorista do grupo.



2

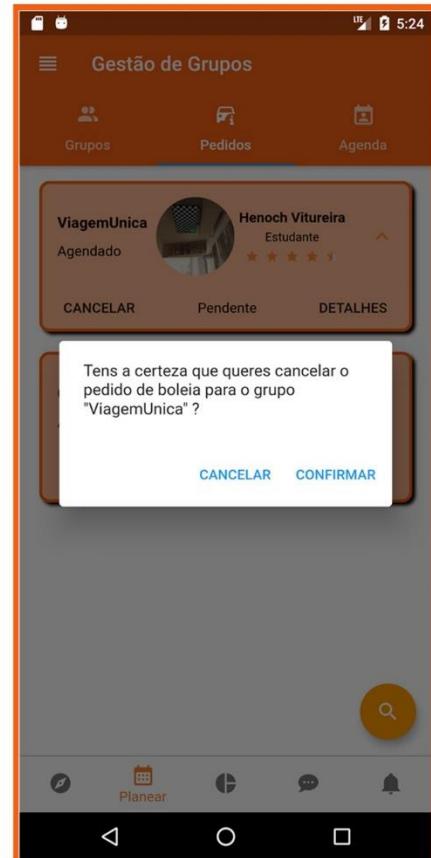
Os estados do seu pedido de boleia podem ser "Agendado", "À espera de passageiros" e "Iniciado". Após expandir um cartão ao clicar no botão mais à direita do mesmo, no centro, pode visualizar o estado do seu pedido, que pode ser "Aceite" ou "Rejeitado". À direita encontra-se o botão "DETALHES", que abre a janela de detalhes do pedido de boleia. Caso um pedido seu seja rejeitado, perde acesso aos botões disponíveis no cartão.



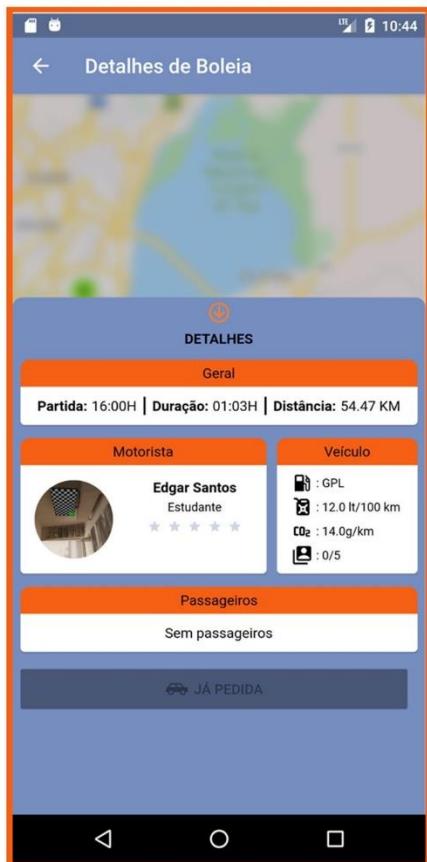
### 3.3. Gerir Pedidos de Boleia

3

Ao clicar no botão "CANCELAR" é apresentado um diálogo de confirmação do cancelamento do pedido. Esta funcionalidade neste momento não tem efeito, pois faz parte duma próxima versão da aplicação.



4



Ao clicar no botão "DETALHES" é apresentada o mesmo género de janela de detalhes que no painel de gestão de grupos mas com destino a detalhar a viagem do pedido de boleia em questão. Os detalhes são os mesmos que o com a diferença de que é não é possível ver o botão de datas, e existe um para pedir boleiam, que se encontra desabilitado, pois como faz parte dos nossos pedidos, já foi pedida.

## 3.4. Agendamentos

1

Para aceder ao painel de agenda de viagens/boleias, certifique-se de que se encontra no modo Passageiro, clique no segundo ícone da barra de navegação inferior e na terceira opção da barra superior, "Agenda". Neste painel pode ver um calendário que destaca o dia atual. Todos os dias que contêm um ícone amarelo com um carro azul são os dias em que há viagens para o utilizador (como visto na legenda). Caso haja alguma viagem no dia atual, o ícone é vermelho em vez de amarelo. O calendário é interativo, pelo que permite navegação por meses e dias ao clicar à esquerda ou direita no seu cabeçalho, ou arrasto para um dos lados.



2

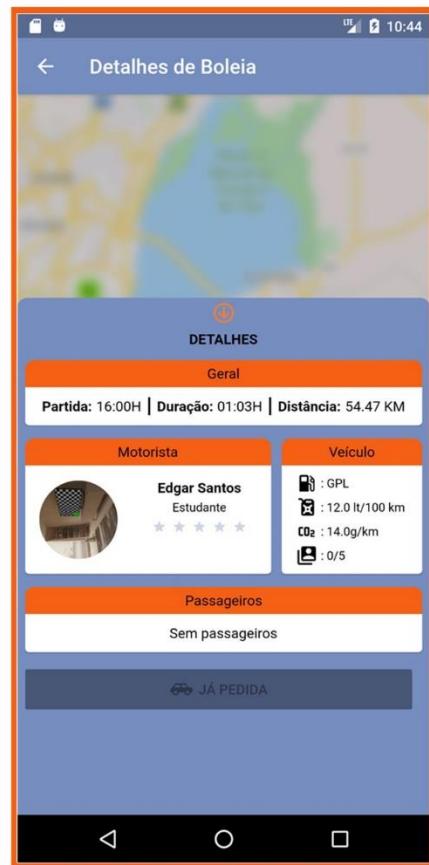
Se clicar em algum dos dias do calendário em que há boleias, aparecerá uma lista abaixo em que cada item corresponde a uma viagem nesse mesmo dia. Cada item da lista contém à esquerda a data da viagem, e à direita, o nome do grupo que realiza essa viagem, e a hora de partida.



### 3.4. Agendamentos

3

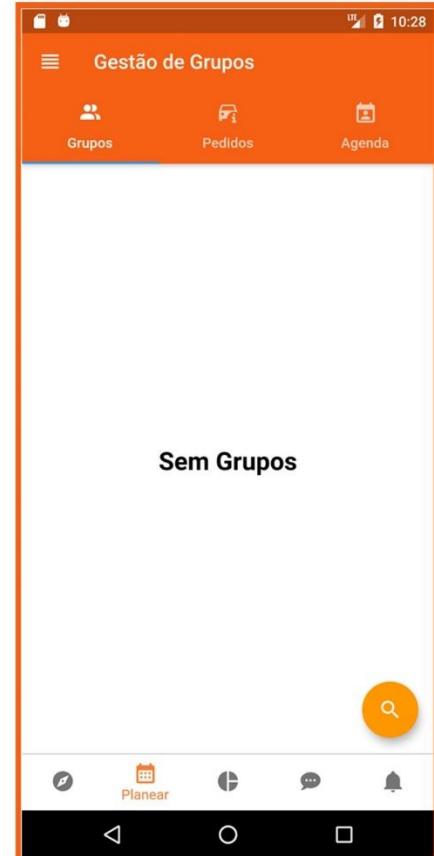
.Se clicar na parte esquerda de um item da lista, é aberta a mesma janela de detalhe que no painel de gestão dos seus pedidos de boleia.



### 3.5. Procurar Boleia

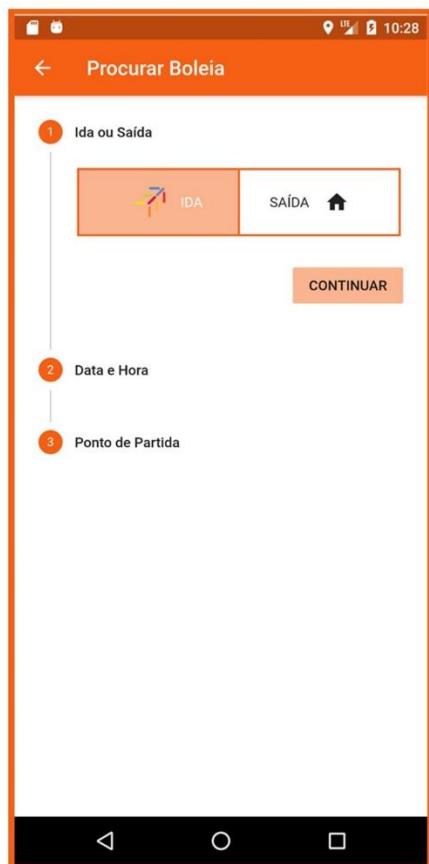
1

Para poder procurar por uma viagem certifique-se de que se encontra no modo Passageiro e clique no segundo ícone da barra de navegação inferior. Em qualquer uma das opções da barra de navegação superior pode visualizar um botão suspenso cor-de-laranja com o ícone de lupa. Para começar a procurar uma viagem ao seu gosto, clique no botão.



2

A procura por uma viagens está dividida por passos. O primeiro é decidir que a viagem é uma ida para o IPS, ou para um destino privado com partida no IPS. Escolha a opção esquerda para o primeiro caso, ou a da direita para o segundo. Para continuar, clique no botão "CONTINUAR".



### 3.5. Procurar Boleia

3

O segundo passo consiste na definição da data e hora em que deseja realizar a viagem e da sua tolerância de tempo quanto à hora de partida da mesma.



4

Ao clicar no campo de texto de data e hora, poderá visualizar uma interface assistente à seleção, onde pode escolher qualquer data dentro da próxima semana , e qualquer hora dentro das 24. Quando estiver satisfeito, clique no botão de confirmação.



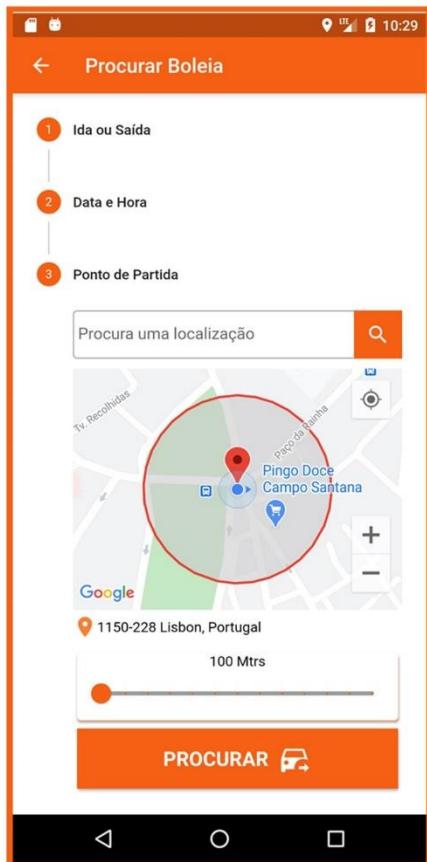
### 3.5. Procurar Boleia

5

A data escolhida irá aparecer no campo de texto. Agora poderá selecionar a sua tolerância de tempo referente à hora que escolheu, ou seja, se selecionar 10, poderá procurar por viagens que partem 10 minutos antes ou depois da hora que definiu. Ao clicar no campo à direita do campo de texto da data e hora, irá conseguir ver as tolerâncias disponíveis e escolher uma com um clique. Pode voltar ao passo anterior através do botão "ANTERIOR", e continuar para o seguinte no botão "CONTINUAR".



6

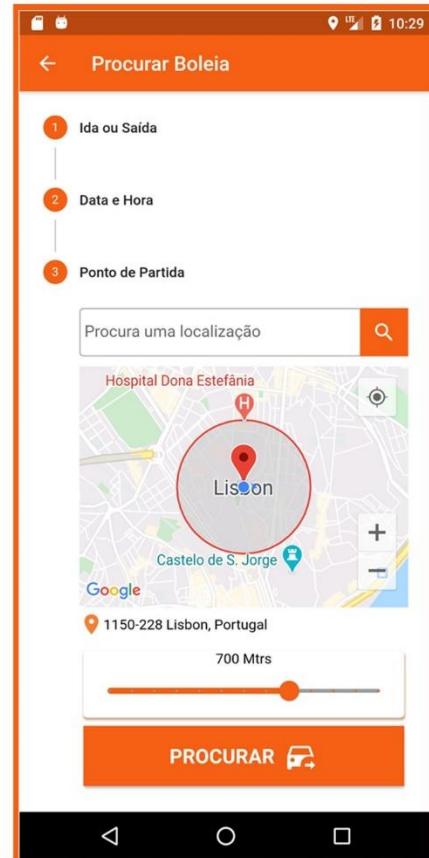


O próximo passo destina-se à definição do ponto de encontro da viagem, caso seja com destino ao IPS, ou ponto de destino, caso seja uma viagem com partida no campus do IPS. O ponto pode ser escolhido através do mapa interativo visível. Pode arrastar e clicar no mapa e nos seus botões de ampliação para decidir um ponto geográfico, ou inserir uma morada no campo de texto acima para procurar um ponto à sua escolha através do botão com o ícone lupa.

### 3.5. Procurar Boleia

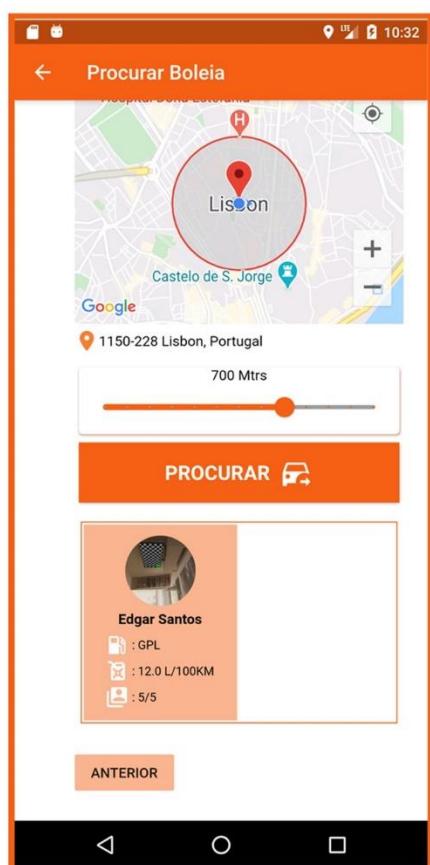
7

Depois de escolher o seu ponto de encontro ou destino, através do botão deslizando abaixo, pode definir o raio em que pretende procurar por uma boleia, podendo adaptar a sua capacidade de deslocação independente de 100 até 1000 metros. Quando estiver satisfeito, procure por uma viagem através do botão "PROCURAR".



8

Ao clicar no botão "PROCURAR", o sistema irá pesquisar por uma viagem à sua medida e apresentar-lhe as encontradas. Caso não haja nenhuma viagem de acordo com os parâmetros que definiu, o aplicação apresenta o texto "Sem grupos encontrados". Caso encontre uma viagem, esta aparecerá num formato retangular com a foto e nome do motorista da mesma, assim como as informações do seu veículo, nomeadamente, o tipo de combustível, o consumo, e os lugares disponíveis na viagem. Pode clicar no retângulo laranja para ver os detalhes da viagem.



### 3.5. Procurar Boleia

9

Ao clicar no retângulo da viagem, irá ser-lhe apresentada a mesma janela de detalhes que o da gestão dos seus pedidos como Passageiro, com a exceção de que o botão para pedir boleia já não se encontra desabilitado. Assim que clicar no botão, a sua boleia irá ser solicitada.

