

**Protrip**  
Sistema de cálculo eficiente  
do custo de uma viagem de automóvel

PROJETO DE MESTRADO

**Paulo Diogo da Silva Mendez**  
MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

*A Nossa Universidade*

[www.uma.pt](http://www.uma.pt)

outubro | 2020

**Protrip**  
Sistema de cálculo eficiente  
do custo de uma viagem de automóvel  
PROJETO DE MESTRADO

**Paulo Diogo da Silva Mendez**  
MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTAÇÃO  
Frederica Margarida Camacho Gonçalves

CO-ORIENTAÇÃO  
Eduardo Miguel Dias Marques

## Resumo

A grande maioria das pessoas quando quer fazer uma viagem de automóvel para um destino distante, comparado com o seu ponto de partida, normalmente apenas tem em conta os custos gerais, como o combustível ou as portagens. Para que um condutor tenha uma expectativa aproximada do custo real, este tem de contar com os gastos relacionados com o veículo, como por exemplo: impostos, manutenções gerais e seguros. No âmbito da área do transporte privado de passageiros, saber todos os custos relacionados com uma frota de veículos é muito importante para uma gestão eficiente dos rendimentos e assim calcular preços mais competitivos para os seus clientes.

Este projeto tem o objetivo de tornar este processo mais simples através da criação de uma aplicação que permita ao utilizador introduzir todos os gastos com o seu veículo e, assim, fornecer uma estimativa de quanto é que uma determinada viagem irá custar. Esta aplicação abrange todos os casos de utilização quer sejam a nível pessoal, quer como profissional da indústria do transporte privado de passageiros.

A abordagem tomada para o desenvolvimento deste software foi desenvolver uma aplicação que consiga calcular de forma eficaz o custo de uma viagem de automóvel.

Os resultados demonstraram que a diferença entre valores calculados na aplicação e os valores calculados a partir dos dados fornecidos pelo veículo são de 1.2%.

## Abstract

The vast majority of people when they want to make a trip by car to a distant destination compared to their starting point usually only take into account general costs such as fuel or tolls. In order for a driver to have an approximate expectation of the real cost, he has to rely on the expenses related to the vehicle, such as: taxes, vehicle maintenance, and / or other expenses related to vehicle wear (small repairs or improvements). In the area of private passenger transport, knowing all costs related to a vehicle fleet is very important for an efficient management of yields and thus calculating more competitive prices for your customers.

This project aims to make this process simpler by creating an application that allows the user to enter all expenses with his vehicle and thus provide an estimate of how much a given trip may cost. This application covers all use cases, whether personal or professional in the private passenger transport industry.

The approach taken to develop this software was to develop an application that can efficiently calculate the cost of a car trip.

The results showed that the difference between values calculated in the application and the values calculated from the data provided by the vehicle are 1.2%.

## Agradecimentos

Em primeiro lugar quero agradecer à Professora Doutora Frederica Gonçalves que não só me acompanhou durante o meu ensino secundário e universitário, mas também por ter-me dado a oportunidade de fazer este projeto dando-me a liberdade de construir um projeto com utilidade. Em segundo agradecer ao Professor Doutor Eduardo Marques por todo o seu tempo dedicado a este projeto e todos os conselhos e feedback dados não só neste projeto, mas também no meu percurso académico.

Quero agradecer também às pessoas mais importantes da minha vida que são os pais, à minha irmã e aos meus avós que sempre deram a força necessária durante todo o meu percurso pessoal e sempre me apoiaram em tudo, seja em projetos pessoais ou profissionais.

Por fim quero agradecer aos meus amigos mais próximos por terem estado ao meu lado em todos os momentos bons e maus da minha vida e que sempre apoiaram e criticaram construtivamente as minhas decisões.

# Índice

<b>Índice de Figuras .....</b>	<b>6</b>
<b>Índice de Tabelas.....</b>	<b>9</b>
<b>Índice de Equações .....</b>	<b>10</b>
<b>1. Acrónimos .....</b>	<b>11</b>
<b>2. Introdução.....</b>	<b>12</b>
2.1. Contextualização do Problema .....	12
2.2. Motivação.....	15
2.3. Objetivos .....	16
2.4. Estrutura do relatório .....	17
<b>3. Estado Da Arte .....</b>	<b>18</b>
3.1. Tipos de Algoritmos. ....	18
3.2. Cálculo do preço das viagens.....	18
3.2.1. Algoritmo da Uber .....	18
3.2.2. Algoritmo do Táxi.....	23
3.2.3. Comparação entre o Algoritmo da Uber e do Taxímetro.....	27
3.3. Cálculo do Trajeto Ótimo.....	30
3.3.1. Algoritmo de Dijkstra .....	31
3.3.2. Inovações na área do cálculo do trajeto mais rápido. ....	34
3.4. Análise a Aplicações existentes no mercado .....	35
3.4.1. Estudo da Aplicação da Uber.....	35
3.4.2. Estudo da Aplicação da Taxiin .....	36
<b>4. Desenvolvimento da Aplicação.....</b>	<b>39</b>
4.1. Requisitos funcionais e não funcionais .....	39
4.1.1. Requisitos funcionais.....	39
4.1.2. Requisitos não funcionais .....	40
4.2. Arquitetura Geral do Sistema.....	41
4.3. Casos de utilização .....	42
4.4. Protótipos de baixa e alta fidelidade.....	42
4.4.1. Protótipos de baixa Fidelidade.....	42
4.4.2. Protótipos de Alta Fidelidade.....	45
4.5. Mapa de navegação .....	47
4.6. Aplicação Final .....	48
4.7. Base de Dados .....	54
4.7.1. API .....	55
4.7.1.1. RegisterAccount.php .....	55
4.7.1.2. CreateVehicleDetails.php.....	56
4.7.1.3. createUserTrip.php .....	57
4.7.1.4. getUserTripHistory.php.....	57
4.7.1.5. login.php.....	58
4.8. Desenvolvimento do Algoritmo para o cálculo do custo de uma viagem .....	58
4.9. Desenvolvimento da aplicação iOS .....	60
<b>5. Testes .....</b>	<b>67</b>
5.1. Testes de usabilidade.....	67

<b>5.2. Testes à aplicação .....</b>	<b>73</b>
<b>6. Conclusão e Trabalho Futuro .....</b>	<b>78</b>
<b>6.1. Trabalho Realizado .....</b>	<b>78</b>
<b>6.2. Trabalho Futuro .....</b>	<b>79</b>
<b>7. Referências: .....</b>	<b>81</b>
<b>ANEXOS .....</b>	<b>84</b>

## Índice de Figuras

<b>Figura 1</b> - Trajeto entre Câmara de Lobos e Estreito de Câmara de Lobos .....	13
<b>Figura 2</b> - Trajeto entre Funchal e Lido .....	14
<b>Figura 3</b> - <i>Surge Period</i> após concerto da Ariana Grande em Nova York [7].....	21
<b>Figura 4</b> - Comparação entre pedidos de viagens aceites, tempo estimado para a recolha e o rácio pedidos-viagens após concerto da Ariana Grande em Nova York [7]. .....	21
<b>Figura 5</b> - Comparação entre pedidos de viagens aceites, tempo estimado para a recolha e o ratio pedidos-viagens em Nova York no ano novo de 2015. ....	22
<b>Figura 6</b> - Exemplo de um Taxímetro nos Estados Unidos da América.....	23
<b>Figura 7</b> - Trajeto entre “Lincoln Ctr for the Performing Arts (Lincoln Center for the Performing Arts)” e “Bronx Behavioral Health Center” .....	26
<b>Figura 8</b> - Viagem a partir do Porto até Vila Real através da Uber. ....	28
<b>Figura 9</b> - Exemplo de um grafo para aplicação do algoritmo de Dijkstra .....	31
<b>Figura 10</b> – (a) Ecrã inicial da aplicação da Uber e (b) Ecrã representativo do percurso escolhido pelo utilizador .....	35
<b>Figura 11</b> - Ecrã inicial da Taxiin .....	37
<b>Figura 12</b> - Arquitetura Geral do Sistema.....	41
<b>Figura 14</b> - Casos de utilização da Aplicação. ....	42
<b>Figura 15</b> - Protótipo baixa Fidelidade: Ecrã principal para o requisito 1 .....	43
<b>Figura 16</b> - Protótipo baixa Fidelidade: Ecrã principal para o requisito 2 .....	44
<b>Figura 17</b> - Protótipo baixa Fidelidade: Ecrã principal para o requisito 3 .....	44
<b>Figura 18</b> - Protótipo Alta Fidelidade: Ecrã principal para o requisito 1 .....	45
<b>Figura 19</b> - Protótipo Alta Fidelidade: Ecrã principal para o requisito 2.....	46
<b>Figura 20</b> - Protótipo Alta Fidelidade: Ecrã principal para o requisito 3 .....	46
<b>Figura 21</b> -Mapa de navegação da aplicação. ....	47
<b>Figura 22</b> -Explore Screen .....	48
<b>Figura 23</b> - Ecrã de Login e Registo .....	49
<b>Figura 24</b> -Formulário de registo de um novo veículo.....	50
<b>Figura 25</b> - Home Screen .....	51
<b>Figura 26</b> - Ecrã Viagem em Tempo real.....	52
<b>Figura 27</b> – Ecrã de Registrar Viagem .....	52
<b>Figura 28</b> – Ecrã de Histórico das Viagens.....	53

<b>Figura 29</b> – Ecrã de Histórico das Viagens .....	54
<b>Figura 30</b> – Estrutura da base de dados da aplicação .....	54
<b>Figura 31</b> -Código Swift utilizado para a implementação do algoritmo .....	60
<b>Figura 32</b> - Projeto Protrip no IDE Xcode .....	61
<b>Figura 33</b> - iPhone 8 e iPhone 11 Pro .....	61
<b>Figura 34</b> – Ficheiro xib para o ecrã HomeViewController .....	62
<b>Figura 35</b> – Ficheiro XIB do ecrã Viagem Tempo Real .....	63
<b>Figura 36</b> – Ligação dos elementos visuais ao código .....	63
<b>Figura 37</b> – Função setupView e StartTravel.....	64
<b>Figura 38</b> – Função runTimer.....	65
<b>Figura 39</b> – Função locationManager .....	66
<b>Figura 40</b> – Função drawRoute .....	66
<b>Figura 41</b> - BMW F30 318d, veículo utilizado nos testes .....	74
<b>Figura 42</b> - Alguns testes Realizados dentro do Protrip.....	75
<b>Figura 43</b> - Protótipo de Baixa Fidelidade - Ecrã de Registo do Automóvel 1 .....	84
<b>Figura 44</b> - Protótipo de Baixa Fidelidade - Ecrã de Registo do Automóvel 2 .....	84
<b>Figura 45</b> - Protótipo de Baixa Fidelidade - Ecrã de Registo do Automóvel 3 .....	85
<b>Figura 46</b> - Protótipo de Baixa Fidelidade - Ecrã de Registo de uma Viagem 1 .....	85
<b>Figura 47</b> - Protótipo de Baixa Fidelidade - Ecrã de Registo de uma viagem 2 .....	86
<b>Figura 48</b> - Protótipo de Baixa Fidelidade - Ecrã de Registo de uma viagem 3 .....	86
<b>Figura 49</b> - Protótipo de Baixa Fidelidade - Ecrã de Viagem em Tempo Real 1 .....	87
<b>Figura 50</b> - Protótipo de Baixa Fidelidade - Ecrã de Viagem em Tempo Real 2 .....	87
<b>Figura 51</b> - Protótipo de Baixa Fidelidade - Ecrã de Viagem em Tempo Real 3 .....	88
<b>Figura 52</b> - Protótipo de Baixa Fidelidade - Ecrã de Histórico de Viagens.....	88
<b>Figura 53</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 1 .....	89
<b>Figura 54</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 2 .....	89
<b>Figura 55</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 3 .....	90
<b>Figura 56</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 4 .....	91
<b>Figura 57</b> - Protótipo de Alta Fidelidade - Ecrã Inicial da Aplicação.....	92
<b>Figura 58</b> - Protótipo de Alta Fidelidade - Ecrã Principal do Protrip .....	93
<b>Figura 59</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem 1 .....	94
<b>Figura 60</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem 2 .....	95
<b>Figura 61</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem em Tempo Real 1 .....	96

<b>Figura 62</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem em Tempo Real 2 .....	97
<b>Figura 63</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem em Tempo Real 3 .....	98
<b>Figura 64</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de Login.....	99
<b>Figura 65</b> - Protótipo de Alta Fidelidade - Ecrã de Registo de uma nova conta.....	100
<b>Figura 66</b> - Feedback Fornecido pelos utilizadores à tarefa 1 dos testes de usabilidade .....	100
<b>Figura 67</b> - Feedback Fornecido pelos utilizadores à tarefa 2 dos testes de usabilidade .....	100
<b>Figura 68</b> - Feedback Fornecido pelos utilizadores à tarefa 3 dos testes de usabilidade .....	101
<b>Figura 69</b> - Feedback Fornecido pelos utilizadores à tarefa 4 dos testes de usabilidade .....	101
<b>Figura 70</b> - Resultado da Pergunta 1 do questionário de usabilidade .....	101
<b>Figura 71</b> - Resultado da Pergunta 2 do questionário de usabilidade .....	102
<b>Figura 72</b> - Resultado da Pergunta 3 do questionário de usabilidade .....	102
<b>Figura 73</b> - Resultado da Pergunta 4 do questionário de usabilidade .....	102
<b>Figura 74</b> - Resultado da Pergunta 5 do questionário de usabilidade .....	103
<b>Figura 75</b> - Resultado da Pergunta 6 do questionário de usabilidade .....	103
<b>Figura 76</b> - Resultado da Pergunta 7 do questionário de usabilidade .....	103
<b>Figura 77</b> - Resultado da Pergunta 8 do questionário de usabilidade .....	104
<b>Figura 78</b> - Resultado da Pergunta 9 do questionário de usabilidade .....	104
<b>Figura 79</b> - Resultado da Pergunta 10 do questionário de usabilidade .....	104

## Índice de Tabelas

<b>Tabela 1</b> -Tabela de tarifas fixas na cidade de Lisboa [4].....	22
<b>Tabela 2</b> - Tabela de tarifas fixas dos táxis na cidade de New York [10].....	24
<b>Tabela 3</b> - Comparação de custos de um percurso entre um veículo elétrico e um a gasolina [11].....	29
<b>Tabela 4</b> - Pesos para os trajetos para todos os vértices a partir de A.....	32
<b>Tabela 5</b> - Requisitos Funcionais .....	39
<b>Tabela 6</b> - Requisitos não funcionais .....	40
<b>Tabela 7</b> -Tabela de entradas e saídas da chamada RegisterAccount.php .....	56
<b>Tabela 8</b> -Tabela de entradas e saídas da chamada CreateVehicleDetails.php .....	56
<b>Tabela 9</b> -Tabela de entradas e saídas da chamada createUserTrip.php .....	57
<b>Tabela 10</b> -Tabela de entradas e saídas da chamada getUserTripHistory.php.....	57
<b>Tabela 11</b> -Tabela de entradas e saídas da chamada login.php.....	58
<b>Tabela 12</b> - Respostas dadas ao questionário SUS.....	72
<b>Tabela 13</b> - "Score" de cada questionário realizado no teste de usabilidade .....	72
<b>Tabela 14</b> - Testes realizados ao algoritmo desenvolvido.....	75
<b>Tabela 15</b> – Resultados dos testes ao percurso “Estreito de Câmara de Lobos – Ribeira Brava” .....	76

## Índice de Equações

Equação 1 - Cálculo do preço de uma viagem de um ponto A -> B na Uber .....	19
Equação 2 - Cálculo do preço de uma viagem de um ponto A -> B num táxi na cidade de New York [11].....	25
Equação 3 – Exemplo prático do cálculo do preço de uma viagem utilizando o algoritmo do Táxi convencional 1.....	26
Equação 4 – Exemplo prático do cálculo do preço de uma viagem utilizando o algoritmo do Táxi convencional 2.....	26
Equação 5 - Cálculo para o custo total da viagem .....	58
Equação 6 - Preço por quilómetro.....	58
Equação 7 - Preço por minuto .....	59

## 1. Acrónimos

AMT – Autoridade da Mobilidade e dos Transportes

API – Interface de Programação de Aplicações

App – Termo comum utilizado para identificar uma aplicação móvel.

GPS – Sistema de Posicionamento Global

HTTP – Protocolo de Transferência de Hipertexto

iOS – Sistema operativo utilizado nos dispositivos móveis da Apple (iPhone, iPad, iPod)

JSON – Tipo de estrutura utilizada para a disponibilização da informação de uma chamada à API

MVC – Arquitetura Model View Controller

OBD – On-Board Diagnostics

PHP – Linguagem utilizada no lado do Servidor para a criação da API

RAM - Região Autónoma da Madeira

SQL– Linguagem utilizada para a comunicação entre a Base de Dados e a API

SO – Sistema Operativo

SUS – System Usability Score

UI -Interface de Utilizado

## 2. Introdução

Neste capítulo irá ser descrito a contextualização do problema, a motivação, os objetivos a serem alcançados nesta tese e por fim toda a estrutura deste relatório.

### 2.1. Contextualização do Problema

O serviço de táxi, também conhecido como Transporte de Aluguer em Veículos Automóveis de Passageiros é um serviço que se foca no transporte privado de clientes. Um serviço de táxi tem as seguintes características em Portugal [1]:

- Ser um automóvel de passageiros, de lotação até nove lugares, incluindo o motorista;
- Estar equipado com taxímetro;
- Estar equipado com dispositivo luminoso identificativo de táxi;
- Dispor de caixa fechada;
- Ter uma distância mínima entre eixos de 2,5m;
- Dispor de distintivos de identificação da licença (número e identificação da freguesia ou concelho) colocados nos guarda-lamas da frente e na retaguarda do veículo.

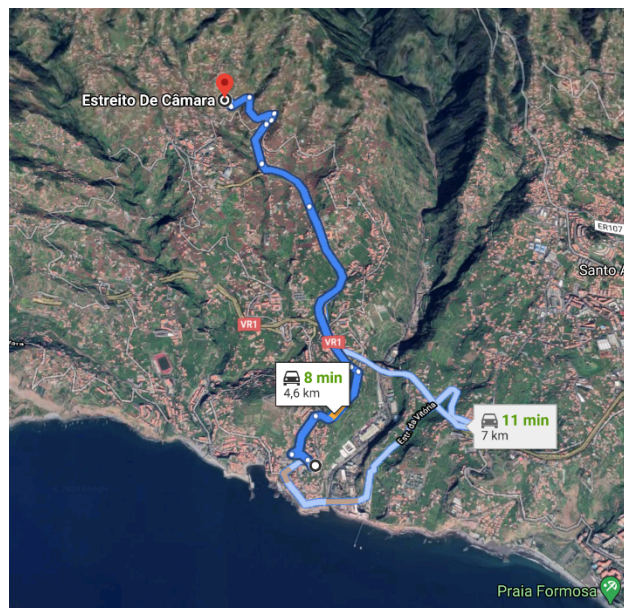
Na Região Autónoma da Madeira (RAM) são respeitadas as mesmas características para território nacional, com a exceção que o veículo regional tem uma pintura amarela com riscas azuis.

O serviço de táxi da RAM é um serviço tradicional e muitas vezes suportado a nível económico, na sua maioria, pela área do Turismo. No entanto, os táxis não servem apenas para transporte privado de pessoas, mas também para transporte de doentes não graves para hospitais/centros de saúde e algumas vezes também realizam excursões pela ilha. De momento na RAM existem 3 formas de requisitar o serviço de táxi:

- Simplesmente dirigir-se a uma das praças de táxi, presentes em todas as freguesias da ilha;
- Ligar para uma das centrais e pedir um táxi para um local específico;
- Marcar com antecedência um lugar e uma hora para que o taxista esteja pronto para levar o passageiro para o seu destino. Normalmente este tipo de requisição é feito por estrangeiros quando chegam ao aeroporto e querem dirigir-se imediatamente

para o hotel. Neste caso a marcação pode ser feita por telefone ou então enviando um email para uma das empresas que oferecem este serviço.

Apesar dos preços tabelados serem regulamentados por uma entidade pública, a Autoridade da Mobilidade e dos Transportes (AMT)<sup>1</sup>, e os taxistas ou empresas de táxi conseguirem prever quanto é que irão faturar, pode ser um desafio calcular o custo total de todas as viagens realizadas num mês, o que, por vezes, leva a uma perceção errada de qual o lucro ou prejuízo. Não existe atualmente uma forma eficiente de calcular os custos de uma viagem realizada, com um erro mínimo associado. Por exemplo, uma viagem do centro do Funchal até ao Lido pode ter um custo para o cliente de 10€ e um custo para o taxista de 2 a 3 euros, enquanto que uma viagem do centro do Câmara de Lobos para o Estreito de Câmara de Lobos poderá custar ao cliente também 10€ mas o custo para o taxista será de 3 a 4 euros, pois, e apesar do trajeto ser semelhante, as diferentes variáveis, como inclinação da estrada, irão ter um impacto grande nos custos de manutenção do veículo [24].



**Figura 1** - Trajeto entre Câmara de Lobos e Estreito de Câmara de Lobos

---

<sup>1</sup> AMT website - <https://www.amt-autoridade.pt>



**Figura 2** - Trajeto entre Funchal e Lido

A grande maioria dos profissionais normalmente considera os principais gastos como sendo gasóleo e impostos, mas existem muitos mais fatores a ter em consideração para calcular os verdadeiros custos como: manutenções gerais, impostos e seguros [25]. Todos estes fatores têm um peso significativo anual nos custos reais de um Profissional da área do transporte privado de passageiros. Apesar de ser possível ao motorista somar todos os custos no fim do ano e subtrair ao total faturado, os profissionais não conseguem ter uma noção exata de quanto realmente custou uma determinada viagem.

Embora este seja um problema muitas vezes associado ao serviço de táxi, também poderá ser um problema, e talvez mais relevante para as plataformas de transporte privado como a Uber<sup>2</sup>, Bolt<sup>3</sup>, ou mesmo para as pessoas que entregam comida ao domicílio através de plataformas como a Uber Eats<sup>4</sup>, Glovo<sup>5</sup>, entre outras.

Para estes profissionais o problema é ainda mais visível pois, enquanto os taxistas normalmente trabalham a tempo integral na indústria do táxi, os motoristas da Uber [26] fazem este trabalho muitas vezes como um extra e é complicado de analisar num curto espaço de tempo se este *part-time* está a trazer lucro ou prejuízo ao fim de algumas viagens. Na Madeira, e devido à nossa orografia, as viagens de 10€ podem ser vantajosas para os utilizadores, mas apenas dar um lucro de 1 a 2 euros aos motoristas após o cálculo do custo real. Outro problema existente no mercado dos táxis e das plataformas digitais é

<sup>2</sup> Uber website: <https://www.uber.com/pt/en/>

<sup>3</sup> Bolt website: <https://bolt.eu/pt-pt/>

<sup>4</sup> Uber eats website: <https://www.ubereats.com/pt>

<sup>5</sup> Glovo website: <https://glovoapp.com/en/fun/>

ter previsões de quanto é que uma viagem irá custar ao motorista na sua realidade [27]. Uma funcionalidade como esta poderia otimizar os lucros dos profissionais como por exemplo:

Imaginemos que um condutor tem que escolher entre duas viagens (referentes às Figuras 1 e 2):

**Viagem 1:** Funchal – Lido, 9 Euros

**Viagem 2:** Câmara de Lobos - Estreito de Câmara de lobos, 10 euros

Se um algoritmo conseguisse prever que a viagem 1 teria um custo de 2 euros devido as condições do percurso serem melhores (ser plano, condições da estrada) e na viagem 2 o custo seria de 5 euros devido a uma maior inclinação da estrada e consequentemente um maior gasto de travões a descer depois do Monte por exemplo. A escolha seria óbvia e o condutor escolheria a viagem 1 caso pretendesse otimizar a sua margem de lucro.

## 2.2. Motivação

*"Quanto é que custa, realmente, ir da minha casa a um determinado sítio?"* Esta é uma pergunta que provavelmente todos nós já tivemos a curiosidade de saber a resposta exata. Normalmente o que a maioria das pessoas faz para responder a esta pergunta é simplesmente ir ao Google Maps<sup>6</sup> ver a distância do percurso, calcular quanto é que o veículo irá consumir a nível de combustível durante a viagem, multiplicar pelo preço do gasóleo/gasolina atual e obtemos a resposta. Mas será que o custo de uma viagem é apenas calculado pelo valor gasto em combustível? O que é que acontece com todos os gastos que não são contabilizados e que anualmente estão associados como as manutenções gerais, os impostos e seguros?

Normalmente, a grande maioria dos profissionais da área do transporte privado de passageiros não têm o hábito de assumir estes gastos para o cálculo de uma viagem em específico, mas sabendo exatamente o custo total poderá ter um impacto positivo na vida profissional das pessoas/empresas. Por exemplo, será que compensa um condutor da Uber realizar um serviço de recolha de um passageiro ao Monte para o Funchal considerando que o condutor está no Funchal? Por um lado, o valor para o passageiro será baixo pois a

---

<sup>6</sup> Google Maps website: <https://www.google.pt/maps/>

Uber oferece viagens a baixo custo, mas será que o desgaste do veículo, combustível gasto e as comissões permitem ao condutor ter lucro? Estas são algumas das questões que esta tese se irá focar e terá como objetivo desenvolver um projeto inovador que permitirá ao utilizador ter em conta todos os gastos que este tem com o seu veículo e assim saber com certeza de quanto é que efetivamente irá custar uma determinada viagem.

### 2.3. Objetivos

O principal objetivo desta tese é criar um software que permita aos profissionais de transporte de passageiros (Táxi) e aos motoristas independentes (Uber, Bolt, etc.) conseguirem calcular o seu lucro/despesa de uma viagem realizada no âmbito da sua atividade profissional. Para o efeito será criada uma aplicação que consiga prever os custos associados de uma viagem através de duas maneiras:

- ✓ Prever o custo de uma viagem antes de esta se realizar;
- ✓ Estimar o custo de uma viagem em tempo real enquanto a viagem está a se realizar.

Para prever este tipo de dados é necessário um dispositivo portátil que disponha de acesso a dados provenientes do GPS fazendo com que seja possível calcular por exemplo, distâncias percorridas e velocidade atual do veículo. Assim sendo uma alternativa para aceder a estes dados de forma fiável, é através da criação de uma aplicação móvel.

A maneira mais eficaz de conseguir garantir que os valores estimados pela aplicação são de confiança será por realizar várias viagens em que são registados todos os valores do computador de bordo do veículo e também os valores estimados pela aplicação que será desenvolvida. Quando for aplicada a fórmula proposta nesta tese aos valores fornecidos pelo computador de bordo, irão ser comparados pelos valores fornecidos pela aplicação e caso a diferença seja mínima (irá ser utilizado um valor simbólico de 5% para a diferença máxima) podemos assumir que os custos estimados fornecidos pela aplicação são de confiança. Toda a parte dos testes realizados à aplicação estará explicada com mais detalhe no capítulo 4.

## 2.4. Estrutura do relatório

Este relatório está estruturado em 5 capítulos principais, sendo estes os seguintes:

**Introdução:** neste capítulo será descrito toda a motivação, os problemas que podemos encontrar no mundo real que necessitem de um sistema como o que irá ser desenvolvido durante esta tese, os objetivos propostos e por fim a estrutura do relatório.

**Estado da Arte:** será descrita a contextualização desta tese refletindo os algoritmos necessários para este estudo bem como os serviços que servirão de comparação como a Uber e o Táxi.

**Desenvolvimento da Aplicação:** todo o processo de desenvolvimento está neste capítulo, incluindo os requisitos funcionais e não funcionais, os protótipos de baixa e alta fidelidade, o código necessário para as funcionalidades mais importantes, as capturas de ecrãs do resultado final e, por fim, a descrição do algoritmo proposto.

**Testes à aplicação:** neste capítulo são apresentados os testes realizados ao algoritmo e a comparação com valores obtidos através da aplicação com valores fornecidos pelo computador de bordo do veículo. Também será realizado testes de usabilidade com um grupo de utilizadores de diferentes áreas académicas, para obter feedback do que pode ser melhorado no software desenvolvido.

**Conclusão e Trabalho Futuro:** as conclusões retiradas com a realização deste projeto, a verificação se os objetivos propostos foram cumpridos e a análise de melhorias possíveis numa futura implementação serão descritos neste último capítulo.

## 3. Estado Da Arte

Neste capítulo serão descritos alguns algoritmos pesquisados para o cálculo do custo de uma viagem em específico, como também algoritmos que permitiram hoje em dia calcular trajetos eficientes para uma viagem de um ponto A a um ponto B. Irão ser estudado os algoritmos da Uber e o algoritmo usado num táximetro, sendo de seguida efetuadas comparações entre os mesmos. Por fim será realizada uma análise a algumas soluções que fornecem serviços de transporte através de uma aplicação móvel.

### 3.1. Tipos de Algoritmos.

As categorias de cálculo para os algoritmos das viagens são:

- 1- Cálculo do preço;
- 2- Calcular o trajeto ótimo.

### 3.2. Cálculo do preço das viagens

De entre os algoritmos para o cálculo do preço das viagens será explorado em primeiro lugar o algoritmo usado pela empresa Uber e, por último, o cálculo feito no serviço tradicional de Táxi. O algoritmo da Uber foi escolhido para este estudo por esta ser a pioneira na digitalização da reserva do transporte de passageiros no sector privado. O serviço tradicional de Táxi, que sendo o mais conhecido e o mais antigo, ainda é o ponto de referência para os novos serviços digitais emergentes.

O principal objetivo desta secção é demonstrar como é que são calculados os custos de viagem para cada opção acima descritas, e fazer uma comparação entre elas com o intuito de demonstrar as vantagens e desvantagens de cada uma, referenciando qual delas é a melhor para o utilizador e para o motorista.

#### 3.2.1. Algoritmo da Uber

A ideia de utilizar aplicações móveis para efetuar a reserva de transporte de passageiros semelhantes a um serviço de táxi não é nova. A Uber, criada por Travis Kalanick e Garrett Camp, foi um serviço criado em 2009 e o seu principal foco era fazer o mesmo que o serviço de táxi, mas de uma forma conveniente e simples. O conceito por detrás da Uber permitiu que qualquer pessoa com um veículo pessoal começasse transportar passageiros

e assim capitalizar o uso do seu veículo pessoal. Esta ideia resultou numa mudança na indústria do transporte privado de passageiros, onde até a data da introdução da Uber era dominada pelo serviço de Táxi [2].

O serviço da Uber funciona de uma maneira muito simples. O utilizador, com uma simples aplicação móvel, pode requisitar um serviço de transporte de um ponto A para um ponto B.

Quando o utilizador pretende realizar uma viagem a aplicação redireciona o seu pedido para o motorista mais próximo, sendo que, quando o motorista aceita a viagem a aplicação indica quanto tempo o carro irá demorar até chegar ao ponto de recolha marcado pelo utilizador. O utilizador enquanto espera pelo motorista terá acesso a todas as informações sobre o mesmo como o nome do motorista, a marca, modelo e matrícula do veículo para que seja fácil de identificar qual o veículo que o irá recolher.

No fim da viagem a aplicação irá pedir ao utilizador e ao motorista para dar um feedback da viagem com o intuito de classificar a mesma. Desta maneira é possível criar um sistema de confiança dentro da plataforma de modo a descartar motoristas e/ou utilizadores que sejam problemáticos [3] [26].

Os custos para cada viagem na plataforma da Uber são calculados tendo em conta diversas tarifas e variáveis sendo estas [4]:

- **Tarifa base:** A tarifa base é determinada pela duração e pela distância de uma viagem.
- **Taxa de reserva:** Dependendo da localização poderá ser adicionada uma taxa fixa a cada viagem para ajudar a suportar os custos operacionais, regulamentares e de segurança.
- **Horas e áreas mais concorridas:** Quando existem mais utilizadores do que motoristas os preços aumentam temporariamente até que a oferta e a procura voltem a estar equilibrados.

Com base na informação disponível e não oficial, a fórmula de cálculo do preço da viagem na Uber poderá ser representada pela seguinte equação (a equação oficial nunca foi disponibilizada pela Uber) [5]:

$$Pt = (Tb + Tr + (M \times Vm) + (D \times Vd)) \times (1 + Vhc)$$

**Equação 1** - Cálculo do preço de uma viagem de um ponto A -> B na Uber

Sendo que as variáveis acima representadas são:

$Pt$  = Preço total para a viagem de um ponto A -> B.

$Tb$  = Tarifa base na cidade onde vai acontecer a viagem.

$Tr$  = Tarifa de reserva na cidade onde vai acontecer a viagem.

$M$  = Número de minutos total da viagem de um ponto A -> B.

$Vm$  = Valor por minuto na cidade onde vai acontecer a viagem.

$D$  = Distância total da viagem de um ponto A -> B.

$Vd$  = Valor por distância (km) na cidade onde vai acontecer a viagem.

$Vhc$  = Valor em percentagem representando a taxa extra para horas concorridas.

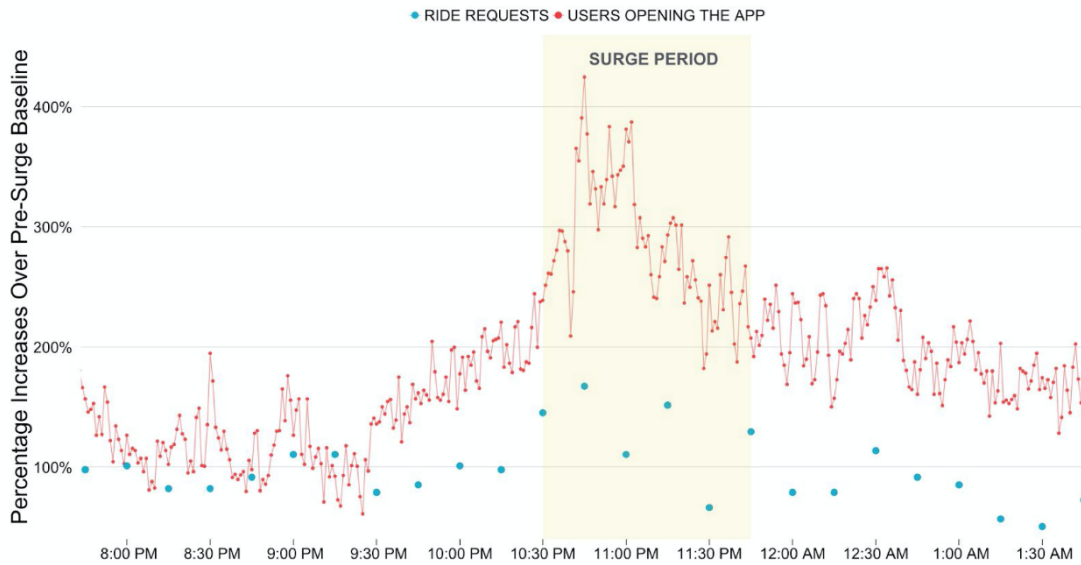
O  $Vhc$  é o parâmetro mais interessante desta fórmula. Este parâmetro é calculado e incrementado quando a procura é maior do que a oferta.

Este algoritmo tem também como objetivo, equilibrar a oferta e a procura sendo que este funciona de modo a aumentar o preço da viagem para que os utilizadores que vejam estes preços inflacionados procurem outro tipo de serviço de transporte ou então esperam que o *'surge period'*<sup>7</sup> acabe para conseguirem viagens mais acessíveis. Com a ajuda deste parâmetro a Uber consegue garantir sempre a maximização do rácio entre pedidos e respostas, maximizando assim também os lucros da empresa [7].

Durante este período de falta de motoristas para a quantidade de utilizadores a requisitarem o serviço, a própria empresa aumenta o valor que um motorista ganha por hora tentando assim conseguir com que motoristas fora de serviço ou fora das áreas de *'surge period'* se desloquem para aquela área para proteger os lucros e o rácio de pedidos e resposta. Na Figura 3 podemos ver um exemplo de um *'surge period'* depois de um concerto esgotado da Ariana Grande em Nova York [7].

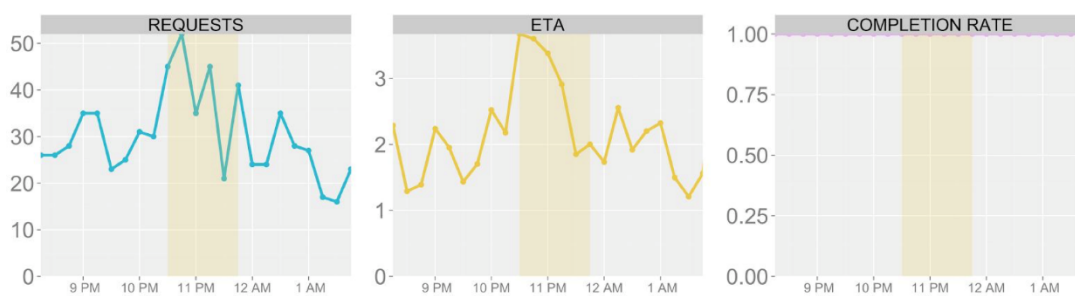
---

<sup>7</sup> Período de Hora de tráfego intenso



**Figura 3 - Surge Period** após concerto da Ariana Grande em Nova York [7]

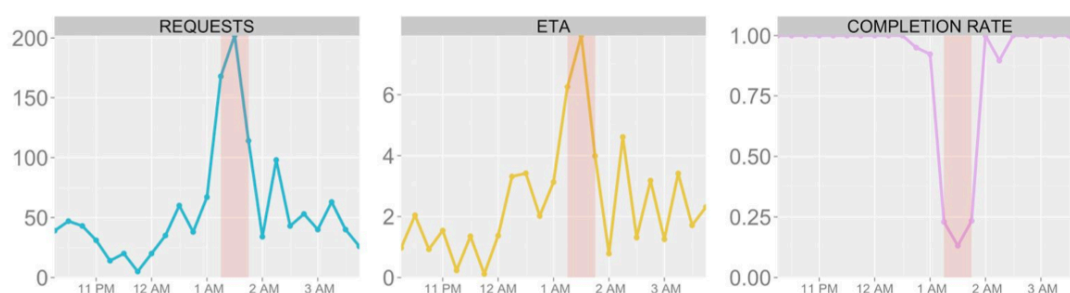
Como podemos observar na Figura 3, o número de utilizadores que abriram a App da Uber (a vermelho) para requisitar uma viagem teve um pico muito mais significativo quando comparado com as restantes horas do dia. No primeiro pico foi ativado o "surge period" onde os preços foram aumentados no seu máximo 6 vezes ao preço normal do trajeto levando a que os utilizadores optassem por outro tipo de transporte ou então esperassem alguns minutos para que este período acabasse. Se analisarmos o gráfico, e graças ao parâmetro *Vhc*, o número de pedidos (pontos azuis) para uma viagem não foi muito superior quando comparamos com as restantes horas do dia, e a Uber continuou com o rácio entre pedidos e viagens acima de 98% maximizando assim os lucros da empresa e a capacidade de resposta dos motoristas. Os dados acima descritos podem ser verificados nos gráficos presentes figura 4, onde demonstram a relação entre pedidos (Requests), rácio entre pedidos e viagens (Completion Rate), e por fim o tempo estimado para a chegada de um veículo (ETA).



**Figura 4 - Comparação** entre pedidos de viagens aceites, tempo estimado para a recolha e o rácio pedidos-viagens após concerto da Ariana Grande em Nova York [7].

Como podemos observar, o algoritmo tem um efeito muito poderoso dentro da aplicação, mas como seria caso acontecesse uma situação semelhante à anterior, mas sem a aplicação do algoritmo? O seguinte exemplo ilustra exatamente isso.

No ano novo de 2015 em Nova York aconteceu uma situação onde existia demasiada procura para a oferta de motoristas que havia na altura. A diferença entre estes dois exemplos foi que devido a um problema nos servidores da Uber, o algoritmo não foi aplicado para este acontecimento.



**Figura 5** - Comparação entre pedidos de viagens aceites, tempo estimado para a recolha e o ratio pedidos-viagens em Nova York no ano novo de 2015.

Como podemos observar na figura 5, o bloco a vermelho nos três gráficos representa o período onde o algoritmo não atuou. Efetuando uma comparação entre as duas situações (Concerto da Ariana Grande e Ano novo de 2015) podemos observar que na segunda situação o número de pedidos foi 4 vezes superior, o tempo estimado de viagem foi 2 vezes maior e o rácio entre pedidos e viagens baixou para 10%. Podemos assim aferir que o algoritmo tem grande peso no cálculo final dos preços atribuídos às viagens de modo a maximizar sempre o rácio entre pedidos e viagens.

De forma resumida e relacionando um exemplo de uma viagem na área de Lisboa temos o seguinte preçário fixo para uma viagem normal:

**Tabela 1**-Tabela de tarifas fixas na cidade de Lisboa [4]

Descrição da Taxa	Valor em euros (€)
Tarifa base	1.00€
Taxa de Reserva	0.00€
Tarifa mínima	2.5€
Valor por minuto	0.10€
Valor por km	0.65€
Taxa de não comparência do utilizador	2.5€

Taxa normal de cancelamento iniciado pelo utilizador	2.50€
------------------------------------------------------	-------

A tarifa base é adicionada ao custo total no momento da recolha do utilizador.

Durante a viagem é adicionado ao preço final a taxa de reserva (fixa), a tarifa mínima (fixa), o valor por minuto e por fim o valor por quilómetro.

Apesar de a Uber ser considerada uma plataforma inovadora no mercado, tal não significa que seja perfeita. A Uber suporta toda a sua operação com base no seu algoritmo e decide todos os preços unilateralmente, o que leva muitas vezes a que os motoristas se sintam abandonados pela empresa pelo fato de serem estes a pagar todos os custos associados com esta profissão [6].<sup>8</sup>

### 3.2.2. Algoritmo do Táxi

Um taxímetro, apresentado na Figura 6, é um aparelho obrigatório para todos os táxis e cujo principal objetivo é determinar, a partir do tempo gasto e distância percorrida, um preço final para o viajante por uma viagem. Os preços são definidos por área, o que significa que um trajeto com a mesma distância e mesmo tempo podem ter preços diferentes em diferentes áreas [9].



**Figura 6** - Exemplo de um Taxímetro nos Estados Unidos da América

<sup>8</sup> É complicado por vezes os motoristas conseguirem obter lucro no fim do ano pois, quando somam as despesas do automóvel como gasóleo, seguro, desgaste do interior e do exterior, o investimento num carro novo, verificam que o ganho muitas vezes não supera o gasto, o que pode se tornar preocupante para o futuro da plataforma no mercado [6].

As taxas também podem ser diferentes consoante o número de passageiros, se existe ou não bagagem ou ainda se o trajeto inclui zonas com taxas diferentes. O taxímetro funciona com base em impulsos elétricos para medir tempo e distância [9].

Para o cálculo da distância o taxímetro necessita de um sensor conectado à transmissão do veículo, em que este transmite um sinal elétrico sempre que uma determinada distância é percorrida (configurável de país para país). Para o cálculo do tempo, o taxímetro possui um contador interno para que quando uma certa quantidade de tempo é atingida este manda também um sinal elétrico para informar o computador interno que necessita de adicionar um valor ao preço atual da viagem.

Por exemplo, um taxímetro pode cobrar uma taxa sempre que 100 metros são percorridos, mas o sensor irá enviar um impulso elétrico sempre que o veículo percorra 1 metro. Assim sendo o taxímetro pode ser configurado para que quando o número de impulsos seja igual a 100 (100 impulsos = 100 metros, 1 impulso por metro) adiciona um valor monetário ao preço atual da viagem [9].

Tendo em conta um exemplo real, em Nova York, só o facto de um passageiro entrar no táxi já tem um custo base de \$2,5. Para cada 0,2 milhas são adicionados 50 cêntimos ao preço total. Além disto, podem ser adicionadas taxas de acordo com a hora específica de quando é feito o serviço, como por exemplo das 4PM-8PM (16h00 - 20h00) é adicionado \$1 à tarifa base e depois das 8PM (20h00) é adicionado \$0,5 à tarifa base [10]. Fazendo uma análise aos preços praticados em Nova York podemos obter a seguinte tabela:

**Tabela 2** - Tabela de tarifas fixas dos táxis na cidade de New York [10]

<b>Descrição</b>	<b>Preço (\$)</b>
Tarifa Base	2,5
1/5 milhas acima de uma velocidade de 12 mph	0,5
60 segundos quando o veículo está imóvel	0,5
Sobretaxa durante a noite das 20:00 às 06:00.	0,5
Hora de ponta das 16h às 20h	1

Encontrar uma fórmula geral para o cálculo do preço de um taxímetro é impossível pois cada cidade tem as suas taxas e maneiras de cálculos diferentes e por essa razão cada taxímetro tem de ser configurado para uma cidade específica respeitando as legislações dessa mesma cidade. Tenhamos em atenção que esta fórmula não é oficial pois as

fabricantes dos taxímetros não revelam a fórmula exata para o cálculo de uma viagem. Contudo, a equação abaixo descrita ira refletir aproximadamente o cálculo do preço na cidade de Nova York para uma viagem de um ponto A -> B [11].

$$Pt = Tb + (Pm \times M) + (Mi \times Pi) + Prh + Tn$$

**Equação 2** - Cálculo do preço de uma viagem de um ponto A -> B num táxi na cidade de Nova York [11]

Sendo que as variáveis acima representadas são:

$Pt$  = Preço total para a viagem de um ponto A -> B.

$Tb$  = Tarifa base em Nova York.

$Pm$  = Preço por milha.

$M$  = milhas percorridas na viagem.

$Mi$  = Número de minutos que o táxi permaneceu imóvel.

$Pi$  = Preço por cada minuto que o táxi permaneceu imóvel.

$Prh$  = Sobretaxa por hora de ponta.

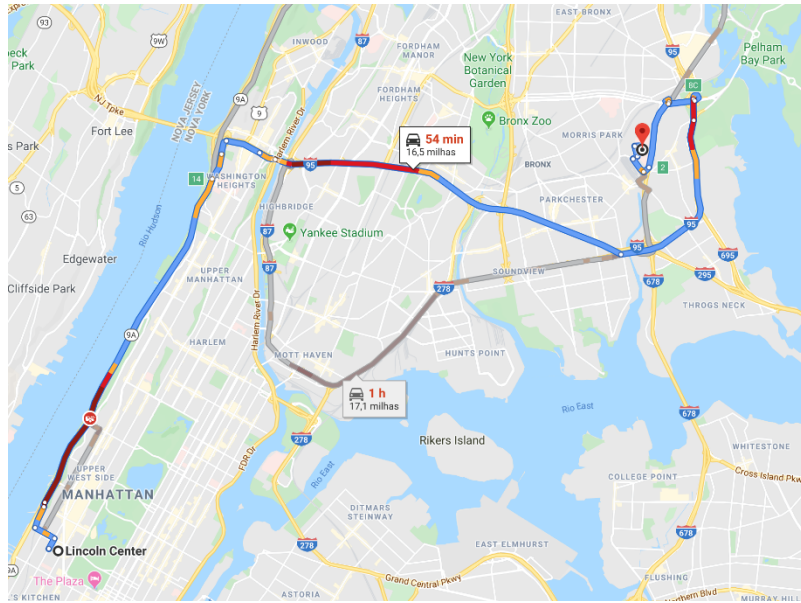
$Tn$  = Sobretaxa noturna.

Vamos ter em conta o seguinte exemplo: Estou em Nova York e pretendo fazer uma viagem desde “*Lincoln Center for the Performing Arts*” para “*Bronx Behavioral Health Center*” às 18h<sup>9</sup>. A distância entre dois pontos é de 17 milhas na rota mais rápida e demorará cerca de 54 minutos numa típica terça-feira em plena hora de ponta.

Vamos também assumir que estaremos parados no trânsito durante 3 minutos durante o tempo total de viagem devido ao trânsito. Na figura 7 é apresentado o trajeto a seguir pelo taxista.

---

<sup>9</sup> Link para o percurso de “*Lincoln Center for the Performing Arts*” para “*Bronx Behavioral Health Center*” Website : <http://tiny.cc/ndy4tz>



**Figura 7** - Trajecto entre “Lincoln Ctr for the Performing Arts (Lincoln Center for the Performing Arts)” e “Bronx Behavioral Health Center”

Considerando o exemplo apresentado acima, obtemos um preço aproximado de quanto iria pagar o passageiro pela viagem utilizando a seguinte equação.

$$P_t = T_b + (P_m \times M) + (M_i \times P_i) + P_{rh} + T_n$$

**Equação 3** – Exemplo prático do cálculo do preço de uma viagem utilizando o algoritmo do Táxi convencional 1

$T_b = 2,5\$$  (respeitando os valores da tabela 2)

$P_m = \text{Preço por } 0,2 \text{ milhas} / 0,25 \text{ milhas} = 2,5\$$  (respeitando os valores da tabela 2)

$M = 17 \text{ milhas}$  (segundo o exemplo acima descrito)

$M_i = 3 \text{ minutos}$  (segundo o exemplo acima descrito)

$P_i = 0,5\$$  (respeitando os valores da tabela 2)

$P_{rh} = 1\$$  (respeitando os valores da tabela 2)

$T_n = 0\$$  (segundo o exemplo acima descrito)

$$P_t = 2,5 + (2,5 \times 17) + (3 \times 0,5) + 1 + 0 = \$47,5$$

**Equação 4** – Exemplo prático do cálculo do preço de uma viagem utilizando o algoritmo do Táxi convencional 2

Segundo a equação acima o passageiro irá pagar cerca de 47.5 dólares para a viagem descrita no exemplo anterior.

A fórmula descrita acima para o cálculo do preço de uma viagem de táxi tem a vantagem de não ser cobrado valores extras ao cliente caso exista muita procura, mas pouca oferta. Também como os valores utilizados para efetuar o cálculo do custo final são tabelados pelas entidades locais, garante um certo nível de confiança ao consumidor pois assim caso este suspeite que o valor cobrado é demasiado alto poderá sempre confirmar com as entidades responsáveis que fiscalizam o serviço de táxi da área onde foi realizada a viagem e assim confirmar se o valor pago está dentro dos valores comuns para o mesmo trajeto ou trajetos semelhantes entre o ponto de origem e o ponto final da viagem.

### 3.2.3. Comparação entre o Algoritmo da Uber e do Taxímetro

A principal diferença entre os serviços foca-se no método de pedir uma viagem e na comodidade que a Uber oferece.

Podemos ainda perceber que os algoritmos para o cálculo dos preços não são assim tão distintos.

Tanto a Uber como o Táxi calculam os seus preços a partir de uma tarifa base e por distância percorrida. Contudo, podemos encontrar algumas diferenças como:

- A Uber calcula os seus preços com base em distância percorrida e tempo decorrido deste o ponto de partida, onde o serviço de táxi apenas cobra pelo tempo quando o veículo se encontra imóvel, isto é, quando o táxi está em andamento este não adiciona qualquer tipo de custo associado ao tempo da viagem.
- Quando estamos presentes num período mais concorrido a Uber usa o seu multiplicador  $Vhc$  para aumentar o seu preço com o intuito de maximizar a taxa de resposta mesmo que o preço possa aumentar exponencialmente. Esta estratégia não acontece no serviço de táxi pois os preços são tabelados e em teoria o táxi num período como este terá preços muito mais acessíveis.

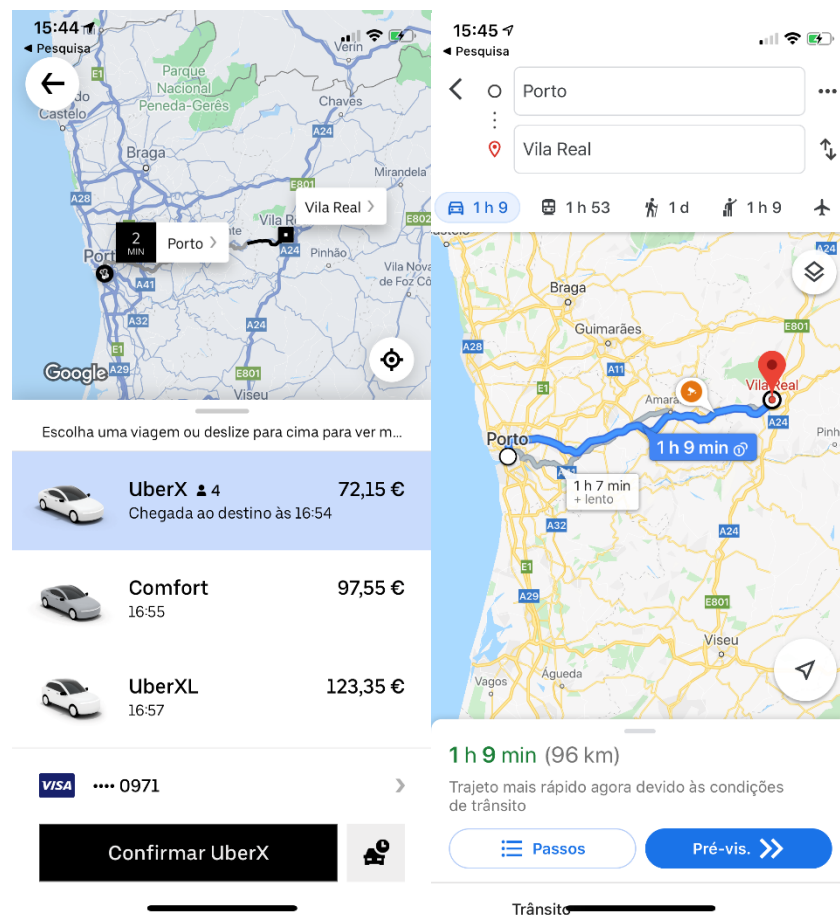
Se compararmos ambos os algoritmos com base no preço, é impossível dizer que o algoritmo da Uber é melhor do que o do táxi, ou vice-versa, pois este algoritmo tem como base calcular um preço final consoante leis, percentagens de lucro esperado para as empresas e custos fixos nas viagens. Portanto não existe um algoritmo melhor, mas sim

dois algoritmos diferentes com o mesmo propósito, fazer com que a empresa maximize os seus lucros [28].

Um problema com estes algoritmos é o facto de realizarem cálculos iguais para veículos diferentes. É o fator que não é tido em conta e que pode levar ao abandono de motoristas, pois as próprias viagens no fim das contas podem não compensar e por vezes levam a prejuízos dos próprios motoristas [6].

Vamos ter em consideração o seguinte exemplo: Temos dois veículos que irão percorrer uma distância de aproximadamente 100 km numa viagem do Porto a Vila Real (Figura 8).

Dos dois veículos um é movido a gasolina e o outro é movido a energia elétrica. Assumindo que esta viagem irá se realizar na Uber obtemos as seguintes informações.



**Figura 8** - Viagem a partir do Porto até Vila Real através da Uber.

Como podemos observar na Figura 8, a viagem proposta tem cerca de 100 km, demorará perto de 1h e 9 minutos e terá um custo de 72.15 euros. A partir deste valor analisemos qual o valor líquido que ambos os motoristas recebem [11].

**Tabela 3** - Comparação de custos de um percurso entre um veículo elétrico e um a gasolina [11].

Parâmetros a considerar	Veículo Elétrico	Veículo a Gasolina	Notas
<b>Valor Bruto</b>	72,15€	72,15€	-
<b>Comissão para a Uber</b>	-18,03€	-18,03	A Uber cobra uma comissão de 25% por cada viagem realizada dentro da plataforma [12]
<b>Combustível Gasto</b>	-3,75€	-11,05€	O consumo medio de um veículo elétrico é de 15 kWh / 100 km onde o preço por cada kW/h é em média 0,25€. [13] [16] O consumo médio de um veículo a gasolina é de 4l/100 km onde o preço por cada litro de gasolina 95 simples é de 1,579€ à data. [15]
<b>Seguro + Imposto único de circulação (IUC) + Inspeção</b>	-2€	-3,25€	Assumindo um valor para ambos os seguros de 190€/ano (seguro contra terceiros). [15] IUC veículo elétrico = 0€. IUC veículo gasolina = 136,72€ [14]  Inspeção para ambos os veículos = 30€  Para este exemplo assumimos que um veículo em média percorre 30km/dia (referencia) o que corresponde a um total de 10950 km / ano.  Valor para veículo a gasolina para este parâmetro por km = 0,032€/km  Valor para veículo elétrico para este parâmetro por km = 0,02€/km
<b>Manutenção: Filtros de ar, troca de filtros do óleo e mudança de óleo</b>	0€	-2,19€	Para este parâmetro assumimos a revisão mais comum que um português realiza no seu veículo. Cada revisão tem um custo medio de 120€ e normalmente um condutor realiza duas revisões ao ano. Assumindo que um condutor realiza 10950km ao ano isto dá um valor de 0,00219€/km. [15]
<b>Lucro Motorista</b>	48,37€	37,63€	-

No exemplo acima foram usado os dados de um Tesla Model 3 para o veículo elétrico e um Renault Mégane para o veículo a gasolina, ambos os veículos são usados frequentemente na plataforma da Uber.

Neste exemplo são consideradas as variáveis mais comuns dentro dos custos normais que uma pessoa comum tem com o seu veículo durante a sua vida útil.

Como podemos observar na Tabela 3 o condutor de um veículo elétrico ganha mais 22% numa viagem de 100km quando comparado com o condutor de um veículo a gasolina.

Se a Uber otimizasse o seu algoritmo para ter em conta o desgaste do veículo e os custos reais específicos de cada veículo, não só poderia influenciar os próprios motoristas a investir em veículos elétricos para baixarem o custo para o utilizador como para os motoristas, como também levaria a viagens mais acessíveis para o utilizador e consequentemente a mais viagens realizadas pelos motoristas [29].

Se utilizássemos o serviço de táxi para realizar este teste iríamos chegar ao mesmo resultado pois ambos os algoritmos têm como base os mesmos parâmetros como explicado no início desta secção.

Para concluir este raciocínio podemos admitir que ambos são justos por não calcularem diferentes preços dependendo dos veículos utilizados, mas, leva a certos motoristas a ganharem mais do que outros. Podemos admitir também que apesar de justos, os algoritmos numa perspectiva de sistema e de otimização não são os mais eficientes pois para o ser deveriam ter em conta mais variáveis como por exemplo as referidas na tabela 3.

De uma forma geral os algoritmos foram pensados e desenhados a partir de médias de tempos e por distâncias percorridas numa determinada cidade, daí existirem preços diferentes entre cidades distintas. Os condutores da Uber são basicamente “contratados” por um algoritmo, porque todos os estes condutores não têm nenhum vínculo contratual com a empresa, o que os torna freelancers e assim usar a plataforma da Uber para conseguir ganhar algum dinheiro extra. Estes algoritmos como já explicados anteriormente, são muito otimizados para maximizar o lucro da empresa e não o lucro dos motoristas, o que leva muitas vezes os próprios condutores a abandonar a Uber e dedicar-se a outro trabalho para que consiga obter mais rendimentos [28].

### 3.3. Cálculo do Trajeto Ótimo

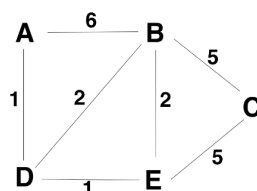
Quando usamos o *Google Maps* ou o *Apple Maps* para encontrarmos o caminho mais curto para o local que pretendemos ir, muitas vezes não nos apercebemos na quantidade de processamento envolvido para conseguir algo tão simples como uma rota de um ponto

A para um ponto B. Graças aos nossos smartphones e a serviços como os acima referenciados, permitem a criação de novas ideias de negócio como aplicações de procura de restaurantes, encontrar um hotel quando formos de férias e outros. Estes novos negócios são muitas vezes suportados pelo facto de estes aparecerem e serem recomendados pelas próprias aplicações de mapas, pois o utilizador está próximo de um destes locais. Estes serviços não só nos ajudam a ir para um novo local eficientemente, como também tem um contributo importante destes pequenos negócios de restauração até grandes cadeiras hoteleiras [30].

Estes serviços só são possíveis devido ao desenvolvimento de algoritmos criados por matemáticos e cientistas da computação durante o século XX. Neste subcapítulo irá ser explorada a base de todos os algoritmos deste género e as últimas inovações neste sector que permitem usar estes serviços facilmente.

### 3.3.1. Algoritmo de Dijkstra

O algoritmo de Dijkstra é a base de todos os serviços de mapas que conhecemos atualmente [31]. Este algoritmo foi criado por Edsger Dijkstra na Holanda em 1959 durante o seu trabalho numa máquina de computação paralela designada ARMAC [32]. Este algoritmo foi criado para demonstrar que esta máquina era capaz de resolver um determinado problema no mundo real. Dijkstra afirmou que *“Para realizar uma demonstração para pessoas fora da área da computação precisas de te um problema que estas pessoas possam entender”* e tendo em conta isto, percebeu que a resposta da máquina teria de ser simples e facilmente entendida por todos. Para isto foi criado um programa que calculasse a rota mais curta entre duas cidades da Holanda criando assim a seguinte pergunta: *“Qual a rota mais curta para viajar de Rotterdam a Groningen”* [18]: Vamos considerar o seguinte grafo presente na Figura 9:



**Figura 9** - Exemplo de um grafo para aplicação do algoritmo de Dijkstra

O principal objetivo deste algoritmo é encontrar o caminho mais curto entre dois vértices, onde este usa o peso de cada caminho como facto principal para obter a resposta. Tendo em conta o exemplo representado na Figura 9 vamos assumir que pretendemos encontrar o caminho mais curto entre o vértice A ao E [19]. Para esta rota vamos apenas considerar 2 caminhos possíveis:

$$A \rightarrow B \rightarrow E \text{ e } A \rightarrow D \rightarrow E$$

Se olharmos para o caminho  $A \rightarrow B \rightarrow E$  observamos que de  $A \rightarrow B$  temos um peso de 6. Sendo que B tem um caminho direto para E o seu custo para percorrer o caminho  $B \rightarrow E$  terá um peso de 2. Se juntarmos os dois pesos relativos a este caminho temos um peso total de 8.

Agora detalhando o segundo caminho  $A \rightarrow D \rightarrow E$  verificamos que de  $A \rightarrow D$  tem um peso de 1 e de  $D \rightarrow E$  tem um peso de 1 também, portanto este caminho tem um peso total de 2, que por sua vez é muito menor quando comparado com o primeiro caminho.

Se pretendermos obter os caminhos mais curtos entre todos os vértices desde A, obtemos a seguinte tabela (Tabela 4).

**Tabela 4** - Pesos para os trajetos para todos os vértices a partir de A.

<b>Vértice</b>	<b>Distância mais curta desde A</b>	<b>Vértice anterior</b>
<b>A</b>	0	-
<b>B</b>	3	D
<b>C</b>	7	E
<b>D</b>	1	A
<b>E</b>	2	D

De modo simplificado, o algoritmo acaba quando visita todos os vértices dentro do grafo. Normalmente temos como base uma tabela como a representada na Tabela 4 que começa sempre com infinito para todos os vértices não visitados. O algoritmo possui também duas listas onde uma serve para seguir os vértices visitados e outra para seguir os vértices não visitados. Quando a lista dos vértices não visitados estiver vazia quer dizer que o algoritmo já encontrou o caminho mais curto para todos os vértices a partir de um vértice inicial (neste exemplo A).

Usando pseudocódigo podemos escrever o algoritmo de Dijkstra de seguinte forma [32]:

1. Introduzir a distância do vértice inicial para o vértice inicial = 0
2. Introduzir a distância do vértice inicial para todos os restantes vértices = Infinito

REPETIR

3. Visitar o vértice não visitado com a menor distancia conhecida do vértice inicial
4. A partir do vértice atual, examinar os seus vizinhos.
5. A partir do vértice atual, calcular a distância dos vértices vizinhos desde o vértice original
6. Se a distancia calculada do vértice é menor que a distância já conhecida, atualizar a distância mais curta.
7. Atualizar o vértice anterior para cada distancia que foi atualizada.
8. Adicionar o vértice atual à lista dos vértices visitados.

REPETIR ATÉ TODOS OS VERTICES SEREM VISITADOS

Este algoritmo tem imensas aplicações no mundo real como por exemplo [20]:

- É usado no Google Maps.
- É usado na rede telefónica.
- É usado no roteamento de IPs.

Apesar das vantagens e implicações que este algoritmo tem na nossa sociedade, este tem algumas desvantagens [20][32]:

- Faz pesquisa cega e por isso perde imenso tempo no processamento.
- Não consegue encontrar o caminho mais rápido quando os pesos são negativos. Isto leva a grafos acíclicos e na maioria das vezes não é possível obter o caminho mais curto correto.
- A eficiência/complexidade deste algoritmo pode ser expressa numa notação Big-O
- Se a *Fibonacci heap* for usada em então a complexidade será  $O(|E| + |V| \log |V|)$
- Se usarmos a pilha binária como referência então a complexidade será  $O(|E| \log |E|)$

Este algoritmo como já foi referido anteriormente é a base de muitos serviços importantes para o quotidiano das nossas vidas. Mas existem outras aplicações para este algoritmo como por exemplo no planeamento de trajetos na área de robótica, que permitem aos robôs encontrar o percurso mais curto para chegar ao destino. Apesar deste algoritmo ter sido criado em 1959, o seu uso e aplicações estão bem presentes hoje em dia em muitas áreas de negócio e de investigação [33].

### 3.3.2. Inovações na área do cálculo do trajeto mais rápido.

Hoje em dia o algoritmo utilizado nos serviços de mapas tem de ter em conta muitos mais fatores do que os previstos por Dijkstra. [31] Para obter o trajeto mais rápido para um determinado destino o algoritmo tem de ter em consideração não só o trajeto, mas obras na estrada, desvios, se existe portagens ou não, polícia e trânsito.

Um exemplo é o Google Maps, sendo o serviço mais avançado até a data. Para o Google Maps conhecer quais os percursos com mais e menos trânsito este depende de duas fontes de informação: Dados de viagens semelhantes, tempo médio da viagem no mesmo traçado a horas semelhantes e através de dados enviados em tempo real pelos sensores de movimento dos *smartphones* dos utilizadores.

Nas versões iniciais do Google Maps este usava os dados de aparelhos de deteção de trânsito inseridos pelo governo maioritariamente nas autoestradas e estradas primárias, o que se mostrou uma limitação pois não havia dados suficientes para todas as estradas do País.

Em 2009 a Google começou a recolher os dados dos seus utilizadores com *smartphones android* para melhorar a precisão das suas previsões de trânsito. Sempre que estes usavam o Google Maps o telefone enviaria dados anonimamente sobre a sua velocidade média do seu veículo [21].

Outra plataforma também interessante é a Waze. Esta tem serviços muito semelhantes ao Google Maps, mas consegue detetar outro tipo de informações úteis para o condutor como por exemplo Operações-Stop, estado da estrada, condições meteorológicas, etc. A principal diferença entre o Waze e o Google Maps é que esta plataforma usa a sua base de utilizadores para adicionar este tipo de informações onde estas são recompensadas por pontos dentro da aplicação [22].

Os serviços acima descritos só são possíveis graças as investigações e desenvolvimento realizado durante o século XX, muitos anos antes da ideia da internet ou dos smartphones. Conforme o explicado acima, todos estes serviços têm como base de alguma forma o algoritmo de Dijkstra, mas todos os serviços e possibilidades que estes fornecem e podem fornecer no futuro é graças ao trabalho de engenheiros de software que utilização cada vez mais as vantagens de um mundo interligado para melhorar a vida dos seus utilizadores.

### 3.4. Análise a Aplicações existentes no mercado

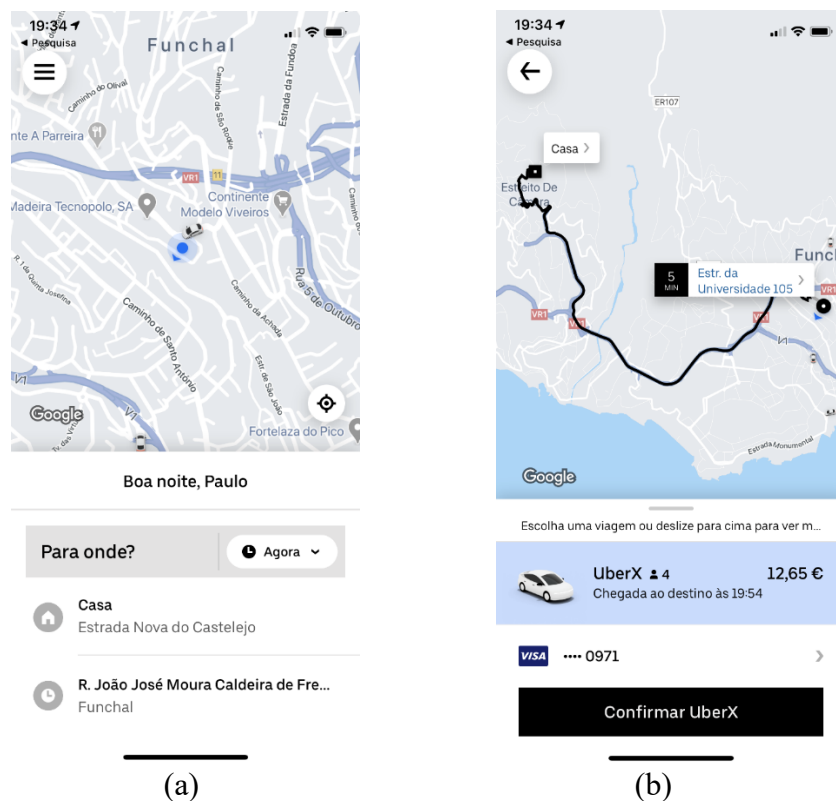
Nesta secção iremos proceder à análise de diferentes Aplicações Móveis existentes no mercado no âmbito de explicar como elas funcionam na ótica do utilizador.

Serão estudadas as aplicações da Uber e da Taxiin<sup>10</sup>. Estas duas plataformas foram escolhidas pelo facto de ambas operarem na Ilha da Madeira no momento do desenvolvimento deste trabalho.

No âmbito desta secção só iremos estudar as principais *interfaces* pelo que vamos descartar écrans secundários (Por exemplo: Login, Registo, Promoções e histórico) pois estes não têm relevância para este projeto.

#### 3.4.1. Estudo da Aplicação da Uber

Apresentamos seguidamente a aplicação tendo como objetivo a requisição de um serviço de transporte na mesma.



**Figura 10** – (a) Ecrã inicial da aplicação da Uber e (b) Ecrã representativo do percurso escolhido pelo utilizador

<sup>10</sup> Taxiin website: <https://www.taxiin.pt>

Na Figura 10 (a) temos representado o ecrã inicial da aplicação da Uber. Este será o ecrã onde o utilizador irá requisitar um veículo para o levar para um determinado destino. Como podemos observar a aplicação contém um mapa central para mostrar ao utilizador onde é que este se encontra no momento. Neste mapa podemos encontrar 2 ícones informativos essenciais para o utilizador:

**-Círculo Azul:** Este ícone representa a localização atual do utilizador e a posição de orientação (direção para que este está virado).

**Veículo:** Este ícone representa um motorista que se encontra disponível para efetuar uma viagem.

Abaixo do mapa podemos encontrar uma área para inserção de texto e uma lista de locais favoritos/recentes.

Quando um utilizador pretende efetuar uma viagem este pode optar por escolher uma morada favorita ou então inserir uma nova morada na área de inserção de texto. Após escolhida a morada iremos ser conduzidos para o ecrã representado na Figura 10 (b).

No ecrã representado na Figura 10 (b) o utilizador irá confirmar a sua viagem. No mapa central é apresentado o percurso a efetuar e o tempo que o motorista irá demorar a chegar ao utilizador.

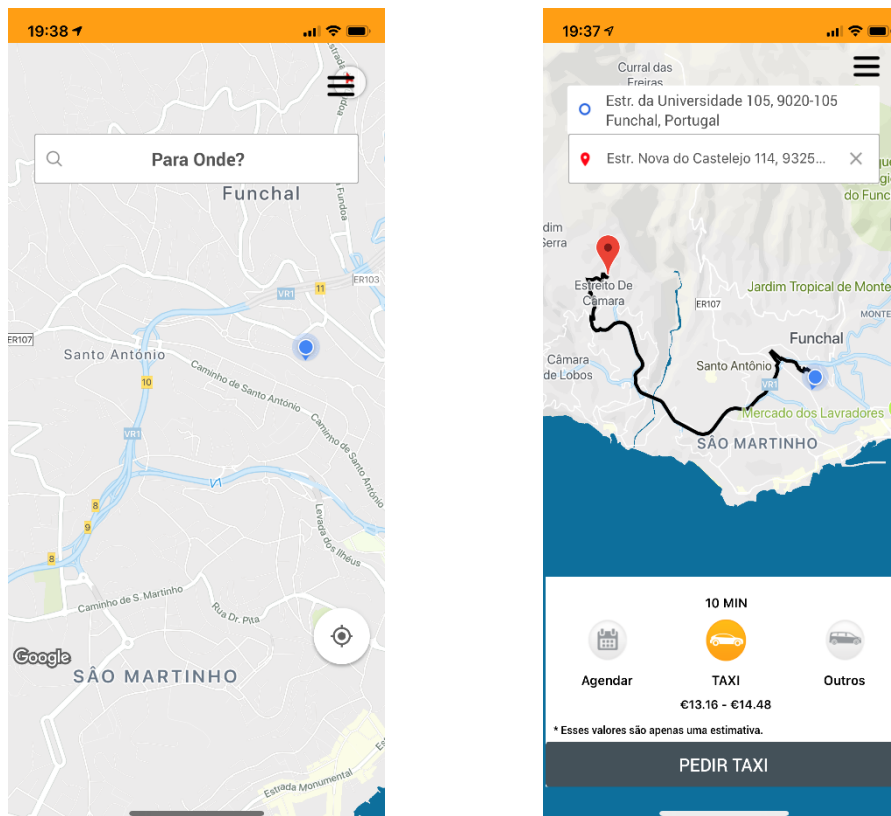
Na secção abaixo do mapa temos o campo para a escolha do serviço onde neste caso apenas temos um serviço disponível que é a Uber X e tem uma capacidade máxima de 4 pessoas. Na mesma secção é possível também encontrar o preço final para esta viagem, qual o método de pagamento que será usado para pagar a viagem e por fim um botão de confirmação. Quando o utilizador estiver pronto a solicitar o serviço, bastará clicar no botão “Confirmar Uber X” para confirmar a o mesmo.

#### 3.4.2. Estudo da Aplicação da Taxiin

A Taxiin é uma aplicação desenvolvida por uma empresa Madeirense, onde tenta oferecer o mesmo serviço que a Uber, mas em vez dos motoristas serem privados este utiliza os profissionais de táxis disponíveis. O principal foco desta aplicação é diversificar o mercado do Táxi para que estes acompanhem a tendência tecnológica. Assim sendo esta funciona de forma muito semelhante à Uber, onde o utilizador pode requisitar um táxi para que este o leve para o lugar desejado.

Como podemos observar na Figura 11, esta aplicação tem uma estrutura muito semelhante à da aplicação da Uber, mas com algumas diferenças. O ecrã principal contém

um mapa central com um ícone a representar a localização do utilizador e uma caixa de texto onde é possível inserir o local para onde o utilizador pretende ir. Podemos observar alguns problemas na UI desta aplicação nomeadamente no canto superior direito onde temos o ícone do menu sobreposto ao ícone de orientação do mapa, tornando assim impossível de virar o mapa para a direção que pretendemos.



**Figura 11** - Ecrã inicial da Taxiin

Após o utilizador escolher a morada para onde pretende ir, a aplicação irá mostrar o ecrã para a visualização do Trajeto pretendido. Como podemos observar na Figura 11, este disponibiliza o Trajeto pretendido, o preço estimado da viagem e opções relativas a reservar esta viagem para o futuro ou escolher outro tipo de veículo com mais capacidade. A Taxiin permite o transporte de 4 a 8 pessoas, transporte executivo e transporte para pessoas de mobilidade reduzida.

A plataforma apenas permite a que os profissionais de táxi (taxistas) a utilizem, sendo que para quem é taxista torna-se fácil integrar-se na plataforma.

Comparando as duas aplicações podemos observar uma diferença notória na sua interface, onde no caso da Uber esta usa uma UI muito mais simples e agradável à vista utilizando cores neutras. A Uber também permite guardar os locais favoritos para que

caso seja um percurso recorrente o utilizador não perca tempo a inserir a morada todas as vezes que necessita se deslocar para esse local, funcionalidade que a Taxiin não permite. A nível dos preços podemos observar que a Uber é tendencialmente mais barata do que a Taxiin pelo menos para o percurso e hora selecionada. Apesar de ambas apresentarem um preço de viagem selecionada, a Taxiin apenas dá uma estimativa, o que pode se revelar mais caro do que o preço apresentado na aplicação. Já na Uber isto não acontece, o preço apresentado será o preço final cobrado ao utilizador o que demonstra mais confiança na aplicação. Apesar de existirem casos pontuais em que o preço na Uber poderá ser mais caro do que apresentado, a própria Uber devolve a diferença caso o acréscimo não tenha sido responsabilidade do utilizador. Na Ilha da Madeira a Taxiin tem a vantagem de permitir o transporte de pessoas com mobilidade reduzida e/ou um grupo maior do que a Uber de momento. Apesar de esta funcionalidade também existir na Uber esta não esta disponível na Região Autónoma da Madeira (RAM) até à data.

## 4. Desenvolvimento da Aplicação

Este capítulo irá descrever todos os aspetos do desenvolvimento deste projeto. Aqui será documentado todo o percurso durante o desenvolvimento focando-se nas seguintes categorias: Requisitos Funcionais e Não Funcionais, Protótipos de baixa e alta fidelidade, Desenvolvimento do Software e por fim o produto final.

### 4.1. Requisitos funcionais e não funcionais

Nesta secção serão introduzidos os requisitos funcionais e não funcionais para este projeto.

Na base dos Requisitos Funcionais queremos que estes exemplifiquem numa forma especifica como é que o projeto irá alcançar os objetivos deste relatório, e para que o utilizador final consiga utilizar a aplicação com todas as funcionalidades esperadas.

Os Requisitos Não Funcionais vão exemplificar todos os requisitos a nível de hardware e desenvolvimento Back-End que permita que a aplicação funcione de forma mais eficiente e possível. Aqui é expectável que a aplicação consiga funcionar de maneira muito rápida para que a qualidade dos dados seja a maior possível. Para isto utilizaremos as versões dos SO e com as funcionalidades mais recentes. A nível de Base de Dados teremos também de ter uma ligação rápida para que todos os dados consigam entrar e sair da aplicação de maneira eficiente. Para isto utilizaremos as últimas versões do MYSQL e uma API desenvolvida em PHP puro sem Framework para que a velocidade do processamento dos dados não seja prejudicada por essas mesmas Framework.

#### 4.1.1. Requisitos funcionais

**Tabela 5** - Requisitos Funcionais

# Requisito	Descrição do Requisito
RF1	Permitir ao utilizador registar uma viagem feita em aplicações como Uber e Bolt e também feitas num táxi.
RF2	Ter um modo onde o utilizador consiga registar o percurso atual de início ao fim e disponibilizar o valor gasto nessa viagem.
RF3	Sempre que o utilizador termina uma viagem/insere uma viagem feita anteriormente apresentar um ecrã com o custo total da viagem.

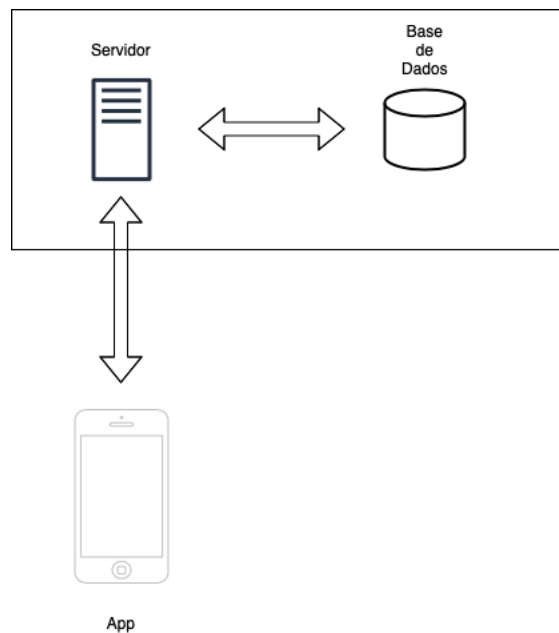
RF4	Guardar um histórico de todas as viagens realizadas pelo utilizador.
RF5	Permitir ao utilizador editar as informações de uma viagem.
RF6	Disponibilizar um ecrã com relatórios diários/mensais/semanais e anuais de todos os custos e ganhos realizados durante esse período.
RF7	Permitir ao utilizador criar múltiplos veículos dentro da aplicação.
RF8	Durante uma viagem que o utilizador registe o Trajeto dentro da App esta terá de ter uma interface apropriada à condução que permita a sua interação sem pôr em risco o condutor e os seus ocupantes.

#### 4.1.2. Requisitos não funcionais

**Tabela 6 - Requisitos não funcionais**

# Requisito	Descrição do Requisito
RNF1	A Aplicação tem que ser implementada utilizando a última linguagem da Apple (Swift).
RNF2	O serviço de mapas terá de ser o Google Maps.
RNF3	A Aplicação tem que suportar iPhones com iOS 11 até a mais recente.
RNF4	A Aplicação terá de suportar todas as variações de ecrã presentes nos iPhones entre o iOS 11 – iOS 13.
RNF5	Para a criação de utilizadores a aplicação terá de usar o “Sign in with Apple” para um registo simplificado sem formulários.
RNF6	A API usada para suportar as funcionalidades da aplicação e para guardar os dados terá de ser implementada em PHP e a estrutura dos dados terá de ser JSON.
RNF7	A Base de dados terá de ser implementada via MySQL.
RNF8	A API terá de ser escalável caso no futuro possa ser implementada uma aplicação Android.
RNF9	A Aplicação necessita de ser muito rápida para que consiga processar todos os dados sem muita margem de erro associada a falta de performance.

## 4.2. Arquitetura Geral do Sistema



**Figura 12** - Arquitetura Geral do Sistema

A arquitetura do sistema desenvolvido (Figura 12) contém uma base de dados, onde são guardados todos os dados necessários para o bom funcionamento do sistema, o código PHP utilizado para a criação da API e que permite a comunicação entre a base de dados e a aplicação iOS, e por fim a aplicação iOS que disponha a interface gráfica que possibilita a interação entre o utilizador e o sistema.

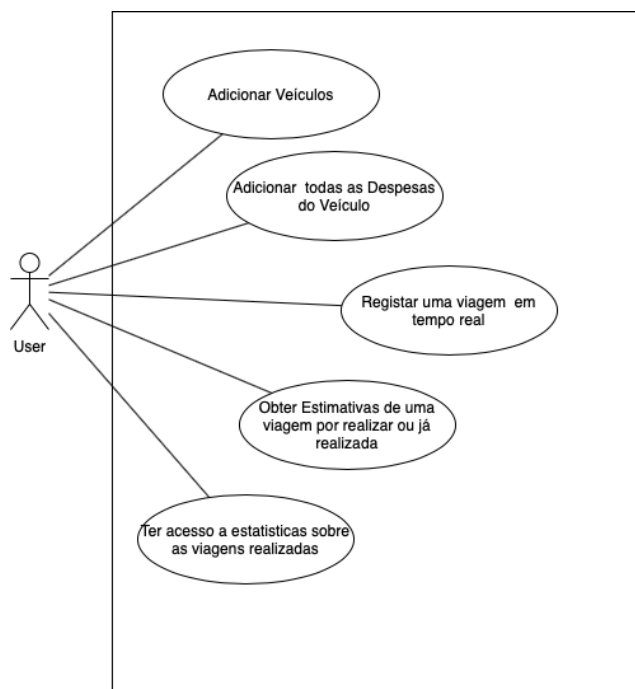
A aplicação iOS foi desenvolvida utilizando o padrão de desenho MVC, que por sua vez é o mais adequado para o desenvolvimento deste projeto devido ao facto de necessitar de comunicação com um servidor para enviar e receber informações atualizadas como por exemplo, todos os dados relativos ao utilizador e do seu veículo, preços atualizados dos combustíveis para que o algoritmo expresse o resultado mais fiável possível e para enviar todas os dados das viagens realizadas dentro da aplicação para que sejam guardadas na base de dados.

Dentro do servidor existe todo o código necessário para que todas as informações dentro da aplicação estejam constantemente atualizadas. A linguagem utilizada para a criação da API foi PHP, que como é uma linguagem simples permite tempos de resposta rápidos entre o servidor e a aplicação iOS.

Por fim é utilizado SQL para a criação e para o envio das *queries* entre a API e a Base de Dados. A Interface Gráfica utilizada para a visualização das tabelas e das informações dentro da base de dados é o PHPMYAdmin.

### 4.3. Casos de utilização

Nesta secção será apresentado os principais casos de utilização dentro da aplicação.



**Figura 13** - Casos de utilização da Aplicação.

### 4.4. Protótipos de baixa e alta fidelidade

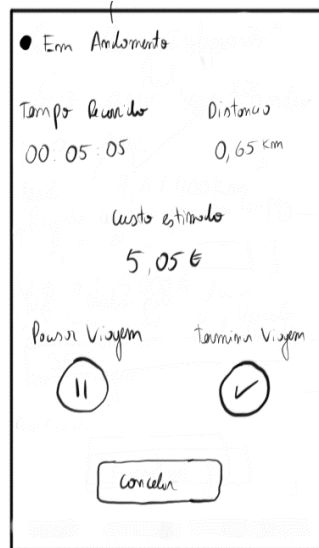
Nesta secção serão apresentados os protótipos de baixa e alta fidelidade. Esta é uma secção importante para este projeto pois será aqui que vamos ter uma primeira visualização de como é que a aplicação final poderá se assemelhar.

#### 4.4.1. Protótipos de baixa Fidelidade

Os protótipos de baixa fidelidade (pbf) são o primeiro esboço de como é que o desenvolvedor pretende que a aplicação final se comporte. Aqui numa primeira interação foram criados alguns écrans de acordo com os requisitos pretendendo sendo que estes são o primeiro esboço da aplicação. Nesta subsecção apenas iremos mostrar os protótipos

para 3 requisitos importantes sendo que os restantes estão disponíveis no Anexo A deste relatório.

**Requisito 1:** “Durante uma viagem que o utilizador registre o Trajeto dentro da app esta terá de ter uma interface apropriada à condução que permita a sua interação sem pôr em risco o condutor e os seus ocupantes.”



**Figura 14** - Protótipo baixa Fidelidade: Ecrã principal para o requisito 1

**Requisito 2:** “Permitir ao utilizador registar uma viagem feita em aplicações como Uber e Bolt e também feitas num táxi”

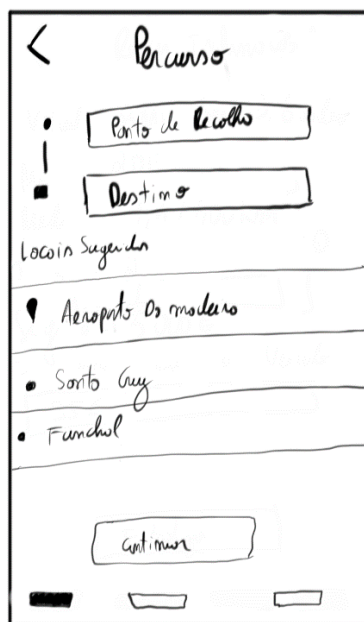


Figura 15 - Protótipo baixa Fidelidade: Ecrã principal para o requisito 2

**Requisito 3:** “Permitir ao utilizador introduzir todos os gastos que este tem com o veículo através de um formulário simples e eficaz.”

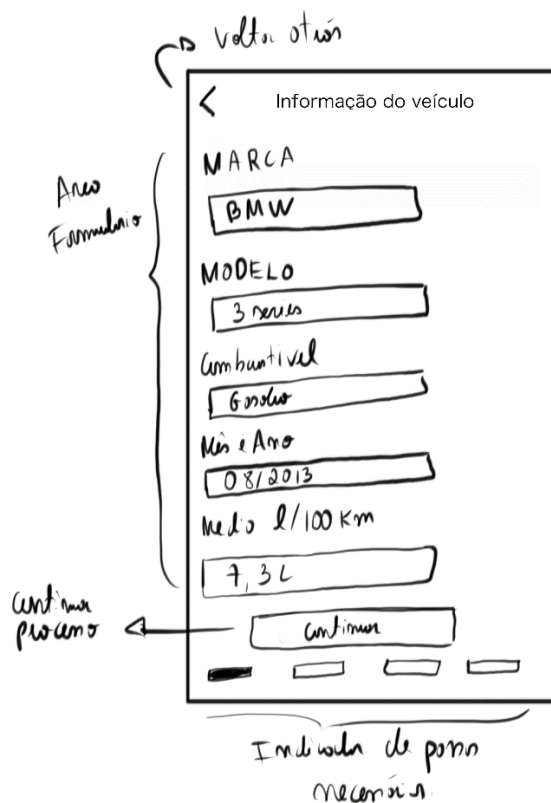


Figura 16 - Protótipo baixa Fidelidade: Ecrã principal para o requisito 3

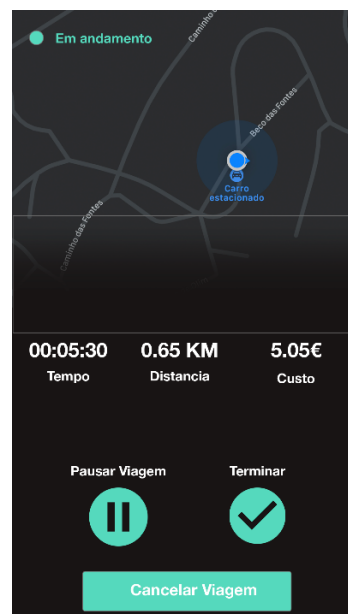
Os Protótipos de baixa fidelidade são muito importantes pois são o primeiro esboço de como poderá se parecer a aplicação final. Claro que estes esboços iniciais são em norma

muito rudimentares, mas caracterizam um dos primeiros passos no desenvolvimento de software.

#### 4.4.2. Protótipos de Alta Fidelidade

Após a demonstração dos protótipos de baixa fidelidade, passamos agora para os protótipos de alta fidelidade (paf). Após a criação dos requisitos e dos pbf os engenheiros e designers já tem uma noção de como a aplicação se deve de comportar e assim criar um protótipo de alta fidelidade que se aproxime muito da aplicação/produto final. Para estes protótipos vamos seguir os mesmos exemplos demonstrados na secção dos protótipos de baixa fidelidade e mostrar os paf dependendo dos 3 requisitos principais da aplicação apresentada nesta tese, sendo que todos os paf criados para este projeto estarão disponíveis na íntegra no Anexo B deste relatório.

**Requisito 1:** “Durante uma viagem que o utilizador registre o Trajeto dentro da App esta terá de ter uma interface apropriada à condução que permita a sua interação sem pôr em risco o condutor e os seus ocupantes.”



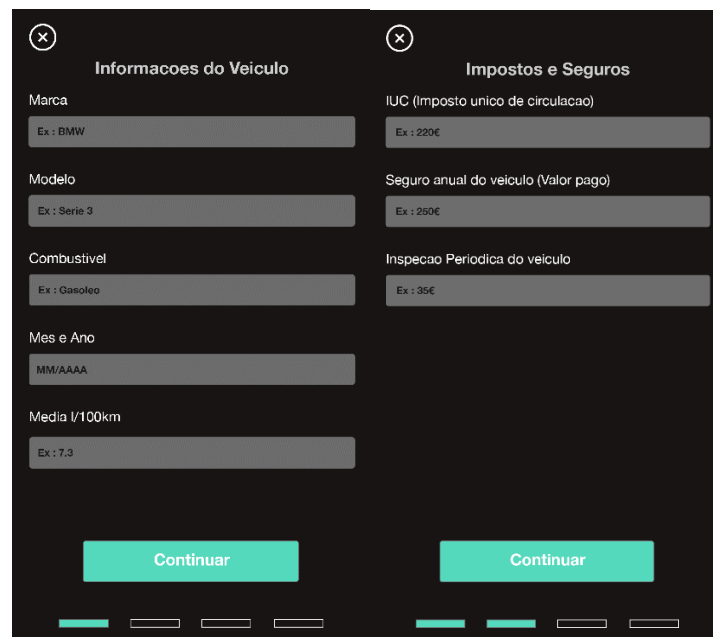
**Figura 17** - Protótipo Alta Fidelidade: Ecrã principal para o requisito 1

**Requisito 2:** “Permitir ao utilizador registar uma viagem feita em aplicações como Uber e Bolt e também feitas num táxi”.



**Figura 18** - Protótipo Alta Fidelidade: Ecrã principal para o requisito 2

**Requisito 3:** “Permitir ao utilizador introduzir todos os gastos que este tem com o veículo através de um formulário simples e eficaz.”

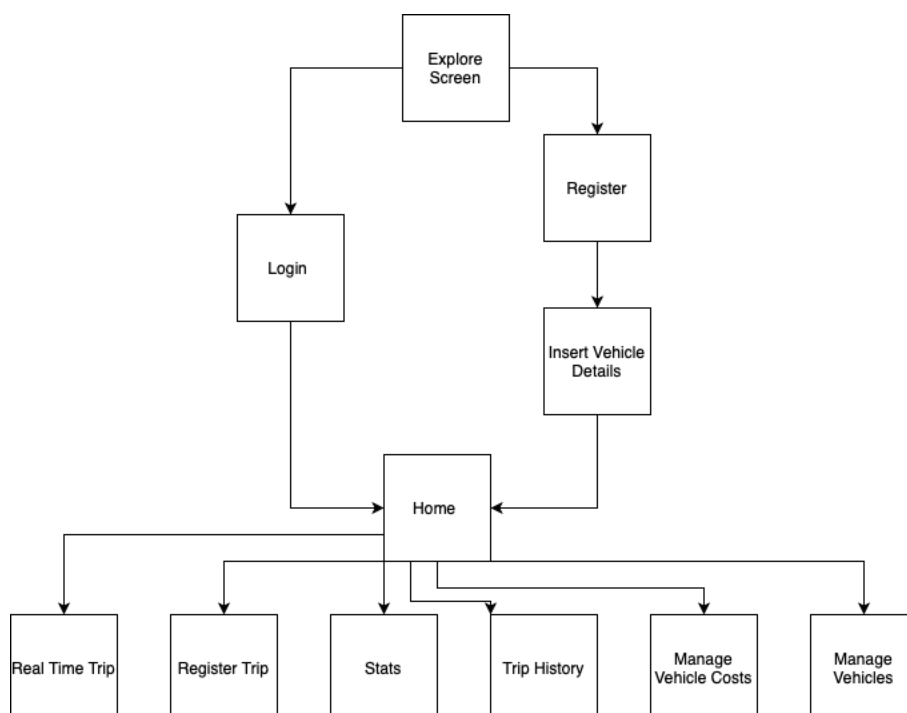


**Figura 19** - Protótipo Alta Fidelidade: Ecrã principal para o requisito 3

Como podemos ver nesta secção os PAF já demonstram a expectativa de como é que a aplicação final deve se assemelhar. Apesar de os PAF não serem uma cópia exata da aplicação final, é um *design* aproximado onde os programadores se podem guiar ao construir a aplicação. Quando comparamos os PBF e os PAF vemos que a estrutura dos objetos se mantém, mas nos PAF já são aplicados estilos e cores e alinhamentos a pensar na UI da aplicação. São nestes protótipos onde a *user experience* do utilizador é pensada e planeada com cuidado para que quando a aplicação chegar a fase de implementação não seja necessário realizar de fazer grandes alterações a UI para aplicar uma melhor UX.

#### 4.5. Mapa de navegação

Nesta secção será apresentado o mapa de navegação da aplicação. Com base neste mapa conseguimos ver como a aplicação será estruturada e que écrans e funcionalidades esta irá ter.



**Figura 20** -Mapa de navegação da aplicação.

## 4.6. Aplicação Final

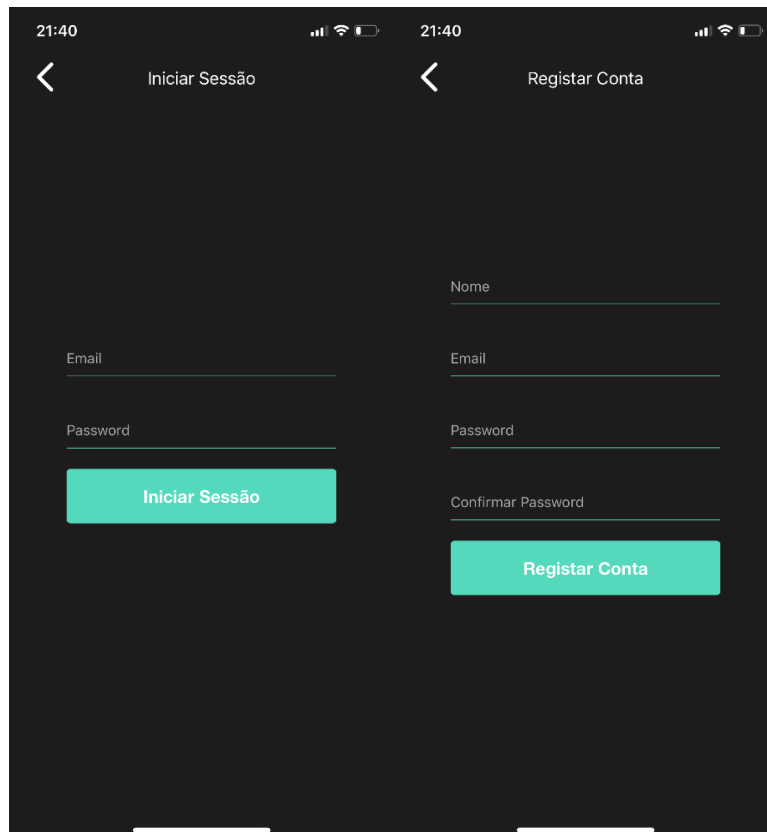
Nesta secção será apresentada a aplicação final com a caracterização de cada ecrã de forma a exemplificar o funcionamento e a UI do projeto desenvolvido.

O ecrã abaixo (*ver Figura 21*) é o primeiro ecrã que o utilizador irá visualizar. Este fornece algumas informações essenciais a um novo utilizador para que este tenha conhecimento do que é suposto a aplicação fazer e os botões propriamente ditos para entrar na aplicação.



**Figura 21** -Explore Screen

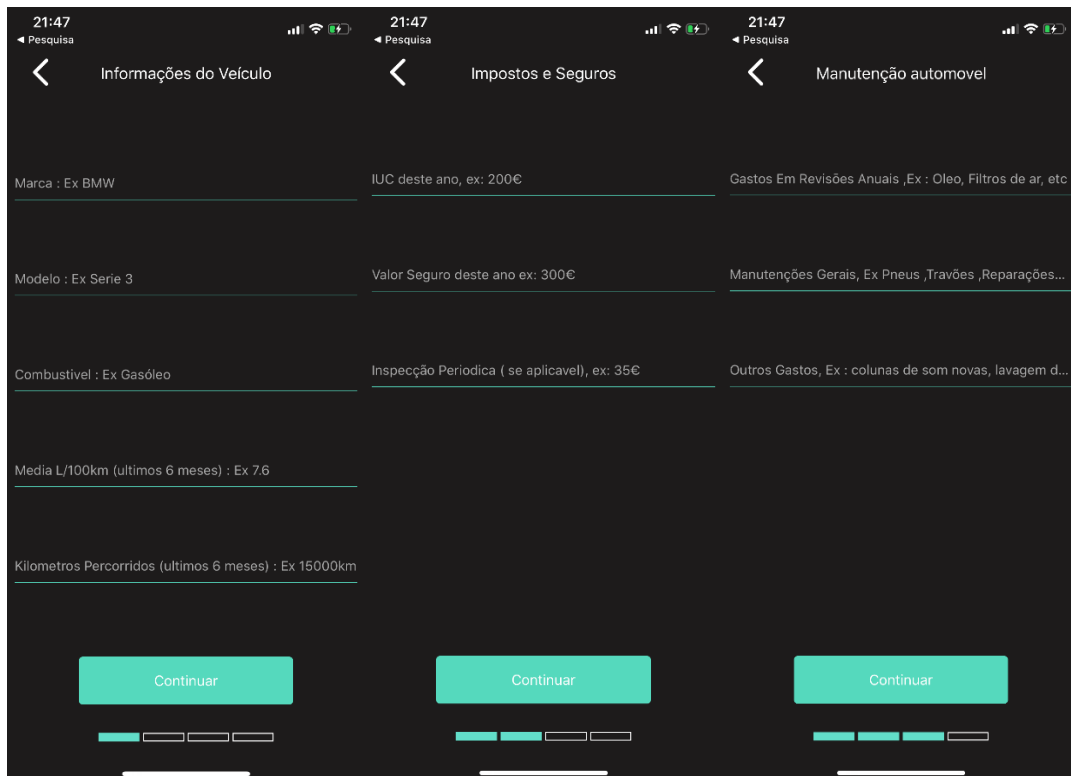
De seguida podemos ver o ecrã de Login e o Ecrã de Registar uma nova conta (*ver Figura 22*). Sendo a simplicidade o foco desta tese só pedimos ao utilizador para fornecer o seu Nome, email e uma password que é encriptada utilizando uma Hash.



**Figura 22** - Ecrã de Login e Registo

Quando o utilizador regista uma conta na aplicação é pedido que preencha um pequeno formulário em que este terá que disponibilizar informações necessárias do seu veículo para que a aplicação e o algoritmo funcionem de forma eficiente. Na figura 23 podemos encontrar os ecrans desse formulário.

Neste formulário o utilizador tem que preencher os dados básicos do seu veículo como: Marca, Modelo, Combustível, Media de combustível e os quilómetros percorridos nos últimos 6 meses. Após os dados do veículo é pedido ao utilizador que preencha os dados relativos aos Impostos e Seguros do veículo e à Manutenção automóvel.



**Figura 23-**Formulário de registo de um novo veículo

Após a conta criada e os dados do veículo devidamente preenchidos irá ser apresentado ao utilizador o Home Screen. Este é o principal ecrã da aplicação onde o utilizador pode encontrar um gráfico com os gastos diários e o custo aproximado de todas as viagens realizadas ao longo do dia. O utilizador irá também encontrar todas as opções relativas as funcionalidades da aplicação. Este ecrã está representado na Figura 24.

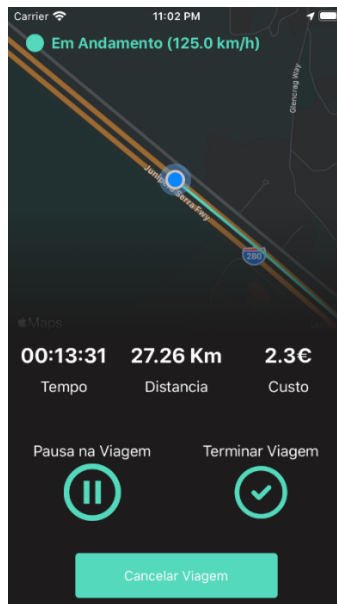


**Figura 24 - Home Screen**

Uma das funcionalidades mais importantes da aplicação é a Viagem em Tempo Real. Neste modo o utilizador poderá seguir em tempo real o custo estimado durante a sua viagem. Neste ecrã é mostrado apenas a informação essencial para que o utilizador não se distraia durante a sua viagem. Neste modo apresentamos o Estado da Viagem (Em andamento, Viagem em Pausa e Parado), a velocidade atual do veículo, um mapa que desenha o percurso do utilizador, o tempo decorrido, a distância percorrida e o custo aproximado da viagem. O utilizador possui 3 ações neste ecrã:

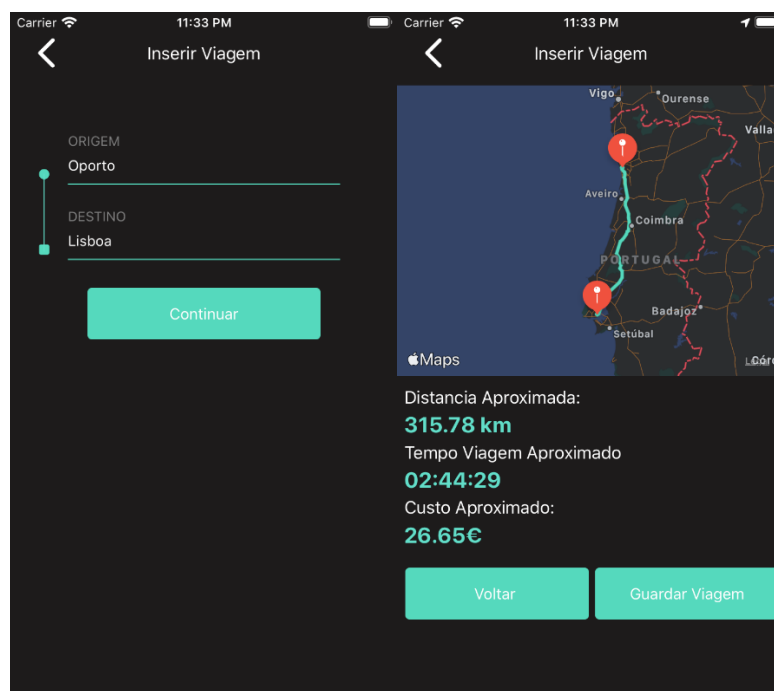
- **Pausa na viagem:** Esta ação foi desenhada para o caso que o utilizador pretenda parar um posto de gasolina, mas não queira terminar a viagem. Com esta ação a viagem entra em modo de pausa em que deixa de contar tempos, distâncias e custos.
- **Terminar Viagem:** Este modo como o nome indica termina a viagem.
- **Cancelar Viagem:** Também como o nome indica, esta ação cancela a viagem.

O ecrã acima descrito esta representado na Figura 25.



**Figura 25 - Ecrã Viagem em Tempo real**

A segunda funcionalidade principal desta aplicação é registar uma viagem. Neste modo permitimos ao utilizador que registre uma viagem feita anteriormente apenas por inserir a morada de origem e a morada de destino. Sabendo as duas moradas a aplicação consegue calcular o custo aproximado da viagem usando apenas os dados fornecidos pelo utilizador. Os ecrans em que esta funcionalidade se encontra está representado na Figura 26.



**Figura 26 – Ecrã de Registrar Viagem**

Tendo duas opções para registar as viagens, a aplicação teria de ter um modo em que permitisse ao utilizador visualizar um histórico das suas viagens. Assim sendo a aplicação contém um ecrã dedicado ao Historial do utilizador. A base de dados guarda a data e hora da viagem, a distância percorrida, o tempo decorrido e o custo da mesma. Também é disponibilizado uma imagem do percurso registado. Poderá visualizar este ecrã na Figura 27.



**Figura 27** – Ecrã de Histórico das Viagens

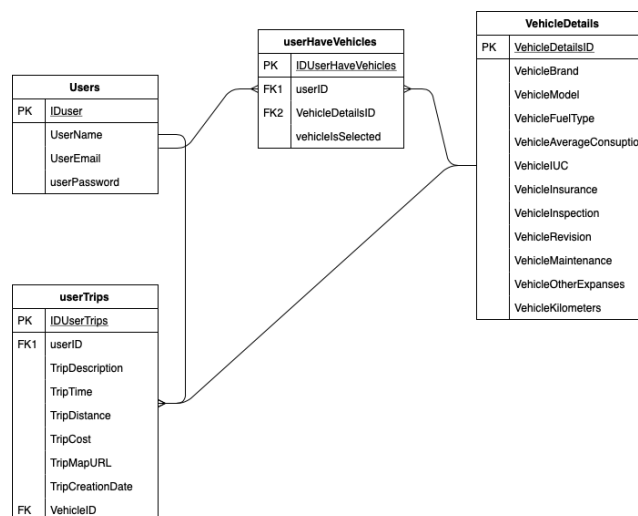
Para concluir esta secção temos por fim uma funcionalidade que permite ao utilizador atualizar ou mudar os custos fixos do seu veículo. Neste próximo ecrã da Figura 28, permite que sempre que aconteça uma despesa extra, o utilizador consegue adicionar essa mesma despesa ao seu veículo para que o cálculo do custo se mantenha eficiente e aproximado ao custo real.



**Figura 28** – Ecrã de Histórico das Viagens

#### 4.7. Base de Dados

Para que o sistema funcione corretamente e para que todos os dados sejam guardados de maneira a facilitar o uso da aplicação por um utilizador de forma normal e eficaz, foi desenvolvida uma pequena base de dados em SQL utilizando a interface PHPMyAdmin. Na Figura 29 podemos encontrar a estrutura da base de dados utilizada para esta tese.



**Figura 29** – Estrutura da base de dados da aplicação

Como podemos observar na Figura 29 esta base de dados tem apenas 4 tabelas: Users, UserHaveVehicles, VehicleDetails e userTrips.

A tabela users guarda toda a informação respetiva aos utilizadores. A tabela UserHaveCars guarda os veículos de cada utilizador. A aplicação está preparada de maneira a possibilitar que um utilizador possua mais do que um veículo pois no cenário imaginado admitimos a possibilidade de por exemplo um taxista acabar o seu serviço e querer registar as viagens realizadas pelo seu veículo particular. A tabela VehicleDetails é responsável por guardar todos os dados de um determinado veículo de um utilizador. É a partir desta tabela que os dados necessários para o algoritmo funcionar são guardados. Por fim temos a tabela UserTrips que é responsável por guardar todas as viagens que um utilizador realizou de modo a conseguir apresentar um histórico das viagens na aplicação.

#### 4.7.1. API

Para que a Aplicação e a base de dados comuniquem entre si foi necessário criar uma API. Esta API foi desenvolvida através de PHP puro e MySQL queries para garantir a performance da aplicação. Durante o desenvolvimento em média cada chamada à API demora entre 0.5 a 0.8 segundos a responder, o que é um tempo aceitável para uma API simples como a implementada neste projeto. Nesta secção irão ser apresentados os ficheiros da API, o seu objetivo e as *queries* SQL necessárias para que a comunicação aconteça.

##### 4.7.1.1. RegisterAccount.php

Este ficheiro é responsável pela criação de um novo utilizador. Esta chamada utiliza o método HTTP POST para esconder os parâmetros pois contem informação privada. Nesta chamada primeiro verificamos se existe algum utilizador já criado com o mesmo email na base de dados para evitar que existam utilizadores duplicados através do seguinte código PHP:

**Tabela 7-**Tabela de entradas e saídas da chamada RegisterAccount.php

Método HTTP	Dados Entrada	Dados Saída
<b>POST</b>	<ul style="list-style-type: none"><li>• Nome do utilizador</li><li>• Email do utilizador</li><li>• Password do utilizador</li></ul>	<ul style="list-style-type: none"><li>• Mensagem de Erro (caso alguma coisa falhe no processo de registo)</li><li>• Mensagem de Sucesso (caso a conta seja criada com sucesso)</li></ul>

Caso não exista nenhum utilizador com o mesmo email, podemos proceder com a criação da conta.

A *password* do utilizador é encriptada utilizando uma hash e caso a criação proceda com sucesso é enviada para a aplicação um JSON com a mensagem de sucesso, o que irá permitir o continuar do processo.

#### 4.7.1.2. CreateVehicleDetails.php

Este ficheiro é responsável pela criação de um veículo e a associação deste a um utilizador em específico. Esta criação é feita da seguinte forma:

**Tabela 8-**Tabela de entradas e saídas da chamada CreateVehicleDetails.php

Método HTTP	Dados Entrada	Dados Saída
<b>POST</b>	<ul style="list-style-type: none"><li>• Marca Veículo</li><li>• Modelo Veículo</li><li>• Combustível</li><li>• Média de combustível</li><li>• IUC</li><li>• Seguro Anual</li><li>• Inspeção Periódica</li><li>• Gastos em Revisões</li><li>• Manutenções</li><li>• Outros Gastos</li><li>• Quilómetros do veículo</li></ul>	<ul style="list-style-type: none"><li>• Mensagem de Erro (caso alguma coisa falhe no processo de adicionar o veículo)</li><li>• Mensagem de Sucesso (caso o veículo seja criado com sucesso)</li></ul>

#### 4.7.1.3. createUserTrip.php

createUserTrip.php é o ficheiro responsável pela criação de uma nova viagem realizada pelo utilizador. Este ficheiro tanto consegue criar viagens em tempo real como viagens feitas anteriormente.

**Tabela 9**-Tabela de entradas e saídas da chamada createUserTrip.php

Método HTTP	Dados Entrada	Dados Saída
<b>POST</b>	<ul style="list-style-type: none"><li>• Descrição da Viagem</li><li>• Tempo da Viagem</li><li>• Distância de Viagem</li><li>• Custo da Viagem</li><li>• URL da fotografia do trajeto percorrido</li><li>• ID do utilizador</li><li>• ID do veículo</li></ul>	<ul style="list-style-type: none"><li>• Mensagem de Erro (caso alguma coisa falhe no processo de adicionar a viagem)</li><li>• Mensagem de Sucesso (caso a viagem seja criada com sucesso)</li></ul>

#### 4.7.1.4. getUserTripHistory.php

Como o nome deste ficheiro indica, getUserTripHistory.php, este ficheiro é o que fornece o histórico de todas as viagens registadas na aplicação pelo utilizador. A nível de complexidade esta chamada é muito simples onde apenas é feita a seleção de todas as viagens associadas a um utilizador e a um veículo.

**Tabela 10** -Tabela de entradas e saídas da chamada getUserTripHistory.php

Método HTTP	Dados Entrada	Dados Saída
<b>GET</b>	<ul style="list-style-type: none"><li>• ID do utilizador</li><li>• ID do veículo</li></ul>	<ul style="list-style-type: none"><li>• Mensagem de Erro (caso alguma coisa falhe no processo de obter o histórico das viagens realizadas)</li></ul> <p>Em caso de sucesso é retornado uma lista contendo:</p> <ul style="list-style-type: none"><li>• ID da Viagem</li><li>• Data da criação da viagem</li><li>• Descrição da Viagem</li><li>• Tempo de Viagem</li><li>• Distância da Viagem</li><li>• Custo da Viagem</li><li>• URL da fotografia do percurso</li><li>• ID do utilizador</li><li>• ID do veículo</li></ul>

#### 4.7.1.5. login.php

Por fim, mas não menos importante este ficheiro é o responsável pela validação do utilizador ao fazer login.

**Tabela 11**-Tabela de entradas e saídas da chamada login.php

Método HTTP	Dados Entrada	Dados Saída
<b>GET</b>	<ul style="list-style-type: none"><li>• Email do utilizador</li><li>• Password do utilizador</li></ul>	<ul style="list-style-type: none"><li>• Mensagem de Erro (caso alguma coisa falhe no processo de validar o login)</li></ul> <p>Em caso de sucesso é retornado os seguintes dados:</p> <ul style="list-style-type: none"><li>• ID do utilizador</li><li>• Uma Flag que informa se o utilizador já tem um veículo associado à sua conta</li><li>• Uma mensagem de sucesso.</li></ul>

#### 4.8. Desenvolvimento do Algoritmo para o cálculo do custo de uma viagem

Para que a aplicação consiga apresentar um custo aproximado para uma determinada viagem, seja ela em tempo real ou registada posteriormente, foi necessário a criação de um algoritmo. Este algoritmo tem uma base semelhante aos algoritmos estudados durante esta Tese, mas com umas pequenas alterações (Equação 5).

$$CT = (Pkm \times TKm) + (Pm \times TM)$$

**Equação 5** - Cálculo para o custo total da viagem

$$Pkm = \frac{mci}{100} \times Pc$$

**Equação 6** - Preço por quilómetro

$$Pm = \frac{\sum Td}{356 \times 24 \times 60}$$

**Equação 7** - Preço por minuto

**Legenda:**

- **CT:** Custo total da viagem
- **Pkm:** Custo por quilómetro
- **Tkm:** Total de quilómetros percorridos
- **Pm:** Custo por minuto
- **TM:** Total de minutos decorridos
- **mci:** média de consumo do veículo que pode ser a média instantânea introduzida pelo utilizador após o fim da viagem ou então a média do veículo introduzida no registo inicial.
- **Pc:** Preço do combustível no momento da viagem (depende do combustível do veículo e do dia)
- **Td:** Total de todas as despesas combinadas

A fórmula utilizada tem uma base muito semelhante as das estudadas no Estado da Arte (Capítulo 2 desta tese). A principal diferença entre a fórmula proposta e as estudadas anteriormente é que os valores dos parâmetros são totalmente dinâmicos dependendo sempre das informações fornecidas pelo utilizador.

O algoritmo proposto divide-se em duas partes fundamentais: **Custo por quilómetro** e **Custo por minutos**. Na parte do *custo por quilómetro* foi decidido calcular este valor a partir do consumo de combustível do veículo visto que este é o fator com mais peso na viagem. O *custo por minuto* de uma viagem tem como base todas as despesas que um veículo tem atualmente no prazo de um ano. Por exemplo, os parâmetros utilizados para o cálculo deste custo são os seguintes: IUC (imposto único de circulação), Seguro, custo da inspeção, custo de todas revisões realizadas ao veículo, custo total de todas as

manutenções efetuadas no veículo e por fim outros custos. De referir que todos estes custos são aplicados apenas ao ano atual, não aos anos do veículo. De seguida é representado o código em SWIFT do cálculo deste valor aplicado ao mundo real (Figura 30).

```
func calculateEstimatePrice(consumo : Float) -> Double
{
    let travelDistanceKm = Float(traveledDistance)
    let minutesPassed : Float = Float(timePassedSeconds)/Float(60)
    let totalCost = (calculatalePricePerKm(consumo: consumo) * Double((travelDistanceKm))) + (calculatePricePerMinute() * Double(minutesPassed))
    return totalCost.rounded(toPlaces: 2)
}

func calculatalePricePerKm(consumo : Float) -> Double
{
    let mediaCombustivel = consumo

    let pricePerKm = ((mediaCombustivel) * (Helper.getPrecoCombustive(combustivel: mainDelegate.currentUserDetails.combustivel ?? "")))/100
    return Double(pricePerKm)
}

func calculatePricePerMinute() -> Double
{
    let iuc = mainDelegate.currentUserDetails.IUC ?? 0
    let seguro = mainDelegate.currentUserDetails.seguroAnual ?? 0
    let inspecao = mainDelegate.currentUserDetails.inspecaoPeriodica ?? 0
    let revisoes = mainDelegate.currentUserDetails.gastosRevisoes ?? 0
    let manutencoes = mainDelegate.currentUserDetails.manutencoes ?? 0
    let outrosCustos = mainDelegate.currentUserDetails.outrosGastos ?? 0

    let sumCosts = iuc + seguro + inspecao + revisoes + manutencoes + outrosCustos
    let pricePerMinute = sumCosts/(365*24*60)
    return Double(pricePerMinute)
}
```

**Figura 30**-Código Swift utilizado para a implementação do algoritmo

#### 4.9. Desenvolvimento da aplicação iOS

A aplicação iOS foi desenvolvida utilizando o IDE Xcode e a linguagem de programação Swift versão 5.0 (a mais recente até a data). A aplicação é nativa e completamente otimizada para o hardware Apple, o que fornece ao utilizador performance e uma curva de aprendizagem baixa pois esta App utiliza muitas funcionalidades disponibilizadas pelo próprio sistema operativo.

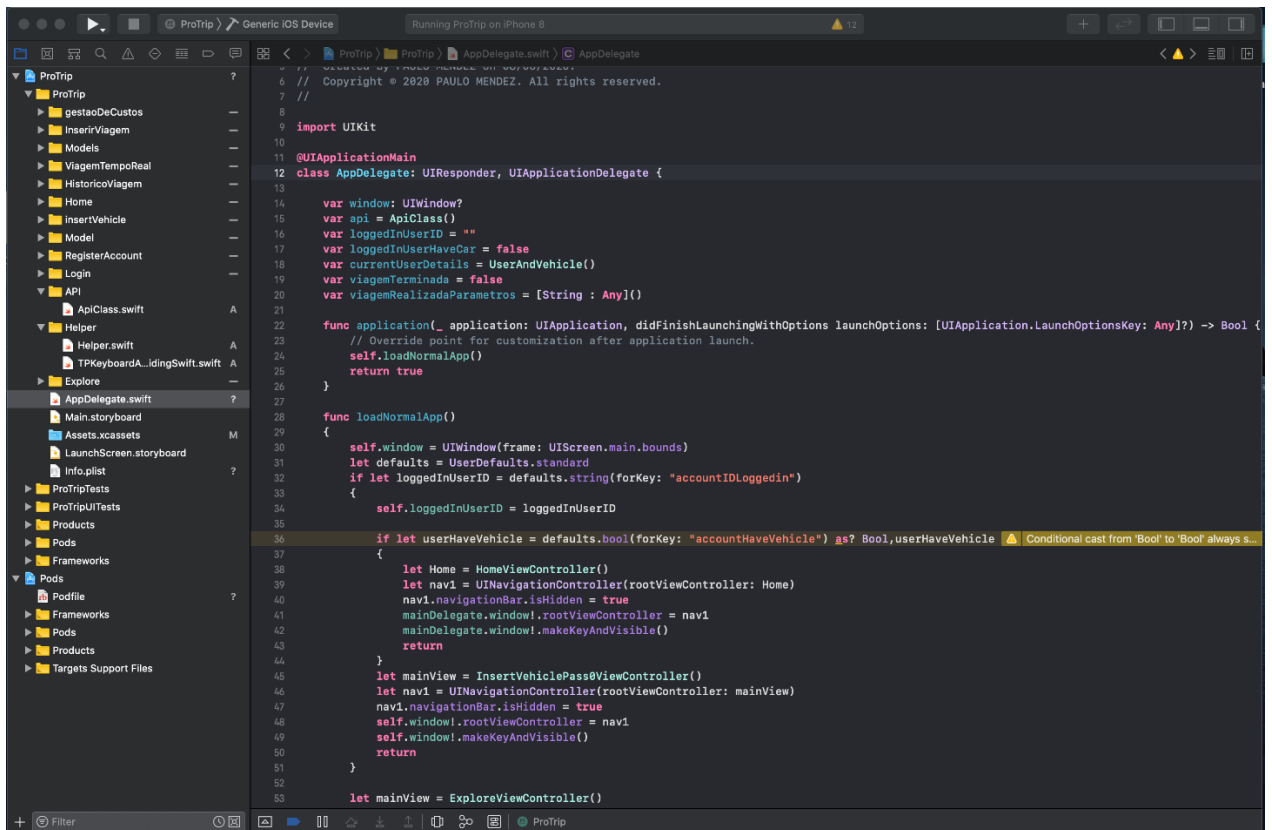


Figura 31 - Projeto Protrip no IDE Xcode

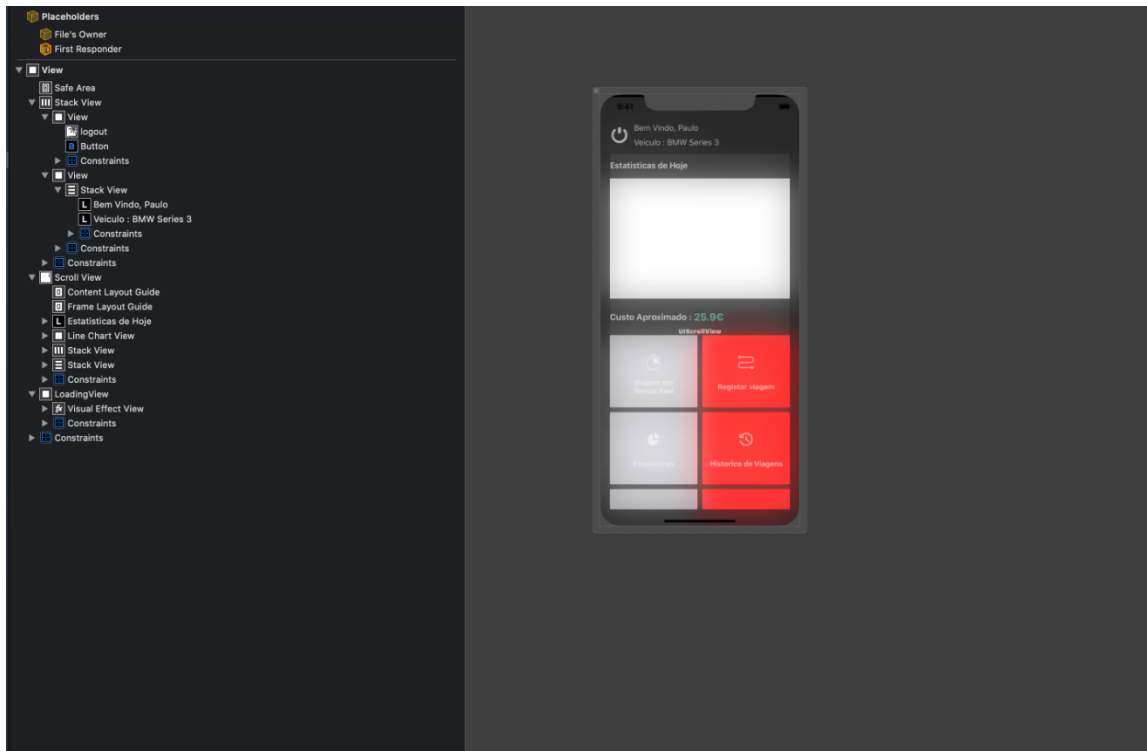
Para testar a aplicação foram utilizados 2 dispositivos, o iPhone 11 Pro e o iPhone 8. Estes dois dispositivos são muito importantes pois representam dois tamanhos de ecrã muito utilizados pelos utilizadores dentro do ecossistema Apple.



Figura 32 - iPhone 8 e iPhone 11 Pro

A arquitetura escolhida para este projeto foi a MVC (Model View Controller) que não só é das mais utilizadas, mas também é das mais simples para criar um novo projeto em iOS.

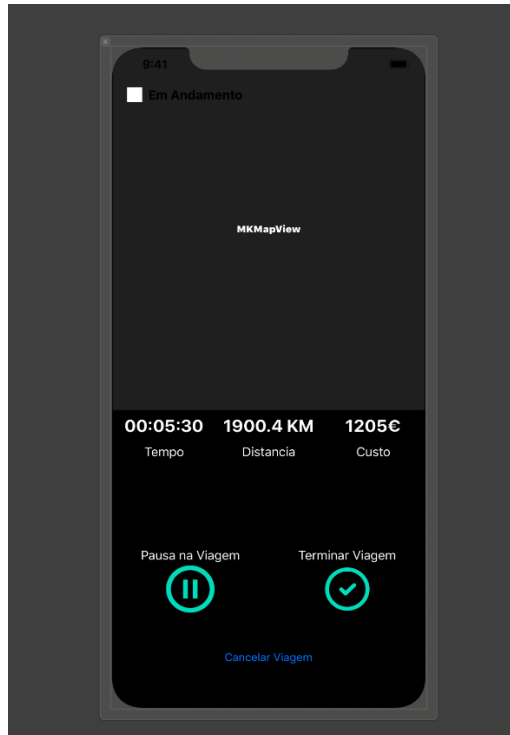
O Xcode torna o processo de criar as *views* e respetivos *controllers* muito mais facilitado quando comparado com outros IDEs. Apesar do Xcode permitir a criação de *views* através de código XML (como em Android) é muito mais fácil utilizar a interface *buidar* para a criação das mesmas, onde são fornecidos os elementos visuais básicos e onde o *developer* possui ferramentas para as customizar para a UI pretendida. As *views* são criadas e guardadas em ficheiros xib e o seu formato esta disponível abaixo.



**Figura 33** – Ficheiro xib para o ecrã HomeViewController

O projeto final contém no total 46 ficheiros e aproximadamente 4221 linhas de código. Atendendo a grande quantidade de código e de ficheiros apenas será explicado como foi desenvolvido umas das funcionalidades principais desta aplicação, a Viagem em tempo real.

Como já foi explicado anteriormente a funcionalidade Viagem em tempo real permite ao utilizador registar uma viagem em tempo real e ter o custo aproximado a cada momento da viagem. Na figura 35 esta representado o xib deste ecrã.



**Figura 34** – Ficheiro XIB do ecrã Viagem Tempo Real

Os principais elementos na *view* são os seguintes: Uma *label* para o estado da viagem, um *MKMapView* para disponibilizar os dados do Mapa, uma secção dedicada para a disponibilização das informações como o tempo, a distância e o custo aproximado. Para finalizar o ecrã, temos a secção dos Botões onde permite ao utilizador pausar a viagem, cancelar e terminar a mesma.

A ligação destes elementos visuais do xib ao código é feito através do código representado na próxima figura (Figura 36).

```
@IBOutlet weak var mapView: MKMapView!  
@IBOutlet weak var driveStatusView: UIView!  
@IBOutlet weak var driveStatusLabel: UILabel!  
@IBOutlet weak var timeTravelledLabel: UILabel!  
@IBOutlet weak var distanceTravelledLabel: UILabel!  
@IBOutlet weak var aproxCostLabel: UILabel!  
@IBOutlet weak var infoToMapTopConstraint: NSLayoutConstraint!  
@IBOutlet weak var cancelViagemButton: UIButton!  
@IBOutlet weak var pausePlayButton: UIButton!  
@IBOutlet weak var pausePlayLabel: UILabel!
```

**Figura 35** – Ligação dos elementos visuais ao código

Após termos os elementos visuais ligados ao código fazemos a inicialização e o Setup de tudo o que precisamos para pôr os elementos funcionais. O código necessário está

representado em duas funções principais, A `setupView()` e o `startTravel()`, ambas representadas na figura seguinte (Figura 37).

```
func setupView()
{
    self.view.backgroundColor = UIColor(red: 29/255, green: 27/255, blue: 27/255, alpha: 1)
    driveStatusView.layer.cornerRadius = driveStatusView.frame.width/2
    driveStatusView.layer.masksToBounds = true
    driveStatusView.isHidden = true
    driveStatusLabel.isHidden = true
    timeTravelledLabel.text = "00:00:00"
    distanceTravelledLabel.text = "0.0 Km"
    aproxCostLabel.text = "0.0€"
    applyMaskToMap()
    Helper.setupButtonStyle(button: cancelViagemButton)
}

func startTravel()
{
    self.locationManager = CLLocationManager()
    self.locationManager.requestAlwaysAuthorization()

    // For use in foreground
    self.locationManager.requestWhenInUseAuthorization()

    if CLLocationManager.locationServicesEnabled() {
        locationManager.delegate = self
        locationManager.desiredAccuracy = kCLLocationAccuracyBest
        locationManager.startUpdatingLocation()
    }

    mapView.delegate = self
    mapView.mapType = .standard
    mapView.showsUserLocation = true
    mapView.isZoomEnabled = true
    mapView.isScrollEnabled = true
    self.mapView.setUserTrackingMode(.followWithHeading, animated: true)
    startTimer()
}
```

**Figura 36** – Função `setupView` e `StartTravel`

As informações no ecrã são todas atualizadas a cada segundo através da função `runTimeTimer`, representada na figura abaixo.

```

@objc func runTimeTimer()
{
    if tripPause
    {
        return
    }
    var minutes = 0
    var hours = 0

    var secondsTemp = timePassedSeconds

    if secondsTemp > 0
    {
        while(secondsTemp >= 60)
        {
            minutes = minutes + 1
            if minutes % 60 == 0
            {
                hours = hours + 1
                minutes = 0
            }
            secondsTemp = secondsTemp - 60
            if secondsTemp < 60
            {
                break
            }
        }
    }
    let hoursString = hours < 10 ? "0\{(hours)" : "\{(hours)"
    let minutesString = minutes < 10 ? "0\{(minutes)" : "\{(minutes)"
    let secondsString = secondsTemp < 10 ? "0\{(secondsTemp)" : "\{(secondsTemp)"
    timeTravelledLabel.text = "\{(hoursString)}:\{(minutesString)}:\{(secondsString)"
    timePassedSeconds = timePassedSeconds + 1
    calculateEstimatePrice()
}

```

**Figura 37** – Função runTimeTimer

Para que seja possível desenhar o trajeto no mapa consoante o percurso do utilizador em todos os momentos da viagem, é necessário ter as coordenadas do utilizador a todos os segundos. As funções responsáveis para esta funcionalidade são a *locationManager()* e a *drawRoute()*.

```

func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {
    if tripPause
    {
        return
    }
    let locValue:CLLocationCoordinate2D = manager.location!.coordinate
    if cordenadasArray.count < 1
    {
        cordenadasArray.append(locValue)
        mapView.mapType = MKMapType.standard
        let annotation = MKPointAnnotation()
        annotation.coordinate = locValue
        annotation.title = "Inicio"
        annotation.subtitle = "current location"
        mapView.addAnnotation(annotation)
    }
    let speedKilometersHour = locationManager.location!.speed * 3.6
    if speedKilometersHour > 3
    {
        if lastlocation == nil
        {
            lastlocation = locations.first
        }
        else
        {
            if let location = locations.last
            {
                traveledDistance += lastlocation.distance(from: location)
                updateDistanceTraveled()
                lastlocation = location
            }
        }
        cordenadasArray.append(locValue)
        changeStateToDrivingMode(speed: speedKilometersHour.rounded(toPlaces: 0))
        drawRoute()
    }
    else
    {
        changeStateToStopMode()
    }
}

```

**Figura 38** – Função locationManager

```

func drawRoute()
{
    mapView.removeOverlay(pointsCordenadas)
    pointsCordenadas = MKPolyline(coordinates: cordenadasArray, count: cordenadasArray.count)
    mapView.addOverlay(pointsCordenadas)
    //Zoom to user location
    if let userLocation = locationManager.location?.coordinate {
        let viewRegion = MKCoordinateRegion(center: userLocation, latitudinalMeters: 800, longitudinalMeters: 800)
        mapView.setRegion(viewRegion, animated: true)
    }
}

```

**Figura 39** – Função drawRoute

Todas as funções representadas neste capítulo são as essenciais para garantir o bom funcionamento desta funcionalidade. Existem outras funções também necessárias, mas não são tão relevantes como as anteriormente descritas. Apesar de existir mais écrans e funcionalidades na aplicação, essas ou não são tão importantes ou são muito semelhantes às descritas. Por exemplo a funcionalidade de Registrar uma Viagem usa muitas das funções como o *locationManager ()* e *drawRoute ()*.

## 5. Testes

Nesta secção será descrito todo o processo de testes realizado à aplicação para garantir a sua eficiência e obter feedback sobre a experiência do utilizador ao utilizar o sistema e de que forma a aplicação poderia ser melhorada.

Irão ser abordados 2 testes principais que serão:

- Testes de usabilidade: irão se focar apenas na experiência do utilizador.
- Testes à aplicação: que irão testar se os dados fornecidos na aplicação e os cálculos que a aplicação realiza são consistentes com os valores de referência que foram calculados a partir dos dados fornecidos pelo próprio veículo.

### 5.1. Testes de usabilidade

Para efetuar os testes de usabilidade à aplicação foi reunido 10 pessoas (6 Homens e 4 Mulheres), com idades entre os 21 e os 26 anos em que foram requisitados a realizar 4 pequenas tarefas de forma a testar a interface gráfica com o objetivo do programador obter feedback dos utilizadores sobre a navegação dentro da aplicação. Dos 10 utilizadores escolhidos 3 já foram condutores da Uber e os restantes 7 são utilizadores comuns habituados a aplicações do género.

As 4 tarefas escolhidas foram as seguintes:

- **Tarefa 1:** Registe-se na aplicação e preencha todos os campos referentes à identificação do veículo.
- **Tarefa 2:** Imagine que se encontra dentro do veículo e pretende saber, em tempo real, quanto é que a viagem lhe irá custar. A partir das funcionalidades dentro da aplicação, registe uma viagem em tempo Real
- **Tarefa 3:** Imaginando que realizou uma viagem do Funchal a Machico e por esquecimento não utilizou o modo “Viagem em tempo real”. Como a aplicação permite registar uma viagem depois dela acontecer, registe a viagem Funchal – Machico dentro da aplicação

- **Tarefa 4:** O seu veículo teve uma avaria no motor e o seu arranjo foi de 250€. Adicione esta despesa à categoria de Manutenções Gerais de forma aos valores dentro da aplicação estarem atualizados.

Após a conclusão de cada uma das tarefas foi requisitado ao utilizador para fornecer feedback do que poderia ser melhorado, se a tarefa era fácil de completar e quais as dificuldades que encontraram no decorrer da mesma.

Para a primeira tarefa foi registado o tempo que cada utilizador levou a completar a mesma. Após a conclusão da tarefa foi pedido um feedback como já explicado anteriormente. O tempo médio para a conclusão da primeira tarefa foi de aproximadamente 76 segundos para o registo da conta e para introduzir as informações do veículo.

Todos os 10 utilizadores forneceram feedback importante, mas destacou se os seguintes pontos que poderiam ser melhorados:

- Para introduzir a marca e modelo do veículo em vez de usar uma caixa de texto, a aplicação deveria apresentar uma lista para todas as marcas possíveis e aquando da seleção da marca, apresentar a lista de todos os modelos dentro dessa marca, de forma a facilitar o processo
- Alguns utilizadores acharam que o formulário é um pouco extenso, e por vezes não tinham bem noção dos valores que eram pedidos e demoravam um pouco mais de tempo a preencher a informação. Outro ponto relacionado com este problema é que pelo menos 1 utilizador referiu como era o pai a pagar todas as despesas do veículo, este não tinha a mínima noção do que preencher no formulário da aplicação.
- Um utilizador sugeriu que fosse possível introduzir a matrícula do veículo e assim preencher automaticamente algumas informações requisitadas como: combustível, marca, modelo, ano e cilindrada.

No geral o teste correu sem nenhum inconveniente, todos os utilizadores conseguiram completar a tarefa sem nenhum problema e sem grandes dúvidas. A grande maioria dos

utilizadores gostou da interface e da maneira como a navegação é feita para preencher as diferentes secções de informação.

A segunda tarefa não foi realizada da maneira pretendida pois, como os testes foram realizados com distanciamento social, não foi possível que os utilizadores testassem o modo de Viagem em Tempo Real da maneira correta, por isso foi mostrada a interface e explicada como ela funcionava e em seguida enviada uma pequena demo (um vídeo a demonstrar a funcionalidade a funcionar como expectável). Neste ponto o feedback dos utilizadores foi coerente entre todos, gostaram da funcionalidade, acharam que uma funcionalidade como esta poderia ajudar a controlar os seus custos de uma maneira mais eficiente, gostaram de como os elementos da UI estão apresentados no ecrã e no geral acharam a funcionalidade muito interessante e não apontaram nenhuma melhoria a ser feita no futuro ou alguma preocupação com o uso desta funcionalidade com o veículo em andamento

A terceira tarefa foi a mais criticada das 4 pedidas. Apesar da funcionalidade prever o custo estimado para uma viagem, os utilizadores queixaram-se da maneira como é introduzido a informação relativa ao ponto de origem e do ponto de destino. Abaixo estão apresentados os pontos com mais destaque:

- Era difícil introduzir uma morada porque os utilizadores estão muito habituados ao “autocomplete” como o Google Maps ou o Apple maps. Resumidamente os utilizadores estavam a espera que quando escrevessem a palavra “Fun”, e aplicação automaticamente mostrada uma lista com os possíveis locais que comessem com essas 3 letras, neste caso “Funchal”. 6 dos 10 utilizadores deram exatamente o mesmo feedback.
- Um utilizador referiu que para o exemplo que apesar do exemplo para testar “Funchal” - “Machico” funcionou perfeitamente bem, quando ele começou a explorar outras localizações e moradas mais complexas ele notou que o sistema não conseguia encontrar muitas vezes trajetos para o que foi introduzido pelo utilizador
- Os utilizadores acharam o modo de introduzir os pontos de origem e de destino, difíceis de utilizar no geral. Apesar de gostarem da funcionalidade em si, tiveram imensos problemas e não foi uma tarefa muito bem conseguida na maior parte dos

utilizadores. Basicamente estavam a espera que a introdução deste tipo de informação fosse muito semelhante ao que encontram no Google Maps, Apple Maps e mesmo no Instagram<sup>11</sup>.

A tarefa 4 foi a mais fácil requisitada aos utilizadores. Todos os utilizadores acharam a tarefa extremamente fácil e acessível, e que todo o processo e a maneira de como as opções estão posicionadas no ecrã é muito intuitivo chegar ao objetivo da tarefa. A única sugestão era fornecida por um utilizador foi o facto de ele se questionar de como é que ele conseguiria mudar a média de combustível gasto pelo seu veículo caso no futuro quisesse atualizar a mesma, opção que não existe de momento dentro da aplicação e que deveria ser implementada em trabalho futuro.

Concluindo esta parte dos testes, com base em conversa verbal com os utilizadores, estes acharam a aplicação intuitiva e fácil de utilizar, gostaram das funcionalidades implementadas no sistema e admitiram que estas eram úteis para o seu quotidiano. Todo o feedback e críticas à aplicação foram bem recebidas, pois estes pontos referidos anteriormente não foram considerados durante o desenvolvimento, e todas as sugestões e críticas seriam para ser implementadas e melhoradas num trabalho futuro para facilitar o processo de utilização da aplicação. Todas as respostas e feedback fornecidas pelos utilizadores estarão disponíveis em Anexo.

Para concluir os testes de usabilidade, foi pedido aos utilizadores para preencherem um pequeno questionário baseado no “System Usability Scale” (SUS) [34], de forma a ter métricas e números concretos para medir a usabilidade da aplicação.

O questionário contém 10 perguntas com uma estrutura de escolha múltipla, em que as opções vão de “Discordo completamente” até “Concordo completamente”. Abaixo encontra-se as perguntas contidas no questionário:

- **Pergunta 1:** Acho que gostaria de usar este sistema com frequência.
- **Pergunta 2:** Achei o sistema desnecessariamente complexo.

---

<sup>11</sup> Instagram Website: <https://www.instagram.com>

- **Pergunta 3:** Achei o sistema fácil de usar.
- **Pergunta 4:** Acho que precisaria do apoio de um técnico para poder usar este sistema.
- **Pergunta 5:** Achei que as várias funções deste sistema estavam bem integradas.
- **Pergunta 6:** Achei que havia muita inconsistência neste sistema.
- **Pergunta 7:** Eu imagino que a maioria das pessoas aprenderia a usar esse sistema muito rapidamente.
- **Pergunta 8:** Achei o sistema muito complicado de usar.
- **Pergunta 9:** Eu me senti muito confiante usando o sistema.
- **Pergunta 10:** Eu precisava aprender muitas coisas antes de começar a usar este sistema.

O sistema SUS foi utilizado para conseguir fazer um teste simples e barato de forma a medir a usabilidade de um sistema, neste caso a aplicação desenvolvida nesta tese. [34]. Para efetuar uma análise aos resultados e obter uma estimativa do quão simples é a utilização da aplicação vamos usar um sistema de pontos que nos vai dizer de forma rápida caso a aplicação tem um nível de usabilidade bom. Segundo estudos realizados sobre o SUS, temos de ter em conta os seguintes intervalos e resultados, numa escala de 0 a 100 [35]:

- 80.3 ou mais alto é um A. As pessoas gostaram da aplicação e provavelmente irão recomendar aos amigos.
- Entre 68 e 80.2 é um C. Significa que a aplicação é OK, mas tem espaço para melhorias.
- 51 ou abaixo é um F. Significa que a aplicação é complexa e que necessita de uma revisão a nível de usabilidade o mais rápido possível.

A tabela 12 contém as respostas a cada uma das perguntas:

**Tabela 12-** Respostas dadas ao questionário SUS

# Pergunta	Discordo completamente (%)	Discordo (%)	Neutro (%)	Concordo (%)	Concordo completamente (%)
1	0	10	20	60	10
2	60	30	10	0	0
3	0	0	20	70	10
4	80	20	0	0	0
5	0	0	10	90	0
6	20	80	0	0	0
7	0	0	30	70	0
8	30	60	10	0	0
9	0	0	60	40	0
10	0	70	30	0	0

A tabela 13 contém os “scores” para cada um dos questionários que posteriormente será feito uma média final de todos os “scores” e assim obter o resultado final dos testes de usabilidade. Os cálculos necessários para obter este resultado final foram conseguidos a partir de uma ferramenta web, que tem como objetivo calcular o “score” de cada um dos questionários realizados individualmente [36].

**Tabela 13-** "Score" de cada questionário realizado no teste de usabilidade

# Questionário	Resultado obtido
1	82.5
2	77.5
3	77.5
4	80
5	85
6	72.5
7	67.5
8	72.5
9	80
10	52.5

A média dos “scores” é de aproximadamente 75, que segundo os intervalos é um C alto. Com base neste resultado e nas informações obtidas pelos estudos realizados dentro do conceito do SUS, este 75 significa que a aplicação é boa, tem uma boa usabilidade no geral, mas existem aspetos que tem de ser melhorados [35]. De acordo com os testes de usabilidade realizados, já conseguíamos ver essa tendência, em que com base no feedback dos utilizadores, existem aspetos na aplicação que podem ser melhorados e sugestões que devem ser implementadas numa fase futura do desenvolvimento. Os “scores” obtidos confirmam exatamente o feedback geral dos utilizadores aquando da realização das tarefas. A escolha por utilizar o SUS foi importante pois é um questionário simples de implementar, com perguntas que não são técnicas e que os utilizadores responderam sem qualquer problema ou dificuldade. Apesar de existirem melhores maneiras de medir métricas de usabilidade, o SUS é um passo inicial importante para avaliar num primeiro ponto o nível de usabilidade de um sistema, e que foi útil para confirmar os resultados obtidos testes de usabilidade realizados.

## 5.2. Testes à aplicação

Para testar o algoritmo e garantir que este nos fornece o cálculo aproximado do custo real de uma viagem, foram realizadas várias viagens e apontados os custos reais fornecidos pelo computador de bordo e os valores fornecidos pelo algoritmo criado dentro da aplicação.

Devido à limitação de dados fornecidos pelo computador de bordo, vamos assumir o custo real como sendo o valor gasto em combustível pelo veículo em cada viagem. Os valores finais da aplicação são os custos calculados tendo em conta não só o combustível gasto, mas também o valor estimado para os diferentes gastos que um utilizador tem com o seu veículo durante o ano (manutenções gerais, impostos e seguros). O veículo utilizado para o teste foi um BMW 318d (F30) de 2013 a gasóleo.



**Figura 40** - BMW F30 318d, veículo utilizado nos testes

Todos os valores respetivos a manutenções gerais, impostos e seguros e outros foram adicionados na aplicação, conforme descritos seguidamente:

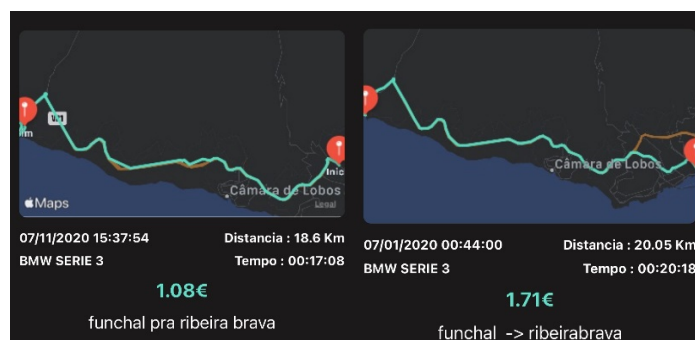
- **Preço do Gasóleo:** 1.13€ ao litro
- **IUC:** 260€
- **Seguro Anual:** 230€
- **Inspeção Periódica:** 36€
- **Gastos em Revisões Anuais:** 400€
- **Manutenções Gerais:** 260€

Para que existisse diversidade nos testes realizados foram feitos 4 percursos diferentes (disponíveis na tabela 14). Dentro destes percursos foi realizado um em específico para que fosse encontrado um padrão e os resultados fossem consistentes para testar realmente se o algoritmo funciona consoante o expectável. O percurso em questão é o Estreito de Câmara de Lobos – Ribeira Brava. Os parâmetros escolhidos para os testes foram os seguintes:

- **Distância e tempo registados no computador de bordo do veículo:** São os valores registados pelo computador de bordo quando realizamos a viagem. Durante o decorrer da viagem o computador de bordo vai registando o tempo e a distância percorrida desde o início do percurso até ao fim do mesmo. Estes valores servem para comparar se a aplicação está a registar valores corretos e assim realizar os cálculos necessários para fornecer o valor do custo da viagem.
- **Distância e tempo registado na app:** Como o nome indica, estes são os valores registados da distância percorrida e do tempo total da viagem dentro da aplicação, utilizando o GPS e o relógio interno para obter estes dados.

- **Consumo Médio:** Este parâmetro é registado pelo computador de bordo e fornece o consumo médio de combustível que foi gasto durante a viagem. A partir deste parâmetro e da distância percorrida é possível calcular o combustível gasto durante a viagem e assim calcular o custo do combustível gasto na mesma.
- **Velocidade média:** A velocidade média é utilizada como parâmetro indicativo de quais foram as condições da viagem.
- **Custo Real Aproximado:** O custo real aproximado é calculado utilizando a mesma fórmula do algoritmo dentro da aplicação, mas utilizando os valores reais fornecidos pelo computador de bordo.
- **Custo Calculado na App:** O custo calculado na aplicação é utilizando a fórmula proposta nesta tese utilizando todos os valores obtidos dentro da aplicação.

Na Figura abaixo (Figura 41), podemos verificar alguns testes realizados dentro da App - Protrip.



**Figura 41** - Alguns testes Realizados dentro do Protrip

Na tabela 14 estão representados todos os valores utilizados nos testes realizados dos percursos.

**Tabela 14** - Testes realizados ao algoritmo desenvolvido

#	Percurso	Distância e tempo registados no computador de bordo	Distancia e tempo registados na App	Consumo Médio	Velocidade Média	Custo Real aproximado	Custo calculado na App
1	Estreito de Câmara de Lobos - Ribeira Brava	17 km 19 minutos	16.78 Km 18.44 minutos	4.2 L/100km	54.6 Km/h	0.84 €	0.84€
2	Ribeira Brava - Estreito de Câmara de Lobos	16 km 18 minutos	16.23 Km 17.14 minutos	9.4 L/100km	55.6 Km/h	1,73 €	1,74€

3	Estreito de Câmara de Lobos – Ribeira Brava	17 Km 16 Minutos	17.46 Km 16.21 minutos	5 L/100km	66.2 Km/h	0.98€	1.02€
4	Ribeira Brava – Canhas	10 Km 16 minutos	10.39 Km 16.48 minutos	5 L/100km	40.8 Km/h	1.17€	1.15€
5	Funchal – Ribeira Brava	19 Km 16 minutos	18.6 Km 17.08 minutos	5 L/100km	70 km/h	1.09€	1.08€
6	Estreito Camara de Lobos – Ribeira Brava	18 Km 19 minutos	17.46 Km 18.12 minutos	4.8 L/100km	58.4 km/h	1.00€	0.98€
7	Estreito Camara de Lobos – Ribeira Brava	18 Km 14 minutos	17.59 Km 13.6 minutos	7.5 L/100km	80.6 km/h	1.54€	1.54€
8	Estreito Camara de Lobos – Ribeira Brava	17 Km 14 minutos	17.49 Km 13.89 minutos	7.2 L/100km	73.6 km/h	1.40€	1.40€

Foram realizados vários percursos para testar o algoritmo e verificar se funciona com diferentes rotas, distâncias, tempos e consumos. Foram realizados apenas 8 testes onde 5 desses foram semelhantes. Para garantir a eficácia e o erro máximo de discrepância entre o algoritmo e o custo real aproximado teve-se em consideração o percurso “Estreito de câmara de Lobos – Ribeira Brava”. Os resultados dos testes podem ser consultados na Tabela 15.

**Tabela 15** – Resultados dos testes ao percurso “Estreito de Câmara de Lobos – Ribeira Brava”

#	Custo Real Aproximado	Custo Calculado pela aplicação	Diferença
1	0.84 €	0.84€	0.00€
2	0.98€	1.02€	0.04€
3	1.00€	0.98 €	0.02€
4	1.54€	1.54€	0.00€
5	1.40€	1.40€	0.00€

Calculando a Média dos custos reais aproximados ( $M=1.152€$ ) e Média dos custos calculados na App ( $M=1.156€$ ), dos resultados obtidos (Tabela 12) obteve-se uma diferença em percentagem de 1.2%.

O custo Real aproximado e o custo calculado pela aplicação utilizam a mesma fórmula para calcular o custo total de uma viagem. A principal diferença entre estes dois testes está na forma como é obtido os tempos e a distância percorrida ao longo da viagem. Estes parâmetros são obtidos de forma diferente, mas os valores são aproximados e assim os

custos são semelhantes. Seguidamente será explicado o processo de obter os custos e os dados em cada um dos ambientes de teste.

### **Custo Real Aproximado:**

- O computador de bordo foi preparado para começar a contar a partir do momento que a viagem começa
- Após a chegada ao destino é registado os valores da distância percorrida, o tempo total e o consumo médio.
- Por fim, com os valores obtidos é calculado o custo total da viagem através da fórmula proposta neste projeto.

### **Custo Calculado pela aplicação:**

- O utilizador inicia uma viagem em tempo real
- Após a chegada ao destino, o utilizador termina a viagem e caso queira introduzir a média de combustível gasto poderá o fazer (se tiver acesso a estes dados)<sup>12</sup>
- A aplicação irá calcular o custo automaticamente.

O principal objetivo dos testes realizados é garantir que a aplicação forneça um custo de uma viagem, recolhendo todos os dados necessários automaticamente sem que o utilizador tenha que realizar alguma ação em específico. Como um dos objetivos deste projeto é facilitar o processo para calcular o custo de uma viagem de forma fácil e intuitiva para o utilizador, necessitámos de garantir que o algoritmo e os dados obtidos estavam de acordo com os valores obtidos na realidade. Dito isto, e tendo em conta os resultados analisados, observamos que a diferença entre os custos reais aproximados e os custos calculados na aplicação é em média de 1.2% de margem de erro no percurso estudado (“Estreito de Câmara de Lobos – Ribeira Brava”), e tendo em conta que o objetivo seria de 5% como margem de erro, garantimos assim com esta margem obtida, que a aplicação consegue obter dados fiáveis e calcular os custos totais de uma viagem sem que o utilizador necessite de registar qualquer informação sobre a viagem externamente à aplicação.

---

<sup>12</sup> Conforme descrito no capítulo 3 desta tese.

## 6. Conclusão e Trabalho Futuro

Neste capítulo será explicado em retrospectiva todo o trabalho realizado ao longo deste projeto e o trabalho futuro que poderá ser realizado para melhorar a experiência do utilizador ao utilizar a aplicação.

### 6.1. Trabalho Realizado

Este projeto começou pela criação de objetivos práticos que justificassem a criação de um projeto como este e em que maneira é que poderia trazer uma melhoria à vida das pessoas, de empresas e de trabalhadores por conta própria dentro do sector do transporte privado de passageiros. Após a criação dos objetivos foi feita uma investigação de como é que a Uber e o Táxi calculam as suas tarifas e assim obter uma base para a criação do algoritmo utilizado nesta tese. O algoritmo proposto nesta tese tem algumas semelhanças com os algoritmos estudados da Uber e do taxímetro, mas a principal diferença é que o proposto tem como objetivo obter os custos para o condutor (foco no condutor) enquanto que o outro tem como objetivo maximizar o lucro.

Acabando a investigação no estado da arte, para o desenvolvimento da aplicação começou-se por definir alguns requisitos funcionais e não funcionais, passando depois a partir destes requisitos serem criados protótipos de baixa fidelidade e obter uma primeira impressão de como é que a aplicação iria se comportar. Depois de analisado os protótipos avançou-se para a criação dos protótipos de alta fidelidade, tendo em conta que a criação destes protótipos tem de ser realizada com atenção e cuidado, pois estes já são um exemplo muito aproximado de como é que a aplicação se irá parecer/comportar numa versão final.

A versão final da aplicação foi na maioria dos casos semelhante aos protótipos de alta fidelidade, mas após uma avaliação de usabilidade, e detetado alguns problemas, os mesmos foram corrigidos através de uma mudança subtil no design da aplicação. Após a conclusão do desenvolvimento, esta foi testada várias vezes para garantir que a aplicação era estável e que não apresentava problemas de performance através de alguns testes realizados pelo o programador. Após a sua finalização a nível de desenvolvimento, estava na hora de levar a aplicação para a estrada e realizar os testes necessários para garantir que o utilizador pode confiar nos dados fornecidos pela mesma. Em primeiro lugar foi utilizado o computador de bordo do veículo para obter os dados necessários para o algoritmo calcular o custo real, mas este cálculo foi realizado manualmente. Como o

algoritmo considera valores como impostos, manutenções gerais, seguro e combustível gasto o cálculo foi um pouco mais complexo do que o normal, onde a grande maioria das pessoas apenas conta com o combustível gasto, o que por si só leva ao custo não ser o real na sua grande maioria. O objetivo dos testes era garantir que os valores calculados manualmente não falhavam muito dos valores calculados automaticamente pela aplicação e foi o que foi verificado onde em média a diferença entre os custos calculados era de apenas 1.2%, o que quando comparado com o objetivo de ficar abaixo de 5% observamos que o resultado se encontra dentro do intervalo proposto.

Um dos principais focos deste projeto era facilitar o processo do cálculo do custo total associado a uma viagem. Calcular estes custos manualmente é a realidade de muitos profissionais de transporte de passageiros e para muitas empresas. Através da aplicação desenvolvida é possível facilitar os passos e a conveniência em que estes custos são calculados, onde o utilizador apenas terá que iniciar o modo de registar viagem tempo real e a aplicação irá solicitar todos os parâmetros necessários para obter um custo para a viagem realizada. Em retrospectiva de todo o trabalho realizado posso afirmar que toda a experiência desde a delineação dos objetivos iniciais até a finalização da aplicação foi enriquecedora. Os objetivos propostos foram cumpridos, o projeto desenvolvido é funcional e pode ser lançado para o mercado e ser utilizado em casos reais.

## 6.2. Trabalho Futuro

Apesar da aplicação desenvolvida ter uma base sólida existem falhas a nível de conveniência que podiam ser melhoradas no futuro. Por exemplo o facto de dependermos do utilizador para introduzir todos os gastos relacionados com o veículo admitimos que possa ser um processo um pouco maçador, por tanto o que seria proposto no futuro seria um modo que a aplicação conseguisse ler faturas e através de Modelos de identificação adicionar o valor à categoria correta.

Outro facto que seria decisivo na precisão exata no cálculo de um custo seria em vez de trabalhar com médias de combustível ter essa informação em tempo real através da porta OBD do veículo e assim saber em todos os pontos da viagem o consumo real do veículo. Através destes dados era possível aumentar a eficiência e eficácia do algoritmo como também poupar alguns passos ao utilizador, tornando assim a experiência muito mais agradável e menos intrusiva.

Também seria interessante construir esta aplicação em Android para que os dois ecossistemas estivessem combinados para maximizar o número de utilizadores.

## 7. Referências:

- [1] - Características dos táxis no território Português–Website (Visitado a 14/04/2020):  
[https://pt.wikipedia.org/wiki/Táxis\\_em\\_Portugal#Caracter%C3%ADsticas\\_dos\\_táxis](https://pt.wikipedia.org/wiki/Táxis_em_Portugal#Caracter%C3%ADsticas_dos_táxis)
- [2] – Criação da Uber (Visitado a 14/04/2020): <https://www.investopedia.com/articles/personal-finance/111015/story-uber.asp>
- [3] - Funcionamento da uber (Visitado a 16/04/2020) <https://help.uber.com/riders/article/como-funciona-a-uber?nodeId=738d1ff7-5fe0-4383-b34c-4a2480efd71e>
- [4] – Tabela de preços da Uber (Visitado a 18/04/2020) <https://www.uber.com/pt/pt-pt/price-estimate/>
- [5] – Fórmula para o cálculo do preço da Uber (Visitado a 18/04/2020)  
<https://ride.guru/content/newsroom/how-is-my-uber-fare-calculated>
- [6] – Uber Driver Pros & Cons (Visitado a 20/04/2020)  
<https://uberpeople.net/pages/prosandcons/>
- [7] – The Effects of Uber’s Surge Pricing: A Case Study (Visitado a 20/04/2020)  
[http://1gluem2nc4jy1gzhn943ro0gz50.wpengine.netdna-cdn.com/wp-content/uploads/2016/01/effects\\_of\\_ubers\\_surge\\_pricing.pdf](http://1gluem2nc4jy1gzhn943ro0gz50.wpengine.netdna-cdn.com/wp-content/uploads/2016/01/effects_of_ubers_surge_pricing.pdf)
- [8] – Tabelas de preços nos táxis em Lisboa (Visitado a 22/04/2020)  
<https://lisbonlisboaportugal.com/pt/taxi-lisboa-br.html>
- [9] – Aquila Electronics. "How Do Taximeters Work?" (March 9, 2011) (Visitado a 26/04/2020) [http://www.aquila-electronics.co.uk/how\\_work.html](http://www.aquila-electronics.co.uk/how_work.html)
- [10] – New York City Taxi and Limousine Commission. "Passenger Information: Rate of Fare." New York City Taxi and Limousine Commission. (March 9, 2011) (Visitado a 26/04/2020) [http://www.nyc.gov/html/tlc/html/passenger/taxicab\\_rate.shtml](http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml)
- [11] – Taxi Calculator (Visitado a 26/04/2020) <https://www.taxi-calculator.com/taxi-fare-estimation>
- [12]- Track your earnings (Visitado a 26/04/2020)  
<https://www.uber.com/gh/en/drive/basics/tracking-your-earnings/>
- [13] - How much power does an electric car use? (Visitado a 26/04/2020)  
<https://www.energuide.be/en/questions-answers/how-much-power-does-an-electric-car-use/212/>
- [14] – Imposto Único de Circulação (Visitado a 26/04/2020)  
<https://www.acp.pt/automovel/documentos-e-fiscalidade/imposto-unico-de-circulacao>
- [15] – Custo medio de um carro a gasolina (Visitado a 26/04/2020)  
<https://www.economias.pt/quanto-lhe-custa-o-carro-por-ano/>

- [16] – Consumo medio de energia de um Tesla Model 3 (Visitado a 26/04/2020) - <https://ev-database.org/car/1060/Tesla-Model-3-Standard-Range>
- [17] - Consumo medio de gasolina de um Renault Mégane (Visitado a 26/04/2020) - <https://www.ultimatespecs.com/pt/car-specs/Renault/74759/Renault-Megane-4-15-dCi-110-GT-Line.html>
- [18] – Algoritmo de Dijkstra’s (Visitado a 1/05/2020)  
[https://www.vice.com/en\\_us/article/4x3pp9/the-simple-elegant-algorithm-that-makes-google-maps-possible](https://www.vice.com/en_us/article/4x3pp9/the-simple-elegant-algorithm-that-makes-google-maps-possible)
- [19] - Graph Data Structure 4. Dijkstra’s Shortest Path Algorithm (Visitado a 1/05/2020)  
<https://www.youtube.com/watch?v=pVfj6mxhdMw>
- [20] – REPORT DIJKSTRA’S ALGORITHM (Visitado a 1/05/2020) -  
<https://cs.nyu.edu/courses/summer07/G22.2340-001/Presentations/Puthuparampil.pdf>
- [21] - How does Google Maps predict traffic? (Visitado a 1/05/2020)  
<https://electronics.howstuffworks.com/how-does-google-maps-predict-traffic.htm>
- [22] - Como o Waze funciona? (Visitado a 4/05/2020) -  
<https://support.google.com/waze/answer/6078702?hl=pt-BR>
- [23] - REQUISITOS DO MOTORISTA-PARCEIRO (Visitado a 4/05/2020) -  
<https://uberportugal.pt/portugal/como-ser-motorista-ou-parceiro-da-uber/>
- [24] – Tabela de tarifas praticadas pelo serviço de táxi na região autónoma da Madeira (Visitado a 4/05/2020) - <https://www.aitram.pt/aitram/tabela-precos>
- [25] – Olga Teixeira, “Quanto custa ter um carro? Veja a que despesas não pode fugir”, 3 de Junho de 2020 - <https://www.e-konomista.pt/quanto-custa-ter-um-carro/>
- [26] – ADRIAANSE, Mark Lievisse. Profits and precarity: Uber and the crisis of work. Leiden University-Leiden, Países Baixos, 2016.
- [27] – ZOEPF, Stephen M., et al. The economics of ride-hailing: Driver revenue, expenses and taxes. CEEPR WP, 2018, 5.
- [28] – KENNEDY, Elizabeth J. Employed by an algorithm: Labor rights in the on-demand economy. Seattle UL Rev., 2016, 40: 987.
- [29] – JENN, Alan. Emissions benefits of electric vehicles in Uber and Lyft ride-hailing services. Nature Energy, 2020, 5.7: 520-525.
- [30] – LU, Jiaying, et al. Goodbye maps, hello apps? Exploring the influential determinants of travel app adoption. Current issues in Tourism, 2015, 18.11: 1059-1079.
- [31] – ZHAN, F. Benjamin. Three fastest shortest path algorithms on real road networks: Data structures and procedures. Journal of geographic information and decision analysis, 1997, 1.1: 69-82.

- [32] – JAISWAL, Nikita; CHAKRAWART, Rajesh Kumar. Increasing no. of nodes for Dijkstra algorithm without degrading the performance. International Journal of Engineering and Computer Science, 2013, 2.03.
- [33] – WANG, Huijuan; YU, Yuan; YUAN, Quanbo. Application of Dijkstra algorithm in robot path-planning. In: 2011 second international conference on mechanic automation and control engineering. IEEE, 2011. p. 1067-1069.
- [34] – System Usability Scale (SUS), (Visitado a 24/09/2020) Available at : <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>
- [35] – Nathan Thomas, How to Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website, (Visitado a 24/09/2020) Available at: <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>
- [36] – SUS calculator, (Visitado a 24/09/2020) Available at: <https://uiuxtrend.com/sus-calculator/>

# ANEXOS

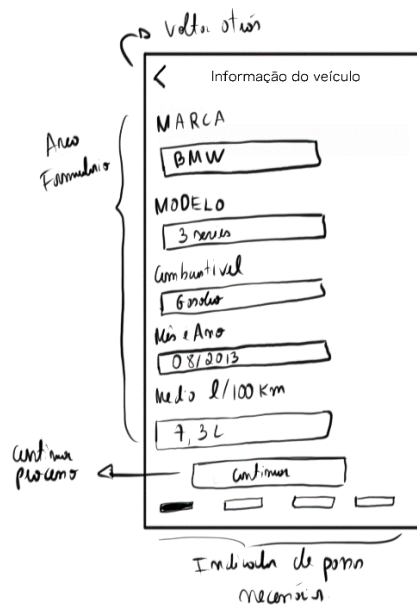


Figura 42 - Protótipo de Baixa Fidelidade - Ecrã de Registo do Automóvel 1

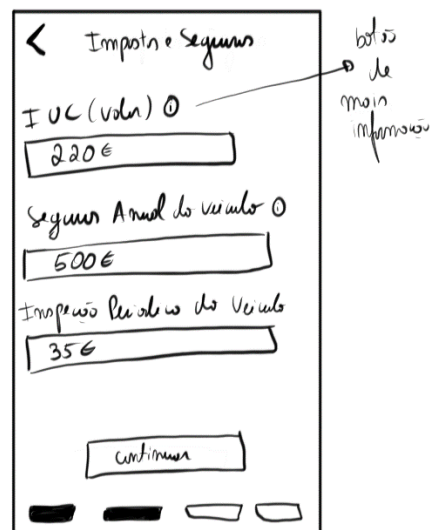


Figura 43 - Protótipo de Baixa Fidelidade - Ecrã de Registo do Automóvel 2

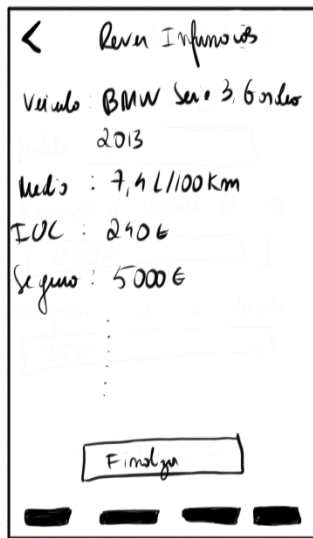


Figura 44 - Protótipo de Baixa Fidelidade - Ecrã de Registo do Automóvel 3

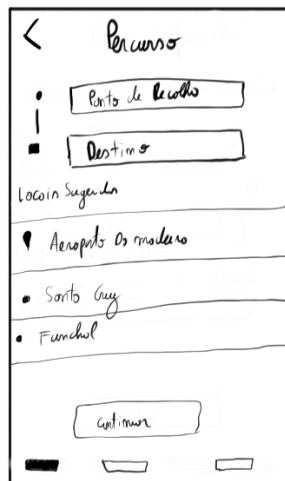
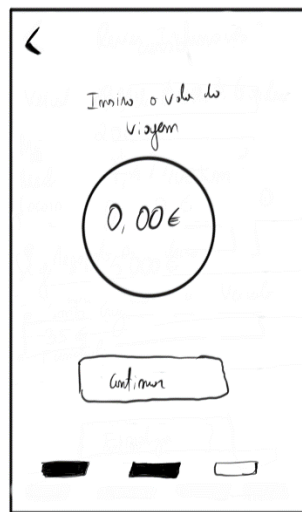
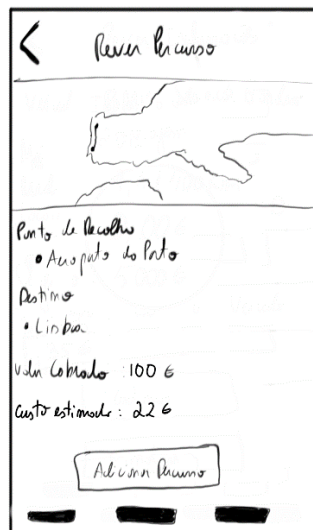


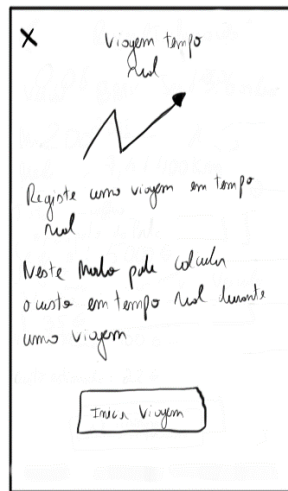
Figura 45 - Protótipo de Baixa Fidelidade - Ecrã de Registo de uma Viagem 1



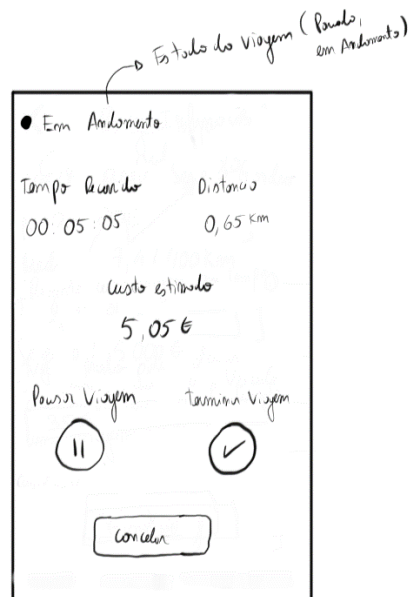
**Figura 46** - Protótipo de Baixa Fidelidade - Ecrã de Registo de uma viagem 2



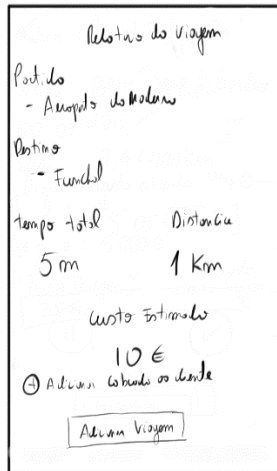
**Figura 47** - Protótipo de Baixa Fidelidade - Ecrã de Registo de uma viagem 3



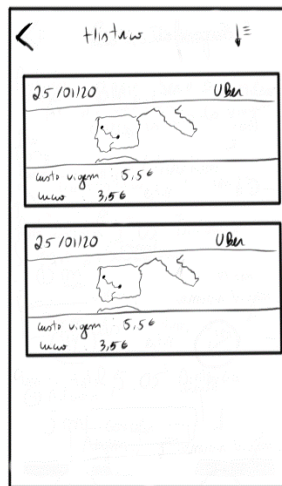
**Figura 48** - Protótipo de Baixa Fidelidade - Ecrã de Viagem em Tempo Real 1



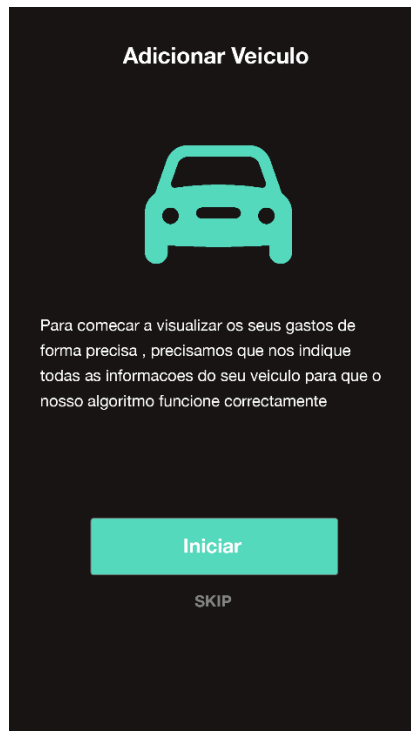
**Figura 49** - Protótipo de Baixa Fidelidade - Ecrã de Viagem em Tempo Real 2



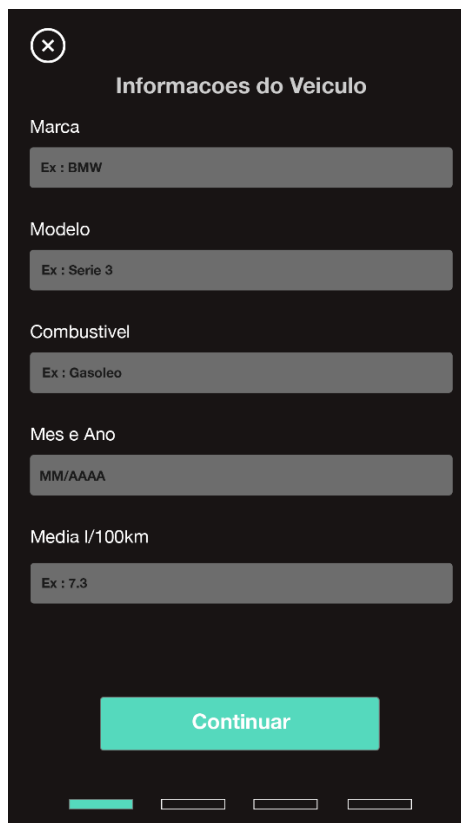
**Figura 50** - Protótipo de Baixa Fidelidade - Ecrã de Viagem em Tempo Real 3



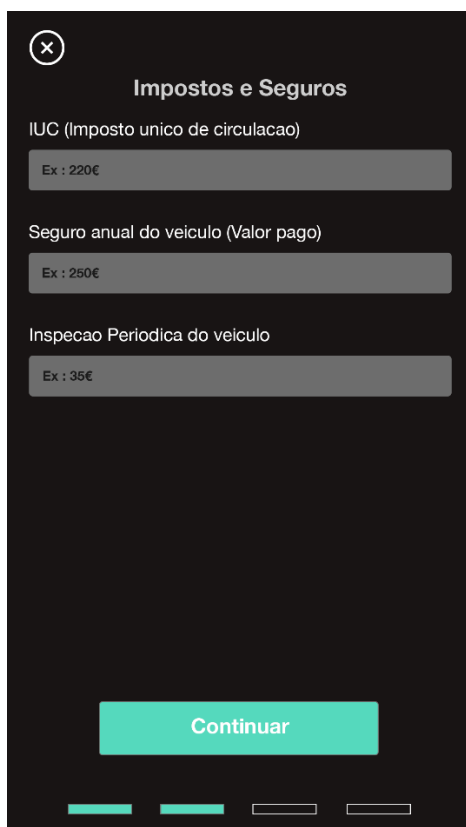
**Figura 51** - Protótipo de Baixa Fidelidade - Ecrã de Histórico de Viagens



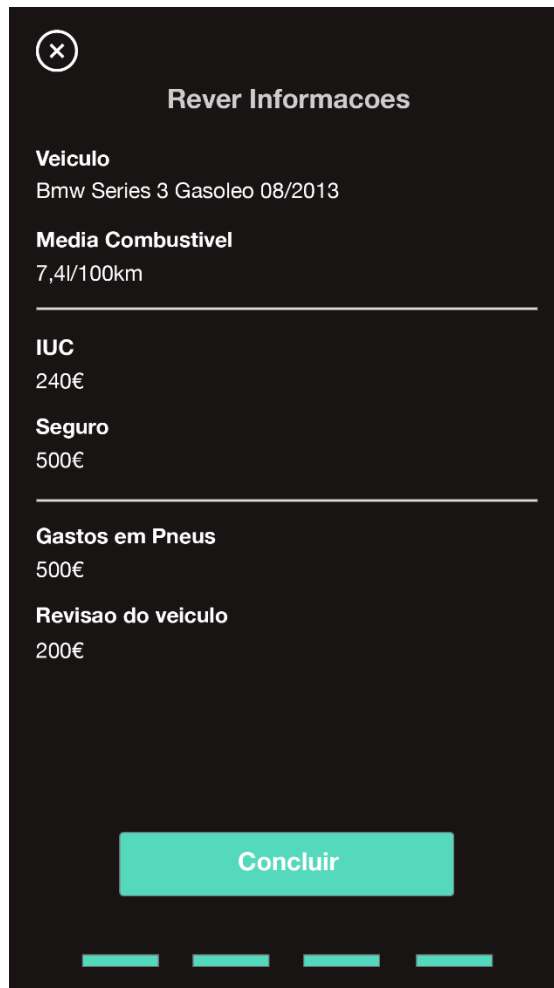
**Figura 52** - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 1



**Figura 53** - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 2



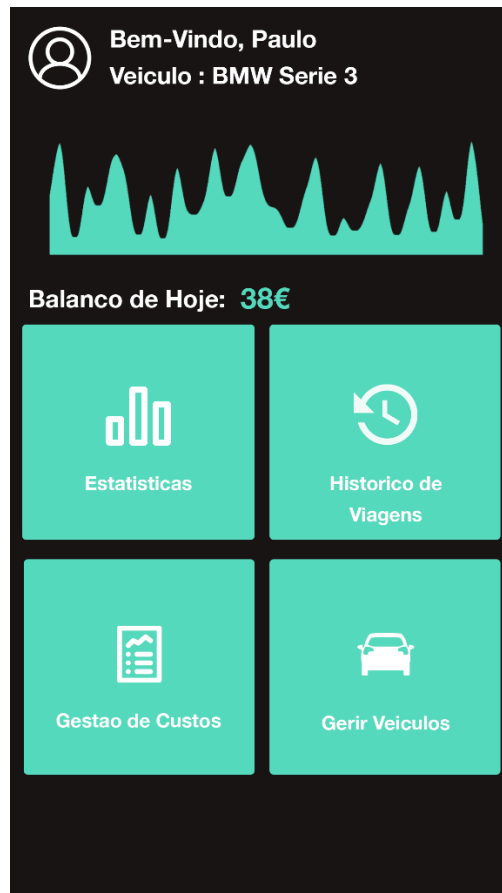
**Figura 54** - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 3



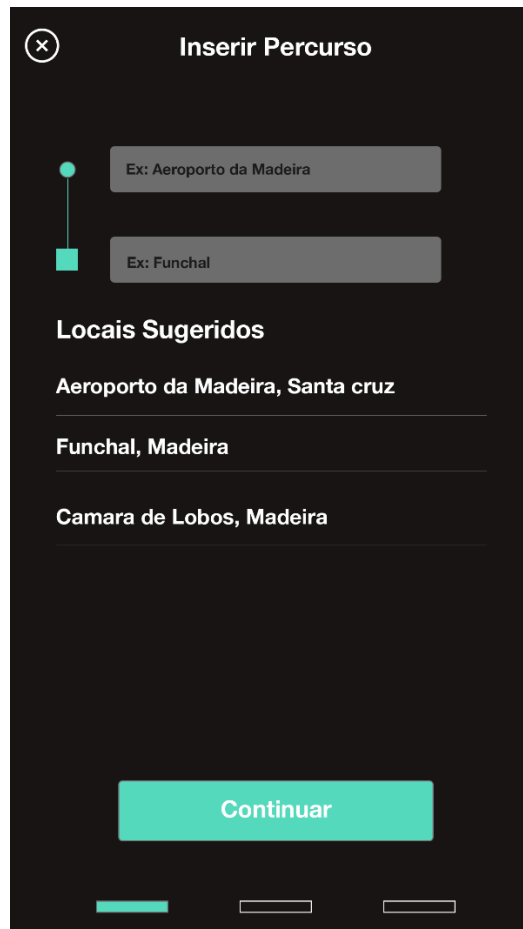
**Figura 55** - Protótipo de Alta Fidelidade - Ecrã de Registo de um automóvel 4



**Figura 56** - Protótipo de Alta Fidelidade - Ecrã Inicial da Aplicação



**Figura 57** - Protótipo de Alta Fidelidade - Ecrã Principal do Protrip



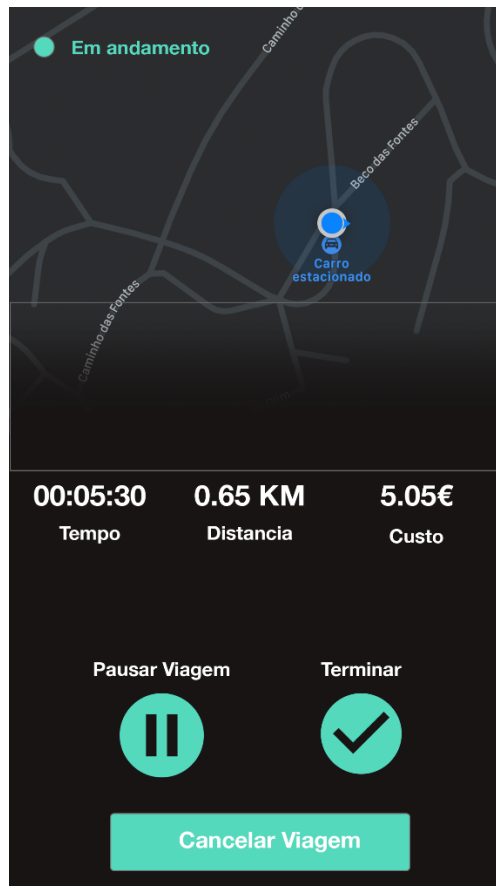
**Figura 58** - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem 1



**Figura 59** - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem 2



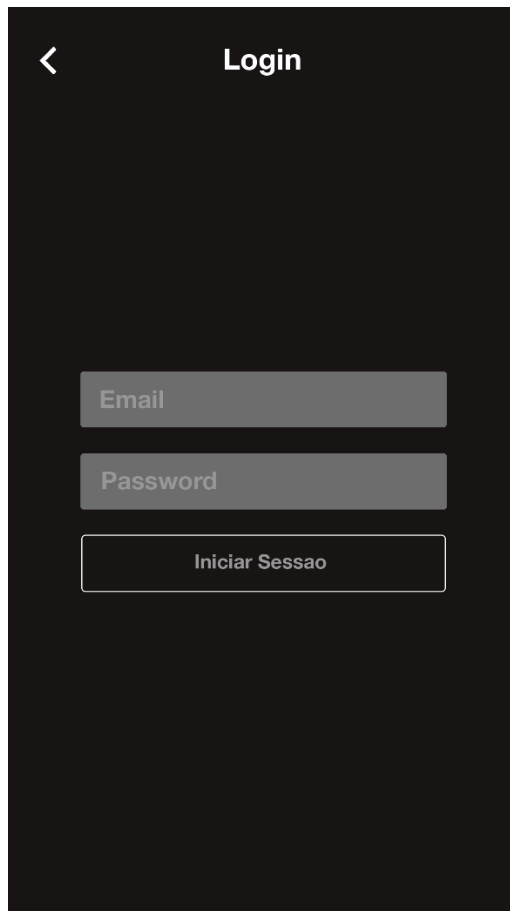
**Figura 60** - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem em Tempo Real 1



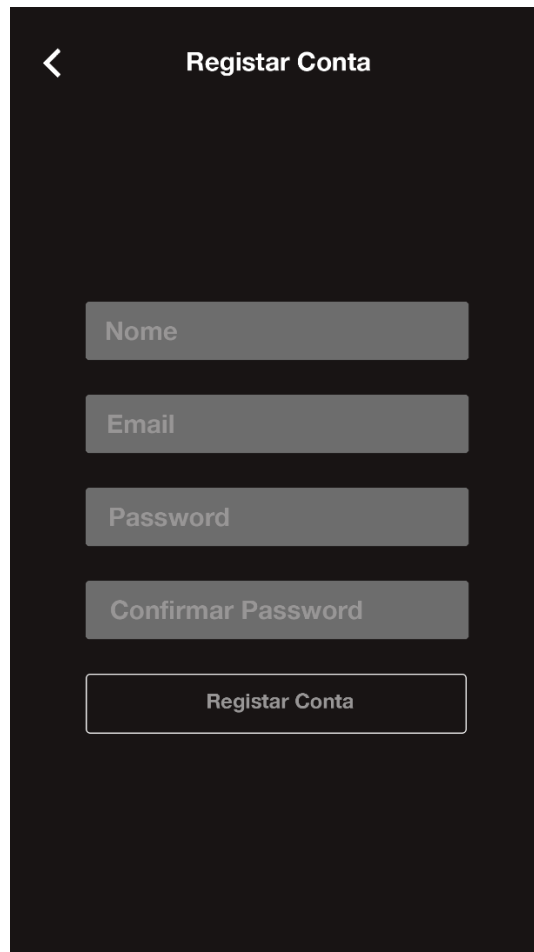
**Figura 61** - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem em Tempo Real 2



**Figura 62** - Protótipo de Alta Fidelidade - Ecrã de Registo de uma viagem em Tempo Real 3



**Figura 63** - Protótipo de Alta Fidelidade - Ecrã de Registo de Login



**Figura 64** - Protótipo de Alta Fidelidade - Ecrã de Registo de uma nova conta

teste1	tempo para realizar o teste	comentarios
		as caixas para as marcas e modelo dos carros deveria preencher automatico , ou mostrar uma lista com os carros
user1	01:06	
user2	01:30	so achei dificil o facto de ter que saber todas as informacoes do veiculo assim derrepente , pois algumas nao tenho a certeza e precisava de confirmar no carro
user3	01:21	dificil de saber os custos exatos das manutencoes pq é o meu pai que paga
user4	01:57	demasiada informacao para ser preenchida
user5	01:15	ta bonito , ta simples e a informacao que é pedida é facil de por uma estimativa
user6	01:10	so acho que a parte de meter a marca e o modelo devia ser uma lista com opcoes fixas , assim era mais facil de preencher pelo menos esta parte
user7	01:13	acho que deveria ser possivel procurar o carro pela matricula , assim algumas informacoes ja ficavam automaticamente preenchidas
user8	01:01	a interface é facil de mexer , apesar de ser necessario preencher muita informacao , os ecrans estao divididos em seccoes o que permite ter uma nocao de progresso
user9	01:05	apenas meter a pesquisar pela a matricula que assim informacoes como cilindrada, combustivel , marca modelo , estariam todas preenchidas automaticamente
user10	01:03	facil utilizacao , so acho que e preciso preencher muita informacao , mas e rapido e facil de preencher

**Figura 65** - Feedback Fornecido pelos utilizadores à tarefa 1 dos testes de usabilidade

teste2	tempo para realizar o teste	comentarios
user1	-	a interface esta muito fixe, as informacoes apresentadas e o mapa no ecran do registo da viagem ta muito bom
user2	-	esta funcionalidade esta porreira , o mapa e o facto de poder apresentar o trajecto feito ta muito fixe
user3	-	gosto do ecran do registo de viagem em tempo real , tem tudo bem apresentado e muito bonito
user4	-	ta bem pensado
user5	-	gosto do facto de ser possivel pausar uma viagem caso tenha de fazer uma paragem durante a viagem , tipo para meter gasolina no carro
user6	-	penso que pode falhar dentro dos tuneis , pois ha muitos que nao tem rede , mas parece que funciona bem
user7	-	no ecran de finalizar as informacoes , o meu carro nao tem a informacao de quanto gastou na viagem , por isso penso que o resultado pode falhar um pouco
user8	-	ta fixe ! Gosto do mapa e do facto de depois a viagem ficar registada no historico
user9	-	gosto desta funcionalidade , acho que usaria na minha vida para controlar melhor os meus custos
user10	-	muito fixe, e como se fosse a app da uber mas para o nosso veiculo

**Figura 66** - Feedback Fornecido pelos utilizadores à tarefa 2 dos testes de usabilidade

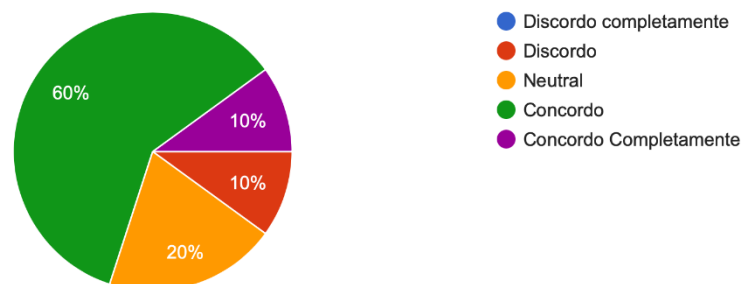
teste3	tempo para realizar o test	comentarios
user1	-	a parte das localizacoes deveria preencher automaticamente pela a morada que esta a ser introduzida, falhou me muitas vezes encontrar um trajeto
user2	-	o que foi pedido no teste funciona bem (funchal - Machico), mas quando tentei inventar um ponto de origem e de destino muitas vezes diz que nao consegue encontrar um trajeto
user3	-	gostei do facto que se procurar Lisboa - Funchal, a app manda me ir de aviao xD, apenas acho que devia ter uma lista que preenche ou ajuda a escolher o que queremos mostrar como o instagram faz
user4	-	sinto que nesta funcionalidade é difícil procurar por morada, porque as vezes estou na duvida e gosta que isto desse opcoes de diferentes moradas para me lembrar de qual é a certa
user5	-	devia de preencher ou pelo menos sugerir a localizacao dependendo do que vou escrevendo na parte de meter o texto
user6	-	talvez podia ter um botao que automaticamente pusesse a nossa localizacao atual, caso queira prever quando e que vai custar uma viagem
user7	-	difícil de usar porque nao da recomendacoes de moradas
user8	-	o instagram da um autocomplete quando tentamos procurar uma morada ou um sitio, acho que devia ser igual nesta parte
user9	-	acho que podia ter um mapa em baixo onde pudessesmos por um pin exactamente de onde saímos e de onde queremos ir, tipo um preview antes de avancar para a parte que estima a viagem
user10	-	esta interessante esta funcionalidade porque nao so permite registar uma viagem que foi feita no passado, mas tambem da para "prever" quanto e que vai custar um a viagem caso eu queira ir com os meus amigos

**Figura 67** - Feedback Fornecido pelos utilizadores à tarefa 3 dos testes de usabilidade

teste4	tempo para realizar o test	comentarios
user1	-	bastante intuitivo mudar os valores do veiculo, o botao na home page é bastante explicito e é facil la chegar
user2	-	deveria dar para mudar tambem os valores da media do combustivel no ecran das gestao custos
user3	-	otimo, bastante facil realizar este teste
user4	-	é bastante intuitivo, facil percepcao e de facil acesso realizar esta tarefa
user5	-	nao e dificil, tem uma utilizade extrema pois tenho sempre os meus valores atualizados e assim ter sempre os valores dos custos do meu veiculo
user6	-	muito facil de utilizar e a tarefa foi bastante intuitiva de realizar nesta aplicacao
user7	-	facil realizar a tarefa
user8	-	nada dificil, bastante facil de adicionar valores ou atualizar valores do meu veiculo
user9	-	nao custou nada realizar a tarefa, muito facil chegar la
user10	-	gostei da experiencia pois achei facil mudar este valor em questao

**Figura 68** - Feedback Fornecido pelos utilizadores à tarefa 4 dos testes de usabilidade

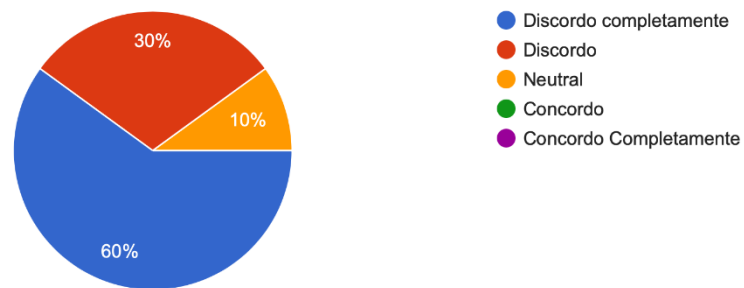
Acho que gostaria de usar este sistema com frequência.  
10 respostas



**Figura 69** - Resultado da Pergunta 1 do questionário de usabilidade

Achei o sistema desnecessariamente complexo.

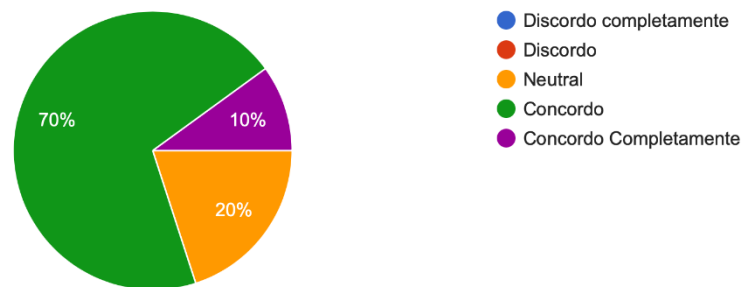
10 respostas



**Figura 70** - Resultado da Pergunta 2 do questionário de usabilidade

Achei o sistema fácil de usar.

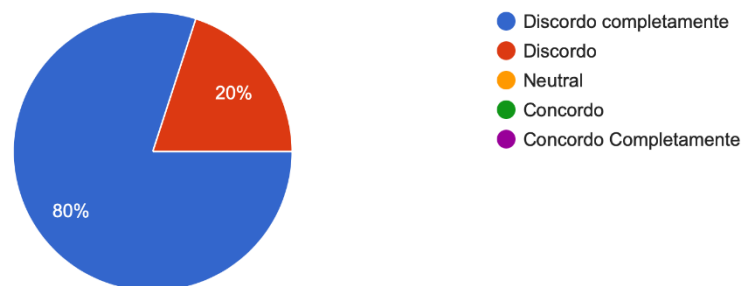
10 respostas



**Figura 71** - Resultado da Pergunta 3 do questionário de usabilidade

Acho que precisaria do apoio de um técnico para poder usar este sistema.

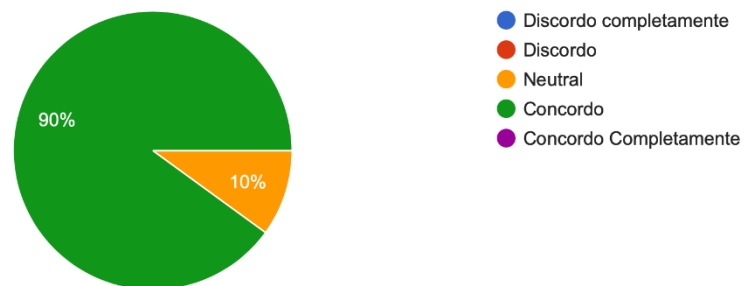
10 respostas



**Figura 72** - Resultado da Pergunta 4 do questionário de usabilidade

Achei que as várias funções deste sistema estavam bem integradas.

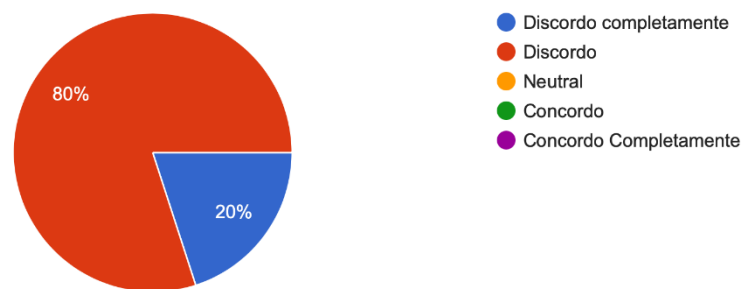
10 respostas



**Figura 73** - Resultado da Pergunta 5 do questionário de usabilidade

Achei que havia muita inconsistência neste sistema.

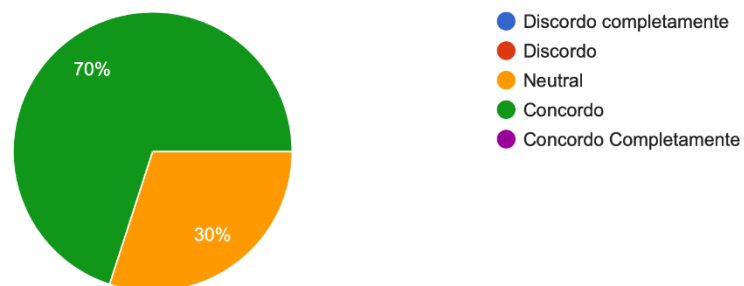
10 respostas



**Figura 74** - Resultado da Pergunta 6 do questionário de usabilidade

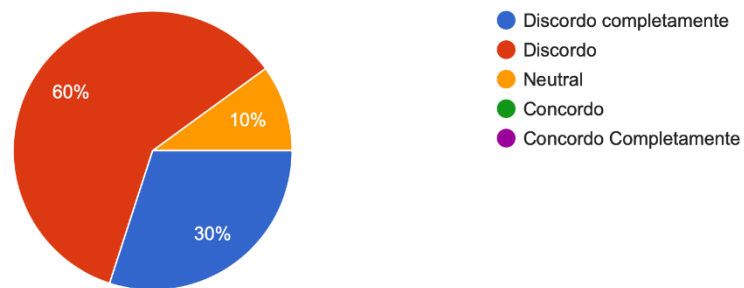
Eu imagino que a maioria das pessoas aprenderia a usar esse sistema muito rapidamente.

10 respostas



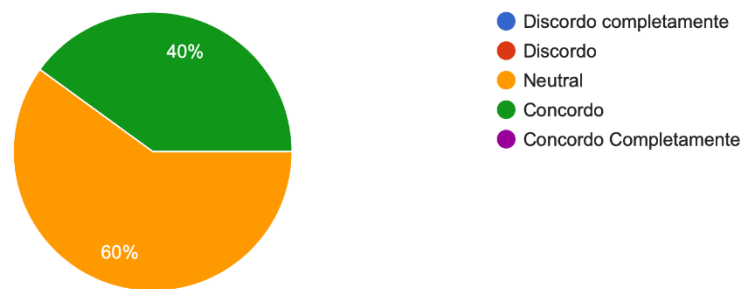
**Figura 75** - Resultado da Pergunta 7 do questionário de usabilidade

Achei o sistema muito complicado de usar.  
10 respostas



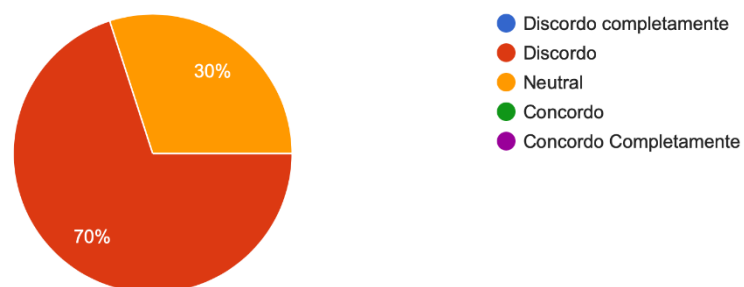
**Figura 76** - Resultado da Pergunta 8 do questionário de usabilidade

Eu me senti muito confiante usando o sistema.  
10 respostas



**Figura 77** - Resultado da Pergunta 9 do questionário de usabilidade

Eu precisava aprender muitas coisas antes de começar a usar este sistema.  
10 respostas



**Figura 78** - Resultado da Pergunta 10 do questionário de usabilidade

-----PÁGINA DEIXADA EM BRANCO PROPOSITADAMENTE -----