



Plataforma simplificada para gestão de frota

MANUEL DOMINGOS DE SOUSA PINHEIRO

Outubro de 2022

Plataforma simplificada para gestão de frota

Manuel Domingos de Sousa Pinheiro

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Computacionais**

Orientador: Joaquim Filipe Peixoto dos Santos

Supervisor: Carlos Magalhães

Júri:

Presidente:

Vogais:

Porto, outubro de 2022

Dedicatória

Dedico esta dissertação aos meus pais.

Resumo

A *Gisgeo Information Systems Lda* é uma empresa dedicada ao desenvolvimento de sistemas de informação geográfica, em que oferece a *Geocar*, uma solução para gestão de frotas de veículos automóveis que permite às empresas e particulares o planeamento e uso eficiente dos seus recursos. A gestão de frotas é efetuada com a instalação de equipamentos capazes de reportar dados georreferenciais através da rede GSM/GPRS nas viaturas dos clientes. Tratando-se de viaturas automóveis é pertinente a extração de mais dados acerca do estado das mesmas com equipamentos de localização capazes de interpretar CAN e OBD. Uma vez que solução *Geocar* é muito dispendiosa, o problema a ser resolvido é encontrar formas de reduzir os custos que esta contém, mas mantendo o desempenho.

É pretendido no âmbito deste projeto de dissertação o desenvolvimento de uma nova solução, designada por a *Geocar Lite*, que recorra a alternativas de forma a remover a utilização dos serviços externos pagos (serviços da *Here*, da *Google*, da *Ez4u* e da *IntouchBiz*) utilizados pela solução atual e/ou reduzi-los ao máximo.

Para a realização deste projeto foram feitos um estudo e uma análise comparativa dos serviços externos que poderiam ser utilizados para o desenvolvimento da nova solução. Para além disso foi feito uma investigação em relação ao funcionamento do equipamento de localização que será integrado para testar a nova solução.

Após a análise foi realizado e avaliado a solução quanto à usabilidade, desempenho e custo obtido em relação à solução atual, a *Geocar*.

Palavras-chave: Gestão de Frotas, *Geocar*, *Geocar Lite*, Desenvolvimento de software, GPRS, CAN

Abstract

GisGeo Information Systems Ltd is a company dedicated to the development of geographic information systems, in which it offers Geocar, a solution for managing vehicle fleets that allows companies and individuals to plan and use their resources efficiently. Fleet management is carried out by installing equipment capable of reporting georeferential data via the GSM/GPRS network in customer's vehicles. In the case of cars, it is pertinent to extract more data about their condition with location equipment capable of interpreting CAN and OBD. Since the Geocar solution is very expensive, the problem to be solved is to find ways to reduce the costs it contains, while maintaining performance.

It is intended within this dissertation project to develop a new solution, called the Geocar Lite, that uses alternatives to remove the use of the external paid services (services from Here, Google, Ez4u and IntouchBiz) used by the current solution and/or reduce them as much as possible.

To carry out this project, a study and comparative analysis of external services that could be integrated for the development of this new solution was made. In addition, an investigation was made regarding the operation of the localization equipment that will be integrated to test the new solution.

After this analysis, the solution was performed and evaluated regarding usability, performance, and cost obtained in relation to the current solution, Geocar.

Keywords: Fleet Management, Geocar, Geocar Lite, Software Development, GPRS, CAN

Agradecimentos

Agradeço à minha família, em particular, aos meus pais, embora não estejam presentes, e aos meus padrinhos que me apoiaram durante todo o meu percurso académico.

Agradeço também ao orientador Joaquim Filipe dos Santos e ao supervisor Carlos Magalhães pela disponibilidade que ofereceram e orientação prestada no decorrer da realização desta dissertação.

Aproveito para agradecer à empresa *GisGeo Information Systems* e todos os seus colaboradores, por todo o apoio e disponibilidade prestada durante a realização do estágio.

Agradeço ainda aos meus colegas que foram um grande suporte em momentos menos fáceis do percurso académico e aos professores que me acompanharam durante o mestrado e que me ajudaram a crescer academicamente.

Por último, gostaria de agradecer ao ISEP por toda contribuição para a minha formação, ajudando-me nesta etapa de vida.

Muito obrigado.

Índice

1	Introdução	1
1.1	Motivação e Contexto	1
1.2	Problema	2
1.3	Objetivos	2
1.4	Estrutura do Documento	3
2	Contexto e Estado da Arte	4
2.1	Enquadramento	4
2.2	Sistemas de Gestão de Frotas	5
2.3	Tecnologias de Localização	5
2.3.1	Global Navigation Satellite System	6
2.3.1.1	GPS Assistido (A-GPS)	8
2.3.2	Dead Reckoning	8
2.4	Tecnologias e Protocolos de Comunicação	9
2.4.1	Sistema GSM	9
2.4.1.1	Sistema GPRS	11
2.4.2	Transmission Control Protocol (TCP) & User Datagram Protocol (UDP)	12
2.5	Ligação a Periféricos	12
2.5.1	Controller Area Network (CAN)	12
2.5.1.1	Arquitetura CAN	15
2.5.1.2	Standard CAN e Extended CAN	16
2.5.1.3	Codificação	18
2.5.1.4	Método de Endereçamento	19
2.5.2	OBD, On Board Diagnostic	20
2.6	Enquadramento tecnológico	22
2.6.1	Node.js	22
2.6.2	Angular	22
2.6.3	C#	22
2.6.4	Leaflet	23
2.6.5	PostgreSQL	23
2.7	Análise de serviços externo existentes	24
2.7.1	Serviços de Mapas	24
2.7.1.1	OpenStreetMaps	24
2.7.1.2	MapQuest	25
2.7.1.3	Google Maps	25
2.7.1.4	HERE Maps	26
2.7.2	Geocodificação e Geocodificação reversa	26
2.7.2.1	HERE Geocoding and Search API	26
2.7.2.2	Google Maps Geocoding API	27

2.7.2.3	Bing Search API	27
2.7.3	Serviço de comunicação	27
2.7.4	Comparação dos serviços descritos.....	28
2.8	Soluções existentes de gestão de frota.....	29
2.8.1	Frotcom - Intelligence Fleets	29
2.8.2	Webfleet Solutions	30
2.8.3	Verizon Connect	31
2.8.4	Comparação das plataformas analisadas	33
3	Análise de Valor.....	34
3.1	Processo de Inovação	34
3.2	New Concept Development	35
3.2.1	Identificação da Oportunidade	36
3.2.2	Análise de Oportunidade	36
3.2.3	Geração de Ideias	38
3.2.4	Seleção de Ideias	38
3.3	Valor da Solução.....	47
3.3.1	Valor, Valor para o Cliente, Valor Percecionado	47
3.3.2	Proposta de Valor.....	48
3.3.3	Businees Model Canvas.....	48
3.3.4	Quality Function Deployment	50
4	Desenho e Análise da solução.....	54
4.1	Arquitetura geral do Sistema	54
4.2	Equipamento de Localização.....	56
4.3	Camada de Recepção.....	57
4.4	Desenho da Arquitetura-Plataforma	60
4.5	Modelo de Domínio	62
4.5.1	Principais conceitos	63
4.6	Engenharia de Requisitos	64
4.6.1	Requisitos Funcionais	64
4.6.2	Requisitos não funcionais.....	73
4.7	Modelo Relacional	74
5	Implementação da solução	76
5.1	Equipamento de Localização.....	76
5.1.1	Componentes e Características.....	77
5.1.2	Configuração do Equipamento	78
5.2	Descrição da implementação	79
5.2.1	Implementação da Camada de Recepção-High Level	80
5.2.1.1	GeoAlerta.....	80
5.2.1.2	GeoData.....	81
5.2.1.3	GeoProcessor	81
5.2.1.4	GeoTask.....	82

5.2.1.5	GeoVeiculo.....	82
5.2.1.6	ConfigServer.....	83
5.2.2	Implementação da Camada de Receção-Low Level.....	84
5.3	Implementação Back-End da Plataforma Web.....	84
5.3.1	Implementação da criação do veículo.....	85
5.3.2	Implementação da visualização do percurso de um veículo.....	85
5.3.3	Implementação das coordenadas de um percurso.....	86
5.3.4	Implementação dos alertas de velocidade e ignição.....	86
5.4	Interface gráfica.....	87
5.5	Testes.....	91
5.5.1	Testes de Integração.....	91
5.5.2	Testes de Desempenho.....	92
6	Avaliação e Experimentação.....	94
6.1	Métricas.....	94
6.2	Hipóteses.....	94
6.2.1	Métrica 1.....	94
6.2.2	Métrica 2.....	95
6.2.3	Métrica 3.....	95
6.3	Metodologia.....	95
6.4	Análise de Resultados.....	95
6.4.1	Inquérito de satisfação do utilizador.....	95
7	Conclusão.....	100
7.1	Objetivos alcançados.....	100
7.2	Limitações e trabalho futuro.....	101
7.3	Apreciação final e pessoal.....	102
Anexo 1	Diagramas auxiliares.....	109
Anexo 2	Modelo Relacional.....	117
Anexo 3	Quality Funtion Deployment.....	118
Anexo 4	Análise de usabilidade da <i>Geocar Lite</i> (administrador).....	119
Anexo 5	Análise de usabilidade da <i>Geocar Lite</i> (cliente).....	123

Lista de Figuras

Figura 1 – Constelação de satélites GPS [71].....	7
Figura 2 – Dead Reckoning [72]	9
Figura 3 – Rede GSM [18]	10
Figura 4 – Arquitetura GPRS [20].....	11
Figura 5 – Camadas OSI e camadas Modelo CAN [24].....	13
Figura 6 – Níveis de tensão e bits dominantes e recessivos [25]	15
Figura 7 – Relação entre o comprimento da Rede e Velocidade de Transmissão	16
Figura 8 – Mensagem CAN da versão 2.0A (Standard CAN) [28]	17
Figura 9 – Mensagem CAN da versão 2.0B (Extended CAN) [28]	18
Figura 10 – Método de inserção de bit.....	18
Figura 11 – Método de endereçamento.....	19
Figura 12 - O conector J1962 macho (à esquerda) e a tomada J1962 fêmea (à direita) [73]	21
Figura 13 – Divisão do Processo de Inovação [53]	34
Figura 14 – Conceito de NCD e as suas partes [74]	35
Figura 15 – Esquematização Hierárquica de AHP.....	39
Figura 16 – Escala de Saaty [75].....	40
Figura 17 – Business Model Canvas.....	49
Figura 18 – Matriz de Correlação.....	52
Figura 19 – Matriz de Relação	52
Figura 20 – Diagramas de componentes (geral).....	54
Figura 21 – Arquitetura 3-tier [76].....	55
Figura 22 – Esquema do Equipamento de Localização.....	57
Figura 23 – Diagramas de componentes da Camada de Recepção.....	59
Figura 24 – Diagramas de componentes	60
Figura 25 – Padrão MVC [77]	61
Figura 26 – Modelo de Domínio	62
Figura 27 – Casos de Uso	64
Figura 28 – UC1 e UC2 Visualizar Percurso.....	65
Figura 29 - UC5-Ver veículos no mapa.....	68
Figura 30 – Equipamento de Localização FMB920	77
Figura 31 – Esquema do Equipamento FMB920.....	78
Figura 32 – Configurador do equipamento	79
Figura 33 – Implementação de processar os alertas	80
Figura 34 – Excerto da implementação do GeoProcessor.....	81
Figura 35 – Implementação da classe GeoTask	82
Figura 36 – Verificar os alertas de velocidade e de ignição.....	83
Figura 37 – Diferentes operações no ConfigServer	83
Figura 38 – Excerto da implementação do <i>ReceiveServer</i>	84
Figura 39 – <i>Controller</i> do criar veículo.....	85
Figura 40 – <i>Query</i> para encontrar o percurso de um veículo.....	85

Figura 41 – <i>Queries</i> para procurar as coordenadas de um percurso.....	86
Figura 42 – <i>Controller</i> do configurar os alertas.....	86
Figura 43 – <i>Dashboard</i> do Administrador.....	87
Figura 44 – Página de Gerir Equipamentos.....	87
Figura 45 - Página de Edição do Cliente e criação de veículos.....	88
Figura 46 – Página do Cliente com opções.....	88
Figura 47 – Visualização do Percurso Atual.....	89
Figura 48 – Visualização dos Percursos.....	89
Figura 49 – Configurar Alertas.....	90
Figura 50 – Relatório dos percursos e de ignição.....	90
Figura 51 – Testes de Integração à plataforma.....	91
Figura 52 – <i>Requests</i> inseridos.....	92
Figura 53 – Gráfico de teste de desempenho.....	93
Figura 54 – UC3 Visualizar Relatório.....	109
Figura 55 – UC4 Configurar Alertas.....	110
Figura 56 – UC6 Criar Veículos, UC7 Editar Veículos e UC8 Desativar Veículos.....	111
Figura 57 - UC9 Criar Clientes, UC10 Editar Clientes e UC11 Desativar Clientes.....	112
Figura 58 – UC12 Criar Equipamentos, UC13 Editar Equipamentos e UC14 Eliminar Equipamentos.....	113
Figura 59 – Diagrama de sequência do UC6.....	114
Figura 60 – Diagrama de sequência do UC2.....	115
Figura 61 – Diagrama de sequência do UC4.....	116

Lista de Tabelas

Tabela 1 – Comparação entre serviços de mapas	28
Tabela 2 – Análise SWOT da oportunidade	37
Tabela 3 – Comparação entre os critérios	40
Tabela 4 – Matriz de comparação dos critérios	41
Tabela 5 – Matriz de comparação dos critérios	41
Tabela 6 – Tabela de importância dos critérios	41
Tabela 7 – Índice de Consistência Aleatória	42
Tabela 8 – Comparação entre as soluções, segundo critério A	44
Tabela 9 – Matriz de comparação, relativamente ao critério A	44
Tabela 10 – Matriz de comparação normalizada e prioridade relativa (critério A).....	44
Tabela 11 – Comparação entre as soluções, segundo o critério B	44
Tabela 12 – Matriz de comparação, em relação ao critério B	45
Tabela 13 - Matriz de comparação normalizada e prioridade relativa (critério B)	45
Tabela 14 – Comparação entre soluções, segundo o critério C.....	45
Tabela 15 - Matriz de comparação, em relação ao critério C	45
Tabela 16 - Matriz de comparação normalizada e prioridade relativa (critério C)	46
Tabela 17 – Prioridades relativas das soluções.....	46
Tabela 18 – Matriz de Pesos Globais das soluções	46
Tabela 19 – Valores de IC e RC para as matrizes	47
Tabela 20 – Necessidades dos utilizadores classificados por importância.....	51
Tabela 21 – Tabela de Requisitos Funcionais (cliente)	96
Tabela 22 – Tabela de Requisitos Funcionais (administrador)	96
Tabela 23 – Tabela de tempo de resposta	97

Acrónimos e Símbolos

Lista de Acrónimos

MEI	Mestrado em Engenharia Informática
ISEP	Instituto Superior de Engenharia do Porto
GPS	<i>Global Positioning System</i>
ICIAO	<i>International Civil Aviation Organization</i>
A-GPS	<i>Assisted GPS</i>
GNSS	<i>Global Navigation Satellite System</i>
GSM	<i>Global System for Mobile Communications</i>
TDMA	<i>Time Division Multiple Access</i>
GPRS	<i>General Packet Radio Service</i>
CAN	<i>Controller Area Network</i>
ECU	<i>Engine Control Unit</i>
OBD	<i>On-Board Diagnostic</i>
DTC	<i>Diagnostic Trouble Codes</i>
PID	<i>Parameter Identification Number</i>
HTTP	<i>Hypertext Transfer Protocol</i>
M2M	<i>Machine to Machine</i>
SIM	<i>Subscriber Identity Module</i>
IMEI	<i>International Mobile Equipment Identity</i>
BSC	<i>Base Station Controller</i>
GGSN	<i>Gateway GPRS Support Node</i>
SGSN	<i>Serving GPRS Support Node</i>
MSC	<i>Mobile Services Switching Center</i>
TCP	<i>Transmission Control Protocol</i>

UDP	<i>User Datagram Protocol</i>
ISO	<i>International Organization for Standardization</i>
SQL	<i>Structured Query Language</i>
OSI	<i>Open System Interconnection</i>
FFE	<i>Fuzzy Front End</i>
NPD	<i>New Product Development</i>
QDF	<i>Quality Function Deployment</i>
AHP	<i>Analytic Hierarchy Process</i>
MVC	<i>Model-View-Controller</i>

1 Introdução

Neste capítulo é apresentado a motivação e contexto do problema em estudo, os objetivos fundamentais a atingir para esta dissertação, e ainda informação acerca da estrutura organizacional do documento.

1.1 Motivação e Contexto

Este projeto surgiu no âmbito do Estágio Curricular para a conclusão do Mestrado em Engenharia Informática na área de especialização de Sistemas Computacionais e foi proposto pela empresa *GisGeo Information Systems*, com o intuito de desenvolver uma nova solução designada por *Geocar Lite* que seja menos dispendiosa do que a sua plataforma atual, a *Geocar*. Para isso, será necessário analisar e estudar a solução atual e verificar as várias tecnologias e soluções para o desenvolvimento da nova solução.

A seleção do estágio curricular na empresa acima referida deve-se a vários desafios, nomeadamente, o facto de aproveitar o projeto de estágio no sentido de melhorar as competências e aprofundar os conhecimentos já adquiridos ao longo da MEI, com auxílio dos profissionais da empresa. Outro desafio que levou à escolha do referido estágio foi também o desenvolvimento de uma nova solução que será utilizada no mundo real, no qual poderá ajudar a *GisGeo* a ter o mesmo controlo sobre as suas frotas utilizando os menores recursos possíveis e com mesmo desempenho.

Ao longo do projeto de estágio foram utilizados os processos de engenharia abordados no ISEP que permitem compreender melhor as etapas envolvidas no desenvolvimento do mesmo, nomeadamente a engenharia de requisitos, análise e modelação.

1.2 Problema

Empresas cujo modelo de negócio esteja dependente de uma frota de veículos, como o caso de transportadores e outros prestadores de serviços, estão expostas aos custos inerentes do uso e manutenção das suas frotas. A solução atual designada por Geocar é um Sistema de Gestão de Frotas que através de instalação de pequenos equipamentos localização GPS nas viaturas, fornece uma plataforma sólida que permite às empresas saber o estado e localização da sua frota, bem como, mitigar os impactos ambientais da mesma. Uma vez que é uma solução muito dispendiosa, o problema a ser resolvido é encontrar formas de reduzir os custos que esta contém, mas mantendo o desempenho. Para isso é necessário criar uma solução recorrendo a alternativas de forma a remover a utilização dos serviços externos pagos utilizados pelo Geocar e/ou reduzi-los ao máximo.

1.3 Objetivos

A solução atual da GisGeo, o Geocar, é um sistema de Gestão de Frotas que se encontra desenvolvido em diferentes tecnologias, nomeadamente em *php* e Java, que recorre a serviços externos, tais como, os serviços da *Here* [1], da Google [2], da *Ez4u* [3] e também da *IntouchBiz* [4]. Para além disso, o sistema encontra-se alocado numa *Cloud* que além de disponibilizar *webservices* e Aplicações *Web*, também armazena dados de localização enviados por dispositivos de localização autónomos. Os serviços externos e a *cloud* acarretam custos elevados para garantir o funcionamento em pleno do sistema, pretende-se por isso reduzi-los ao máximo. Foram definidos objetivos concretos que pretendem reduzir o número de serviços externos a usar de forma a diminuir as despesas mensais do Geocar, nomeadamente:

- Retirar a utilização de serviços externos para referenciar geograficamente as viaturas num mapa, um dos serviços fornecidos pela Api Google;
- Abdicar da disponibilização de moradas, um dos serviços da Api *Here* usado atualmente para efetuar e traduzir uma determinada coordenada GPS em nome da rua específica;
- Abdicar dos serviços das Api *Ez4u* e da Api *IntouchBiz*, utilizados para envio de SMS pelo sistema de alertas;
- Reduzir o incremento de espaço anual em *Cloud* que é necessário para manter todo o histórico dos dados de localização armazenados pela plataforma;
- Reduzir fluxo de comunicações entre os equipamentos móveis e a respetiva *Cloud*;
- Reduzir número de técnicos de instalação, manutenção e suporte associados à plataforma.

Inicialmente será feita uma pesquisa do estado de arte em sistemas de gestão de frotas, bem como o estudo e análise das tecnologias de GPS, GSM, GPRS e dos sistemas OBDII e CAN. Para além disso será feita uma investigação em relação ao funcionamento do dispositivo de localização que será integrado para testar a nova solução. Por fim, com os objetivos acima referidos e descritos, estes serão analisados, de modo a saber quais os requisitos funcionais e não funcionais a definir para o desenvolvimento da plataforma *web* e *mobile*.

1.4 Estrutura do Documento

Nesta secção será apresentada a estrutura desta tese, fornecendo uma visão geral do documento e dos aspetos abordados.

No primeiro capítulo é descrito o âmbito em que o projeto está envolvido no contexto da organização *GisGeo Information Systems*. Além disso, são descritos o problema e os objetivos do projeto.

No segundo capítulo é realizado o enquadramento teórico e tecnológico, que permite uma melhor compreensão dos conceitos-chave e é também realizada uma análise do mercado referente algumas soluções existentes que resolvem problemas semelhantes.

No terceiro capítulo é descrita a análise de valor do projeto, com o intuito de demonstrar o seu valor de mercado para as partes interessadas.

No quarto capítulo é descrito a análise e o desenho da solução, no qual é feita uma análise da solução do projeto numa perspetiva técnica.

No quinto capítulo é descrito a implementação e os testes da solução, no qual é feita uma descrição pormenorizada das partes que compõe a nova solução, bem como é feito testes à mesma.

No sexto capítulo é descrito uma avaliação da nova solução, onde é apresentado as técnicas adotadas para a avaliação da solução e os resultados da mesma.

No sétimo capítulo é descrito as conclusões relativas ao projeto desenvolvido, mencionando os pontos fortes e as falhas da solução, são referidos e justificados os contratemplos, e são sugeridas abordagens e soluções para um possível projeto a ser utilizado no futuro.

Por fim, o capítulo de Bibliografia contém todas as referências bibliográficas utilizadas ao longo do relatório para suportar o conteúdo elaborado.

2 Contexto e Estado da Arte

Neste capítulo descreve um conjunto de conceitos, nomeadamente, os protocolos e as tecnologias de comunicação e as tecnologias de localização, bem como a explicação destas tecnologias nos dispositivos de localização, que servirão de base para a melhor compreensão do desenvolvimento do trabalho.

2.1 Enquadramento

Desde o final do século XX, com o acentuar da globalização, as empresas encontram cada vez mais dificuldades, num mundo em constante mudança, para encontrar alternativas de mercado estáveis que permitam obter padrões de crescimento sustentáveis, que sirvam de suporte à evolução da organização.

É, portanto, necessário encontrar soluções que possibilitem explorar ao máximo os recursos ao dispor da organização, especialmente no que diz respeito aos meios que levam os produtos/serviços junto do cliente.

Assim, o meio empresarial sente a necessidade de ter uma forma de gerir o transporte, de modo a garantir a segurança dos seus produtos e funcionários, reduzir os custos, os desperdícios e aumentar os lucros. Desta forma surge a gestão de frotas, como forma de otimizar os meios ao dispor. [5]

2.2 Sistemas de Gestão de Frotas

Os sistemas de gestão de frotas como é o exemplo a Geocar, desenvolvido pela GisGeo, são sistemas de informação que permitem ao utilizador interagir com a sua frota de forma ágil e eficiente, sendo característica a existência de mapas digitais onde, em tempo-real, é possível ao utilizador visualizar o estado e posição de todos os seus veículos. Analisando os dados presentes na plataforma o utilizador pode obter informações que permitem a tomada de decisões por forma a melhorar a eficiência nas deslocações da sua frota permitindo assim obter uma maior produtividade e redução dos custos associados com o uso e manutenção das viaturas. Estando os veículos permanentemente ligados ao sistema de gestão permite uma proteção contra o roubo das mesmas, uma vez que a qualquer momento o utilizador pode saber o estado e localização de todos os seus veículos. Estes sistemas estão dependentes da instalação de hardware na respetiva viatura. Os equipamentos instalados são capazes de obter informação acerca da sua localização através de um módulo GPS, bem como o estado do veículo recorrendo há ligação da ficha OBD ou linhas CAN (*Controller Area Network*) presentes no mesmo. Toda esta informação recolhida é enviada para o servidor central através da rede GSM/GPRS ou usada para gerar determinados eventos, tais como, o excesso de velocidade ou excesso de ralenti. Como o funcionamento destes sistemas estão dependentes de equipamento instalado nas viaturas e do servidor central qualquer falha que poderá acontecer compromete a disponibilidade do serviço de gestão.

2.3 Tecnologias de Localização

A localização, nos dias que correm, é um tópico muito abordado e apresenta-se cada vez mais como uma necessidade real do cidadão comum. Esta necessidade nasce, não só, porque as pessoas cada vez mais querem saber onde estão e como pretendem deslocar-se para um determinado local, bem como saberem também onde se encontram determinados bens, pessoas ou animais. E como existe procura, o mercado cria a oferta, existindo neste momento vários sistemas de localização baseados em diferentes tecnologias.

Destes diferentes sistemas comerciais de localização aquele que se encontra mais difundido é o *Global Positioning System* (GPS), uma vez que apresenta uma excelente relação custo/benefício, permitindo uma localização a nível global. Para além do GPS existem outros meios de localização que estão a crescer muito nos últimos anos, nomeadamente os sistemas de posicionamento de rádio terrestre que usam uma rede de pacotes de rádio de baixa frequência, em que têm sido usados como alternativa aos sistemas baseados em GPS.

2.3.1 Global Navigation Satellite System

Sistema Global de Navegação por Satélite (GNSS) refere-se à constelação de satélites que possibilita o posicionamento em tempo real de objetos, bem como a navegação em terra ou mar. Esses sistemas são utilizados em diversas áreas, tais como, o mapeamento topográfico e geodésico, aviação, navegação marítima e terrestre, monitorização de frotas, demarcação de fronteiras, agricultura de precisão, entre outros usos.

O desempenho desses sistemas é avaliado segundo os seguintes critérios [6]:

- **Precisão:** A diferença entre a medição recebida e a posição real;
- **Integridade:** A capacidade de o sistema despoletar um alerta quando detetar uma medição anormal;
- **Continuidade:** Capacidade de o sistema trabalhar sem interrupções;
- **Disponibilidade:** Percentual do tempo que o sistema preenche os requisitos anteriores (precisão, integridade e continuidade).

A terminologia GNSS foi utilizado pela primeira vez em 1991 pela Associação Internacional de Aviação Civil (*International Civil Aviation Organization* – ICAO) para designar os sistemas de posicionamento por satélites artificiais com cobertura mundial. Considera-se que para obter cobertura global, uma constelação de satélites deve possuir um mínimo de 24 satélites posicionados de forma que um determinado recetor sobre a superfície terrestre possa ter um mínimo de quatro satélites no horizonte para serem detetados. Quatro satélites são utilizados para determinar a posição do recetor: o primeiro satélite é utilizado para sincronizar o tempo entre sistema de tempo dos satélites e do recetor, enquanto outros três satélites são utilizados para calcular as coordenadas tridimensionais. Até ao momento, apenas dois sistemas GNSS são considerados plenamente operacionais e com alcance global: sistema GPS e o sistema russo GLONASS. Desde dezembro de 2016, o sistema europeu Galileu, ainda que não de forma completa, tem fornecido também posicionamento com alcance global. [7]

Como já foi referido, um dos sistemas de navegação por satélite mais conhecidos é designado por *Global Positioning System* (GPS) que permite obter informação acerca do tempo e localização na superfície terrestre desde que haja uma linha de visão a pelo menos quatro satélites. O projeto de desenvolvimento do sistema GPS teve início em 1973 nos Estados Unidos com o objetivo de ultrapassar as limitações dos sistemas de navegação anteriores. Este sistema baseado originalmente em vinte e quatro satélites ficou completamente operacional em 1995 e contempla o uso militar, comercial e civil estando este último disponível a qualquer pessoa que possua um dispositivo recetor. Para além deste sistema de navegação por satélite, existem também outras implementações, tais como, o GLONASS, GALILEU e o COMPASS.

O GLONASS, *Global Navigation Satellite System*, é um sistema russo cujo desenvolvimento teve início em 1976 e que utiliza uma constelação de 24 satélites artificiais com cobertura global que orbitam a altitudes inferiores à dos satélites do sistema de GPS [8].

O GALILEU, o sistema de navegação por satélite da União Europeia em que é o único com fins exclusivamente civis, tem prevista uma constelação com trinta satélites, dos quais vinte e sete estarão permanentemente ativos e três de reserva, dispostos em três orbitas terrestres de media altitude (a aproximadamente 23222 Km de altitude em relação ao nível médio da água do mar e uma inclinação orbital em relação ao equador de cinquenta e seis graus), promete precisões superiores aos sistemas já existentes, havendo testes agendados em 2014 e foi prevista uma completa operabilidade do sistema em 2020 [9].

O COMPASS ou *BeiDou-2* é um sistema de navegação por satélite chinês formado por trinta e cinco satélites dos quais cinco são geostacionários [10] para suportar uma implementação anterior *BeiDou-1*, no qual este sistema é composto por 3 satélites e possui aplicações e cobertura limitadas [10]. Atualmente existem no mercado equipamentos que suportam GPS como também GLONASS e estão preparados para receber GALILEO aferindo medições mais rápidas e precisas, pois têm à sua disposição a informação de mais satélites.

A constelação dos satélites do sistema de GPS representado na Figura 1 é composta por seis planos orbitais a 20200 Km de altitude e com uma inclinação de 55° em relação ao equador. Cada satélite GPS contém relógios atômicos e emissores de ondas eletromagnéticas que são utilizadas para o envio para os recetores relativamente à informação sobre o posicionamento e tempo do satélite. [11]

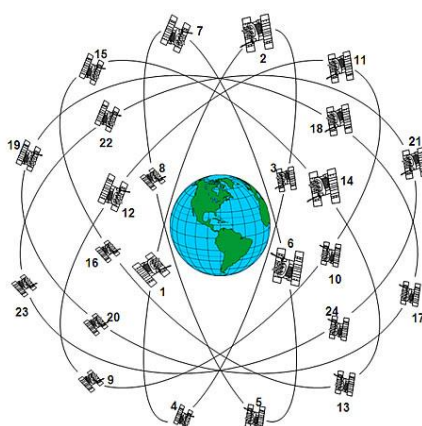


Figura 1 – Constelação de satélites GPS [71]

Estes satélites estão distribuídos de uma forma que um recetor, localizado em qualquer ponto da superfície terrestre, estará sempre ao alcance de pelo menos três deles (quatro ou mais para uma maior precisão). Portanto a localização é baseada em cálculos que ocorrem através de um processo designado por **triangulação**.

No processo de triangulação, os três satélites enviam o sinal para o recetor que calcula o tempo que cada sinal demorou a chegar até ele [12]. Além da sua localização terrestre, o recetor GPS também consegue calcular a altura do recetor em relação ao nível do mar, no entanto é

necessário um quarto satélite. Tanto os satélites como os recetores GPS possuem um relógio interno que regista as horas com uma enorme precisão, em nanossegundos. Quando o satélite emite o um determinado sinal para o recetor, o horário em que este saiu do satélite também é enviado. [12]

Ao captar os sinais dos satélites, o recetor calcula a distância entre eles pelo intervalo de tempo entre instante atual e o instante em que os sinais foram enviados. Tendo em conta a velocidade de propagação do sinal, o recetor pode situar-se na intersecção desses dados, permitindo identificar exatamente onde o aparelho se encontra na Terra. Para que a posição do recetor seja sempre atualizada, os envios desses sinais acontecem a uma velocidade de 300 mil quilómetros por segundo (velocidade da luz) no vácuo. [12]

2.3.1.1 GPS Assistido (A-GPS)

O A-GPS consiste numa versão otimizada do serviço GPS, este utiliza a rede GSM/GPRS de modo a recolher informações da localização fornecidas pela estação base onde o dispositivo se encontra ligado. Ao contrário do serviço GPS, o A-GPS é um serviço pago, pois é necessária uma ligação GPRS/UMTS com a estação para se receber os dados referentes à localização. Este serviço tem especial impacto em zonas fortemente urbanizadas, ou dentro de edifícios onde muitas vezes não se consegue adquirir sinal GPS. Com este sistema é possível melhorar a precisão das localizações nessas circunstâncias. [13]

O sistema A-GPS para funcionar necessita de um dispositivo recetor de GPS na estação base de modo a conseguir captar sinal GPS, esse sinal é depois enviado para o terminal móvel que vai utilizar essa informação como uma referência, o que permite agilizar todo o processo de cálculo da posição, fazendo com que o sistema funcione em locais onde anteriormente não funcionava, como é o caso de prédios e garagens.[13]

Existe uma outra solução utilizada pelos equipamentos recetores GPS que em vez de recorrer ao serviço de A-GPS diretamente, guarda a sua última posição a qual posteriormente é utilizada para o cálculo quando inicia uma nova sessão.

2.3.2 Dead Reckoning

O *Dead Reckoning* é um método de cálculo ou estimação da posição atual com base numa posição anterior bem definida e no conhecimento das variações da direção e velocidade na deslocação efetuada [14], Figura 2.

A precisão deste método de localização está fortemente condicionada pela precisão das medições de velocidade e de direção, devido ao facto de os erros serem acumulativos. Isto é, dado que o cálculo de uma nova localização é sempre baseado no cálculo da posição anterior, o erro resultante vai ser o somatório dos erros de todas as posições.

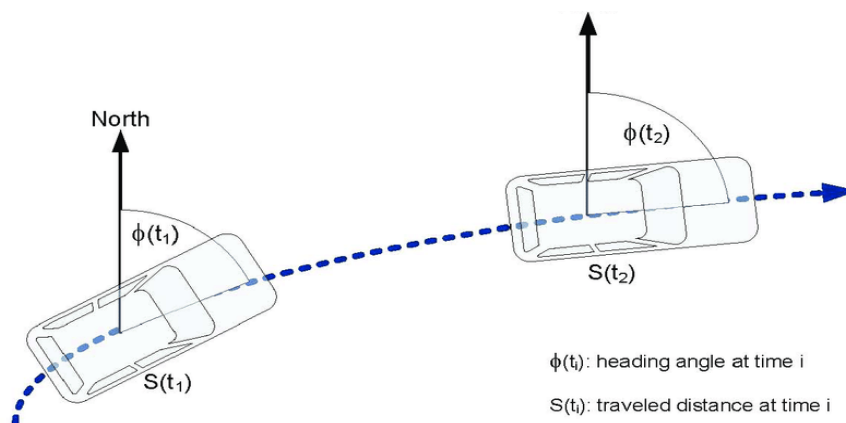


Figura 2 – Dead Reckoning [72]

Atualmente, o *Dead Reckoning* é implementado em alguns sistemas de navegação automóvel com objetivo de superar as limitações das tecnologias GPS /GNSS. Os sinais de micro-ondas por satélite não são captados em garagens e túneis e, muitas das situações é ainda mais difícil a captação em locais que contenham árvores, uma vez que as propagações de sinais ficam bloqueadas para os satélites. Num sistema de navegação com *Dead Reckoning*, o veículo é equipado com sensores que conhecem a circunferência da roda e registam as rotações das rodas e a direção. Frequentemente, esses sensores já estão presentes nos carros com outras finalidades (sistema de travagem anti bloqueio, controlo eletrónico e estabilidade) e podem ser lidos pelo sistema de navegação a partir do barramento da rede de área do controlador. [15]

2.4 Tecnologias e Protocolos de Comunicação

Faz parte de uma solução de gestão de frotas o conhecimento em tempo real do estado e posição das viaturas, daí que é importante que os equipamentos instalados sejam capazes de transmitir essas mesmas informações. Maioritariamente esta comunicação é apenas entre um equipamento e um servidor central, tratando-se então de uma comunicação *Machine to Machine* (M2M). [16]

2.4.1 Sistema GSM

O sistema GSM (*Global System for Mobile Communications*) é um sistema 2G de comunicações móveis. O GSM foi o primeiro sistema de comunicações móveis digital que teve a possibilidade de comunicações de voz e dados (até aproximadamente 13 kbit/s). O GSM tem uma interface rádio que usa TDMA (*Time Division Multiple Access*), que consiste em separar a informação dos diferentes utilizadores através de intervalos de tempo diferentes, funcionando nas bandas de frequência de 0,9 GHz e de 1,8 GHz, com o telefone transmitindo potências máxima de 2 W e com a mínima de 0,003 W em 0,9 GHz, e máxima de 1 W e com a mínima de

0,001 W em 1,8 GHz.[17] Para se ligar a uma rede GSM, um terminal precisa de um cartão *Subscriber Identity Module* (SIM) que possui o número atribuído pela operadora, bem como o *International Mobile Subscriber Identity* (IMSI). Cada terminal é também identificado pelo seu *International Mobile Equipment Identity* (IMEI), identidades que após o utilizador introduzir o seu *Personal Identification Number* (PIN) são usadas pelo terminal para se autenticar na rede GSM permitindo ao mesmo efetuar e receber chamadas e mensagens. [18]

Para efetuar uma chamada de voz o equipamento digitaliza e comprime o sinal que recebe do microfone e envia-a para a *Base Transceiver Station* (BTS) que opera a célula onde o mesmo se encontra. As arquiteturas em células permitem a mobilidade e a partilha do meio com os outros equipamentos devido ao reaproveitamento das gamas de frequência em diferentes células não adjacentes. Por sua vez a estação BTS está ligada a uma estação de controlo, *Base Station Controller* (BSC), que gere a distribuição de ligações de várias BTS. Estes controladores estão ligados à central *Mobile Switching Center* (MSC) onde ocorre a autenticação dos equipamentos móveis e onde é efetuada a ligação das chamadas aos destinatários, conforme se pode visualizar na Figura 3. Com o GSM é possível também o envio de pequenas mensagens de texto designadas por *Short Message System* (SMS) em que possibilita efetuar chamadas de dados que após um determinado período de estabelecimento de ligação permitem o acesso à Internet através do terminal. Este método de ligação oferece uma taxa de transmissão máxima de 9,6 kbit/s requer sempre um período de estabelecimento de ligação a cada nova ligação por não se estabelecer uma ligação permanente à rede.[19]

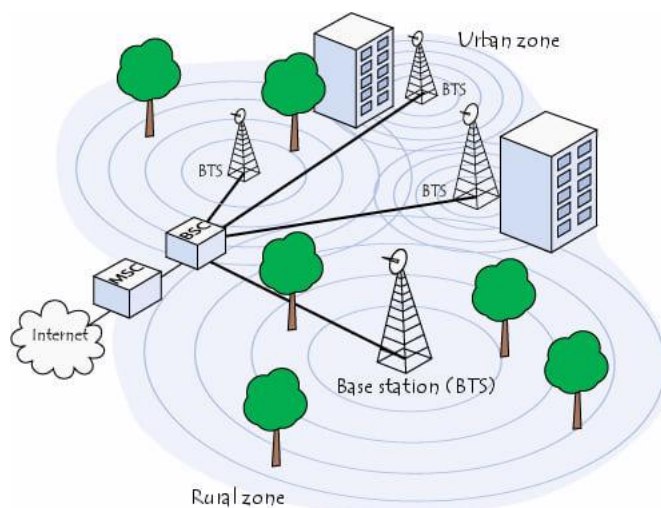


Figura 3 – Rede GSM [18]

2.4.1.1 Sistema GPRS

O *General Packet Radio Service* (GPRS) é uma tecnologia que constitui uma extensão da rede GSM (*Global System for Mobile Communications*). Este foi criado com o objetivo de permitir às operadoras GSM a possibilidade de disponibilizar acesso às redes de transporte de dados por pacotes (comutação por pacotes), como é o caso da Internet.

A arquitetura GPRS utiliza toda estrutura já construída na rede GSM, incluídos novos elementos de rede e interfaces, além de modificar alguns já existentes, como se pode visualizar na Figura 4. Numa rede de comutação de circuitos, como a PSTN, ISDN e mesmo o GSM, durante a comunicação entre dois elementos é criado um canal entre eles por onde a informação circula. Numa rede de comutação de pacotes o canal físico é partilhado por todos os elementos. A informação é “dividida” em pacotes que posteriormente será reencaminhado para o destino. [20]

Sendo uma rede de comutação de pacotes, com o GPRS as operadoras podem cobrar aos clientes em função do volume de dados transferidos, por outro lado, ao GSM tem de ser baseado na duração da ligação. O GPRS permite taxas de transferências de dados até aos 170 kbps, enquanto no GSM é apenas até aos 9 kbps. [21]

Para utilizar o GPRS é necessário a introdução de dois novos elementos na rede GSM, o GGSN (*Gateway GPRS Support Node*) e o SGSN (*Serving GPRS Support Node*). O GGSN é o *gateway* responsável pela ligação às redes IP externas.[22] Relativamente ao SGSN, este funciona ao mesmo nível que o MSC (*Mobile Services Switching Center*) e representa para a parte do sistema referente à comutação de pacotes o mesmo que o MSC para a parte de comutação de circuitos. É responsável pelas questões de segurança e autenticação, acede às bases de dados de apoio e faz a deteção e registo dos novos MS na sua área de cobertura (**Erro! A origem da referência não foi encontrada.**).

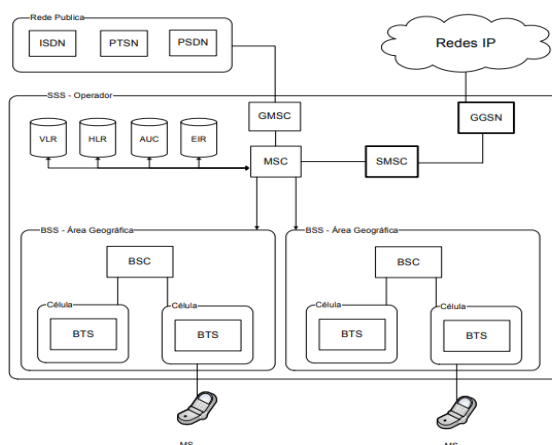


Figura 4 – Arquitetura GPRS [20]

2.4.2 Transmission Control Protocol (TCP) & User Datagram Protocol (UDP)

TCP e o UDP são utilizados para encapsular uma mensagem que se deseja enviar em pacotes possibilitando que a mesma seja enviada através de uma rede IP.

O TCP é um protocolo orientado à conexão que permite que seja estabelecida uma ligação entre duas aplicações em que os pacotes são enviados de forma ordenada com verificação de erros e retransmissão, quando necessário garantindo a integridade do conteúdo enviado. Para tal este protocolo atua em três estados, estabelecimento de ligação (onde o protocolo tenta estabelecer uma sessão fidedigna para que possa haver transmissão de informação), transferência de dados (após a criação da sessão é possível então a transmissão dos conteúdos) e por último, a fase de finalização da ligação (terminada a emissão de dados) é encerrando então a sessão entre os dois clientes. [23]

O UDP, ao contrário do anterior, é um protocolo não orientado à conexão pelo que não garante a entrega dos pacotes nem a sua ordenação ou duplicação, apenas garante a integridade dos dados através de *checksums*, mas não efetua qualquer retransmissão, pois não existe ligação entre o emissor e o recetor.[23] Através deste protocolo as aplicações podem enviar informação sem que seja necessária a criação de uma sessão ou canal exclusivos, o que pode ser explorado em casos em que a perda de um pacote é preferível ao tempo de espera por pacotes em atraso.

2.5 Ligação a Periféricos

Atualmente os veículos modernos possuem diversos módulos de controlo eletrónicos sendo tipicamente o mais importante o do motor. Vários destes módulos podem funcionar de forma independente, formando subsistemas fechados, outros necessitam de comunicar com módulos vizinhos, sensores e atuadores tornando-se necessário que exista uma infraestrutura que permita estas mesmas comunicações. [24]

2.5.1 Controller Area Network (CAN)

O protocolo CAN começou a ser desenvolvido em 1983 pela empresa Robert Bosch GmbH, no entanto, só em 1986 foi lançado oficialmente pela SAE (*Society of Automotive Engineers*), entrando no mercado em 1987.

A sua aplicação vai desde ligações de baixo custo até redes de uma velocidade de 1 Mbit/s, como é o caso da eletrónica automóvel, uma vez que se trata de um protocolo de comunicação serie síncrono capaz de suportar eficazmente o controlo de sistemas distribuídos em tempo real e com um elevado nível de segurança. Durante o seu desenvolvimento existiram inúmeras versões, sendo a última a CAN 2.0 A / B, publicada em 1991. A versão CAN 2.0A utiliza o formato

padrão com um identificador de 11 bits e a CAN 2.0B um formato de identificação alargado de 29 bits. [25]

Em 1993 a *International Organization for Standardization* (ISO) publicou a norma ISO 11898, que mais tarde foi reestruturada em três partes distintas: a norma ISO 11898-1, que abrange a camada *Data Link Layer*, a norma ISO 11898-2 que especifica a camada física de comunicação de alta velocidade da rede CAN e a norma ISO 11898-3, que foi introduzida mais tarde com o objetivo de especificar a camada física de comunicações de baixa velocidade e tolerante a erro. No entanto, as normas responsáveis pelas camadas físicas 11898-2 e 11898-3, não fazem parte das especificações da *Bosh CAN 2.0*. [25]

A compatibilidade entre diversos dispositivos é garantida através de 3 camadas, tendo como base o modelo *Open Systems Interconnection* (OSI):

- Camada de Aplicação (*Object Layer*);
- Camada de Ligação Lógica (*Transfer Layer*);
- Camada de Ligação Física (*Physical Layer*).

Na Figura 5 é possível observar as camadas de aplicação e ligação que compreendem todos os serviços e funções da camada ligação de dados (*data link*) definida pelo modelo OSI. [24]

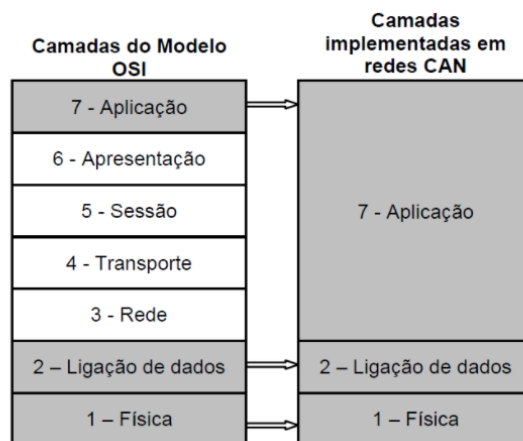


Figura 5 – Camadas OSI e camadas Modelo CAN [24]

Camada de Aplicação

Esta camada tem como finalidade converter a representação da informação para um formato universal e deste modo facilitar as comunicações entre aplicações que residem em ambientes muito diferentes um do outro. [26]

Camada de Ligação de Dados

Nesta camada garante a comunicação entre pontos de uma rede que estão diretamente ligados entre si e usam o mesmo tipo de camada física. Recebe um SDU (*Service Data Unit*) da camada 3 (Camada de Rede), adiciona a sua PCI (*Protocol Control Information*) e passa o resultado à camada 1 (Camada de Ligação Física), sob a forma de bits, os PDU (*Protocol Data Unit*) da camada 2 tomam tipicamente a designação de tramas ou quadros ("*frames*"). Neste contexto a camada 2 pode ter de lidar com o endereçamento, deteção de erros, controlo de erros, controlo de fluxo e acesso ao meio de transmissão. É importante salientar que todas estas operações estão limitadas aos pontos da rede que partilham o mesmo nível físico. [26]

Camada de Ligação Física

Nesta camada os dados provenientes do nível 2 (Camada de Ligação de Dados), na sua forma mais elementar (bits) são transformados em sinais, ou alteração de grandeza de sinais adequados ao meio responsável pela propagação do mesmo desde o emissor até recetor. O recetor monitoriza o sinal e de acordo com as variações detetadas na grandeza produz bits que entrega à camada 2. [26]

As redes CAN apresentam as seguintes características principais: [25]

- Comprimento dos barramentos: varia de 40 metros a 1 quilometro;
- Taxa de transferência de dados: 50Kbit/s até 1 Mbit/s;
- Capacidade de dados: 0 até 8 bytes por mensagem;
- Comprimento das mensagens: máxima de 130 bits;
- Priorização das mensagens;
- Garantia de tempos de latência;
- Flexibilidade de configuração dos sistemas e redes;
- Receção múltipla (*multicast, broadcast*) com sincronização de tempo;
- Sistema de consistência de dados;
- Deteção e sinalização de erros;
- Retransmissão automática de mensagens corrompidas quando o barramento se encontra novamente disponível;
- Distingão entre erros temporários e falhas permanentes de nós e desligamento automático de nós que apresentem falhas.

2.5.1.1 Arquitetura CAN

A rede CAN tem uma arquitetura multi-mestre onde são ligados diferentes módulos, formando assim diferentes nós. No mínimo são necessários dois módulos para formar uma rede CAN, sendo que, atualmente, um veículo pode possuir mais de 70 módulos. Todos os módulos da rede CAN têm direitos iguais, ou seja, num determinado momento qualquer módulo pode enviar uma mensagem, sendo estas enviadas em *multicast*, isto é, são enviadas para todos os módulos existentes na rede.[27]

A rede CAN tem um funcionamento semelhante ao protocolo CSMA/CD + NDA (*Carrier Sense Multiple Access with Non-Destructive Arbitration*), garantindo que todos os módulos verificam o estado do barramento antes de iniciarem a comunicação, analisando assim se outro módulo está a enviar uma mensagem com maior prioridade. Caso isto se verifique, o módulo cuja mensagem for de menor prioridade interrompe a sua transmissão e o módulo com a mensagem de maior prioridade inicia o seu envio. No final o módulo com a mensagem de menor prioridade continua o seu envio sem que este tenha de ser reiniciado. [25]

Os diferentes módulos da rede não possuem endereços. As mensagens que enviam contêm identificadores usados, tanto para definir prioridades no envio, como para os restantes módulos presentes na rede, no qual decidem se a respetiva mensagem em transito deve ser processada ou ignorada. Os identificadores com menor valor binário têm maior prioridade.

Outra característica importante é o conceito *Non Return to Zero (NRZ)*, onde os dados não são representados pelos níveis lógicos 0 ou 1, mas sim por valores de tensão específicos e constantes, ou seja, por bits dominantes e bits recessivos criados em função da condição presente nos fios CAN_H e CAN_L. (Figura 6) A velocidade de transmissão de dados é inversamente proporcional ao comprimento do barramento. A maior taxa de transmissão especificada é de 1 Mbit/s para um barramento de 40 metros e a menor taxa é de 50kbit/s para um barramento de 1000 metros. [25] A Figura 7 mostra a relação entre o comprimento da rede (barramento) e a taxa de transmissão de dados.

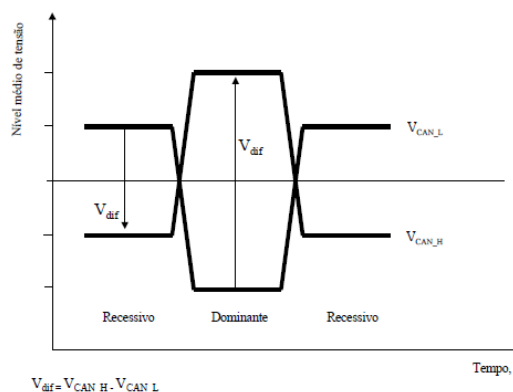


Figura 6 – Níveis de tensão e bits dominantes e recessivos [25]

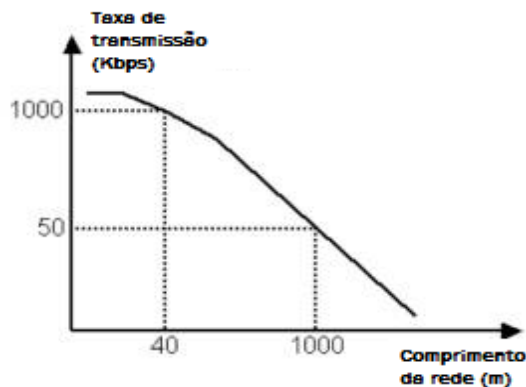


Figura 7 – Relação entre o comprimento da Rede e Velocidade de Transmissão

Em relação à arquitetura e implementação dos barramentos CAN, podem ser utilizados 1,2 ou 4 fios. Nas redes com um fio designa-se por fio de dados ou linha CAN.

As redes com 2 e 4 fios usam como sinais de dados o CAN *High* (CAN_H) e CAN *Low* (CAN_L), sendo que nas redes com 4 fios existem mais dois dedicados à alimentação (VCC) e à massa (GND). Nos barramentos com 2 ou 4 fios, estes não podem ser blindados e devem ser entrançados, de modo que os efeitos causados por interferências eletromagnéticas sejam atenuados. Assim, qualquer interferência é induzida de forma semelhante em ambos os condutores causando perturbação em ambos os sinais no mesmo sentido e com igual intensidade. Desta forma, como os dados são interpretados pelos módulos através da análise da diferença de potencial entre os sinais CAN_L e CAN_H, e esta diferença permanece inalterada devido à atenuação fazendo com que a comunicação não seja prejudicada. [25]

2.5.1.2 Standard CAN e Extended CAN

A primeira versão do protocolo CAN, norma ISO 11519 (*low-speed CAN*), permite apenas comunicações até 125 kbit/s, usando o identificador padrão de 11 bits. A segunda versão, norma ISO 11898 de 1993, manteve os mesmos 11 bits, mas permite comunicações com débitos compreendidos entre os 125 kbit/s e 1 Mbit/s.

Uma atualização, em 1995, introduziu o modo estendido (*extended*) de 29 bits. A norma ISO 11898 de 11 bits é frequentemente citada como *standard CAN* versão 2.0A, enquanto a atualização para 29 bits é citada como *extended CAN* versão 2.0B. A Figura 8 apresenta um exemplo de uma mensagem transmitida no modo *standard CAN* versão 2.0A. [28]



Figura 8 – Mensagem CAN da versão 2.0A (Standard CAN) [28]

A mensagem contém os seguintes campos com os respetivos significados:

SOF – *Start of frame* - É o bit dominante. Marca o início de uma mensagem e é usado para sincronizar todos os nós.

11-Bit Identifier – Identificador padrão de 11 bits. Estabelece a prioridade da mensagem. Quanto menor for o valor binário deste campo maior será a sua prioridade.

RTR – *Remote Transmission Request* - É dominante quando a informação é solicitada de outro ponto. Todos os nós recebem este pedido, mas o identificador determina o nó específico.

IDE – *Identifier Extension* - Bit identificador de extensão dominante. Significa que o identificador CAN standard sem extensão está a ser transmitido.

R0 – Bit reservado.

DLC – *Data Length Code* - Estes 4 bits informam sobre o tamanho de dados que está a ser transmitido (bytes).

DATA – Podem ser transmitidos até 64 bits de dados.

CRC – *Cyclic Redundancy Check* - Contém o *checksum* para deteção de erros. 16 bits (15 bits mais o delimitador).

ACK – *Acknowledgement* - Sempre que um nó recetor subscreve este bit recessivo, indica que a mensagem recebida veio sem erros.

EOF – *End-Of-Frame* - Fim da mensagem.

IFS – *Inter-Frame Space* - É composto por 7 bits que contêm o tempo necessário para o controlador adquirir corretamente os dados recebidos para um *buffer* de receção de mensagem.

Como foi referido anteriormente a atualização da rede CAN bus para a versão 2.0B permitiu a criação de um identificador de 29 bits (Figura 8), sendo agora possível circular na rede cerca de 537 milhões de mensagens. No entanto, os 18 bits adicionais presentes no identificador em relação à rede CAN 2.0A aumentam o tempo de transmissão de cada mensagem, o que pode ser uma desvantagem em determinadas aplicações em tempo-real. A Figura 9 apresenta a estrutura de uma mensagem de CAN versão 2.0B com os respetivos campos. [28]



Figura 9 – Mensagem CAN da versão 2.0B (Extended CAN) [28]

O significado dos campos de bits adicionados e/ou alterados de modo a transitar da **standard CAN** para a **extended CAN** é o seguinte:

SRR – *Substitute Remote Request* - Substitui o bit RTR do CAN 2.0A.

IDE – *Identifier Extension* - Um bit recessivo no IDE indica que há mais bits identificadores a seguir. Os 18 bits da extensão seguem o bit IDE.

R1 – Bit reservado

2.5.1.3 Codificação

Os campos das *frames* de dados e remotas (SOF, arbitragem, controlo, dados e sequência CRC), são codificados pelo método da inserção do bit (*bit stuffing*) de comprimento cinco. Assim, sempre que um emissor detete cinco bits consecutivos, de igual valor, na série de bits a transmitir, ele insere automaticamente um bit complementar na sequência de bits a transmitir. Os restantes campos têm forma fixa e não são sujeitos a este método. As *frames* de erro e sobrecarga possuem forma fixa pelo que também não são codificadas pela regra da inserção do bit. Este processo é representado na Figura 10.

Sequências a transmitir	100000abc	011111abc
Sequências transmitidas	1000001abc	0111110abc
a, b, c ∈ { 0, 1 }		

Figura 10 – Método de inserção de bit

2.5.1.4 Método de Endereçamento

Quando são transmitidos dados utilizando o CAN, não existe endereço fonte ou destino numa mensagem. Os identificadores únicos, das mensagens, servem para caracterizar o conteúdo da mensagem (ex. rpm ou temperatura do motor, no caso do controlo de veículos) sendo da competência de cada nó da rede decidir se a mensagem é ou não válida, para esse nó, realizando para isso um teste de aceitação ao identificador da mesma. Este teste é designado por filtragem, “*Frame acceptance filtering*”, existindo vários dispositivos controladores que permitem diferentes níveis de sofisticação desta filtragem.

Outra característica importante do identificador, para além de definir o conteúdo da mensagem, é a de este estabelecer, também, a prioridade da mensagem. Isto é importante para a atribuição do barramento quando várias estações competem pelo acesso ao barramento.

O processo de transmissão e recepção de mensagens CAN, ilustrado na Figura 11, consiste no seguinte: se a unidade central de processamento, *Central Processing Unit* (CPU), de um dos nós desejar enviar uma mensagem para um ou mais nós da rede, este “passa” os dados a serem transmitidos e o respetivo identificador para o controlador CAN (“Preparar”) desse nó. Isto é tudo o que o CPU necessita realizar para iniciar a transferência de informação. A mensagem é então composta e transmitida pelo controlador CAN. Logo que o controlador consiga acesso ao barramento (“Enviar mensagem”) todos os outros nós na rede CAN tornam-se recetores da mesma (“Receber mensagem”). Cada estação na rede CAN, tendo recebido corretamente uma mensagem, realiza um teste de aceitação para determinar se os dados recebidos são ou não relevantes para essa estação (“Seleção”). Se os dados tiverem significados são processados (“Aceite”), caso contrário são rejeitados (“Não Aceite”). [25]

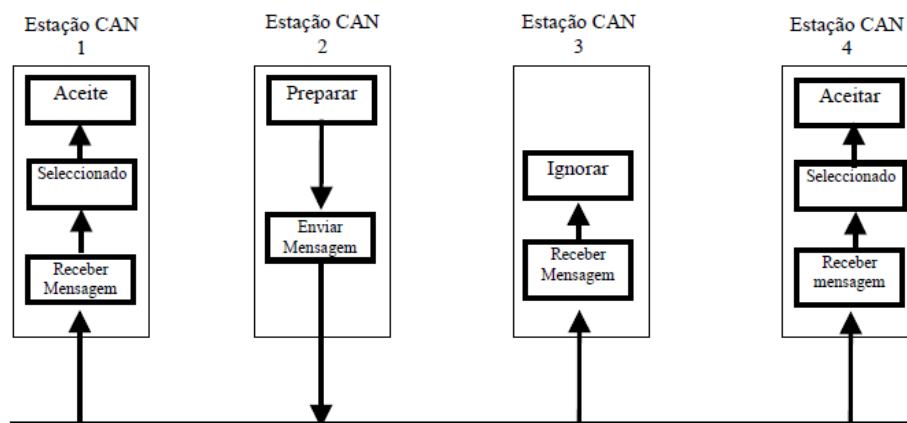


Figura 11 – Método de endereçamento

Desta forma, é atingido um nível superior de flexibilidade do sistema e de configuração devido ao esquema de endereçamento orientado à contenção. É fácil adicionar estações à rede CAN existente sem necessidade de alterações de *hardware* ou *software* dos nós existentes, desde que as novas estações sejam apenas recetoras. Uma vez que o protocolo de transmissão de dados não requer endereços físicos para os componentes individuais, suporta o conceito de eletrónica modular e ao mesmo tempo permite receção múltipla (*multicast, broadcast*) e sincronização de processos distribuídos. Medições, necessárias como informação por vários microcontroladores, podem ser transmitidas pela rede, de tal forma que não é necessário que cada microcontrolador possua o seu sensor, evitando redundância de sensores no caso de redes de aquisição. [25]

2.5.2 OBD, On Board Diagnostic

O OBD (*On-Board Diagnostics*) é um sistema que permite aceder a informações sobre as condições de funcionamento dos veículos.

A quantidade de informação de diagnóstico disponibilizada por este sistema, varia bastante desde as primeiras versões que continham os indicadores *Check Engine* ou MIL – *Malfunction Indicator Light*, no qual acediam no painel de instrumentos do veículo quando um problema era detetado, mas não devolviam qualquer indicação relativamente á origem do problema.

As versões mais recentes já possuem uma porta de comunicações padronizada e códigos de erro DTC (*Diagnostic Trouble Codes*) que permitem a rápida identificação de problemas no veículo e, por sua vez permite a identificação da solução de uma forma mais eficaz.

Na década de noventa, o padrão OBD-I tornou-se obrigatório em alguns países, mesmo não havendo nenhum padrão no que respeita ao método de ligação aos códigos de erro e nem mesmo à identificação do sistema, ainda não tinha sido atribuída a designação OBD. Este cenário acabou por sofrer alterações a partir de janeiro de 1996, quando nos Estados Unidos da América passou a ser obrigatório o sistema OBD-II em todos os veículos automotores. Passado alguns anos, na Europa foi adotado o padrão EOBS e no Japão foi adotada uma variação denominada por JOBD (*Japanese On-Board Diagnostics*). Em relação aos restantes países, a maioria adotou o padrão americano, enquanto os outros ainda não possuem uma legislação específica para estes sistemas. [29]

O protocolo ALDL (*Assembly Line Diagnostic Link*) considerado o precursor do OBD foi criado pela *General Motors* em 1981. A interface variava de modelo para modelo e não havia padronização nem das portas, nem do protocolo digital para comunicação.

O principal objetivo do sistema OBD-I era incentivar os fabricantes da indústria automóvel a projetar sistemas confiáveis que controlassem o volume de emissões de dióxido de carbono, de modo que estes se mantivessem constantes e regulados ao longo do período de vida útil do veículo. Este sistema acabou por não ser aceite pela maioria dos construtores devido há falta

de implementação de normas técnicas entre os fabricantes e da informação específica do sistema.

O padrão OBD-II ou OBD2 constitui uma evolução do OBD-I tanto em funcionalidades como na aplicação de normas. O padrão OBD-II especifica o tipo de tomada utilizada (uma tomada J1962 fêmea com entrada para 16 pinos posicionada no máximo a 0,6 metros do volante da viatura, de modo que esteja sempre ao alcance do motorista), um mapa das funções de cada pino (descritas na norma SAE J196), os sinais elétricos disponíveis, o formato de comunicação, uma lista dos parâmetros do veículo que são monitorizados e por fim as instruções de armazenamento dos dados recolhidos [29]. Na Figura 12 é possível observar o conector J1962 macho e a tomada J1962 fêmea.



Figura 12 - O conector J1962 macho (à esquerda) e a tomada J1962 fêmea (à direita) [73]

É possível utilizar cinco protocolos para sinais na interface OBD2: PWM, VPW, ISO 9141-2, ISO 14230 (semelhantes ao RS-232) e o protocolo de comunicação de redes CAN. A maioria dos veículos utiliza somente um destes protocolos.

O acesso a dados a uma ECU (*Electronic Control Unit*) é uma valiosa fonte de informações relativas a problemas num determinado veículo. A norma SAE J1979 define o método de requisição de dados de diagnóstico e uma lista dos parâmetros disponíveis (rotações por minuto, temperatura do motor, etc.) que estão identificados unicamente através de um PID (*Parameter Identification Number*).

O sistema de requisição e consulta de dados de um PID permite o acesso em tempo real às informações de desempenho do veículo, bem como códigos erros DTC que tenham sido identificados.[29]

Este padrão estabelece ainda uma extensa lista de códigos de diagnósticos de problemas, os códigos de erros indicados pelo sistema (DTC). Como resultado desta uniformização, pode ler-se o sistema de qualquer veículo compatível com o padrão, independentemente da marca.

O padrão EOBD é uma variação do OBD-II exigido na Europa desde 2001 para motores a gasolina e 2003 para motores a gasóleo, baseado no padrão europeu de emissões Euro V e Euro VI, foi desenvolvido para garantir emissões menores que as exigidas pelos seus precedentes Euro III e IV. Alguns fabricantes que utilizam canais adicionais para comunicação com equipamentos denominam o sistema de EOBD2, apesar de ser baseado no OBD-II, tal como no EOBD [29].

2.6 Enquadramento tecnológico

Neste subcapítulo são descritas as tecnologias usadas para o desenvolvimento da solução: *Node.js*, *Angular*, *C#*, *Leaflet* e *PostgreSQL*. Deste modo, é apresentada uma breve descrição de cada tecnologia, bem como as suas características para que se entendam alguns dos conceitos tratados nos próximos capítulos.

2.6.1 Node.js

NodeJS é uma plataforma de desenvolvimento *open-source*, assentado no interpretador *V8 JavaScript Engine*, que foi desenvolvido pela Google em C++.

A grande vantagem do *NodeJS* é de ser uma plataforma baseada em eventos *Input/Output*, o que admite acessos assíncronos.

A plataforma disponibiliza vários módulos core que podem ser utilizados na criação de aplicações. E, ainda usufrui de um grande repositório de pacotes gratuitos que simplificam a construção de aplicações web. [30]

2.6.2 Angular

O Angular (geralmente referido como “Angular 2+” ou “Angular 2”) é uma *framework web open-source* desenvolvida em *TypeScript* – uma linguagem que tem por base o *JavaScript* – pela Google, utilizada para desenvolver o módulo de *front-end* de uma aplicação web na forma de uma *Single Page Application* (SPA). Foi elaborado pela mesma equipa que criou o AngularJS e é considerado uma reformulação da *framework* anterior. O Angular utiliza o conceito de componente como a base da sua arquitetura. Cada componente define uma classe que contém tanto os dados da aplicação como também a sua lógica. [31]

2.6.3 C#

C# é uma linguagem de programação, desenvolvida pela Microsoft para a *framework* .NET. Além disso é considerado uma linguagem de alto nível, que contém bases de diversas linguagens, incluindo o C, C++ e JAVA, usando o melhor de cada uma. Para além destas bases, C# possui características próprias. [32]

As principais características retiradas das outras linguagens são o alto desempenho do C, a estrutura orientada a objetos do C++, o GC e a segurança do JAVA e o desenvolvimento rápido do Visual Basic. [32]

2.6.4 Leaflet

Leaflet é uma biblioteca *JavaScript* de código aberto que foi criada em 2011 por *Vladimir Agafonkin* com o objetivo de contruir *web mapping applications*.

Desde 2011 que a biblioteca *Leaflet* tem sido atualizada constantemente, no qual permite auxiliar os profissionais que trabalham com SIG (Sistemas de Informação Geográficas). A versão 0.6 de junho de 2013, para além de ter sido melhorada a API, de terem sido corrigido erros, melhorado os controlos, também foi introduzida a possibilidade de introduzir o *layer* no formato de *GeoJSON*. Esta atualização é importante, uma vez que é um formato fácil de converter nos programas de SIG, o que torna mais fácil a introdução de um grande volume de dados no script.[33]

O grande concorrente direto do *Leaflet* é o *OpenLayers*, pois ambos são uma biblioteca de *JavaScript*, no qual têm o código aberto e são ferramentas de mapas cliente. A biblioteca *Leaflet* é mais leve que o *OpenLayers* e tem uma base de código mais recente, o que torna este compatível com os últimos recursos do *JavaScript*, HTML 5 e CSS 3. As desvantagens do *Leaflet* em comparação com o *OpenLayers*, é que este possui o recurso *Web Feature Service* e o suporte nativo para projeções. O *Leaflet* também tem concorrentes de código fechado como é o caso da *Google Maps* e *Bing Maps*. [34]

Por fim, o *Leaflet* tem bastantes vantagens para quem pretende desenvolver um software com a biblioteca de *JavaScript*, uma vez que tem uma API bem estruturada, é fácil de usar, é responsiva, utiliza os últimos recursos do *JavaScript*, HTML5 e CSS 3 e para além disso encontra-se bem documentada.

2.6.5 PostgreSQL

O *PostgreSQL* é uma base de dados *open-source* objeto-relacional. Possui mais de 15 anos de desenvolvimento ativo e uma arquitetura que ganhou um elevado relevo devido à sua fiabilidade e a correta integridade dos dados. É compatível com os principais sistemas operativos (*Windows*, *Linux*, *MacOS*, etc.) e é inteiramente aprovada pelo certificado ACID (Atomicidade, Consistência, Isolamento e Durabilidade). A base de dados inclui quase a totalidade dos tipos de dados SQL (*Structured Query Language*), incluindo *integer*, *numeric*, *boolean*, *char*, *varchar*, *date*, *interval* e *timestamp*, para além disto suporta também o armazenamento de grandes objetos binários, tais como, imagem, som ou vídeo. O *PostgreSQL* apresenta ainda interfaces nativas com as linguagens C/C++, JAVA, .Net, Perl, Python, Ruby, PHP, ODBC, entre outros. [35]

O *PostgreSQL* contém processos de recuperação eficientes e eliminação de falhas, como o MVCC (*Multi-Version Concurrency Control*), pontos de restauro, replicação assíncrona e otimizador de *query*. É ainda muito escalável, tanto na quantidade de dados que pode gerir e o número de utilizadores que pode conter. [35]

2.7 Análise de serviços externo existentes

Neste subcapítulo são descritos diversos serviços externos que se encontram desenvolvidos na solução atual Geocar, bem como será feita uma análise de comparação com outros serviços externos existentes no mercado, de modo a escolher as tecnologias/serviços mais adequadas para o desenvolvimento da nova solução, tendo em atenção o custo de utilização e os requisitos definidos.

2.7.1 Serviços de Mapas

Dado que a biblioteca (*Leaflet*) acima referida funciona como interligação entre a aplicação web e os serviços de mapas, torna-se necessário a utilização de um serviço de mapas.

Um dos fatores a ser considerado na gestão de uma frota são os níveis de congestionamento das rotas que os condutores seguem. Uma previsão do fluxo de tráfego pode ser determinante para o cumprimento de horários e a consequente satisfação dos clientes. Assim, salientam-se os prestadores de serviços de mapas que contenham este recurso, que apenas existe na forma de imagens de mapas e, portanto, apenas empresas que forneçam serviços de mapas o possuem.

De seguida é descrito e analisado diversos serviços de mapas e posteriormente será feita uma comparação entre eles de modo a verificar o serviço adequado para o desenvolvimento da nova solução.

2.7.1.1 OpenStreetMaps

OpenStreetMaps é um projeto que funciona através de um mapeamento colaborativo, na elaboração de um mapa mundial livre e editável. Apresenta a vantagem de ser *open-source*, devido ao facto de ser gerida por uma fundação designada por *OpenStreetMap Foundation*, no qual é responsável pela aquisição e manutenção dos servidores que alojam o *OpenStreetMaps*, assim como pela angariação de fundos para suportar essa infraestrutura. A fundação dedica-se também a incentivar o crescimento, o desenvolvimento e a distribuição dos dados geo-espaciais para que qualquer pessoa os possa consultar e usar. [36]

2.7.1.2 MapQuest

A *MapQuest* é uma empresa especializada na criação de soluções geoespaciais. Dispõe de diversas *APIs e Software Development Kits (SDK)* que estabelecem o acesso a uma variedade de mapas e também a funções fundamentais, tais como a tradução de endereços, o cálculo de rotas, a representação do fluxo de tráfego, entre outras.

As APIs estão disponíveis através de *JavaScript*, mas foram criados plugins de integração com a biblioteca *Leaflet*, sendo possível utilizar os recursos da *MapQuest*. Estes plugins convertem os objetos *MapQuest* em extensões de objetos do tipo camada *Leaflet*.

O uso destas APIs necessita de um registo para a obtenção de uma chave API. O registo poderá ser feito de forma gratuita, permitindo o acesso a todos os recursos disponíveis apenas limitado por um número máximo de 15000 transações por dia. No caso de o registo ser associado a um plano pago, o número máximo de transações é aumentado, por exemplo, para 30000 transações por dia, o valor seria 96,70€ por mês.

A *MapQuest* dispõe também de imagens de mapas com informação relativa ao fluxo de tráfego, no entanto, o foco desta empresa é nos Estados Unidos, sendo que por essa razão, este recurso se encontra limitado ou inexistente para outras localizações [37].

2.7.1.3 Google Maps

O *Google Maps* é um dos serviços de mapas mais utilizados no mundo, tanto na forma de mapa interativo, como na forma de API. Esta dispõe de diversas formas de acesso: Android, iOS, *web* (mapas incorporados ou através de *JavaScript*) e serviços web. [38]

É importante referir que quando a API é acedida através de um serviço web, não é possível obter uma resposta de imagens de mapas, assim, a importação de um mapa deverá ser feita previamente através de um mapa incorporado ou através de *JavaScript*.

As APIs apresentam uma variedade de serviços: localização de pontos de interesse, cálculo de rotas, valores de elevação, fuso horário e até mesmo o valor dos limites de velocidade. Para o seu funcionamento é necessário o registo do utilizador e da aplicação de forma a gerar a chave da API.

A sua limitação está no número de acessos possíveis através da versão grátis do serviço (25000 por dia), sendo que a extensão desse limite está dependente da compra de um plano.

Este serviço fornece também a opção de representação do fluxo de tráfego, apenas possível através da API *JavaScript*. [39] É importante salientar que é expressamente proibido utilizar as APIs sem o uso de um mapa do Google.

2.7.1.4 HERE Maps

HERE é uma empresa especializada na construção de diversas soluções para topografia e mapeamento. Oferece a possibilidade de utilizar as suas APIs e SDK, para além dos seus produtos específicos.

Estas APIs podem ser acedidas através de JavaScript ou através de pedidos REST. Ambos os métodos poderão ser utilizados para a grande maioria das APIs, nomeadamente: listagem de pontos de interesse, pesquisa de endereços, representação do fluxo de tráfego, *Geofencing* (serviço baseado em GPS que define limites geográficos virtuais), entre outras.

No caso das APIs JavaScript, note-se que estas terão de iniciar em primeira instância um mapa interativo, a que posteriormente são adicionadas camadas com os recursos das restantes APIs.

No caso dos pedidos REST a resposta será feita em forma de objeto JSON, não sendo necessário utilizar nenhum mapa para obter esta informação. No entanto, a HERE oferece uma solução REST para o carregamento de mapas que poderá ser utilizada numa biblioteca como *Leaflet* ou *OpenLayers*. [40]

2.7.2 Geocodificação e Geocodificação reversa

A geocodificação refere-se à conversão de um endereço físico, como um endereço, em coordenadas geográficas (latitude e longitude), para mostrar um determinado local num mapa. A geocodificação reversa (*reverse geocoder*) faz o cálculo oposto, isto é, encontra um endereço quando recebe um conjunto de coordenadas geográficas. Um *geocoder*, ou sistema de geocodificação, é um software ou serviço web que executa a geocodificação. Um *geocoder* pode ser usado para vários objetivos, tais como, a obtenção de direções, cartografia e identificação geográfica.

De seguida é descrito e analisado diversas APIs e posteriormente será feita uma comparação entre elas, de modo a verificar a necessidade de utilizar no desenvolvimento da “Geocar Lite”.

2.7.2.1 HERE Geocoding and Search API

A *HERE Geocoding and Search API* é uma API que fornece as capacidades de pesquisa e geocodificação da Here para criar aplicações com capacidade de localização que permitem aos utilizadores pesquisar por pontos de interesse, encaminhar e reverter endereços geocodificados e geo-posições a partir do mapa da *Here*, e também beneficiar de atualizações de dados mais rápidos.[41] Para a utilização desta API é necessário um registo que poderá ser feito de forma gratuita, permitindo o acesso a todos os recursos disponíveis apenas limitado por um número máximo de 30000 transações por mês. No caso de o registo ser associado a um plano pago, o número máximo de transações é aumentado, por exemplo, entre 30000 e 5000000 transações por mês, o valor seria 0.60€ por cada 1000 transações efetuadas. [42]

2.7.2.2 Google Maps Geocoding API

A Google Maps Geocoding API que utiliza um geocodificador através de um serviço HTTP. A geocodificação consiste em transformar endereços em coordenadas, bem como também é possível realizar a operação inversa, transformando coordenadas em endereços. Este serviço é utilizado normalmente para geocodificar endereços previamente conhecidos para que torne possível localizá-los num mapa. Esta operação é morosa e acarreta custos, e por isso é necessário, sempre que possível, o programador persistir numa base de dados ou em *cache* os geocódigos de endereços estáticos. Isto faz com que as aplicações sejam mais rápidas, uma vez que melhora o tempo de resposta da aplicação, pois as consultas seriam realizadas com a API, apenas quando necessário. Este serviço da Google está limitado a 2500 consultas por dia e para o pacote de serviços empresarial da Google pode realizar até 100000 consultas por dia que corresponde a um custo de 381,12€ por mês. [43]

2.7.2.3 Bing Search API

A *Locations* API do Bing, é uma API direcionada para a informação de localizações geográficas através do uso de pedidos REST. Esta API poderá receber como parâmetros a morada, um ponto geográfico ou um nome a procurar.[44]

Relativamente a custos, este serviço é gratuito até 1000 transações por mês e caso ultrapasse esse número de transações passará a ser cobrado 6,57€ até 25000 transações.[45]

2.7.3 Serviço de comunicação

Os serviços de comunicação consistem em serviços que têm um papel fundamental na solução Geocar, nomeadamente a utilização de plataformas de envio de comunicações de SMS que auxiliam no sistema de alertas.

As plataformas utilizadas para este efeito são a **Intouchbiz** [4] e a **Ez4u** [3],

A plataforma **Ez4u** desenvolvida pela empresa Ez4u oferece serviço de envio de mensagens personalizadas, calendarização, identificação com a marca da empresa no remetente, contém mecanismos de receção com processamento automático e também permite a integração com sistemas externos através de API *RESTful/JSON*. O cliente pode adquirir através de pré-pagamento, fazendo um carregamento na conta do valor que desejar.

Em relação à plataforma **Intouchbiz** desenvolvida pela empresa Intouchbiz tem um serviço de envio de mensagens personalizadas, no qual uma determinada empresa ou até mesmo um particular pode adquirir através de um custo associado pelo número de mensagens enviadas.

Uma vez que estas plataformas de envio de SMS requerem custos de utilização, as mesmas serão abdicadas no desenvolvimento da nova solução.

2.7.4 Comparação dos serviços descritos

De seguida é feita uma análise comparativa entre os serviços descritos anteriormente, no qual foi feito um estudo das funcionalidades e características que os mesmos oferecem, de modo a verificar qual os serviços adequados para o desenvolvimento da *Geocar Lite*.

Tabela 1 – Comparação entre serviços de mapas

Tecnologias	OpenStreetMaps	MapQuest	GoogleMaps	HERE Maps
Características				
Integração GPS	Sim	Sim	Sim	Sim
API disponível	Sim	Sim	Sim	Sim, pelo desenvolvedor
Resultados de pesquisa dinâmica baseado em <i>dragging on the map</i>	Não	Não	Sim	Sim
Web Browser Supports	IE7+, Firefox 3.5+, Safari 4+, Google Chrome 4+	IE10 e IE11, Firefox 3.5+, Safari 9+, Google Chrome 4+	IE7+, Firefox 2.0.0.8+, Safari 3+, Mozilla 1.7+, Opera 8.0.2+, Google Chrome 1+	IE7+, Firefox 2.0.0.8+, Safari 3+, Mozilla 1.7+, Opera 8.0.2+, Google Chrome 1+
Reverse Direction Service	Sim	Sim	Sim	Sim
Geocoding Web Service	Não	Sim	Sim	Sim
Points of interest (POI)	Sim	Sim	Sim	Sim
Waypoints	Sim	Sim	Sim	Sim
Direction Web Service	Open Route Service API	MapQuest Directions API	Google Direction API	Here Routing API
Tipos de Mapas	Mapa, Terreno, satélite (exclusivo para edição)	Mapa, satélite	Mapa, Terreno, Satélite e Ruas	Mapa, satélite, terreno, 3D com plugin, 3D sem plugin para browsers compatíveis, modo noturno
Custo	<i>Open-source</i>	<i>Open-source</i> , Pago até 856,6€ por mês	<i>Open-source</i> , Pago até 800€ por mês	<i>Open-source</i> , Pago até 428€ por mês

Como se pode visualizar na tabela acima (Tabela 1), está descrito uma pequena síntese sobre cada serviço de mapa já referido. Existem vários aspetos a serem analisados com algum detalhe relativamente a algumas características que se encontram na tabela, nomeadamente o facto de o serviço de mapa *OpenStreetMaps* ser *open-source* e de livre utilização, enquanto nos restantes serviços também são *open-source* para uma determinada quantidade de *requests* à API, mas a partir de um dado valor de *requests* é cobrado um custo para a sua utilização. Uma vez que para o desenvolvimento da nova solução é necessário obter os dados vindos do dispositivo de localização é necessário analisar se o serviço de mapas tem essa possibilidade de integração de GPS, e verifica-se que todos os serviços de mapas analisados são possíveis de integração do mesmo. É importante referir que para o desenvolvimento da nova solução não será necessário incluir funcionalidades que necessitem de serviços específicos, tais como, *GeoCoding Web Service* [41], uma vez que para a sua integração é necessário um determinado custo. Após esta análise dos serviços de mapas descritos, selecionou-se o serviço de mapas *OpenStreetMaps*, pela sua simplicidade e também pelo facto de não implicar custos de utilização e também a possibilidade de integração no *Leaflet*.

2.8 Soluções existentes de gestão de frota

Atualmente os sistemas de informação para gestão de frotas automóveis são essenciais para empresas com frotas, uma vez que permitem uma melhor gestão de recursos e consequentemente a diminuição de custos operacionais, bem como o aumento da satisfação do cliente final.

Nesta secção é apresentada uma análise das soluções existentes no mercado no que diz respeito a plataformas que permitem gerir uma determinada frota de veículos, bem como saber a sua localização e por sua vez auxiliar na manutenção da mesma.

De seguida será feita uma descrição pormenorizada de diversos aspetos dos diferentes softwares existentes e também uma análise comparativa dos mesmos.

2.8.1 Frotcom – Intelligence Fleets

A Frotcom Lusitana é uma distribuidora do sistema inteligente de localização de veículos e gestão de frotas – Frotcom, em Portugal. Tem como objetivo auxiliar as empresas na gestão das suas frotas de uma forma simples e eficaz, resultando num aumento evidente da produtividade, das margens e da qualidade de serviço. [46]

A Frotcom oferece um pacote completo de serviços de Telemática para frotas incluindo a instalação, formação e assistência técnica, para todos os tipos de frotas – automóveis, carrinhas e camiões.

O software da Frotcom contém diversas funcionalidades que se encontram divididas em três fatores importantes: **localização GPS de veículo ativos**, **gestão de condutores** e **segurança**. [47]

Em relação a **localização GPS de veículos ativos** contém as seguintes funcionalidades:

- Possibilidade de visualizar e configurar relatórios de gestão de frotas (horas de condução e trabalho de tacógrafo, temperaturas inadequadas, gestão de combustível, entre outros);
- Localização GPS de veículos e monitorização de sensores;
- Visualização do veículo mais próximo;
- Planeamento e monitorização de rotas.

Relativamente a **gestão de condutores** contém as seguintes funcionalidades:

- Possibilidade de identificação automática de condutores;
- Análise do estilo de condução, nomeadamente o controlo de acelerações e travagem, o funcionamento ao ralenti, e em alguns casos, o excesso de aceleração do motor (RPM) e utilização do controlo de velocidade de cruzeiro;
- Monitorização dos tempos de condução, no qual permite saber os tempos de condução de cada condutor, em qualquer etapa do percurso;
- Comunicação por texto com os condutores.

Por fim, em relação à **segurança**, esta contém as seguintes funcionalidades:

- Alarmes associados a entrada em áreas e saída de áreas, ao excesso de velocidade e funcionamento ao ralenti, entre outros;
- Gestão de combustível;
- Imobilização remota do veículo;
- Controlo de acesso pode proteger o acesso à carga de um determinado veículo.

2.8.2 Webfleet Solutions

A *Webfleet Solutions*, conhecida como *Tom Tom Telematics*, é uma empresa internacional de gestão de frotas, localização GPS de viaturas em tempo real e automóvel interligado. A *Webfleet Solutions* conta com mais de 50000 clientes pelo mundo em 100 países. O seu software gestão de frotas tem como objetivo otimizar o desempenho da frota de um cliente, bem como fornecer informações de modo ajudar a reduzir custos operacionais, poupar tempo e também dar prioridade à segurança.[48]

De seguida é apresentada as funcionalidades do seu software de gestão de frotas [49]:

- **Seguimento de veículos:** Permite saber a posição dos veículos, bem como registar o tempo, a quilometragem e a localização de cada viatura em tempo real.

- **Otimização de frotas:** Melhora a eficiência da frota. Tem como objetivo tomar decisões mais inteligentes com relatórios detalhados e informações em tempo real acerca dos veículos a partir do painel.
- **Gestão de equipas de trabalho:** Permite estabelecer uma ligação entre a solução WEBFLEET e um *Driver Terminal PRO*, no qual permite obter acesso às informações acerca das tarefas e dos condutores. Sistema que serve de apoio à comunicação com uma determinada equipa e ajuda-a a simplificar o fluxo de trabalho.
- **Condução ecológica e segura:** Tem como objetivo reduzir os custos, diminuir as emissões de carbono, melhorar a segurança e melhorar o estilo de condução.
- **Integração na empresa:** Possibilidade de integrar facilmente o software da WEBFLEET num software particular ou aplicações empresariais atuais.
- **Fuga ao trânsito e navegação profissional:** Permite uma navegação profissional que evita o trânsito, de modo a não afetar a produtividade do colaborador, no qual tem um grande impacto no seu serviço e planeamento.

2.8.3 Verizon Connect

A *Verizon Connect* é uma empresa portuguesa especializada na produção e comercialização de soluções de localização via GPS para particulares e empresas [50]. Apresenta-se como sendo fornecedora de informação relevante e conectividade, dando aos seus clientes a possibilidade de estarem mais informados sobre a localização do veículo e do motorista, a eficiência da condução, a segurança, a produtividade e a conformidade com os parâmetros definidos pela mesma. [51]

O software gestão de frotas da *Verizon Connect* contém diversas funcionalidades, nomeadamente: [52]

- **Mapa com todos os veículos:** Mapa online que mostra a localização, o estado e a velocidade atual do veículo monitorizado.
- **Monitorização de veículos:** Permite a visualização da localização GPS e a velocidade dos veículos, no qual os dados são atualizados a cada 90 segundos.
- **Reprodução de rota:** Avaliar os movimentos e o tempo em ralenti de um determinado veículo ou reproduzir a rota percorrida.
- **Limitações geográficas:** Permite receber alertas sobre atividades em horários e locais específicos.
- **Dados de tráfego do Google Maps:** Consultar as informações de tráfego do Google como parte de visualização do mapa.
- **Vista de rua e de satélite do Google:** Possibilidade de aceder à vista de rua e de satélite do Google no modo mapa.
- **Aplicação Manager:** Aplicação móvel que permite aos gestores localizar e gerir motoristas.

- **Relatórios:** Possibilidade de receber dados da frota, incluindo quilómetros diários, utilização de combustível, eventos de excesso de velocidade e horas de chegada.
- **Agendamento e atribuição de serviços:** Permite agendar serviços e fazer a gestão do horário dos técnicos num calendário simples. Para além disso, é possível agendar e atribuir tarefas diretamente aos técnicos.
- **Digital Tachograph:** A utilização de tacógrafos digitais permite descarregar remotamente os dados do tacógrafo ajudando uma determinada empresa a estar em conformidade com a legislação em vigor, reduzir o tempo de inatividade dos veículos da frota e otimizar as atividades diárias.
- **Imobilização:** Permite imobilizar as viaturas remotamente, com o objetivo de ajudar na recuperação das mesmas e reduzir utilizações não autorizadas. A imobilização é feita através de inibição de arranque o que contribui para a segurança dos motoristas, uma vez que apenas se efetiva após a ignição ser desligada.
- **Prova de entrega:** Utilização da localização GPS do veículo para confirmar a entrega ou chegada ao local do serviço.
- **Alertas:** Permite receber alertas de SMS ou de email para eventos importantes, tais como, excesso de velocidade ou ralenti.
- **Atribuição de condutores a veículos:** Utilização de cartões RFID (identificação do condutor) para comunicar a atividade dos veículos e o desempenho dos condutores.
- **Dispositivos de monitorização GPS ligados à ficha OBD:** Ativação fácil para veículos fabricados a partir de 1996 ou mais recentes que tenham uma porta OBD-II.
- **Dispositivos de monitorização GPS instalados na viatura.**
- **Aplicação móvel do condutor:** Uma aplicação móvel criada com o intuito de ajudar a manter a produtividade dos condutores, alertá-los sobre eventos críticos.
- **Agendamentos de manutenção de veículos:** Possibilidade de configurar lembretes de serviços de veículos com base em intervalos de calendário ou de quilometragem.
- **Dashboards:** Visualização de *dashboards* de gestão de frotas, no qual incluem custos de mão de obra, consumo de combustível, ralenti em excesso, métricas de segurança e utilização de ativos.
- **Grupos de veículos e condutores:** Permite organizar a frota de acordo com a estrutura da empresa para facilitar a sua gestão.
- **Limites de velocidade estipulados:** Aceder aos limites de velocidade estipulados para as rotas que os condutores fazem
- **Permissões do condutor.**

2.8.4 Comparação das plataformas analisadas

As potencialidades oferecidas por um sistema de gestão de frotas não são secundárias. No passado, a principal oferta associada a estas soluções baseava-se exclusivamente na capacidade de localização, no entanto, a realidade da gestão exige atualmente, informações adicionais, tanto ao nível das tarefas realizadas, dos percursos, dos condutores, como também a informações sobre os veículos (marca, óleo, matrícula, data de inspeção, seguro, avarias, etc.) que constituem a frota.

Após uma análise algumas alternativas existentes no mercado verificaram-se que estes tipos de sistemas têm disponíveis funcionalidades/características consideradas indispensáveis, tais como:

1. Monitorização dos veículos por GPS;
2. Visualização de percursos;
3. Possibilidade de extração de relatórios de gestão de frotas;
4. Identificação automática dos condutores;
5. Análise do estilo de condução;
6. Planeamento e monitorização de rotas;
7. Detecção de possíveis situações de alarme/alertas ou avaria;
8. Gestão de combustível;
9. Utilização de aplicação móvel e web

Com os desenvolvimentos tecnológicos registados foi possível o *upgrade* nestas soluções, permitindo deste modo a produção de sistemas de gestão de frotas mais flexíveis e à medida das necessidades sentidas pelos clientes.

O leque de soluções apresentadas e disponíveis no mercado é muito variado, no entanto, é possível encontrar desde soluções simples e intuitivas até soluções mais complexas, dependendo da finalidade para a qual se pretende e do tipo de utilizador.

O leque de soluções apresentadas e disponíveis no mercado é muito variado, no entanto, é possível encontrar desde soluções simples e intuitivas até soluções mais complexas, dependendo da finalidade para a qual se pretende e do tipo de utilizador.

3 Análise de Valor

Neste capítulo é feita a análise de valor da solução a ser adotada para a resolução do problema. Inicialmente é feita uma identificação e análise das oportunidades existentes, de acordo com a conjectura atual. De seguida, de acordo com a análise de oportunidades realizada é feita uma seleção de ideias, de modo a escolher a melhor que se enquadra com o projeto a ser realizado.

Para além disso é também definido o valor da solução a implementar, através da utilização da proposta de valor e do modelo de negócio CANVAS. Por fim é utilizado o *Quality Function Deployment* (QDF) para transformar todas as necessidades existentes em valor real.

3.1 Processo de Inovação

Segundo *Koen*, o processo de inovação pode ser dividido em 3 fases distintas, que se encontram representadas na Figura 13:

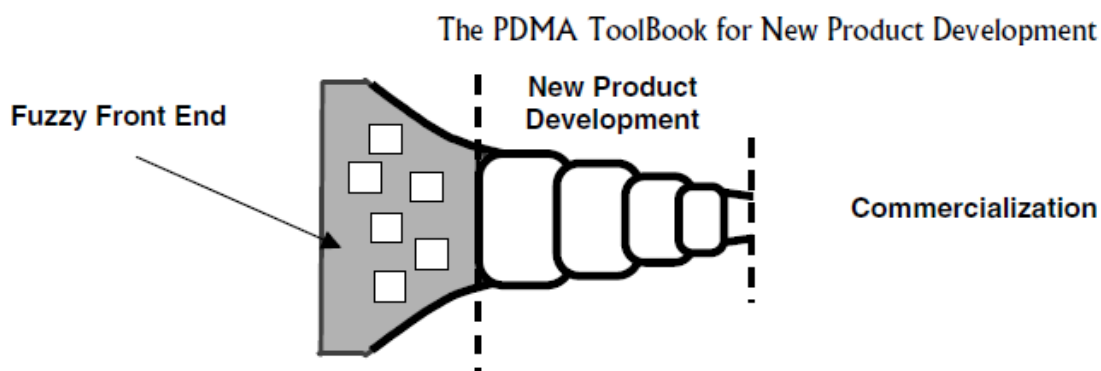


Figura 13 – Divisão do Processo de Inovação [53]

- **Fuzzy Front End (FFE):** A primeira fase, o **Fuzzy Front End**, representa um estado não estruturado, caótico e experimental, onde existem variáveis incertas de comercialização e financiamento e onde se procura a melhoria e fortificação de conceitos de negócio. [53]
- **New Product Development (NPD):** A segunda fase, o **New Product Development**, encontra-se com planos concretos e orientados a os objetos específicos, onde todas as datas são definidas e existem equipas focadas no desenvolvimento e teste de produtos. [53]
- **Comercialização:** A terceira e última fase, a **Comercialização**, representa-se como o resultado das duas fases anteriores, sendo o objetivo final do processo inovativo a comercialização do produto desenvolvido. [53]

3.2 New Concept Development

O *New Concept Development* é um modelo não linear de inovação, criado por *Peter Koen* e a sua equipa, com o objetivo de criar conhecimento e terminologia comum no FFE. Este modelo baseia-se em 3 partes fundamentais, que se encontram ilustradas na Figura 14:

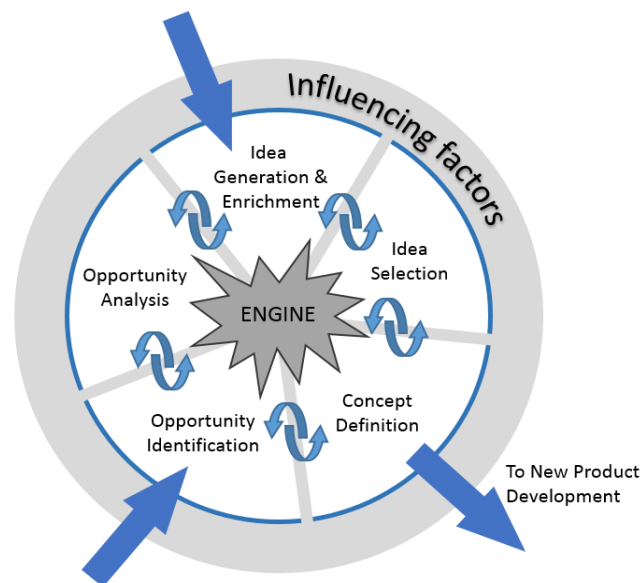


Figura 14 – Conceito de NCD e as suas partes [74]

- **O Motor:** O motor encontra-se relacionado à liderança, o plano estratégico e a cultura existente numa empresa e que vão influenciar diretamente os 5 elementos controláveis. [53]
- **O Ambiente:** O ambiente representa fatores externos não controláveis que afetam todo o processo de inovação, como leis, política, economia e competidores.
- **Os 5 Elementos Controláveis:** Os 5 elementos controláveis representam 5 atividades controladas do FFE, compostas pela identificação de oportunidades, análise de oportunidades, geração de ideias, seleção de ideias e definição de conceitos.

3.2.1 Identificação da Oportunidade

Esta é a fase onde se identificam as diferentes oportunidades existentes, sendo nelas que são alocados os recursos. Estas oportunidades podem ser enquadradas em diferentes contextos, como respostas às concorrências através de processos mais eficazes, melhoria e alterações a serviços já existentes ou até a criação de raiz de novos produtos e processos. [53]

O problema identificado na secção 1.2 representa uma oportunidade de desenvolver uma nova solução que seja menos dispendiosa do que a solução atual, a Geocar, e que contenha o mesmo desempenho.

Contudo, as oportunidades de inovação podem resultar tanto em algo positivo para a empresa como negativo. De forma a analisar que tipo de resultado irá proporcionar esta oportunidade, será feita uma análise da oportunidade.

3.2.2 Análise de Oportunidade

Com a análise de oportunidade pretende-se traduzir a oportunidade encontrada numa oportunidade de negócio para a empresa. Para fazer essa tradução irá ser desenvolvida uma análise SWOT.

A análise SWOT é uma ferramenta que apresenta, de uma forma esquemática, as **Strengths** (Forças), **Weaknesses** (Fraquezas), **Opportunities** (Oportunidades) e **Threats** (Ameaças), permitindo assim fazer uma avaliação estratégica de um produto ou de uma empresa. As Forças e as Fraquezas apresentam os pontos positivos e pontos negativos relativos à empresa perante a concorrência. As Oportunidades e Ameaças possibilitam uma análise externa ao meio envolvente e como isso influencia a gestão de uma empresa. [54]

Tabela 2 – Análise SWOT da oportunidade

Forças	Fraquezas
Utilização de tecnologias <i>open-source</i> ;	Quantidade de funcionalidades reduzidas;
Plataforma intuitiva e funcional;	
Custo de desenvolvimento reduzido;	
Aplicação móvel e web.	
Oportunidades	Ameaças
Facilidade e manutenção do sistema;	Concorrentes existentes;
Aumento de utilização da plataforma.	Alterações legislativas, nomeadamente de proteção de dados.

Com a análise da Tabela 2 pretende-se avaliar a viabilidade da oportunidade de desenvolver uma nova solução para gestão de frotas que seja menos dispendiosa do que a solução Geocar.

No que se refere às forças, foram encontradas quatro, com a necessidade de serem referidas. A primeira, é o facto de ser desenvolvido uma plataforma com tecnologias *open-source*, uma vez que vai ser utilizado um serviço de mapas gratuito e de livre utilização. Outra força diz respeito ao facto de a plataforma ser intuitiva e funcional, que ajuda o cliente a ter possibilidade de gerir os seus veículos de uma forma simples e eficaz. Outra força importante a ser referida é o facto de ser possível o cliente ter o controlo sobre a sua frota de veículos a partir de uma aplicação web e de uma aplicação *mobile*. Por fim a última força identificada foi os custos de desenvolvimento reduzidos, no qual para a criação da nova solução será abdicado os serviços externos pagos, bem como a redução do fluxo de comunicações entre os equipamentos móveis e a respetiva *Cloud*.

Relativamente às fraquezas, o único aspeto é em relação à quantidade de funcionalidades que a plataforma terá. Isto deve-se ao facto de algumas funcionalidades serem necessárias a utilização de APIs que requerem custos e por essa razão não serão desenvolvidas fazendo com que apenas contenha as funcionalidades essenciais.

Em relação as oportunidades, foram identificadas duas. A primeira oportunidade é a facilidade e manutenção do sistema, no qual o que se pretende é a redução do número de técnicos de instalação, bem como um sistema de fácil manutenção, nomeadamente que seja capaz de controlar o histórico dos dados de localização armazenados na base de dados (dados guardados dos últimos 3 meses). A última oportunidade identificada, é a partir do desenvolvimento de uma nova solução, esperar-se que a utilização desta plataforma aumente consideravelmente por se tratar de uma plataforma mais simples e funcional.

Relativamente às ameaças identificadas, existe uma relação direta com a concorrência. O aparecimento de novos concorrentes que tenham plataformas de gestão de frotas com melhor

desempenho e com mais funcionalidades podem ser uma ameaça. É importante perceber as ameaças relacionadas com a concorrência de modo a melhorar continuamente o produto para que os utilizadores não sintam a necessidade de optar por outras soluções. Por último, é importante também manter uma atenção constante na vertente legislativa, nomeadamente em relação à proteção de dados. Alterações legislativas podem tornar ilegais algumas práticas ou funcionalidades do produto e, por esse motivo, é fulcral que exista uma atenção contínua de modo a conseguir alterar essas práticas ou funcionalidades.

Tendo em conta todos os pontos apresentados da análise SWOT é possível concluir que a oportunidade é viável e irá trazer benefícios para a empresa. Alguns deles são, a nível externo, oferece a melhoria do desempenho e da redução dos custos da solução da GisGeo e, a nível interno, tem as funcionalidades e características que marcam a diferença no mercado. De seguida irá ser feita a geração de ideias para o desenvolvimento da solução.

3.2.3 Geração de Ideias

O terceiro passo do *New Concept Development* é a geração de ideias, uma fase em que são feitas sugestões de possíveis soluções para que sejam escolhidas as melhores entre elas. Resultando um conjunto de ideias que poderão ser executadas.

A ideia do projeto proposto pela organização consiste na redução de custos da solução atual Geocar, e para isso é necessário desenvolver uma nova solução que será utilizada no mundo real, no qual poderá ajudar a GisGeo a ter o mesmo controlo sobre as suas frotas utilizando os menos recursos possíveis e com melhor desempenho.

Para o caso em questão foi feita uma pesquisa de possíveis soluções que cumprissem os critérios da oportunidade, da análise SWOT. Após uma discussão interna conclui-se que existem três hipóteses de solução:

- Reestruturar a solução existente reduzindo funcionalidades;
- Criação de uma nova solução utilizando tecnologias *open-source* e com funcionalidades reduzidas;
- Reestruturar a solução existente com tecnologias *open-source*;

Com as hipóteses de soluções enumeradas, selecionadas e validadas com base nos critérios de resolução do problema, o próximo passo é a seleção das ideias.

3.2.4 Seleção de Ideias

Com a identificação da oportunidade, a análise de oportunidade e a geração de ideias concluídas e validadas, segue-se a seleção de ideias. Com esta análise pretende-se escolher quais as ideias que irão trazer mais valor para o consumidor e o negócio. O modelo utilizado

para fazer a seleção das ideias foi o *Analytic Hierarchy Process* (AHP) e construiu-se a árvore hierárquica de decisão de acordo com a Figura 15.

Divisão Hierárquica

O *Analytic Hierarchy Process* (AHP) foi criado em 1970, por *Thomas L. Saaty*. É um modelo utilizado para a tomada de decisões que apresentam muitas variáveis e cenários complexos. Para utilizar o modelo é necessário definir um objetivo, critérios e soluções.[55]

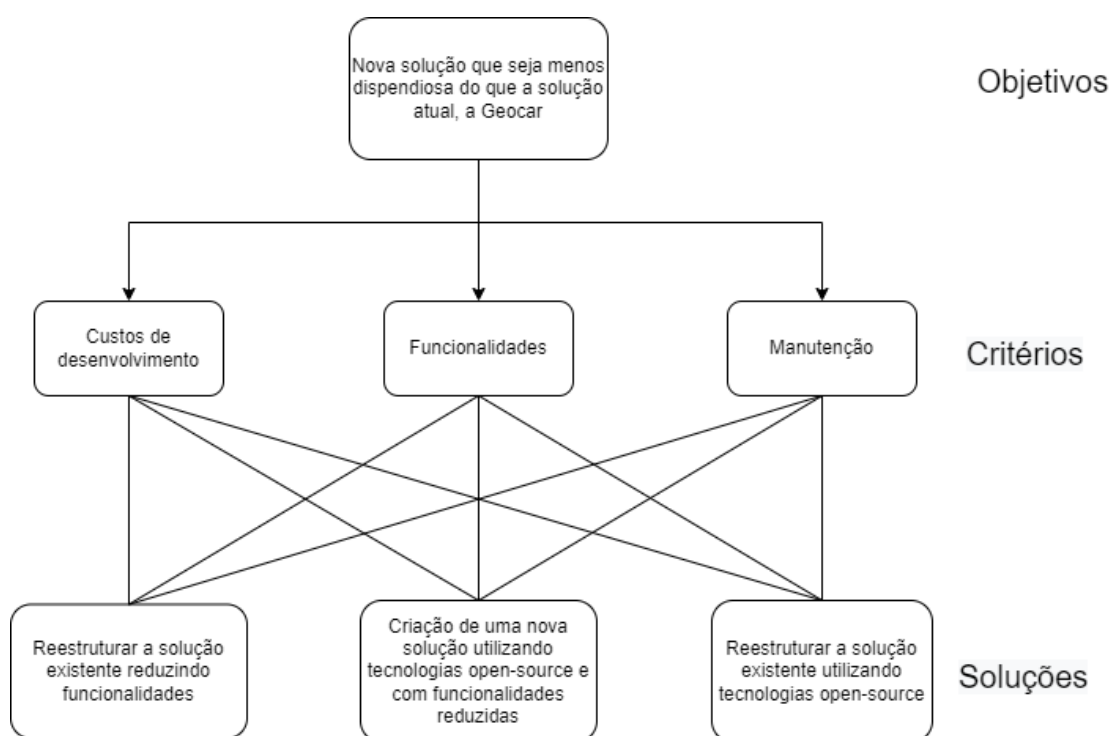


Figura 15 – Esquematização Hierárquica de AHP

No âmbito do problema em questão, o objetivo definido é desenvolver uma nova solução de gestão de frotas que não implique muitos custos em relação à solução atual. Os **critérios** são os custos de desenvolvimento, as funcionalidades e a manutenção da plataforma. Relativamente aos custos de desenvolvimento considera-se um fator importante, uma vez que pretende-se criar uma solução que seja menos dispendiosa do que a solução atual, a Geocar, e para isso é necessário abdicar de serviços externos (*Here, Google, Ez4u e IntouchBiz*) e reduzir o fluxo de comunicações entre os equipamentos móveis e a respetiva *Cloud*. Em relação às funcionalidades é um aspeto a ter em conta, pois verifica-se se os requisitos funcionais enunciados foram cumpridos. Por fim, em relação à manutenção, a solução será feita de modo que seja fácil de manter a plataforma usável.

Definição de Prioridades dos Critérios

O primeiro passo para a definição de prioridades é a comparação dos critérios. A comparação entre dois elementos, pelo AHP, pode ser feita de múltiplas formas, contudo a mais comum é pela escala de importância (escala de *Saaty*). Os valores, da escala, estão compreendidos entre 1 e 9, e representam a importância de um elemento relativamente a outro.

Nível de importância	Definição	Explicação
1	Igual importância	As duas atividades contribuem igualmente para o objetivo
3	Fraca importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito forte importância	Uma atividade é muito fortemente favorecida em relação a outra
9	Importância absoluta	A evidência favorece uma atividade em relação a outra com o mais alto grau de certeza
2,4,6,8	Valores intermediários	Quando se procura uma condição de compromisso entre duas definições

Figura 16 – Escala de Saaty [75]

Como é possível observar na Figura 16, o valor mínimo, 1 representa igual importância e, o valor máximo, 9 equivale a importância absoluta. Tendo em conta a escala e os critérios obtiveram-se os seguintes valores:

Tabela 3 – Comparação entre os critérios

			Importância
Custo de desenvolvimento	relativamente	Manutenção	5
Funcionalidades		Manutenção	3
Custo de desenvolvimento		Funcionalidades	1

Pela tabela de comparação é possível notar que o custo de desenvolvimento comparado com a manutenção tem muito forte importância. O custo de desenvolvimento confrontado com as funcionalidades tem a mesma importância. E por fim as funcionalidades comparadas com a manutenção tem um valor de fraca importância.

Com a tabela de comparação dos critérios definida é necessário colocar os valores numa matriz (matriz de comparação).

Considerando:

- A – Custo de desenvolvimento
- B – Funcionalidades
- C – Manutenção

Tabela 4 – Matriz de comparação dos critérios

	<i>A</i>	<i>B</i>	<i>C</i>
<i>A</i>	1	1	5
<i>B</i>	1	1	3
<i>C</i>	1/5	1/3	1
<i>Total</i>	2,2	2,333	9

O passo seguinte é normalizar os valores da matriz.

Tabela 5 – Matriz de comparação dos critérios

	<i>A</i>	<i>B</i>	<i>C</i>
<i>A</i>	0,454	0,428	0,555
<i>B</i>	0,454	0,428	0,333
<i>C</i>	0,091	0,143	0,111

Por fim efetuou-se o cálculo do *EigenValue* (Prioridade relativa), onde foi efetuada a média de cada uma das linhas da matriz de comparação normalizada.

Tabela 6 – Tabela de importância dos critérios

	<i>Prioridade relativa</i>	<i>Importância</i>
<i>A</i>	0,479	Maior
<i>B</i>	0,405	Intermédio
<i>C</i>	0,115	Menor

Através da Tabela 6 é possível concluir que o critério manutenção é o de menor importância, com uma percentagem de 11,5%, o critério de custo de desenvolvimento é o de maior

importância, com uma percentagem de 47,9% e o critério funcionalidades é de importância intermédia, com uma percentagem de 40,5%. Com os cálculos das prioridades, dos critérios, finalizados, é essencial verificar a veracidade dos valores.

Consistência dos Critérios

Uma vez que os valores atribuídos na comparação são subjetivos e baseados em opiniões, é necessário verificar a sua coerência. Esta é feita por uma análise de consistência da matriz apresentada na Tabela 11. Os cálculos seguiram a seguinte sequência [56]:

- 1) Soma dos valores de cada coluna e multiplicar pela respetiva prioridade relativa, obtendo assim o $\lambda_{m\acute{a}x}$;
- 2) Cálculo do Índice de Consistência (IC), da seguinte forma $\frac{\lambda_{m\acute{a}x}-n}{n-1}$ no qual o n representa o número de linhas;
- 3) Cálculo da Relação de Consistência (RC), da seguinte forma $\frac{IC}{IAM}$.

O valor de Inconsistência Aleatória Média (IAM) é um valor pré-definido que está dependente da dimensão da matriz em questão (Tabela 6). No caso em questão a nossa matriz tem a dimensão 3, logo o valor do IAM é 0,58. A consistência é validada se o valor final for inferior ou igual 0,1.

Tabela 7 – Índice de Consistência Aleatória

Dimensão da matriz	1	2	3	4	5	6	7	8	9	10
IAM	0,00	0,00	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49

Posto isto, foi executado o cálculo do primeiro ponto da sequência:

$$\begin{bmatrix} 1 & 1 & 5 \\ 1 & 1 & 3 \\ \frac{1}{5} & \frac{1}{3} & 1 \end{bmatrix}$$

$$SC1 \quad SC2 \quad SC3$$

Em que SCn é a soma da coluna e o n o número da coluna.

Obtendo os seguintes valores:

- $SC1 = 2,2$
- $SC2 = 2,333$
- $SC3 = 9$

Com todos os valores para o $\lambda_{m\acute{a}x}$, foi feito o seu calculo:

$$\lambda_{m\acute{a}x} = 2,2 * 0,479 + 2,333 * 0,405 + 9 * 0,115 = 3,034$$

Seguiu-se o ndice de Consistencia:

$$IC = \frac{3,034 - 3}{3 - 1} = 0,017$$

E a Relatao de Consistencia:

$$RC = \frac{0,017}{0,58} = 0,029$$

Em conclusao como o RC e menor que 0,1 significa que a matriz utilizada para a comparaao dos criterios e valida e nao e necessario realizar calculos novamente.

Definiao de prioridades das Soluoes

Apos a definiao de prioridade dos criterios e da verificaao da sua consistencia, e necessario fazer a definiao das prioridades das soluoes, relativamente aos criterios.

Para a definiao de prioridades das soluoes, seguiu-se as seguintes etapas:

- 1) Comparou-se as soluoes, traduziu-se a informaao para uma matriz de comparaao, normalizou-se e calculou-se a prioridade relativa, para cada um dos criterios.
- 2) As prioridades relativas foram colocadas numa matriz e multiplicou-se pelos valores das prioridades relativas dos criterios.

Considerando:

- Soluao A - Reestruturar a soluao existente reduzindo funcionalidades;
- Soluao B - Criaao de uma nova soluao utilizando tecnologias *open-source* e com funcionalidades reduzidas;
- Soluao C - Reestruturar a soluao existente com tecnologias *open-source*;

Foi feita a comparaao entre as soluoes, relativamente ao criterio de **custo de desenvolvimento** (criterio A).

Tabela 8 – Comparação entre as soluções, segundo critério A

			Importância
Solução A	relativamente	Solução C	1/5
Solução B		Solução C	3
Solução A		Solução B	1/7

Resultando na matriz de comparação seguinte:

Tabela 9 – Matriz de comparação, relativamente ao critério A

	<i>Solução A</i>	<i>Solução B</i>	<i>Solução C</i>
<i>Solução A</i>	1	1/7	1/5
<i>Solução B</i>	7	1	3
<i>Solução C</i>	5	1/3	1
<i>Total</i>	13	1,476	4,2

De seguida, normalizou-se os seus valores e calculou-se a prioridade relativa:

Tabela 10 – Matriz de comparação normalizada e prioridade relativa (critério A)

	<i>Solução A</i>	<i>Solução B</i>	<i>Solução C</i>	<i>Prioridade Relativa</i>
<i>Solução A</i>	0,077	0,097	0,047	0,074
<i>Solução B</i>	0,538	0,677	0,714	0,643
<i>Solução C</i>	0,385	0,226	0,238	0,283

Relativamente ao critério **funcionalidades** (critério B) resultou a seguinte comparação:

Tabela 11 – Comparação entre as soluções, segundo o critério B

			Importância
Solução A	relativamente	Solução C	1/6
Solução B		Solução C	2
Solução A		Solução B	1/8

E da comparação surgiu a seguinte matriz de comparação:

Tabela 12 – Matriz de comparação, em relação ao critério B

	<i>Solução A</i>	<i>Solução B</i>	<i>Solução C</i>
<i>Solução A</i>	1	1/8	1/6
<i>Solução B</i>	8	1	2
<i>Solução C</i>	6	1/2	1
<i>Total</i>	15	1,625	3,167

Sendo o passo seguinte normalizá-lo e calcular a prioridade relativa com base nos valores da normalização.

Tabela 13 - Matriz de comparação normalizada e prioridade relativa (critério B)

	<i>Solução A</i>	<i>Solução B</i>	<i>Solução C</i>	<i>Prioridade Relativa</i>
<i>Solução A</i>	0,067	0,077	0,053	0,066
<i>Solução B</i>	0,533	0,615	0,632	0,593
<i>Solução C</i>	0,4	0,317	0,316	0,344

No que se refere à comparação das soluções, tendo em conta o critério da **manutenção** (critério C), obteve-se os seguintes valores:

Tabela 14 – Comparação entre soluções, segundo o critério C

			Importância
Solução A	relativamente	Solução C	1
Solução B		Solução C	3
Solução A		Solução B	1/5

Resultando a seguinte matriz de comparação:

Tabela 15 - Matriz de comparação, em relação ao critério C

	<i>Solução A</i>	<i>Solução B</i>	<i>Solução C</i>
<i>Solução A</i>	1	1/5	1
<i>Solução B</i>	5	1	3
<i>Solução C</i>	1	1/3	1
<i>Total</i>	7	1,533	5

De seguida, normalizou-se a matriz e calculou-se a prioridade relativa.

Tabela 16 - Matriz de comparação normalizada e prioridade relativa (critério C)

	<i>Solução A</i>	<i>Solução B</i>	<i>Solução C</i>	<i>Prioridade Relativa</i>
<i>Solução A</i>	0,143	0,130	0,2	0,157
<i>Solução B</i>	0,714	0,652	0,6	0,655
<i>Solução C</i>	0,143	0,217	0,2	0,186

A matriz das prioridades relativas das soluções, é a seguinte:

Tabela 17 – Prioridades relativas das soluções

	<i>Critério A</i>	<i>Critério B</i>	<i>Critério C</i>
<i>Solução A</i>	0,074	0,066	0,157
<i>Solução B</i>	0,643	0,593	0,655
<i>Solução C</i>	0,283	0,344	0,186

Tendo em conta a matriz das prioridades relativas, em relação aos critérios e a matriz das dos critérios, efetuou-se o cálculo das prioridades relativas das soluções (descrito anteriormente):

$$\begin{bmatrix} 0,074 & 0,066 & 0,157 \\ 0,643 & 0,593 & 0,655 \\ 0,283 & 0,344 & 0,186 \end{bmatrix} * \begin{bmatrix} 0,479 \\ 0,405 \\ 0,115 \end{bmatrix}$$

Do cálculo efetuado obteve-se as seguintes prioridades relativas:

Tabela 18 – Matriz de Pesos Globais das soluções

	<i>Peso Global</i>
<i>Solução A</i>	0,0802
<i>Solução B</i>	0,6234
<i>Solução C</i>	0,2963

Consistência das Soluções

Com a definição das soluções concluída, é necessário validar a consistência dos valores. Foram efetuadas as análises de consistência para todas as matrizes de comparação das soluções com os critérios (Tabela 11,14 e 17). A validação segue o mesmo procedimento do cálculo da consistência dos critérios.

Obteve-se os seguintes valores de IC e RC das matrizes:

Tabela 19 – Valores de IC e RC para as matrizes

<i>Matriz</i>	<i>IC</i>	<i>RC</i>
<i>Critério A</i>	0,0498	0,086
<i>Critério B</i>	0,0217	0,037
<i>Critério C</i>	0.0645	0,028

Como é possível observar, todos os valores de RC são inferiores a 0,1, logo é possível concluir que os valores das matrizes são consistentes.

Conclusões

Com a consistência dos cálculos comprovada é possível considerar os valores obtidos. Os resultados das soluções são os seguintes:

- Reestruturar a solução existente reduzindo funcionalidades – 8,02%;
- Criação de uma nova solução utilizando tecnologias *open-source* e com funcionalidades reduzidas – 62,34%;
- Reestruturar a solução existente com tecnologias *open-source* – 29,63%;

Provando que a solução de criar uma solução utilizando tecnologias *open-source* e com funcionalidades reduzidas é a melhor para o problema em questão.

3.3 Valor da Solução

3.3.1 Valor, Valor para o Cliente, Valor Percecionado

De forma a compreender a importância de um produto, é necessário analisar o seu valor, valor para o cliente e o valor percebido.

O valor de um produto é a importância de um produto ou serviço na perspectiva de um cliente. É um conceito fundamental para o desenvolvimento de um produto e na definição de um preço de mercado [57]. Alguns exemplos de ações que aumentam o valor de um produto são novas funcionalidades, a confiabilidade, fácil usabilidade e a eficiência.

O valor para o cliente é o valor da satisfação dos clientes, perante um produto ou serviço, sendo este subjetivo e uma avaliação pessoal [58]. O valor da satisfação de um cliente varia consoante a sua utilização e a sua experiência, com o mesmo. Na perspectiva do projeto, referenciado neste documento, com a facilidade de gerir uma frota de veículos através de uma plataforma com

uma interface simples e intuitiva, é esperado que o cliente apresente um valor de satisfação elevado.

O valor percebido é uma comparação entre os benefícios e os sacrifícios de um produto ou serviço, pelo cliente [59]. Relativamente ao projeto, os benefícios da solução, para o cliente, são o aumento da produtividade, redução de custos, a simplificação de processos e a centralização de informação. No entanto a adaptação para com a utilização da ferramenta pode ser considerada um sacrifício para o cliente. É de salvaguardar que o valor percebido é variável consoante a opinião pessoal de cada cliente.

3.3.2 Proposta de Valor

A proposta de valor expõe, de uma forma clara, objetiva e transparente, a relevância da utilização de um serviço ou da compra de um produto para o cliente [60]. Representa assim uma parte fulcral de um negócio, uma vez que sem um conceito bem executado é difícil obter sucesso.

Aplicando este conceito ao projeto, a proposta de valor traduz-se numa solução de gestão de frotas que não implique muitos custos como acontece com a solução atual, a Geocar, e que fornece ao cliente a possibilidade de gerir os seus veículos e saber a suas localizações de uma forma simples e dinâmica.

3.3.3 Business Model Canvas

O modelo de negócio Canvas é uma ferramenta vantajosa para a definição do modelo de negócio de uma empresa ou de um projeto. Permite uma assistência visual para a compreensão do funcionamento de um negócio[61]. Responde a questões como:

- Quem irão ser os principais parceiros e fornecedores?
- Quais são as atividades mais importantes?
- Quais são os recursos que o negócio necessita?
- Quais são os custos?
- Qual é o problema a ser resolvido?
- Quem são os clientes?
- Qual é a relação estabelecer com os clientes?
- Como conseguimos alcançá-los?

Este modelo é constituído por parceiros, atividades e recursos chave, proposta de valor, relação com os clientes, canais, segmentos de clientes, estrutura de custos e fontes de receita [61].

A Figura 17 ilustra o modelo de negócio Canvas elaborado para o projeto desta tese.

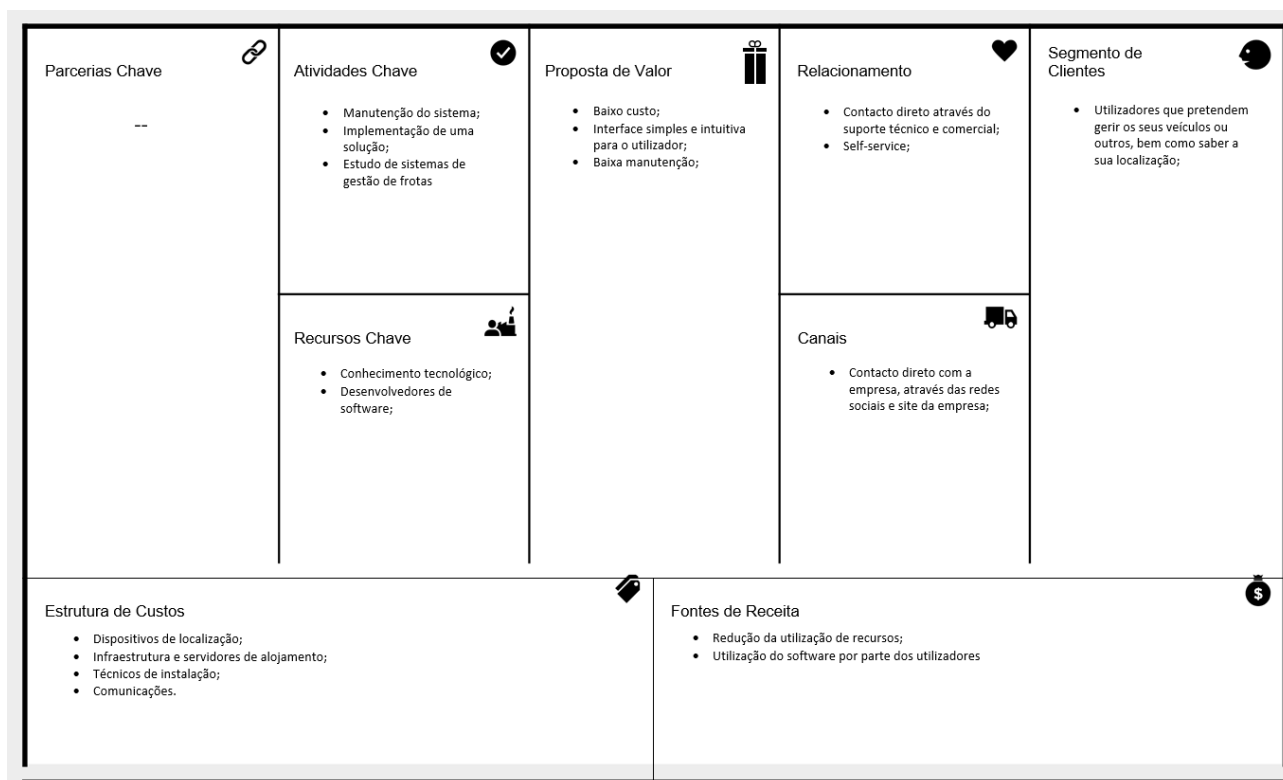


Figura 17 – Business Model Canvas

Como é possível observar na figura acima, cada quadrado corresponde a uma área do negócio, de seguida irão ser explicados, cada um deles, com maior detalhe:

- **Parceiras Chave** – nesta área é pretendido que se enumere as alianças estratégicas com fornecedores ou com outras empresas, que irão possibilitar ajuda para o desenvolvimento do produto. Uma vez que o produto é uma plataforma interna, não existem parcerias chave.
- **Atividades Chave** – neste campo são enumerados os processos fundamentais para a eficácia do modelo de negócio. No caso do presente projeto, as atividades são o estudo de sistemas de gestão de frotas, de modo a perceber os pontos essenciais para analisar e a melhor forma de executar o desenvolvimento, a implementação de uma solução e a manutenção do sistema.
- **Recursos Chave** – este campo visa perceber quais as necessidades a nível de recursos que o negócio exige. Os recursos podem ser de carácter humano, físico, intelectual e financeiro. No âmbito do projeto é necessário o conhecimento tecnológico e programadores para implementar a plataforma.

- **Proposta de Valor** – nesta área é apresentado o que irá ser oferecido aos clientes. É favorável que a oferta se destaque pela diferença, relativamente à concorrência. No âmbito deste projeto, a proposta de valor consiste, baixo custo e baixa manutenção para o cliente, bem como apresentar uma interface simples e intuitiva para o utilizador, uma vez que as outras plataformas semelhantes são complexas.
- **Relacionamento** – este campo pretende expor como irá ser formada a relação com os clientes. No âmbito deste projeto é pretendido que os clientes sejam independentes na utilização da plataforma, mas terão sempre o apoio através do suporte técnico e comercial.
- **Canais** – pretende-se, com este ponto, saber como irão ser alcançados os clientes. No âmbito do projeto, o único canal de comunicação é o contacto direto com a empresa, através das redes sociais e site da própria empresa.
- **Segmento de Clientes** – esta área tem como objetivo enumerar quem são os clientes que irão usufruir do produto. No caso deste projeto serão os utilizadores que pretendem gerir os seus veículos ou outros, bem como saber as suas localizações.
- **Estrutura de Custos** – com este ponto é pretendido que se identifique os custos de maior importância inerentes ao negócio. No projeto em questão são os dispositivos de localização, a infraestrutura e servidores de alojamento, os técnicos de instalação dos dispositivos de localização e as comunicações dos dispositivos para o servidor.
- **Fontes de Receita** – nesta área é esperada que se enumere as formas como irá ser conseguido o retorno do investimento no negócio. No âmbito do projeto é obtido uma redução da utilização de recursos, nomeadamente, a implementação da nova solução e também a utilização do software por parte dos utilizadores.

3.3.4 Quality Function Deployment

O QDF (*Quality Function Deployment*) é um sistema feito para o desenho de produtos de acordo com as necessidades dos consumidores. O seu processo teórico começou a ser desenvolvido na Mitsubishi, em 1972, procurando desenvolver os seguintes fatores:

- Melhor entendimento das necessidades dos consumidores.
- Melhoria da organização e desenvolvimento de projetos.
- Diminuição dos problemas.
- Aumento da produtividade.

O primeiro passo consiste na definição das necessidades dos utilizadores. Estas necessidades devem partir de testemunhos de clientes. Para o contexto do projeto, e tal como referido na secção 3.2.2 (Análise da Oportunidade), existem pedidos dos clientes para o produto em análise, pelo que é possível fazer um levantamento direto das necessidades do cliente a partir daí. As necessidades identificadas foram:

- **Visualização dos veículos em tempo real:** Possibilidade de o utilizador poder visualizar os seus veículos em tempo real;
- **Rápido a carregar (mapa):** A plataforma deve mostrar o mapa de forma fluida;

- **Visualização do percurso no mapa:** O utilizador deve ter a possibilidade de ver o percurso do seu veículo no mapa;
- **Visualizar o estado do veículo (ignição ligado/desligado) :**O utilizador deve ter a possibilidade de ver o estado do seu veículo, nomeadamente, quando a ignição está ligada ou desligada;
- **Notificações dos alertas:** A plataforma deve ser possível mostrar as notificações dos alertas aos utilizadores;
- **Aplicação web e mobile:** A plataforma deve estar disponível para web e mobile.

A cada uma destas necessidades foram atribuídos valores de importância absoluta, de 1 a 5, onde representa a mínima importância para o utilizador e 5 representa a importância máxima. De seguida foi feita a sua normalização, obtendo-se os resultados apresentados na Tabela 20:

Tabela 20 – Necessidades dos utilizadores classificados por importância

<i>Necessidades</i>	<i>Importância (Absoluta)</i>	<i>Percentagem</i>
<i>Visualização dos veículos em tempo real</i>	5	20,83%
<i>Rápido a carregar (mapa)</i>	3	12,5%
<i>Visualizar o percurso do mapa</i>	4	16,7%
<i>Visualizar o estado do veículo</i>	4	16,7%
<i>Notificações dos alertas</i>	5	20,83%
<i>Aplicação web e mobile</i>	3	12,5%

As características de engenharia identificadas que procuram responder às necessidades dos utilizadores foram as seguintes:

- **Sessão de autenticação (login):** Cada sessão efetuada por cada utilizador, deve ser possível mostrar somente os seus veículos associados.
- **Integração de serviços de mapas:** Característica associado ao mapa que é utilizado pela plataforma, de forma a poder mostrar os veículos;
- **Integração com dispositivos de localização GPS:** Característica associado aos dados que são mostrados no mapa vindos dos equipamentos de localização GPS
- **Serviços de alertas em tempo real:** Característica associado aos alertas (velocidade/ignição) que são configurados na camada de receção da solução.
- **Acesso à base de dados:** Característica associado aos dados de consulta e de armazenamento para efetuar determinadas operações.

As Figuras 18 e 19 apresentam duas escalas, que permitem respetivamente representar as relações entre as necessidades dos consumidores e as características de engenharia e a correlação entre as diferentes características de engenharia:




Matriz de Relação		
	Relação Forte	9
	Relação Média	3
	Relação Fraca	1
	Sem Relação	0

Figura 19 – Matriz de Relação

Matriz de Correlação	
++	Fortemente positiva
+	Positiva
-	Negativa
--	Fortemente negativa
	Sem correlação

Figura 18 – Matriz de Correlação

De acordo com análise do Anexo 3 é possível retirar as seguintes ilações:

- A **Integração com dispositivos de localização GPS** é a característica de engenharia que apresenta maior grau de importância.
- O **acesso à base de dados** é a característica de engenharia que apresenta menor grau de importância.
- As características de engenharia **integração com dispositivos de localização GPS** e a **integração de serviços de mapas** possuem uma correlação fortemente positiva.
- As características de engenharia **integração com dispositivos de localização GPS** e o **acesso à base de dados** possuem uma correlação fortemente positiva.
- Em relação à concorrência, estas possuem resultados idênticos em todas necessidades dos utilizadores.

4 Desenho e Análise da solução

Este capítulo descreve os conceitos principais da solução, assim como é analisado o modelo de domínio do projeto em questão. Também é feita a engenharia de requisitos onde são apresentados os requisitos funcionais e não funcionais. Para além disso, é analisado e esquematizado a arquitetura geral do sistema, bem como o modelo relacional da solução a desenvolver.

4.1 Arquitetura geral do Sistema

A arquitetura de software de um sistema consiste na definição dos componentes de software, nas suas propriedades externas, como também nas suas interligações com outros softwares. Apresenta-se de seguida na Figura 20 um diagrama de componentes do sistema em geral que foi planeado com os seus principais componentes.

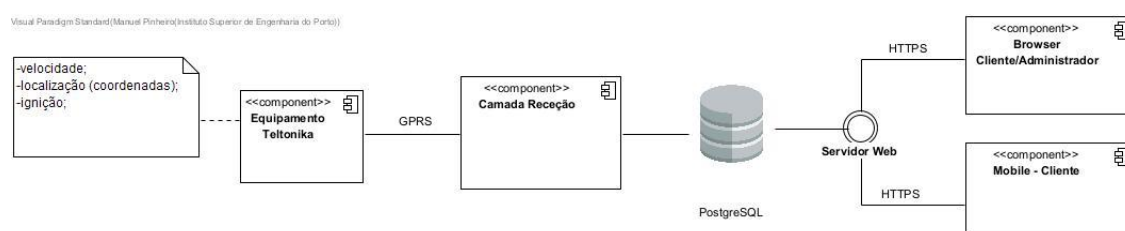


Figura 20 – Diagramas de componentes (geral)

No diagrama representou-se duas plataformas, em que uma delas é uma plataforma *web* e a outra é uma aplicação *mobile*. A plataforma *web* é utilizada pelos administradores e os clientes, onde estes têm a possibilidade de efetuar as suas operações consoante a sua autenticação. Relativamente à aplicação *mobile*, será desenvolvida para o administrador e o cliente, de modo a poderem ter acesso as mesmas funcionalidades da plataforma *web* através de um dispositivo

móvel. Para além disso, para a recolha dos dados, tais como, as coordenadas geográficas, o estado de ignição e a velocidade é utilizado um **equipamento de localização** (dispositivo da *Teltonika*), que por sua vez comunica com a **camada de receção** através da ligação GPRS. A camada de receção contém comandos SQL, onde insere e/ou seleciona os dados da Base de Dados (**PostgreSQL**) e também onde é feito a tradução e compilação dos dados vindos do equipamento de localização. Além disso, a plataforma *web* e a aplicação *mobile* acedem às funcionalidades através da conexão de *webservices* fornecida pela aplicação, no qual estas conexões são efetuadas em *Hyper Text Transfer Protocol* (HTTP). Para a construção da arquitetura geral do sistema tem como base a **Arquitetura 3-tier**.

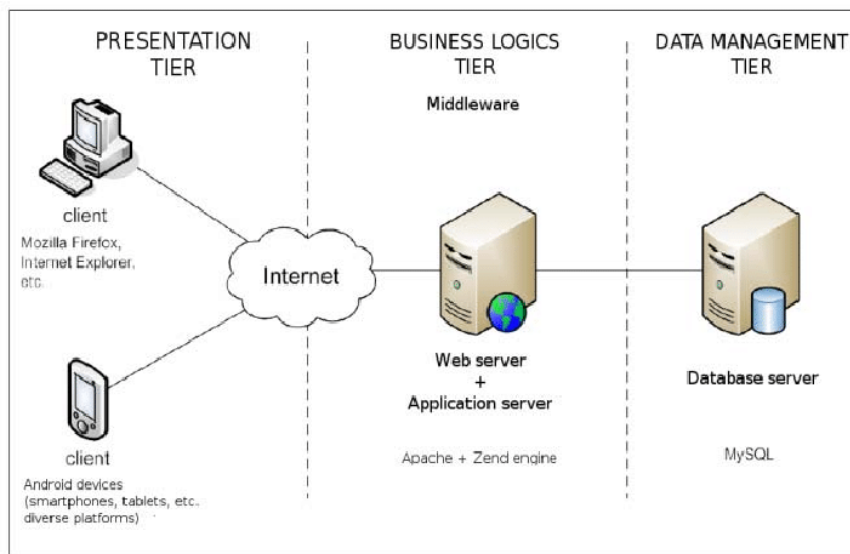


Figura 21 – Arquitetura 3-tier [76]

Como é possível visualizar na Figura 21, arquitetura 3-tier é constituída por 3 camadas [62]:

- **Presentation Tier**

Componente de interface do utilizador. É a camada que interage diretamente com o utilizador

- **Business Logics Tier**

Componente de processamento. Calcula e determina o fluxo de informação entre as outras duas camadas.

- **Data Management Tier**

Componente de armazenamento de dados.

Estas camadas estão conceptualmente separadas, o que não corresponde necessariamente a localizações físicas distintas (por exemplo, as três camadas podem existir em apenas duas máquinas distintas). A **Presentation Tier** é vista como a camada do topo, isto é, de apresentação; numa aplicação típica da Web corresponde às interfaces em HTML. É com esta camada que o

utilizador interage, não tendo a possibilidade de saber como é que os dados estão a ser processados e onde estão armazenados, uma vez que o HTML é gerado pelo servidor. A camada intermédia (**Business Logics Tier**) é a responsável pelas transações e pela manutenção da integridade dos dados, entre a camada 1 (**Presentation Tier**) e a camada 3 (**Data Management Tier**). Esta (camada intermédia) interpreta o pedido do cliente *Web*, formula o *query* à base de dados, e, de acordo com os resultados da consulta, constrói a página em HTML e reenvia-a ao cliente *Web* para ser mostrada ao utilizador. Na **Data Management Tier**, camada 3, encontram-se os dados, que podem estar armazenados no servidor residente, ou em outros servidores de dados [62].

4.2 Equipamento de Localização

A Figura 22 ilustra a arquitetura de *hardware* do equipamento de localização utilizado. Este dispositivo é composto por 4 módulos principais: um **módulo GPS**, um **módulo GSM/GPRS**, **módulo de alimentação** e um **microprocessador**. Para além destes módulos já referidos, existem outros módulos que podem ser utilizados dependendo do modelo do equipamento e da sua finalidade.

- GSM/GPRS Modem – Efetua a comunicação das informações (dados geográficos, a velocidade e o estado de ignição) que são recolhidas do veículo através da ligação OBDII para a camada de receção que se encontra num servidor externo;
- Módulo GPS - Recebe o sinal de GPS;
- Microprocessador - Efetua o controlo e processamento do Sistema (GPS, sensores, GSM, I/Os e GPRS);
- Modulo de alimentação – Fornece a alimentação aos diferentes componentes do sistema;
- Sinais Digitais/Analógicos – Monitorização de parâmetros, tais como, a velocidade, a quantidade de combustível, o sensor de temperatura, o estado da ignição, o sensor das portas, entre outros;
- Bluetooth – Faz a leitura de dados do barramento CAN da ECU (*Engine Control Unit*) do veículo;

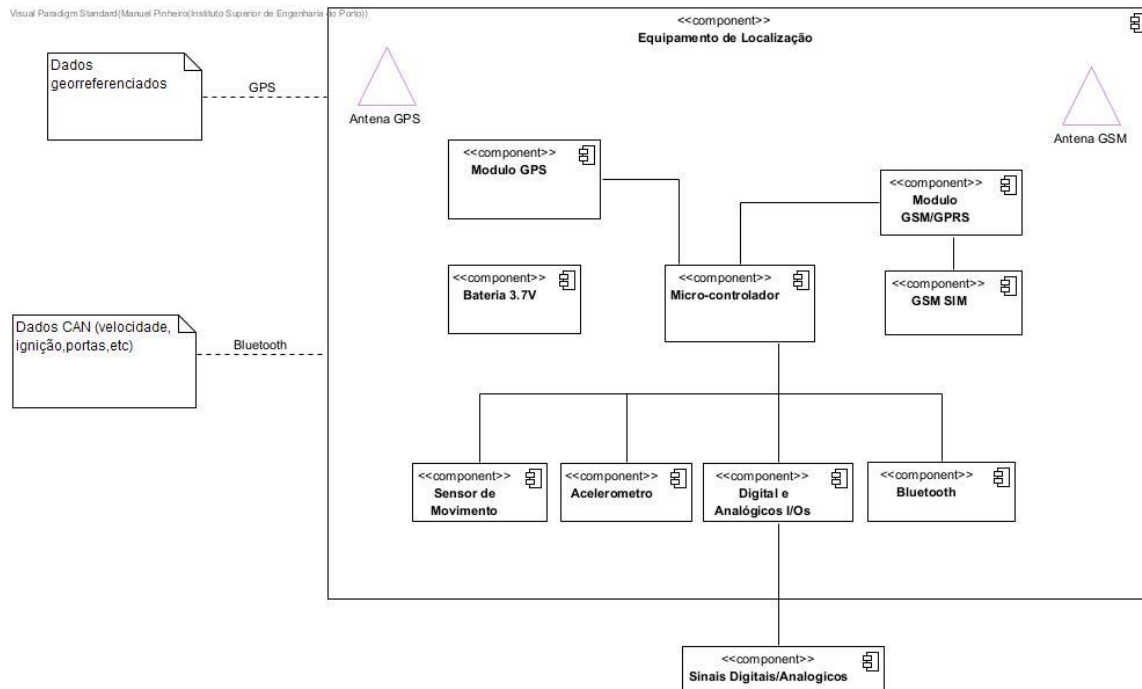


Figura 22 – Esquema do Equipamento de Localização

4.3 Camada de Receção

A camada de receção consiste num conjunto de componentes que possuem mecanismos de envio e receção que permitem a interação do equipamento com a plataforma. Esta camada é responsável por traduzir a informação recebida do equipamento e enviá-la para os servidores internos segundo um protocolo de comunicação específico sobre UDP ou TCP estando à escuta numa determinada porta.

A camada de receção é constituída por **dois packages**, designados por **High Level** e **Low Level**. O package **High Level** é onde é tratado os dados em relação ao veículo, nomeadamente, a configuração dos alertas (velocidade e ignição), registo dos percursos efetuados e a posição atual. Relativamente ao package **Low Level** é direcionado ao dispositivo de localização, uma vez que é feito a tradução e a compilação dos dados (coordenadas geográficas, velocidade, ignição) para o **High Level**.

High Level

Este package é responsável por fazer o tratamento dos dados recebidos pelo *Low Level*. Aqui é feita a transformação dos dados do componente GeoData para o GeoProcessor, no qual este tem como função tratar os dados para cada veículo. Além disso este componente é constituído pelos seguintes subcomponentes: GeoProcessor, GeoTask, GeoData, GeoVeiculo, GeoAlertas e ConfigServer.

GeoProcessor

Este subcomponente tem como objetivo para cada veículo verificar o IMEI do dispositivo de localização e a partir dos dados compilados do GeoData, utilizá-los para verificar e gerar o alerta do estado de ignição e o respetivo alerta de excesso de velocidade. Para além disso também permite atualizar o estado do veículo (ativo/desativo), bem como atualizar o equipamento que está associado a uma viatura.

GeoData

Este subcomponente tem como função criar uma trama de dados recebida pela camada *Low Level* que é constituída pelos seguintes parâmetros: velocidade, estado da ignição, latitude, longitude, altitude, data hora e o id do veículo.

GeoTask

Este subcomponente tem como função controlar os dados disponíveis na base de dados, no qual se definiu que os mesmos serão guardados dos últimos 3 meses.

GeoAlertas

Este componente é responsável por configurar e definir o tipo de alerta (ignição e/ou velocidade) para cada veículo ou para os veículos.

GeoVeiculo

Este subcomponente é responsável por inserir as coordenadas geográficas na tabela de base de dados, bem como definir a posição atual do veículo através dos dados do GeoData. Para além disso é configurado os alertas gerados para cada tipo de alerta, no qual é verificado o alerta de excesso de velocidade do veículo e também o alerta do estado de ignição, por fim é configurado o percurso através do evento de ignição do veículo.

ConfigServer

Este componente é responsável por criar uma *socket* do servidor (*server*) que comunica com a *socket* do cliente da parte do *back-end* da aplicação, no qual são transferidos/enviados os dados necessários (veículo) para uma porta específica, utilizando um protocolo específico (UDP ou TCP).

Low Level

Este package é responsável por estar à escuta numa porta específica (respeitando os protocolos UDP e TCP) dos dados vindo do equipamento de localização através do seu IMEI. Para cada protocolo de transporte está asegurado o modo de funcionamento de como são transferidos os diferentes tipos de pacote AVL que contém dados de localização (latitude, longitude, altitude, velocidade, estado da ignição e data hora) em *bytes*. De seguida estes dados são compilados para o componente **GeoData**.

O **Low Level** é constituído por três subcomponentes principais: **Teltonika**, **ReceiveServer** e **TransportUDP**. Os subcomponentes **Teltonika** e **TransportUDP** tem como função traduzir os dados vindos do equipamento que se encontram em *bytes*, e que estes serão descodificados de modo a serem colocados num objeto do tipo **GeoData** para que estes sejam enviados para a base de dados. O **ReceiveServer** consiste em criar duas *threads* distintas, uma com o protocolo UDP e outra com o protocolo TCP, no qual as mesmas são executadas numa porta específica.

Na Figura 23 encontra-se apresentado o diagramas de componentes da camada de receção do sistema.

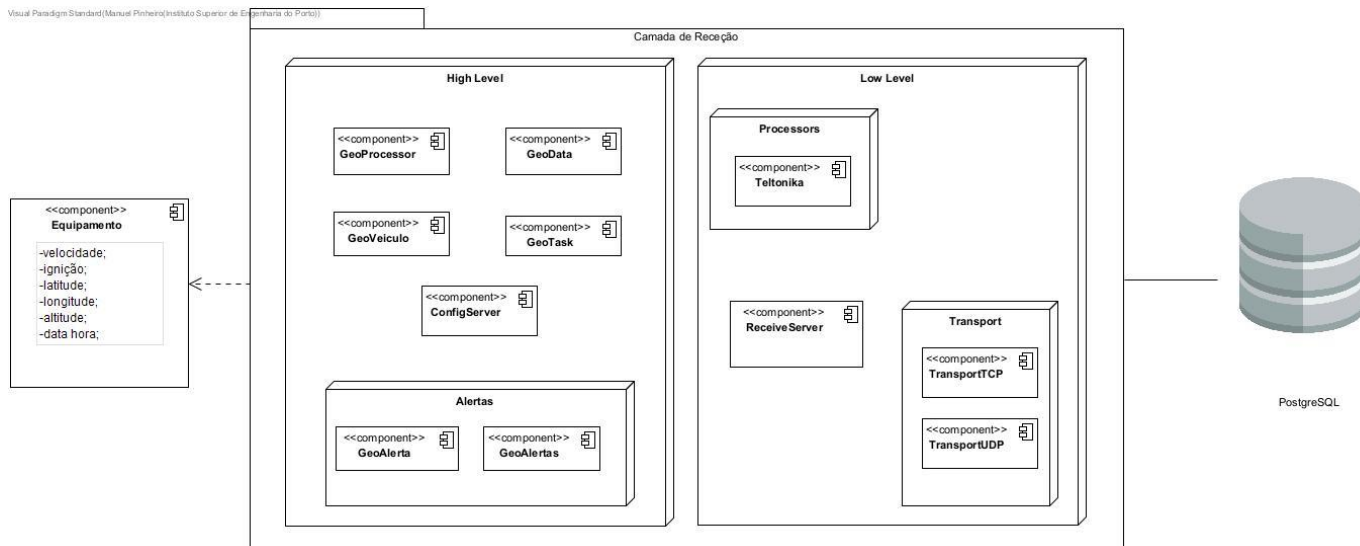


Figura 23 – Diagramas de componentes da Camada de Receção

4.4 Desenho da Arquitetura-Plataforma

O desenho arquitetural é um artefacto que representa a estrutura e organização geral do sistema de *software* a desenvolver. Este sistema é constituído por um conjunto de componentes que se interligam entre si através de relações. A partir deste sistema são gerados vários artefactos que descrevem a arquitetura do sistema a desenvolver e servem de modelo ao processo de implementação.

Na Figura 24 encontra-se apresentado o diagrama de componentes de alto nível construído para o sistema.

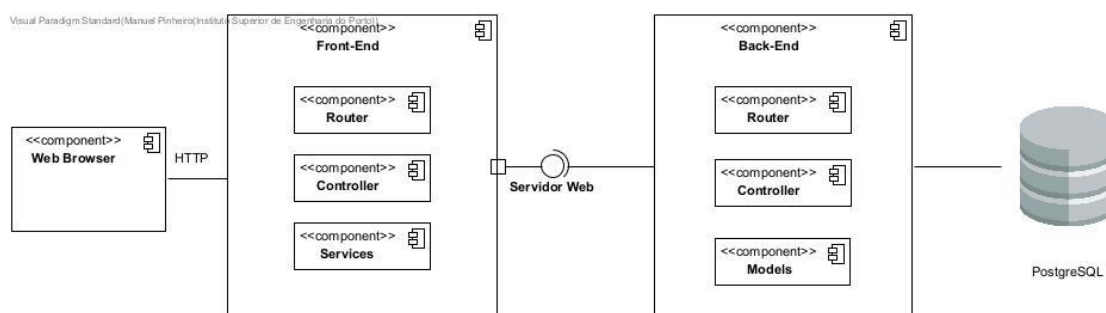


Figura 24 – Diagramas de componentes

Existem quatro componentes principais no sistema:

- **Front-End:** Componente que será desenvolvido em *Angular*, no qual é a Interface com o utilizador da plataforma. A comunicação com o *Back-End* é feita por HTTPS, através do consumo de uma *RESTful Web Services*. O utilizador utiliza esta interface através de um *web Browser* que comunica com a aplicação por HTTPS;
- **Back-End:** Componente que será desenvolvido em *NodeJs*, trata toda a lógica de negócio (*Models e Controllers*) e disponibiliza uma *RESTfull Web Services* para utilização pelo *Front-End*. Este componente comunica ainda com a base de dados.
- **Base de dados:** Componente base de dados trata do armazenamento de dados do sistema e é utilizado pelo componente de *Back-End*.

Para a construção do sistema de *software* foi utilizada uma arquitetura por camadas (*Layered Architecture*) baseado no padrão arquitetural *Model-View-Controller* (MVC). A utilização deste padrão permite que seja mais fácil a manutenção do *software*, com pacotes modulares de rápido desenvolvimento. Com este padrão, o desenvolvimento das funções divididas entre *Models, Views e Controllers* torna a aplicação mais leve e independente.

Além disso o facto de ter um design modular e separado também permite aos programadores e designers trabalharem simultaneamente, de modo que eles possam alterar partes da aplicação/*software* sem afetar outras.

Este padrão é caracterizado por ser constituído por três componentes lógicos: *Model*, *View* e *Controller*. Estes componentes comunicam entre si e permitem menos acoplamento no sistema.

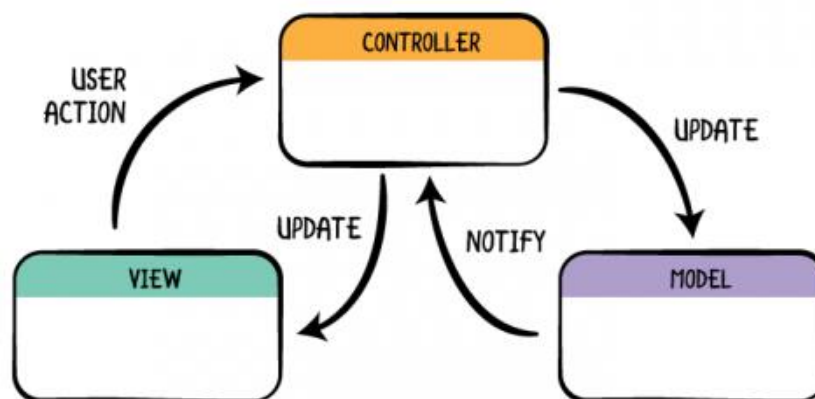


Figura 25 – Padrão MVC [77]

Como é possível visualizar na Figura 25, a estrutura do MVC divide-se em 3 componentes [63]:

- **Model**

Componente responsável pela lógica de negócio e gestão de dados da aplicação. Este componente representa objetos do mundo real e contém as entidades do domínio.

- **View**

Representa as funcionalidades que interagem diretamente com os utilizadores da aplicação.

- **Controller**

Componente responsável por tratar os pedidos do utilizador. Atua sobre o *Model* e apresenta os resultados na forma de *View*.

4.5 Modelo de Domínio

O modelo de domínio é o artefacto mais importante na análise orientada a objetos, que permite representar visualmente um conjunto de classes conceptuais significativas no domínio do problema. Para a construção do domínio é necessário a identificação de conceitos que se encontrem relacionados com os requisitos do sistema, os atributos dos mesmos e as ligações entre si. Portanto é apresentado de seguida o modelo desenvolvido para a resolução (Figura 26) do problema que foi exposto anteriormente na secção 1.2.

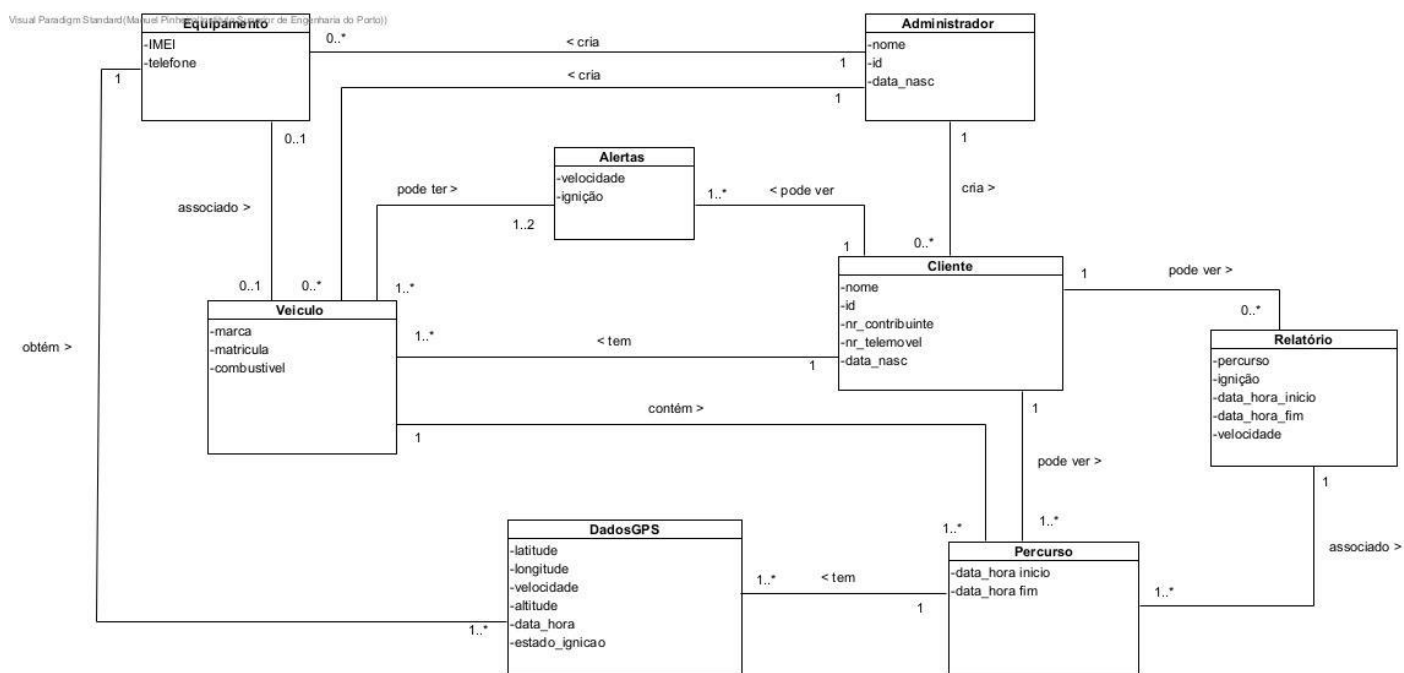


Figura 26 – Modelo de Domínio

No modelo de domínio acima esquematizado, pode-se analisar que o **cliente** pode ter vários **veículos**, em que por sua vez cada um destes contém um equipamento de localização identificado por um IMEI e um número de telefone. O conjunto de dados geográficos (**Dados GPS**) registados pelo equipamento de localização cria/regista o **percurso** efetuado pelo veículo. Para além disso é possível associar um ou dois **alertas(s)** (velocidade e/ou ignição) a vários veículos, de modo o cliente ter um melhor controlo do excesso de velocidade e/ou do estado da ignição no momento. Por fim, o cliente pode visualizar um **relatório** que contém a duração dos percursos efetuados pelos seus veículos, as velocidades máxima, média e mínima nesse percurso. Para além disso também permite saber o número de eventos de ignição num período estabelecido pelo cliente para um determinado veículo.

4.5.1 Principais conceitos

Nesta parte encontram-se descritos os principais conceitos de domínio, para que o trabalho desenvolvido e apresentado nas secções seguintes do relatório seja mais fácil de compreender.

- **Administrador**
Pessoa responsável pela gestão dos veículos, dos clientes e dos equipamentos. Para além disso, é responsável por associar um determinado equipamento a um veículo, bem como cria veículos para um determinado cliente.
- **Cliente**
Sujeito com permissões de ver os seus veículos no mapa, bem como tem a possibilidade de ativar/desativar os alertas de excesso de velocidade e/ou estado de ignição de um dos seus veículos. Além disso também pode ver os percursos efetuados pelos seus veículos num determinado período.
- **Alertas**
Representa os alertas do estado de ignição e os alertas de excesso de velocidade.
- **Relatório**
Possibilidade de extração de um relatório que contém o histórico dos percursos efetuados, nomeadamente a duração e as velocidades média, mínima e máxima desse percurso e também um relatório do número de eventos de ignição num período específico.
- **Percurso**
Representa a rota efetuada pela viatura de um determinado cliente num período específico.
- **Veículo**
Representa a viatura do cliente criada pelo administrador.
- **Equipamento**
Representa o dispositivo de localização identificado por um IMEI e número de telefone que se encontra instalado num veículo do cliente.
- **DadosGPS**
Representa o conjunto de dados geográficos (latitude, longitude, altitude, velocidade, data hora e o estado de ignição) registadas pelo equipamento de localização.

4.6 Engenharia de Requisitos

A Engenharia de Requisitos pode ser definida como o processo de desenvolvimento de software onde são recolhidos, analisados, documentados e compreendidos os requisitos e necessidades do produto a desenvolver. De uma forma geral os requisitos possuem um papel fundamental para o desenvolvimento de software. Os requisitos de software são uma das principais medidas de sucesso de um determinado software, visto que os seus objetivos que tinham sido planeados, forma totalmente de acordo com as necessidades dos clientes.

Neste capítulo vão ser identificados os requisitos funcionais e não funcionais, para uma melhor compreensão das condições do projeto.

4.6.1 Requisitos Funcionais

Os requisitos funcionais determinam as principais características/funcionalidades do sistema. Estes requisitos assumem somente os associados à implementação das características do sistema.

Deste modo é apresentado o diagrama de casos de uso (Figura 27) com os requisitos identificados.



Figura 27 – Casos de Uso

De seguida são apresentados todos os casos de uso num formato breve que descreve, de forma visual, através de um diagrama *System Sequence Diagram* (SSD), e textual a interação do utilizador com o sistema e vice-versa.

UC1: Fazer Login na Plataforma e UC2: Visualizar Percurso

Estes casos de uso têm como ator principal o cliente. O primeiro caso de uso permite o cliente efetuar o login na plataforma para ter acesso as suas funcionalidades. Após estar autenticado é apresentado um mapa com a localização atual de todos os seus veículos. De seguida, seleciona um determinado veículo no mapa e é apresentado os dados do veículo. Depois seleciona a opção “Visualizar Percurso” e de seguida seleciona um determinado período, e por fim é apresentado o percurso no mapa desse veículo. Na Figura 28 encontra-se apresentado o diagrama SSD, a interação do utilizador com o sistema e vice-versa.

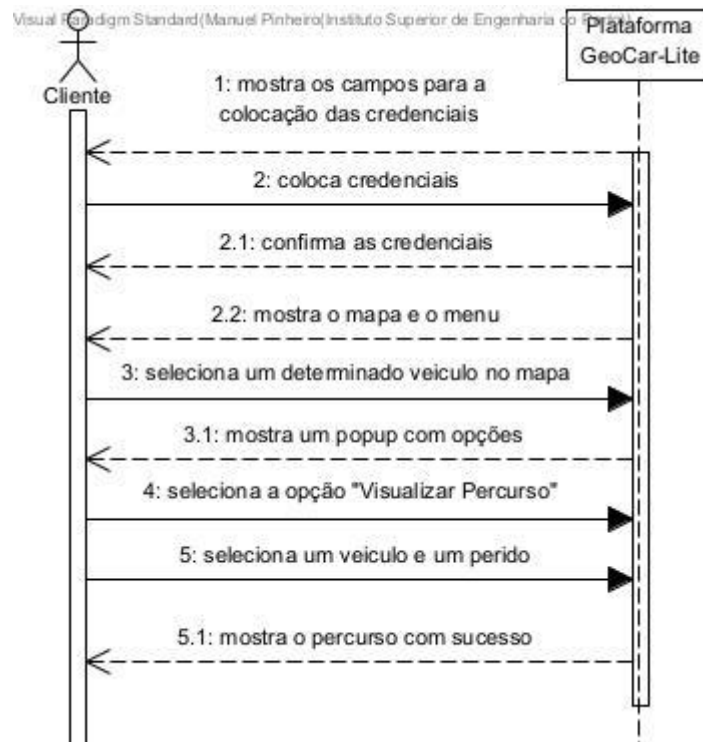


Figura 28 – UC1 e UC2 Visualizar Percurso

UC3: Visualizar Relatórios

Este caso de uso mostra que o cliente pode visualizar relatórios em relação ao seu veículo, tais como, visualizar uma tabela de percursos e ignição num período específico. Uma vez que o SDD deste caso de uso é extenso foi colocado em anexo. (Figura 54)

UC3 – Visualizar Relatórios	
Cenário Principal	<ol style="list-style-type: none">1. O cliente faz login na plataforma;2. A plataforma confirma as credenciais;3. O cliente confirma as credenciais;4. A plataforma mostra o mapa com os veículos em tempo real;5. O cliente seleciona um veículo6. O cliente seleciona “Relatórios”7. A plataforma mostra os tipos de relatório e o período de visualização8. O cliente seleciona um tipo de relatório (percursos, ignição) e o período9. Envio automático para o email do cliente um/uma relatório/tabela e sucesso da operação
Ator Principal	O ator principal neste caso de uso é o cliente
Partes interessadas e os seus interesses	Cliente: pretende visualizar relatórios
Pré-condições	O cliente tem de estar autenticado
Pós-condições	O cliente visualizar relatório

UC4: Configurar Alertas

Este caso de uso mostra que o cliente pode configurar os alertas (ignição e/ou velocidade), no qual na plataforma irá conter uma tabela em que pode configurar os alertas (velocidade e estado da ignição) para o veículo selecionado no mapa (seleção de *checkbox*). Uma vez que o SDD deste caso de uso é extenso foi colocado em anexo. (Figura 55)

UC4 – Configurar Alertas	
Cenário Principal	<ol style="list-style-type: none">1. O cliente faz login na plataforma;2. A plataforma confirma as credenciais;3. O cliente confirma as credenciais;4. A plataforma mostra o mapa com os veículos em tempo real;5. O cliente seleciona um veículo;6. O cliente seleciona “Configurar Alertas”;7. A plataforma mostra ativar/desativar os alertas de velocidade e/ou de ignição;8. O cliente ativa/desativa o(s) tipo(s) de alerta e clica em aplicar;9. A plataforma confirma as configurações e sucesso da operação
Ator Principal	O ator principal neste caso de uso é o cliente
Partes interessadas e os seus interesses	Cliente: pretende configurar alertas
Pré-condições	O cliente tem de estar autenticado
Pós-condições	O cliente configura os alertas

UC5: Ver veículos no mapa

Este caso de uso mostra que o cliente pode ver a localização dos seus veículos em tempo real. Na Figura 29, encontra-se apresentado o diagrama SSD, a interação do utilizador com o sistema e vice-versa.

UC5 – Ver veículos no mapa	
Cenário Principal	<ol style="list-style-type: none"> 1. O cliente faz login na plataforma; 2. A plataforma confirma as credenciais; 3. O cliente confirma as credenciais; 4. A plataforma mostra o mapa com os seus veículos em tempo real; 5. Sucesso da operação.
Ator Principal	O ator principal neste caso de uso é o cliente
Partes interessadas e os seus interesses	Cliente: pretende ver os veículos no mapa
Pré-condições	O cliente tem de estar autenticado
Pós-condições	O cliente vê os veículos em tempo real

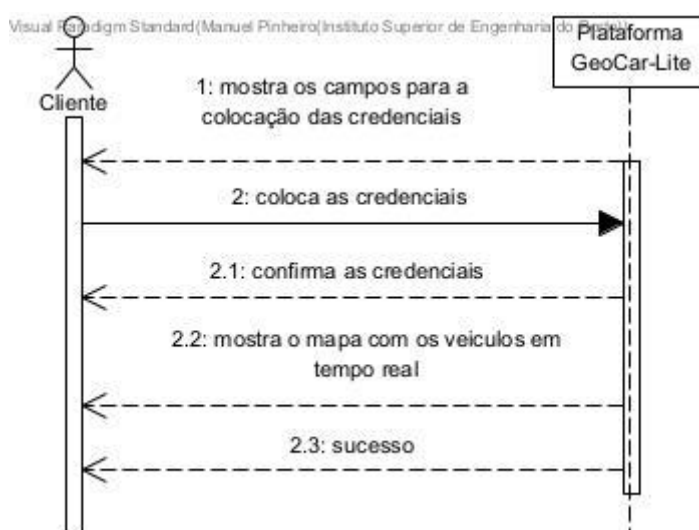


Figura 29 - UC5-Ver veículos no mapa

UC6: Criar Veículos, UC7: Editar Veículos e UC8: Desativar Veículos

Este caso de uso mostra que o administrador pode gerir um determinado veículo, isto é, tem a permissão de criar, editar e desativar o veículo selecionado. Para isso o administrador pode também associar a um equipamento (caso ele exista) ou simplesmente pode desativá-lo. Uma vez que o SDD deste caso de uso é extenso foi colocado em anexo. (Figura 56)

UC6 – Criar Veículos, UC7 – Editar Veículos e UC8 – Desativar Veículos	
Cenário Principal	<ol style="list-style-type: none">1. O administrador faz login na plataforma;2. A plataforma confirma as credenciais;3. O administrador confirma as credenciais;4. A plataforma mostra a lista de clientes;5. O administrador seleciona um determinado cliente6. A plataforma mostra uma página com os detalhes do cliente bem como a sua lista de veículos criados, bem como a opção “Criar Veículo” ou “Editar Veículo” ou “Desativar Veículo”7. O administrador cria ou edita ou desativa um veículo e/ou coloca os dados necessários do veículo, bem como associa a um equipamento (caso esteja criado).8. A plataforma confirma as configurações e sucesso da operação
Ator Principal	O ator principal neste caso de uso é o administrador
Partes interessadas e os seus interesses	Administrador: pretende editar ou desativar ou criar os veículos
Pré-condições	O administrador tem de estar autenticado
Pós-condições	O administrador cria ou edita ou desativa os veículos

UC9: Criar Clientes, UC10: Editar Clientes e UC11: Desativar Clientes

Este caso de uso mostra que o administrador pode gerir um determinado cliente, ou seja, tem permissão de criá-lo ou editá-lo ou desativá-lo. Para isso o administrador tem de preencher os dados necessários para a criação ou editar os dados, bem como associar a um veículo, caso pretenda. Uma vez que o SDD deste caso de uso é extenso foi colocado em anexo. (Figura 57

UC9 – Criar Clientes, UC10 – Editar Clientes e UC11 – Desativar Clientes	
Cenário Principal	<ol style="list-style-type: none">1. O administrador faz login na plataforma;2. A plataforma confirma as credenciais;3. O administrador confirma as credenciais;4. A plataforma mostra a lista de clientes;5. O administrador seleciona a opção de “Criar Novo” ou “Editar Cliente” ou “Desativar Cliente”.6. A plataforma mostra os campos que o administrador deve preencher ou editar ou simplesmente pode desativar.7. O administrador preenche os dados.8. A plataforma confirma as configurações e sucesso da operação.
Ator Principal	O ator principal neste caso de uso é o administrador
Partes interessadas e os seus interesses	Administrador: pretende criar ou editar ou desativar os clientes
Pré-condições	O administrador tem de estar autenticado
Pós-condições	O administrador criar ou editar ou desativar os clientes

UC12: Criar Equipamentos, UC13: Editar Equipamentos e UC14: Eliminar Equipamentos

Este caso de uso mostra que o administrador pode gerir um equipamento, ou seja, tem a possibilidade de criá-lo, editá-lo e eliminá-lo. Para isso o administrador tem de preencher os dados necessários (IMEI, número de telefone) para a criação ou edição, bem como associar a um veículo, caso pretenda, ou simplesmente pode eliminá-lo. Uma vez que o SDD deste caso de uso é extenso foi colocado em anexo. (Figura 58)

UC12 – Criar Equipamentos, UC13 – Editar Equipamentos e UC14 – Eliminar Equipamentos	
Cenário Principal	<ol style="list-style-type: none"> 1. O administrador faz login na plataforma; 2. A plataforma confirma as credenciais; 3. O administrador confirma as credenciais; 4. A plataforma mostra a lista de clientes; 5. O administrador seleciona a opção “Gerir Equipamentos” 6. A plataforma mostra uma página com uma lista de equipamentos criados, bem como a opção “Criar Equipamento” ou “Editar Equipamento” ou “Eliminar Equipamento” 7. O administrador cria ou edita um equipamento e coloca os dados necessários, bem como associa a um veículo ou simplesmente pode eliminá-lo 8. A plataforma confirma as configurações e sucesso da operação
Ator Principal	O ator principal neste caso de uso é o administrador
Partes interessadas e os seus interesses	Administrador: pretende criar ou editar ou eliminar os equipamentos
Pré-condições	O administrador tem de estar autenticado
Pós-condições	O administrador cria ou edita ou elimina os equipamentos

UC15: Apagar Dados da Base de dados

Este caso de uso mostra que na camada de recepção estará desenvolvida de modo a ser possível eliminar o histórico de dados geográficos (latitude, longitude, altitude, velocidade, data hora, ignição) com mais de 3 meses, de modo a não ocupar muito espaço na base de dados.

UC16: Inserir Coordenadas

Este caso de uso mostra que na camada de recepção será responsável por registrar/inserir as coordenadas dos veículos na base de dados.

UC17: Inserir Estado Atual do Veículo

Este caso de uso mostra que na camada de recepção será responsável por registrar/inserir o estado atual do veículo no mapa, isto é, registrar a última localização identificada pelo dispositivo de localização.

UC18: Inserir Percurso

Este caso de uso mostra que na camada de recepção será responsável por registrar/inserir o percurso do veículo num determinado período na base de dados, analisando os pontos geográficos onde ocorreu uma mudança de estado de ignição, ou seja, onde o condutor iniciou um percurso acionando a ignição e de seguida finalizou a rota desligando a mesma.

UC19: Atualizar Percurso

Este caso de uso mostra que na camada de recepção será responsável por atualizar o percurso de um determinado veículo na base de dados, verificando onde houve mudança de estado de ignição.

UC20: Inserir Alertas

Este caso de uso mostra que na camada de recepção será responsável por registrar/inserir os alertas gerados dos veículos na base de dados, nomeadamente, a data hora onde aconteceu uma mudança de estado de ignição, bem como, o excesso de velocidade.

4.6.2 Requisitos não funcionais

Requisitos não funcionais descrevem características não descritivas da solução alvo. Ao contrário dos requisitos funcionais, que especificam as funcionalidades que o sistema deve ser capaz de realizar, os não funcionais definem como este funcionará [64]. Este tipo de requisitos consiste em restrições relacionadas com desempenho, manutenibilidade, funcionamento, segurança ou normas a seguir, entre outros aspetos [64]. Para esta especificação procedeu-se à análise FURPS+, um método de categorização de requisitos que os organiza consoante a sua área de influência no projeto.

1. Usabilidade

Relacionado com a especificação de interfaces com o utilizador e das interações dos utilizadores com o sistema

- Interface gráfica simples e eficiente, no visual;
- Transições das funcionalidades fluidas;
- Deve ser simples e intuitiva a utilização das funcionalidades;
- Diferentes tipos de visualização do mesmo conteúdo numa determinada página;

2. Desempenho

Avalia os requisitos de desempenho do *software*, nomeadamente: tempo de resposta, consumo de memória, utilização da CPU, capacidade de carga e disponibilidade da aplicação.

- O tamanho do visual deverá ser ajustável na aplicação web;
- O visual deverá ser rápido independentemente da dimensão da base de dados.

3. Confiabilidade/Fiabilidade

Determina a capacidade do sistema em lidar com eventos inesperados, isto é, é a probabilidade de o software não causar uma falha num sistema durante um determinado período sob condições específicas.

- Após uma falha, o lado do servidor deve ser capaz de continuar disponível e funcional.

4. Suportabilidade

Estes requisitos contem as principais características: manutenção, testabilidade, configurabilidade, adaptabilidade, nível de suporte e compatibilidade.

- O sistema deve ter suporte para diferentes plataformas (*mobile* e *web*);
- O sistema deve estar apto para vários idiomas.

4.7 Modelo Relacional

No Anexo 2 encontra-se representado o modelo relacional gerado da nova solução. Este modelo contém algumas entidades que são importantes para o desenvolvimento do projeto, e que devem ser mencionadas, tais como, **AlertasGerados**, **Percurso**, **EstadoAtual**, **Coordenadas**.

A entidade **AlertasGerados** é responsável por armazenar os dados do registo do alerta que o cliente pretende efetuar, nomeadamente, o id do seu veículo, a data e a hora que pretende que esse alerta seja gerado, o tipo de alerta (estado da ignição e/ou excesso de velocidade), os parâmetros e o respetivo email do cliente.

Em relação à entidade **Percurso**, esta é responsável por armazenar os percursos efetuados por um veículo através do evento de ignição (ligado/ desligado).

O **EstadoAtual** é uma entidade que regista a posição (coordenadas geográficas) atual do veículo.

Por fim, a entidade **Coordenadas** tem um papel fundamental no funcionamento da plataforma, uma vez que regista todos os dados geográficos (latitude, longitude, altitude, velocidade, data hora e estado de ignição) vindos do equipamento de localização.

5 Implementação da solução

Na fase de implementação são utilizados os artefactos produzidos nas secções anteriores, para a implementação do sistema. Os principais artefactos que resultam desta fase é o código fonte da plataforma e da camada de receção, bem como a configuração do equipamento de localização.

Neste capítulo são demonstrados e explicados excertos do código relativos às várias camadas da plataforma para demonstração dos padrões utilizados, bem como é exibida e descrita a UI da plataforma do administrador e do cliente e por fim é feito testes relativos à solução desenvolvida.

5.1 Equipamento de Localização

O equipamento FMB920 da empresa Teltonika foi escolhido pela GisGeo para ser integrado para testar a nova solução *Geocar Lite* por se mostrar ser uma mais-valia para o serviço devido aos protocolos CAN que implementa e a sua relação qualidade/preço. O FMB920 é um rastreador compacto e inteligente com conectividade *Bluetooth*, antenas internas GNSS e GSM de alto ganho e contém uma bateria de *backup*, no qual permite que o dispositivo funcione sem fonte de energia externa.

O FMB920 é um equipamento capaz de determinar a sua posição, velocidade e direcção através de GPS, obter dados de sensores externos por meio de entradas de sinais digitais e analógicos e consegue efetuar leitura de dados do barramento CAN da ECU (*Engine Control Unit*) do veículo através de *Bluetooth*. O equipamento recorre à rede GSM/GPRS para transmitir informações e dados recolhidos sendo que este também permite, através do seu software, configurar os dados pretendidos ou atualizar as suas configurações e *firmware*. (Figura 30)



Figura 30 – Equipamento de Localização FMB920

5.1.1 Componentes e Características

O FMB920, ilustrado na Figura 30, é uma pequena caixa com 1.2 x 7.9 x 4.3 cm e aproximadamente 54g que pode operar a temperaturas compreendidas entre -40°C e 85°C com consumos inferiores a 25mA a uma tensão de 12V ou inferiores a 6 mA caso o equipamento esteja em modo *sleep*. É importante referir que contém uma antena GNSS que permite a leitura de vários sistemas de navegação por satélite, tais como, GPS, GLONASS, GALILEO, BEIDOU, SBAS, QZSS, DGPS, no qual esta antena tem uma precisão de 2,5 metros e com uma sensibilidade de rastreamento de -165 dBm.

Em relação à interface, este é constituído pelos seguintes componentes (Figura 31):

- **Uma entrada digital (fio amarelo)**, que permite monitorizar o estado de ignição, o sensor da porta e o sensor do cinto de segurança;
- **Uma entrada analógica (fio cinzento)**, que corresponde ao sensor analógico de combustível, ao sensor das portas, bem como ao botão de emergência;
- **Uma saída digital (fio cinzento-claro)**, que corresponde ao *engine control*, ao *starter control* e também ao *remote switch*;
- **Antenas internas GNSS e GSM** de alto ganho, contém uma **bateria de backup integrada**, permite ainda a colocação de um **cartão Micro-SIM** e por fim contém uma memória interna de 128MB.
- **Suporte para a colocação de um MicroSD**, que permite armazenar os dados geográficos do veículo. [65]

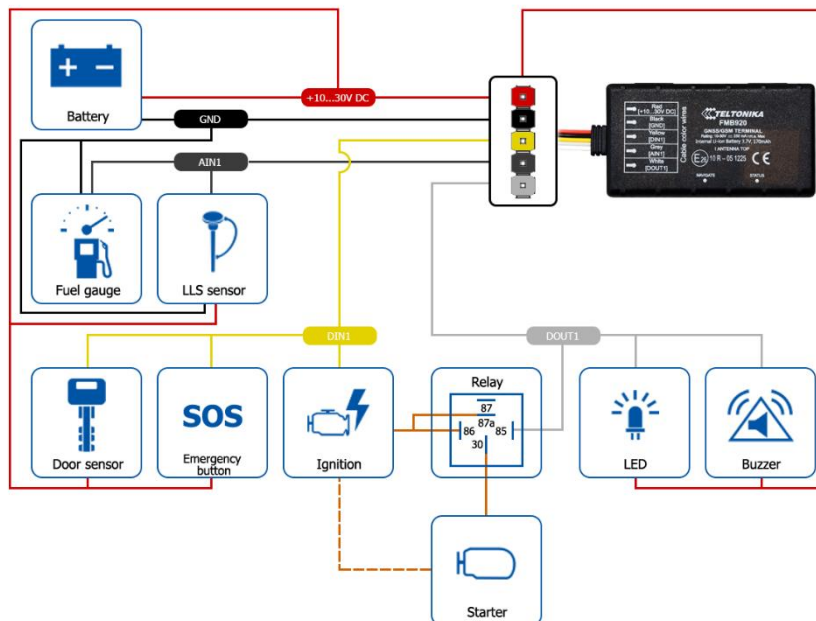


Figura 31 – Esquema do Equipamento FMB920

5.1.2 Configuração do Equipamento

Como já foi referido, foi utilizado para testes o equipamento de localização o **Teltonika FMB920** e para a sua configuração foi necessário utilizar um software próprio da marca e instalar a respetivos drivers no próprio computador. Inicialmente foi necessário colocar um cartão SIM no dispositivo, de modo o servidor interno conseguisse comunicar com ele através da tecnologia GPRS. De seguida é necessário configurar os I/Os (estado da ignição (*on/off*), velocidade, latitude, longitude, altitude, data hora) que se pretende utilizar. Para além disso é necessário colocar o endereço de ip do servidor onde está alojado a camada receção numa porta específica (55699) e seleccionar um protocolo (UDP/TCP). É importante referir que também é necessário estabelecer um período, no qual os dados são enviados pelo equipamento para o servidor enquanto o veículo se encontra em movimento e/ou parado. Na Figura 32 encontra-se ilustrado um exemplo do configurador do equipamento.

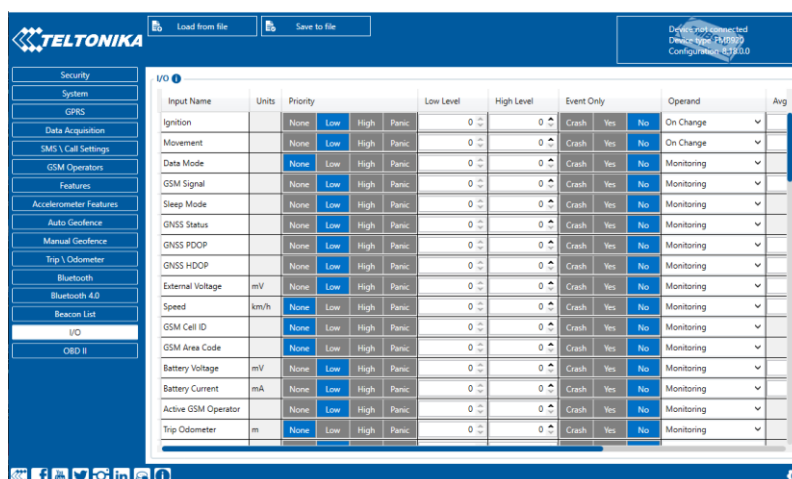


Figura 32 – Configurador do equipamento

Para finalizar esta configuração é necessário enviar comandos com uma determinada nomenclatura através de SMS para o equipamento de localização, de modo a enviar determinados parâmetros, tais como, ip e porta do servidor, tempo de transmissão do movimento do veículo, tempo de transmissão do veículo parado, alteração do ângulo efetuado pelo veículo, o limite de velocidade, ativar aceleração e/ou travagem e/ou curva acentuada. A alteração dos parâmetros faz-se através do comando *setparam*. É possível alterar mais do que um parâmetro de configuração ao mesmo tempo acrescentando ponto e vírgula (“;”), como, por exemplo [66]:

Sintaxe: *setparam* <num_parametro>:<valor>;<num_parametro>:<valor>

5.2 Descrição da implementação

A implementação desta solução é composta pelos componentes principais: a **Camada de Receção**, o **Back-End** e o **Front-End**. Como já foi referido na secção 4.1, a camada de receção foi desenvolvida em C# e consiste em compilar e traduzir os dados vindo do equipamento de localização e que por sua vez são inseridos na base de dados (PostgreSQL). Relativamente ao **Back-End** é constituído pelo *Model* e *Controllers*, e foram desenvolvidos usando a linguagem *NodeJs*. Por fim em relação ao **Front-End** a sua implementação foi feita utilizando a *framework* Angular que é baseado em *Typescript*, HTML5 e CSS3 (cf. secção 4.1).

5.2.1 Implementação da Camada de Receção-High Level

O package **High Level** da Camada de Receção, como já foi referido, é responsável por tratar os dados em relação ao veículo, nomeadamente, a configuração dos alertas (velocidade e estado da ignição), o registo dos percursos efetuados, bem como o controlo dos dados disponíveis na base de dados (dados dos últimos 3 meses).

De seguida é descrito alguns excertos de código utilizado nas principais classes deste *package* da Camada de Receção.

5.2.1.1 GeoAlerta

Na Figura 33 encontra-se um excerto da implementação do gerar alertas e processá-los, bem como após essa operação os mesmos serão enviados por email para o cliente, de modo a terem conhecimento do mesmo. Como é possível visualizar na figura abaixo é selecionado um veículo específico e de seguida é criado um corpo de email com os dados do cliente, no qual serão enviados. Para além disso é atualizado na tabela AlertasGerados os alertas criados (excesso de velocidade e/ou estado da ignição) e também a data e hora que foram espoletados.

```
public Boolean process()
{
    try {
        // Get email by vehicle owner
        DataRow _result = new DB().QUERY("SELECT ve.matricula, ut.email FROM Veiculo AS ve LEFT JOIN Utilizador AS ut ON ve.utilizadorid=ut.id WHERE ve.id="
        + idVeiculo + " AND ve.ativo=TRUE AND ut.ativo=TRUE;").Tables[0].Rows[0];
        String plate = _result[0].ToString(); if (String.IsNullOrEmpty(plate)) return false;
        String email = _result[1].ToString(); if (String.IsNullOrEmpty(email)) return false;

        // Create message & send email
        String alert_description = (type_alert == Type_Alert.ignicao) ? "Ignição" : "Velocidade";
        String subject = "GeocarLite - Alerta de " + alert_description;
        String body = "<b>Alerta de " + alert_description + "</b><br />Viatura " + plate + "<br />" + ((type_alert == Type_Alert.ignicao) ? "Foi ligada em "
        + Utils.ToString(new DateTime(timestamp)) : "Circulou a uma velocidade excessiva de " + parametros + "km/h em " + Utils.ToString(new DateTime(timestamp)));
        Boolean sended = Utils.send_email(email,subject,body); if (!sended) return false;

        // Update on database
        new DB().COMMAND("UPDATE AlertasGerados SET enviado=TRUE WHERE veiculoId=" + idVeiculo + " AND data_hora=" + timestamp +
        " AND tipo_alerta=" + ((int)type_alert) + ";");
        return true;
    } catch (Exception) { return false; }
}
```

Figura 33 – Implementação de processar os alertas

5.2.1.2 GeoData

Nesta classe contém os dados que são recolhidos pelo dispositivo de localização, tais como, a data e hora, o IMEI, o id do evento, a latitude, a longitude, a altitude, o ângulo, a velocidade, nº de satélites, o HDOP (Diluição da Precisão Horizontal) e por fim o IO correspondente. Em relação aos eventos despoletados pelos dispositivos de localização, estes poderão ser configurados através do *software* próprio do dispositivo ou poderão ser configurados na camada de receção. Estes eventos contêm diferentes estados, como, por exemplo, verificar a existência do evento de ignição, o movimento, a distância percorrida ou o ângulo de trajetória do veículo.

Em relação aos IOs, estes correspondem aos *Inputs* e *Outputs* vindos dos dispositivos de localização, tais como, a Ignição, o odómetro, a voltagem da fonte alimentação e a velocidade, em que estes contêm valores específicos correspondentes a cada IO, como, por exemplo, o IO da ignição corresponde ao valor 100. É importante salientar que estes IOs estão configurados no dispositivo de localização e variam de marca para marca.

5.2.1.3 GeoProcessor

Esta classe é importante para o funcionamento dos alertas da camada da receção, uma vez que é a partir desta que é processado os alertas de velocidade e de ignição de cada veículo, como se pode verificar na Figura 34. Para além disso, verifica-se ainda quais os veículos que estão associados a um determinado equipamento através de um IMEI, de modo a processar os respetivos alertas. É importante referir que para além do que foi referido, também permite atualizar o estado do veículo (ativa/desativa), bem como atualizar o IMEI do equipamento que está associado a um veículo.

```
public override void run()
{
    while (true) {

        // Get next GeoData
        GeoData data = pop();
        if (data == null) { Thread.Sleep(1000); continue; }

        // Process data
        int idx = index(data.Imei);
        if (idx >= 0) {
            // Alerts process
            process_alertOnOff(1stViaturaas[idx], data);
            process_alertSpeed(1stViaturaas[idx], data);
            // Insert data on bd
            1stViaturaas[idx].set(data);
        }
    }
}
```

Figura 34 – Excerto da implementação do GeoProcessor

5.2.1.4 GeoTask

Na Figura 35 encontra-se descrita a implementação do controlo das coordenadas armazenadas na base de dados, no qual as mesmas serão guardadas até 3 meses, de modo a não utilizarem muitos recursos, nomeadamente o espaço em base de dados fazendo com que a plataforma não perca desempenho e não implique custo adicionais. Como é possível observar é utilizado um método *execute ()* que contém um comando *DELETE* para eliminar as coordenadas registadas pelo dispositivo de localização, no qual é executado de três em três meses.

```
public class GeoTask : GeoTaskRun
{
    private int Nr_meses = 3;

    0 references | ManuelPinheiro, 248 days ago | 1 author, 2 changes
    public GeoTask(int nr_meses) : base(5000) {
        //Hour = hour;
        Nr_meses = nr_meses;
    }

    2 references | ManuelPinheiro, 248 days ago | 1 author, 2 changes
    public override void execute()
    {
        DB db = new DB();
        DateTime now = DateTime.Now;
        DateTime data_hora = new DateTime(now.Year, now.Month, 1, 1, 0, 0).AddMonths(Nr_meses*(-1));
        // DateTime data_hora = DateTime.Now.AddMonths(Nr_meses * (-1));

        int deleteData = db.COMMAND("DELETE FROM coordenadas WHERE data_hora < " + data_hora.ToString("yyyy-MM-dd HH:mm:ss") + "");
        if (deleteData>0)
        {
            Console.WriteLine("Dados foram apagados!!");
        }
        Console.WriteLine("Estou a iniciar!!");
    }
}
```

Figura 35 – Implementação da classe GeoTask

5.2.1.5 GeoVeiculo

Nesta classe é feito uma manipulação dos dados vindo do GeoData correspondente aos veículos que se encontram associados aos IMEI, no qual é registado os **percursos** efetuados pelos veículos através do evento de ignição, bem como é registado todas as **coordenadas geográficas** dos veículos e também é registado a **posição atual** de cada veículo do cliente. Na Figura 36 encontra-se um excerto da implementação desta classe que mostra como é verificado o alerta do estado de ignição e o alerta de velocidade. Em relação ao alerta de ignição é verificado se o estado do mesmo se encontra a *true* ou *false*, e através desse método auxiliar é utilizado na classe GeoAlerta. Relativamente ao alerta de velocidade, o mesmo é registado através de um número de velocidades registadas (5 dados) na base de dados, em que de seguida é feita uma média dessas velocidades e compara com o valor inserido pelo cliente.

```

// Alert Checks
1 reference | ManuelPinheiro, 10 days ago | 1 author, 1 change
public Boolean checkOnOff(int ignition, long timestamp)
{
    // Pré-Conditions
    if (!alerta_on) return false;
    if (last_on == 0) return false;
    if ((timestamp - last_update) > time_speed) return false;

    // Check transition off to on
    return last_on == GeoVeiculo.off && ignition == GeoVeiculo.on;
}
1 reference | ManuelPinheiro, 10 days ago | 1 author, 1 change
public Boolean checkSpeed(double speed, long timestamp)
{
    // Pré-Conditions
    if (alert_speed <= 0) return false;
    if (last_speed.Count < nr_speed) return false;
    if ((timestamp - last_update) > time_speed) return false;

    // Check speed avg
    return avg_speed(speed) > alert_speed;
}
2 references | ManuelPinheiro, 10 days ago | 1 author, 1 change
public double avg_speed(double speed)
{
    if (last_speed.Count == 0) return speed;

    double t = speed;
    foreach (double sp in last_speed) t += sp;
    return t / (last_speed.Count + 1);
}

```

Figura 36 – Verificar os alertas de velocidade e de ignição

5.2.1.6 ConfigServer

Nesta classe é feita a criação de uma *thread* para o protocolo UDP e para o protocolo TCP, de modo a receber dados da parte do *backend* através de uma porta específica. Para além disso é feito um método auxiliar que permite verificar no *backend* as funções que se pretende fazer, como, por exemplo, caso pretenda criar um veículo é necessário enviar do *backend* uma sintaxe do tipo “set; id_veiculo; imei” que faz com que no GeoProcessor se efetue as operações pretendidas. Na Figura 37 encontra-se o método de diferentes operações (*insert*, *update* e *delete* e *config_alert*) na camada de receção.

```

switch(_parts[0].ToLower()) {
    case "del": //del;vehicle
        if (_parts.Length != 2) throw new Exception("Invalid parameters for command DEL");
        Console.WriteLine("RECEBI O COMANDO!!!!!-->" + _command);
        return geoProcessor.del(long.Parse(_parts[1])) ? "done" : "not done";

    case "set": // set;vehicle;imei
        if (_parts.Length != 3) throw new Exception("Invalid parameters for command SET");
        Console.WriteLine("RECEBI O COMANDO!!!!!-->" + _command);
        return geoProcessor.set(long.Parse(_parts[1]), long.Parse(_parts[2])) ? "done" : "not done";

    case "config_alert": // config_alert;vehicle;ai:true|false or config_alert;vehicle;as;0|120
        if (_parts.Length != 4) throw new Exception("Invalid parameters for command CONFIG");
        return geoProcessor.set_alert(long.Parse(_parts[1]), _parts[2], _parts[3]) ? "done" : "not done";

    case "config_alerts": // config_alerts;vehicle;ai:true|false;as;0|100
        if (_parts.Length != 4) throw new Exception("Invalid parameters for command CONFIG_ALERT");
        return geoProcessor.set_alerts(long.Parse(_parts[1]), _parts[2], _parts[3]) ? "done" : "not done";

    default: throw new Exception("Undefined command");
}

```

Figura 37 – Diferentes operações no ConfigServer

5.2.2 Implementação da Camada de Receção-Low Level

Relativamente ao package **Low Level** é direcionado ao dispositivo de localização, uma vez que é feito a tradução e a compilação dos dados de localização para o **High Level**. As classes mais importantes para este processo de tradução e compilação são designadas por *PTeltonika*, *TransportUDP* e *ReceiveServer*.

De seguida é descrito um excerto da implementação da classe *ReceiveServer* deste package.

```
ReceiveServer _me = this;

// UDP (IPv4) protocol
listenerUDP = new UdpClient(new IPEndPoint(IPAddress.Any, portReceiver));
listenerUDP.Client.SetSocketOption(SocketOptionLevel.Socket, SocketOptionName.ReuseAddress, true);
udp_Thread = new Thread(() => {
    while (true) {
        try {
            IPEndPoint clientUDP = new IPEndPoint(0, 0);
            byte[] bytes = listenerUDP.Receive(ref clientUDP);
            Transport transUDP = Transport.UDP(ref listenerUDP, ref clientUDP, bytes);
            Processor.RUN(ref _me, ref transUDP);
        } catch (Exception ex) { Console.WriteLine("ERROR :: Invalid udp connection detected", ex); }
    }
}); udp_Thread.Start();
Console.WriteLine("Start listener, port " + portReceiver + " (UDP) for " + Processor.parseTP(processorReceiver) + " equipments.");
```

Figura 38 – Excerto da implementação do *ReceiveServer*

Como se pode analisar na figura acima (Figura 38), é feito a execução de uma *thread* utilizando o protocolo UDP numa porta específica recebendo dados em bytes do componente *Teltonika*.

5.3 Implementação Back-End da Plataforma Web

O **Back-End** da plataforma é um componente que trata de toda a lógica de negócio (*Models* e *Controllers*). Aqui é feito o tratamento dos dados que constitui as entidades principais da solução, bem como é o local onde ocorrem as leituras, inserções e o armazenamento dos dados numa base de dados configurada.

Esta secção contém alguns excertos de código utilizado nos principais componentes da plataforma, particularmente, no componente *Model* e no componente *Controller*. De seguida é exposto alguns excertos da implementação para as funcionalidades mais pertinentes da plataforma.

5.3.1 Implementação da criação do veículo

Esta é uma funcionalidade bastante importante para o funcionamento do sistema, uma vez que para mostrar os veículos no mapa é necessário criá-los, colocando os dados necessários (cliente, marca, matrícula, combustível) e associá-los a um equipamento através do seu IMEI e número de telefone.

No *Model* encontra-se o construtor da classe Veículo e a respectivas *queries* para guardar o veículo na base de dados e mostrar todos os equipamentos disponíveis. No componente *controller* tem como função enviar os dados necessários para a camada de recepção através de uma *socket* cliente, conforme a Figura 39. Para uma melhor compreensão deste caso de uso foi feito o diagrama de sequência, conforme a Figura 59.

```
exports.createVeiculo = async (req, res, next) => {
  const errors = validationResult(req.body);
  if (!errors.isEmpty()) return next();
  try {
    const veiculo = {
      utilizadorid: req.body.utilizadorid,
      matricula: req.body.matricula,
      marca: req.body.marca,
      imei: req.body.imei,
      combustivel: req.body.combustivel,
    };
    const veiculoId = await (await Veiculo.create(veiculo)).rows[0].id;
    console.log(veiculoId);
    await sendDataSocket(
      "set;" + veiculoId + ";" + veiculo.imei, 55800,
      "localhost", callback_set1, res, veiculoId);
  } catch (err) {
    if (!err.statusCode) {
      err.statusCode = 500;
    }
    next(err);
  }
};
```

Figura 39 – *Controller* do criar veículo

5.3.2 Implementação da visualização do percurso de um veículo

Na parte de visualização de um determinado percurso de um veículo é necessário o cliente colocar um intervalo de datas (data de início e data de fim), de modo a receber os percursos efetuados por esse veículo. Na Figura 40 mostra a *query* para procurar os percursos num determinado período. Para uma melhor compreensão deste caso de uso foi feito o diagrama de sequência, conforme a Figura 60.

```
static findPercurso(veiculoid, data_hora_inicio, data_hora_fim) {
  return db.query(
    "SELECT P.id, P.data_hora_inicio, P.data_hora_fim, P.veiculoid FROM Percurso P WHERE veiculoid=$1"+
    "AND data_hora_inicio IS NOT NULL AND data_hora_fim IS NOT NULL AND ((data_hora_inicio>=$2 AND data_hora_fim<=$3)"+
    "OR (data_hora_inicio>=$2 AND data_hora_fim<=$3) OR (data_hora_inicio<$2 AND data_hora_fim>$3))",
    [
      veiculoid,
      data_hora_inicio + " 00:00:00.000",
      data_hora_fim + " 23:59:59.999",
    ]
  );
}
```

Figura 40 – *Query* para encontrar o percurso de um veículo

Para além de procurar todos os percursos efetuados para um determinado veículo, também é possível procurar o percurso atual. Para isso é necessário procurar o percurso efetuado pelo veículo em que ainda não tenha data e hora de fim.

5.3.3 Implementação das coordenadas de um percurso

Na Figura 41 encontra-se descritos duas *queries*. A primeira tem como objetivo encontrar todas as coordenadas de um determinado percurso efetuado por um veículo selecionado pelo cliente. Por fim a segunda corresponde a todas as coordenadas do percurso atual de um veículo selecionado pelo cliente na plataforma.

```
//DESENHAR NO MAPA
static allCoordenadasByVeiculoDateInicioFim(percursoid){
  return db.query("SELECT latitude, longitude, speed FROM Coordenadas"+
    "WHERE percursoid=$1 ORDER BY data_hora ASC",[percursoid]);
}
static allCoordenadasByVeiculoPercursoAtual(veiculoId){
  return db.query("SELECT latitude, longitude, speed FROM Coordenadas"+
    "WHERE percursoid IN (SELECT P.id FROM Percurso P WHERE veiculoId=$1 AND data_hora_inicio"+
    "IS NOT NULL AND data_hora_fim IS NULL ORDER BY data_hora_inicio DESC LIMIT 1) ORDER BY"+
    "data_hora ASC",[veiculoId]);
}
```

Figura 41 – *Queries* para procurar as coordenadas de um percurso

5.3.4 Implementação dos alertas de velocidade e ignição

Na Figura 42 está descrito o *controller* do configurar os alertas com a camada de receção. Como é possível analisar é enviado os dados do veículo (id) e os respetivos alertas (*alerta_on_off*, *alerta_speed*) através de uma *socket* cliente utilizando o protocolo UDP numa porta específica. Para uma melhor compreensão deste caso de uso foi feito o diagrama de sequência, conforme a Figura 61.

```
exports.alertasVeiculo = async (req, res, next) => {
  const veiculoId = parseInt(req.params.id);
  const { alerta_on_off, alerta_speed } = req.body;
  try {
    await sendDataSocket(
      "config_alerts;" + veiculoId + ";" + alerta_on_off + ";" + alerta_speed,
      55800, "localhost", callback_alert, res, veiculoId
    );
  } catch (err) {
    if (!err.statusCode) {
      err.statusCode = 500;
    }
    next(err);
  }
};
```

Figura 42 – *Controller* do configurar os alertas

5.4 Interface gráfica

O essencial desta plataforma consiste em localizar os veículos em tempo real, bem como saber o estado da viatura (ignição, velocidade) num determinado período. Neste contexto exemplifica-se a forma como se processa na realidade com o recurso ao *Geocar Lite*.

Como se constata na Figura 43, e sabendo-se que o administrador é um dos intervenientes que tem a possibilidade de criar os veículos, os equipamentos e também os clientes, após a sua autenticação ficará disponível a sua página de acesso.

ID	Email	Nome	Data de Nascimento	Ações
41	teste.simov1@gmail.com	Maria Amelia de Carvalho	December 17, 2000	[X]
43	nelinho_1997@hotmail.com	Sonia Carvalho	April 18, 2022	[X]

Figura 43 – *Dashboard* do Administrador

De seguida, na plataforma o administrador pode gerir os equipamentos, no qual terá de preencher os dados necessários, nomeadamente o seu IMEI e o número de telefone associado, conforme a Figura 44.

ID	IMEI	Telefone	Nome do Cliente	Matrícula do Veículo	Ações
18	123456789882312	913950412			[✓] [X]
29	555555122222222	913911111	Sonia Carvalho	99-BN-84	[✓] [X]
30	333333333333333	966666666	Maria Amelia de Carvalho	49-81-VV	[✓] [X]

Figura 44 – Página de Gerir Equipamentos

Por fim, o administrador tem permissão de criar uma *password* para o cliente, bem como editar os seus dados, como também criar os seus veículos, associando aos equipamentos de localização, de modo que os clientes nas suas sessões (páginas) consigam visualizar as suas viaturas no mapa em tempo real. Na Figura 45 é possível observar um exemplo de uma página de um cliente com os seus respetivos veículos.

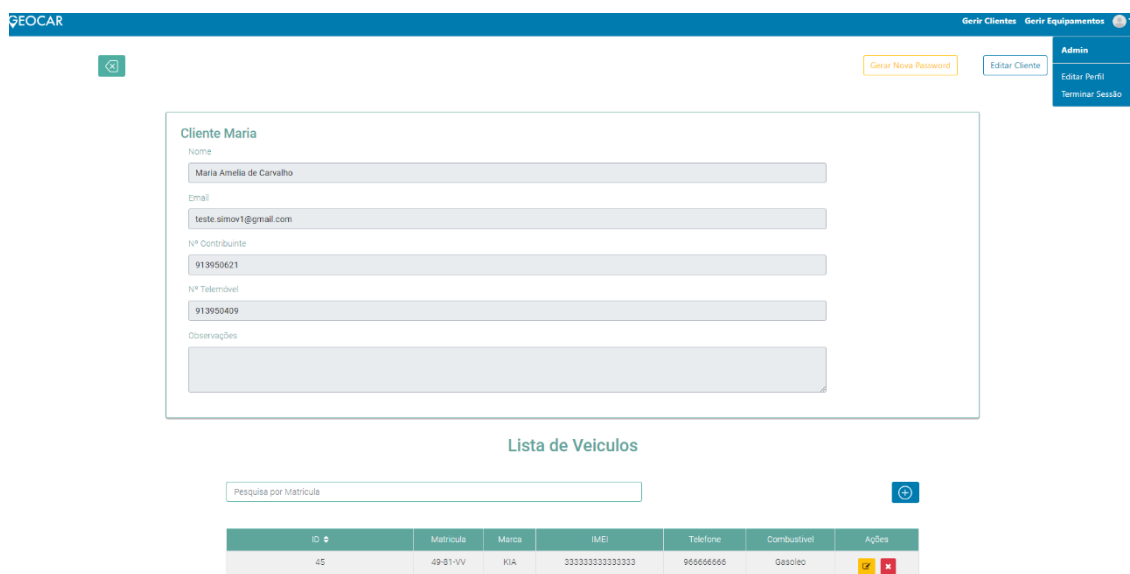


Figura 45 - Página de Edição do Cliente e criação de veículos

Em relação à página de acesso do cliente, a mesma contém um mapa com a localização dos seus veículos em tempo real. O cliente tem a possibilidade de selecionar um dos seus veículos e pode ver os seus detalhes (marca, matrícula, combustível), bem como configurar os alertas (velocidade e/ou ignição), ver os percursos efetuados e também exportar um relatório (ignição/percursos) para o email. Na Figura 46 mostra a página de acesso do cliente com os seus respetivos veículos no mapa em tempo real.

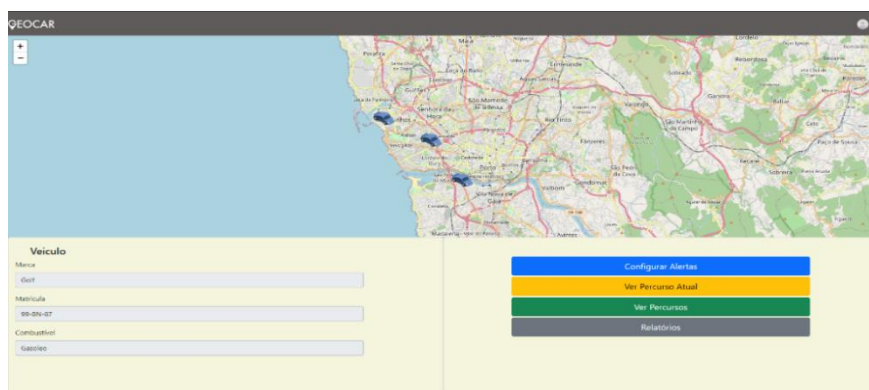


Figura 46 – Página do Cliente com opções

Caso o cliente pretenda visualizar o percurso atual de um dos seus veículos, basta clicar na opção “Ver Percurso atual” e na plataforma é possível observar o percurso feito pelo veículo, bem como os locais onde o mesmo excedeu a velocidade através da linha vermelha e os locais onde a velocidade foi controlada através da linha verde. Na Figura 47 encontra-se um exemplo de um percurso atual de um veículo selecionado pelo cliente.

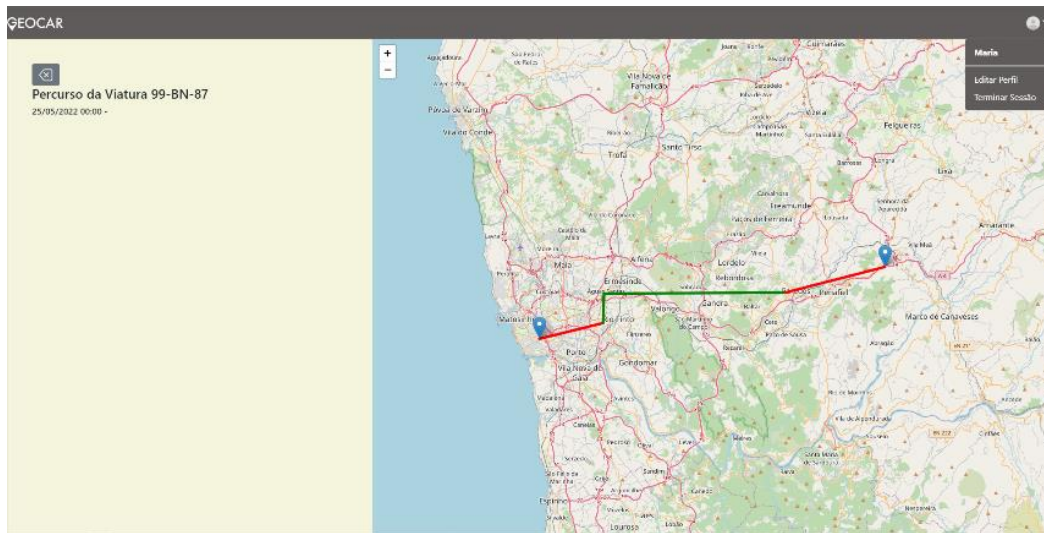


Figura 47 – Visualização do Percurso Atual

É importante salientar que para além da possibilidade anteriormente referida poderá ainda na *Geocar Lite* o cliente visualizar os percursos efetuados pelo veículo selecionado num período definido, bem como os locais onde excedeu a velocidade, como consta na Figura 48.

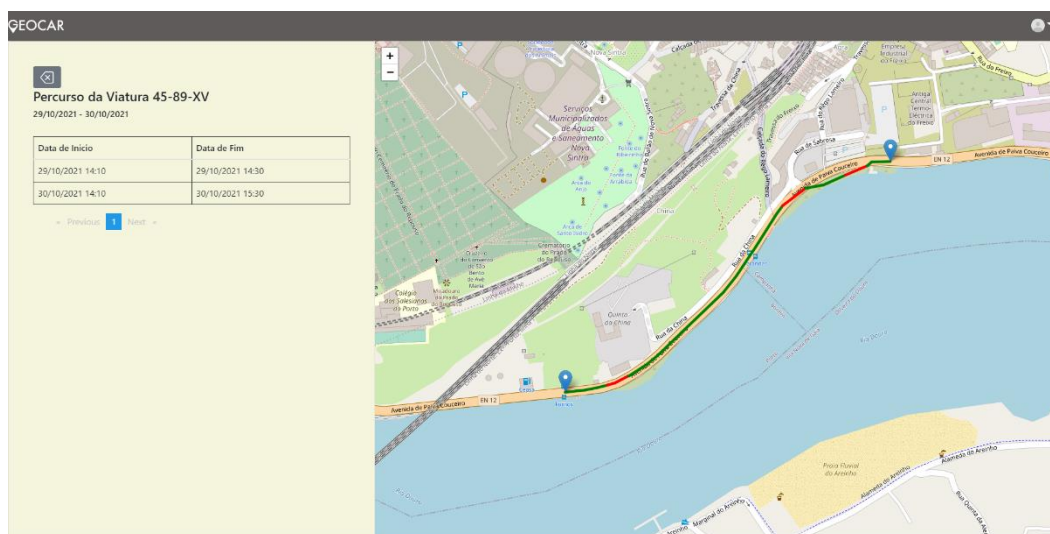


Figura 48 – Visualização dos Percursos

O cliente pode configurar os alertas de um veículo que se encontra no mapa. Para isso é necessário clicar no próprio veículo e de seguida selecionar o botão “Configurar Alertas” e depois selecionar os alertas (ignição/velocidade) que pretende que estejam ativos, estes serão despoletados para o email do cliente e no caso do alerta de velocidade tem a possibilidade de estabelecer um limite de velocidade, conforme a Figura 49.

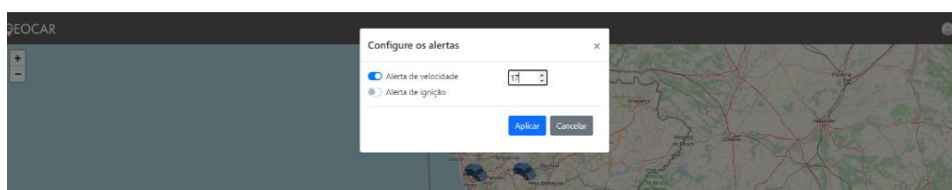


Figura 49 – Configurar Alertas

Por fim, o cliente pode gerar relatórios, isto é, é enviado para um email duas tabelas com os relatórios de percursos efetuados e a tabela do número de eventos de ignição do veículo num período específico, conforme a Figura 50.

teste.simov1@gmail.com para mim

Tabela com relatório de percursos efetuados pelo veiculo
 Parametros do relatório: Veiculo com a matricula 45-89-XV, no periodo de 2021-10-29 a 2021-10-31

Data de Inicio	Data de Fim	Duração	Velocidade Média (Km/h)	Velocidade Máxima (Km/h)	Velocidade Minima (Km/h)
30/10/2021 14:10:00	30/10/2021 15:30:00	1 hour, 20 minutes	68.28571428571429	121	0
29/10/2021 14:10:00	29/10/2021 14:30:00	20 minutes	69.63636363636364	125	0

Tabela com relatório de ignição dos percursos efetuados pelo veiculo
 Parametros do relatório: Veiculo com a matricula 45-89-XV, no periodo de 2021-10-29 a 2021-10-31

Data do Evento
29/10/2021 14:10:00
30/10/2021 14:10:00

Nº de eventos de ignições do periodo: 2

Figura 50 – Relatório dos percursos e de ignição

5.5 Testes

A fase de testes no processo de desenvolvimento de software é essencial. Estes são importantes e tem como objetivo validar a plataforma em relação aos requisitos de negócio obtidos através da especificação funcional e complementados pela especificação técnica.

Foram feitos testes para garantir o bom funcionamento da nova solução, de modo a verificar aspetos pertinentes, tais como, a qualidade de software, o desempenho da plataforma e também a capacidade que esta pode suportar para uma determinada quantidade de dados.

5.5.1 Testes de Integração

Os testes de integração consistem em testar dois ou mais componentes da aplicação que funcionam em conjunto com o intuito de produzir um resultado esperado. Normalmente é utilizado determinado componente, tal como, a base de dados ou sistema de ficheiros ou dispositivos de rede para verificar o sucesso da solicitação. [67]

De seguida na Figura 51 mostra os testes de integração aos componentes da solução a executarem com sucesso. Estes testes verificam se os pedidos HTTP (POST/GET/PUT/DELETE) de cada entidade funcionam corretamente e se existia alterações na respetiva base de dados.

```
Node js + Postgres Testing API Coordenadas
✓ should get all coordenadas (96ms)
✓ Should get all Coordenadas by Veiculo and Data_hora
✓ Should get all Coordenadas by Percurso Id
✓ Should get all Coordenadas by Veiculo

Node js + Postgres Testing API Equipamentos
✓ Should get all equipamentos
✓ Should get equipamento by ID
✓ Should get equipamento by veiculoID
✓ Should create equipamento

Node js + Postgres Testing API Percurso
✓ should get percurso by veiculo and data_hora_inicio e data_hora_fim
✓ Should get percurso atual

Node js + Postgres Testing API Users
teste.simov1@gmail.com
qwerty
✓ should get all users
✓ should get all clientes
✓ should get utilizador by id

Node js + Postgres Testing API Veiculos
✓ Should get all veiculos
✓ Should get veiculo by ID
✓ Should get veiculo by User ID
✓ Should create veiculo

17 passing (395ms)
```

Figura 51 – Testes de Integração à plataforma

5.5.2 Testes de Desempenho

Os testes de desempenho é um tipo de teste de software que ajuda a determinar o desempenho de um software em termos de velocidade, tempo de resposta, escalabilidade, uso de recursos e escalabilidade sob uma determinada carga de trabalho. O objetivo dos testes de desempenho é identificar os períodos onde ocorreu um baixo desempenho de uma determinada plataforma. [68]

Posto isto, para verificar os testes de desempenho à plataforma *Geocar Lite* utilizou-se o *software JMeter Apache* para analisar os pedidos que eram efetuados à base de dados, e verificar os respetivos comportamentos com o auxílio de um gráfico.[69]

Inicialmente para a realização do teste colocou-se determinados parâmetros, nomeadamente, o número de *threads* com o valor a 50, o número de vezes que o plano de testes será executado (*Loop Count*) com o valor a 3 e o período de teste (*Ramp-Up Period*) com o valor a 90. Depois foi inserido os *requests* que se encontram na API da plataforma, como se pode verificar na Figura 52. Por fim foi gerado um gráfico a partir desses *requests* efetuados, como está ilustrado na Figura 53, que contém quatro linhas coloridas:

- **Preta:** O número total de amostras atuais enviadas;
- **Azul:** A media atual de todas as amostras enviadas;
- **Vermelha:** O desvio-padrão atual;
- **Verde:** Taxa de transmissão que representa o número de pedidos por minuto que o servidor tratou.

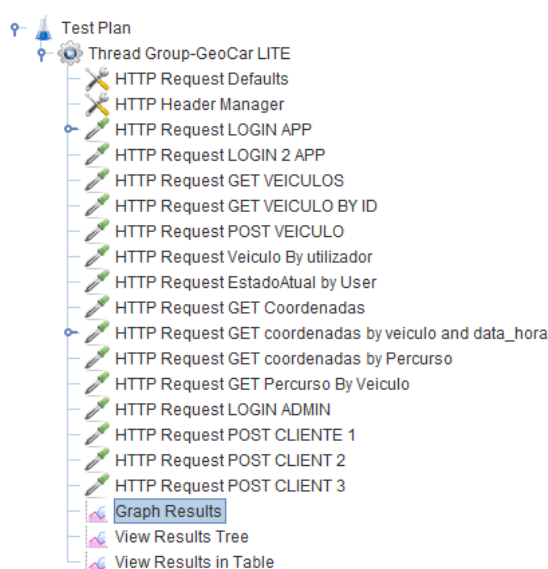


Figura 52 – Requests inseridos

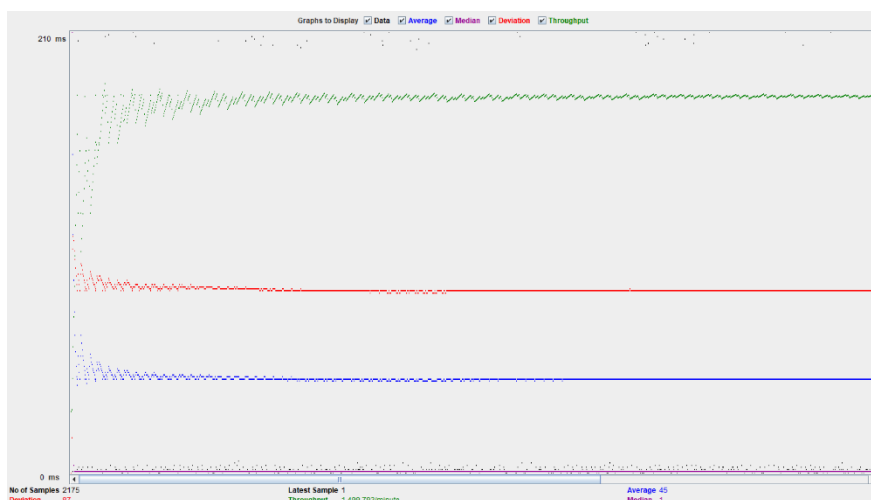


Figura 53 – Gráfico de teste de desempenho

Na Figura 53, encontra-se representado o gráfico de desempenho à plataforma Geocar-Lite, no qual para a análise do desempenho à API, deve-se ter em atenção dois parâmetros importantes, nomeadamente à Taxa de Transferência (linha verde) e ao Desvio (linha vermelha).

A Taxa de Transferência é o parâmetro mais importante, no qual representa a capacidade da API de lidar com uma carga mais pesada. Quanto maior for a Taxa de Transferência, melhor é o desempenho do servidor.

Neste teste, o rendimento da API é de 1499,793/minuto. Isto significa que API pode processar 1499,793 pedidos por minuto. Este valor é bastante elevado para o número de *threads* inseridos, em que se pode concluir que API tem um bom desempenho. Em relação ao desvio, verifica-se que tem um valor favorável, uma vez que não existe grande discrepância em relação à média.

Para além de verificar o desempenho da plataforma, também se verificou o comportamento da mesma quanto ao tempo de resposta da interface gráfica em relação à visualização dos veículos em tempo real utilizando o *Google Chrome Dev Tools*.

6 Avaliação e Experimentação

Este capítulo apresenta o processo de avaliação e experimentação do projeto desenvolvido e os seus resultados.

Inicialmente, serão apresentadas as métricas definidas para avaliar a solução. Em seguida, estão descritas as hipóteses formuladas. No final do capítulo, é apresentado a metodologia utilizada e os resultados da avaliação.

6.1 Métricas

Para uma avaliação global da solução foram consideradas três métricas:

- **Métrica 1:** Satisfação do utilizador quanto à utilização da plataforma;
- **Métrica 2:** Desempenho e tempo de resposta da plataforma;
- **Métrica 3:** Custo obtido pela nova solução em relação à solução atual.

6.2 Hipóteses

Posteriormente à definição das métricas, segue-se a formulação de hipóteses. Assim, por cada métrica foram formuladas 2 hipóteses.

6.2.1 Métrica 1

- **Hipótese 1 (H_0):** A usabilidade da plataforma tem um nível de satisfação superior a 3 numa escala de 0 a 5.
- **Hipótese 2 (H_1):** A usabilidade da plataforma tem um nível de satisfação igual ou inferior a 3 numa escala de 0 a 5.

6.2.2 Métrica 2

- **Hipótese 1 (H_0)**: O sistema demora menos 1 segundo a responder.
- **Hipótese 2 (H_1)**: O sistema demora mais de 1 segundo a responder.

6.2.3 Métrica 3

- **Hipótese 1 (H_0)**: O custo obtido teve uma redução superior a 50% em relação à solução atual.
- **Hipótese 2 (H_1)**: O custo obtido não teve uma redução inferior a 50% em relação à solução atual.

6.3 Metodologia

Definidas as métricas e as suas hipóteses é altura de avaliá-las. Para desempenhar essa avaliação três metodologias distintas serão utilizadas. Para avaliar a Métrica 1 foram realizados inquéritos de satisfação, no qual o primeiro é relativamente à plataforma do administrador e outro em relação à plataforma do cliente. O primeiro inquérito foi disponibilizado à pessoa responsável pelo apoio técnico e à equipa de desenvolvimento e o outro foi disponibilizado só à equipa de desenvolvimento. Apesar de a equipa de desenvolvimento da empresa não ser os principais utilizadores da plataforma da parte do cliente, no entanto conseguem avaliar o impacto da solução no que diz respeito à usabilidade da plataforma. Em relação à Métrica 2 foram realizados testes de integração e testes de desempenho, de modo a avaliar o desempenho e o tempo de resposta da plataforma. Por fim relativamente à Métrica 3, verificou-se o custo que se obteve no seu desenvolvimento em relação à solução atual da GisGeo.

6.4 Análise de Resultados

Aqui é realizada uma análise dos resultados obtidos para cada uma das métricas anteriormente definidas. Além disso, é aferido se a plataforma cumpre com os casos definidos nas hipóteses da secção 6.2.

6.4.1 Inquérito de satisfação do utilizador

De modo a aferir se a plataforma corresponde ao que foi pedido para empresa GisGeo, foi realizada uma apresentação da plataforma e das suas funcionalidades à equipa e depois foi pedido que esta respondesse a dois pequenos questionários. O primeiro questionário foi relativamente à usabilidade da plataforma por parte do cliente (Anexo 5), no qual foi respondido pela equipa de desenvolvimento, e o outro inquérito corresponde à usabilidade da

plataforma por parte do administrador (Anexo 4), onde foi respondido pela equipa de apoio técnico e a equipa de desenvolvimento.

Estes questionários apresentam uma escala de satisfação entre 0 e 5, onde 0 é **INSATISFATÓRIO** e 5 é **BASTANTE SATISFATÓRIO**.

Feedback da usabilidade da plataforma por parte do cliente

O questionário composto por 8 asserções apresentadas na tabela com o valor percentual de respostas para cada valor da escala anteriormente definida.

Tabela 21 – Tabela de Requisitos Funcionais (cliente)

Asserções	0	1	2	3	4	5
Interface gráfica	0%	0%	0%	28,6%	71,4%	0%
Visualização dos veículos em tempo real	0%	0%	0%	42,9%	57,1%	0%
Processo de visualização dos percursos	0%	0%	0%	14,3%	85,7%	0%
Processo de notificações de alertas	0%	0%	14,3%	0%	85,7%	0%
Processo de visualização de relatórios	0%	0%	14,3%	57,1%	28,6%	0%
Tempo para completar as tarefas	0%	0%	0%	28,6%	71,4%	0%
Satisfação da utilização da plataforma	0%	0%	0%	14,3%	85,7%	0%
Recomendação do uso da plataforma	0%	0%	0%	14,3%	85,7%	0%

Feedback da usabilidade da plataforma por parte do administrador

O questionário composto por 14 asserções apresentadas na tabela com o valor percentual de respostas para cada valor da escala anteriormente definida.

Tabela 22 – Tabela de Requisitos Funcionais (administrador)

Asserções	0	1	2	3	4	5
Interface gráfica	0%	0%	0%	0%	50%	50%
Processo de criação de clientes	0%	0%	0%	0%	66,7%	33,3%
Processo de edição de clientes	0%	0%	0%	0%	66,7%	33,3%
Processo de eliminação dos clientes	0%	0%	0%	0%	66,7%	33,3%
Processo de criação de equipamentos	0%	0%	0%	0%	66,7%	33,3%
Processo de edição de equipamentos	0%	0%	0%	0%	66,7%	33,3%
Processo de eliminação dos equipamentos	0%	0%	0%	0%	66,7%	33,3%
Processo de criação de veículos	0%	0%	0%	16,7%	66,7%	16,7%
Processo de edição de veículos	0%	0%	0%	16,7%	66,7%	16,7%
Processo de eliminação dos veículos	0%	0%	0%	16,7%	66,7%	16,7%
Tempo para completar as tarefas	0%	0%	0%	16,7%	66,7%	16,7%
Fluidez da plataforma	0%	0%	0%	16,7%	66,7%	16,7%
Satisfação da utilização da plataforma	0%	0%	0%	0%	66,7%	33,3%
Recomendação do uso da plataforma	0%	0%	0%	0%	66,7%	33,3%

Tendo em consideração as hipóteses formuladas para esta Métrica e os resultados dos inquéritos, onde a média das respostas obtidas é de aproximadamente 4 (3,9), é possível afirmar que há uma satisfação geral na plataforma. Apesar disso, há certos aspetos a melhorar como o processo de gestão de veículos, bem como o processo de notificações de alertas e a visualização de relatórios (percursos e ignição).

Tempo de resposta e desempenho da plataforma

Para avaliar os tempos de resposta, os testes de integração e os testes de desempenho descritos 5.5.1 e 5.5.2, foram executados 10 vezes numa máquina com um processador Intel Core i5 5200U @2.20GHz, com a base de dados e a camada de receção alojados no próprio computador.

Tabela 23 – Tabela de tempo de resposta

Tempo médio (ms)	Menor tempo (ms)	Maior tempo (ms)
779,2	718	1168

Comparando com a hipótese formulada, o tempo médio deve ser menor que 1 segundos, o tempo médio de execução é de 779,2 milissegundos. Isto demonstra que o sistema cumpre com a hipótese definida ao respeitar o tempo limite de 1 segundo. É importante referir que este tempo de resposta varia consoante a capacidade de processamento do computador que foi utilizado.

Custo obtido pela *Geocar Lite* em relação à solução atual

Para avaliar o custo obtido pela *Geocar Lite* teve de se analisar três fatores importantes, nomeadamente, os dados que o dispositivo de localização consome por mês e o alojamento na *web* (plataforma *web* e a base de dados). Em relação aos dados, o dispositivo de localização (cartão SIM) não consome mais do que 1GB de dados por mês, em que 1GB custa pouco mais de 5€. Em relação ao alojamento na *web* (*cloud*) varia muito consoante a quantidade de dados que se pretende armazenar, bem como o desempenho da mesma. Assumindo que segundo os testes efetuados na base de dados (*PostgresSQL*) a quantidade de dados (coordenadas, veículos, utilizador, equipamentos, percursos) que se obteve por dia de um utilizador com um veículo obteve-se **608kb**. Uma vez que a plataforma está preparada para guardar os dados dos últimos 3 meses (90 dias) e considerando que um utilizador contém 25 veículos estima-se que o tamanho será:

$$\text{Armazenamento}_{3\text{meses}} = 90 * 1 * 608\text{kb} * 25 \approx 152000\text{kb}$$

Caso no futuro a plataforma contenha 200 utilizadores estima-se que o tamanho obtido serão:

$$\text{Armazenamento}_{3\text{meses}} = 152000\text{kb} * 200 = 273600000\text{ kb} \approx 273,6\text{Gb}$$

Uma vez que o tamanho obtido pelos 200 utilizadores durante 3 meses será aproximadamente 273,6 Gb, analisou-se as características ideais para uma *cloud*, segundo Amen [70], e concluiu-se que a mesma terá de conter as seguintes características:

- 4 vCPU;
- 8 Gb de RAM;
- 400GB de SSD;
- Tráfego: 3TB/m;
- 1 IPv4;
- Certificado SSL: *Positive SSL DV Certificate*.

O valor aproximado para o alojamento (*cloud*) será 134,86€ por mês, ou seja, num ano seria 1 618,32€.

Podemos concluir, caso o dispositivo em 3 meses consuma 3Gb de dados, o valor a cobrar será aproximadamente 15€ e o valor anual (valor com descontos) para o alojamento será 1 618,32€ por ano.

$$Valor_{final} = 1618,32€ + (4 * 15€) \approx 1678,32€$$

O valor final por ano será aproximadamente 1678,32€. Comparando o valor obtido pela manutenção da *Geocar Lite* e o valor do custo da manutenção da solução atual da *GisGeo*, verifica-se que a hipótese definida (H_0) foi cumprida.

7 Conclusão

No presente capítulo, são apresentados os resultados obtidos no desenvolvimento do projeto de estágio. É feito um resumo da solução desenvolvida, que engloba as suas características e funcionalidades. De seguida são expostos os objetivos atingidos de entre os propostos no início do projeto. Posteriormente, serão referenciadas as limitações encontradas no desenvolvimento e descrição do trabalho a realizar no futuro. Por fim, será dada uma apreciação final sobre o trabalho desenvolvido.

7.1 Objetivos alcançados

Os objetivos finais deste projeto tinham como finalidade o desenvolvimento de uma plataforma que implementasse os requisitos propostos pela empresa que se encontram descritos nas secções dos Requisitos Funcionais e nos Objetivos. Contudo, e tendo em consideração os requisitos funcionais e não funcionais é de seguida descritos os quesitos que foram atingidos:

1. O Utilizador (administrador/cliente) pode fazer login na aplicação e também tem a possibilidade de visualizar ou editar os dados do seu perfil;
2. O Administrador pode editar, desativar e criar veículos;
3. O Administrador pode editar, desativar e criar clientes;
4. O Administrador pode gerar uma password única para o cliente, de modo que ele tenha acesso à sua página;
5. O Administrador pode editar, eliminar e criar equipamentos;
6. O Administrador pode associar um equipamento a um veículo;
7. O Administrador pode criar vários veículos para um determinado cliente;
8. O Administrador/Cliente pode recuperar a sua password;
9. O Cliente pode visualizar os seus veículos no mapa em tempo real;
10. O Cliente pode configurar os alertas de ignição e de velocidade num veículo selecionado, bem como pode visualizar os respetivos alertas no seu e-mail;
11. O Cliente pode visualizar o percurso atual e os percursos num período definido de um veículo selecionado;

12. O Cliente pode gerar relatórios de Percursos e de Ignição efetuados pelo veículo num determinado período e por sua vez são enviados para o seu e-mail;
13. A Camada de Receção contém método auxiliar para eliminar histórico dos dados de localização que contém mais de 3 meses;
14. A Camada de Receção insere os dados dos alertas na base dados;
15. A Camada de Receção insere e atualiza o estado atual da viatura na base de dados;
16. A Camada de Receção insere o percurso de uma determinada viatura na base de dados;
17. A Camada de Receção regista as coordenadas geográficas dos veículos na base de dados.

Para além dos Requisitos Funcionais alcançados, também se verificou através da análise e do estudo dos diferentes serviços de mapas (*OpenStreetMaps*), serviços de comunicação (envio SMS), tecnologias de desenvolvimento (*Angular*, *NodeJs* e *Leaflet*) e a criação de uma base de dados específica, que existe a possibilidade de criar uma plataforma simplificada para gestão de frotas menos dispendiosa do que a solução atual da GisGeo e com mesmo desempenho e também permitir ao cliente ter um controlo sobre os seus veículos de uma forma ágil e eficiente. É importante salientar que a configuração do equipamento de localização também teve um grande impacto na redução dos custos obtidos da nova solução, nomeadamente o tempo definido para o envio dos dados de localização para o servidor, bem como os dados que foram estabelecidos.

7.2 Limitações e trabalho futuro

No geral o projeto realizado satisfaz os requisitos propostos inicialmente, no entanto, ficou por desenvolver quesitos que eram considerados úteis para complementar a solução *Geocar Lite*, nomeadamente o facto de não ter sido desenvolvido a aplicação móvel para o cliente e administrador, devido ao tempo limitado, uma vez que se deu mais importância às funcionalidades gerais da solução.

Para além disso, a plataforma não ficou apta para vários idiomas, bem como a sua interface gráfica do administrador e do cliente não ficou muito apelativa, mas ficou bastante funcional. Em relação aos relatórios gerados de ignição e de percursos poderiam ter mais informação detalhadas, mas devido à restrição dos custos de desenvolvimento, não foi possível acrescentar e configurar mais dados, pois ocuparia mais espaço na Base de Dados.

Por fim, relativamente à visualização dos percursos efetuados pelos veículos de testes, estes não ficaram muito bem delineados no mapa, devido ao facto dos dados geográficos terem sido recolhidos de um **servidor backup**, isto é, os dados obtidos pelos equipamentos de localização eram enviados para dois servidores, um que era o **servidor principal** do Geocar e o outro o **servidor de backup**. A maioria desses dados vindo dos equipamentos de localização eram enviados para o servidor principal através do protocolo UDP, verificando que no servidor backup teria poucos dados armazenados. A razão para isto acontecer é o facto de os dados dos equipamentos de localização serem enviados através do protocolo UDP, no qual não garante

que esses dados sejam transferidos para o servidor principal, passando assim a serem novamente enviados para o servidor backup.

7.3 Apreciação final e pessoal

Considerando todo o trabalho efetuado, desde o estudos e análises efetuadas, bem como a implementação da aplicação até ao momento, considera-se que a maioria dos objetivos foram atingidos.

Este *software* desenvolvido apresenta uma boa estrutura e segue a maioria dos princípios adotados pela empresa fazendo com que as funcionalidades desenvolvidas sejam mais fáceis de alterar ou acrescentar outras funcionalidades.

A fase inicial do estudo e investigação efetuada relativamente às tecnologias utilizadas, bem como as soluções encontradas para o desenvolvimento da solução, revelaram-se ser uma mais-valia para o sucesso do projeto desenvolvido.

Por fim, a execução deste projeto permitiu ganhar experiência e mais conhecimentos relativamente às ferramentas utilizadas, nomeadamente, *Angular, C#, Node Js, PostgreSQL e Leaflet*, tornando-se um grande desafio cumprido.

Referências

- [1] HERE, “HERE Technologies | The world’s #1 location platform,” 2022. <https://www.here.com/> (accessed Feb. 02, 2022).
- [2] Google, “Acerca de – Google Maps,” 2022. <https://www.google.com/maps/about/#/> (accessed Feb. 02, 2022).
- [3] EZ4U, “EZ4U SMS Marketing & Transacional - Portugal e Resto do Mundo,” 2022. <https://www.ez4uteam.com/> (accessed Feb. 02, 2022).
- [4] InTouchBiz, “Intouchbiz - Conected We Stant,” 2022. <https://intouchbiz.com/> (accessed Feb. 02, 2022).
- [5] L. Sulzbacher, “Gestão de frotas: A importância e os benefícios de realizá-la,” 16/07/2021, 2021. <https://blog-pt.checklistfacil.com/gestao-de-frotas/> (accessed Jan. 25, 2022).
- [6] IBGE, “gnss,” 2018. <https://atlascolar.ibge.gov.br/conceitos-gerais/o-que-e-cartografia/sistema-global-de-navegac-a-o-por-sate-lites.html> (accessed Jun. 27, 2022).
- [7] Wikipedia, “Sistema de navegação por satélite – Wikipédia, a enciclopédia livre,” 2022. https://pt.wikipedia.org/wiki/Sistema_de_navegacão_por_satélite (accessed Jun. 27, 2022).
- [8] Canaltech, “O que é GLONASS? - Canaltech,” 2022. <https://canaltech.com.br/produtos/O-que-e-GLONASS/> (accessed Feb. 02, 2022).
- [9] ESA, “ESA - What is Galileo?,” 2022. https://www.esa.int/Applications/Navigation/Galileo/What_is_Galileo (accessed Feb. 02, 2022).
- [10] Canaltech, “O que é Compass ou Beidou-2? - Canaltech,” 2022. <https://canaltech.com.br/curiosidades/O-que-e-Compass-ou-Beidou-2/> (accessed Feb. 02, 2022).
- [11] IME, “Como funciona o GPS? – Derivando a matemática,” 2022. <http://www.ime.unicamp.br/~apmat/o-sistema-gps/> (accessed Feb. 02, 2022).
- [12] SoFísica, “GPS - O que é, como funciona - Só Física,” 2022. <https://www.sofisica.com.br/conteudos/curiosidades/gps.php> (accessed Feb. 02, 2022).
- [13] OFICINA REDAÇÃO, “O que é A-GPS?,” 9/4/2018, 2018. https://www.oficinadanet.com.br/artigo/1185/o_que_e_a-gps (accessed Feb. 02, 2022).
- [14] StringFixer, “Ajuste de contas,” 2022. https://stringfixer.com/pt/Dead_reckoning (accessed Feb. 02, 2022).
- [15] StringFixer, “Localização automática de veículos,” 2022. https://stringfixer.com/pt/Automatic_vehicle_location (accessed Feb. 02, 2022).
- [16] Wikipedia, “Machine to Machine – Wikipédia, a enciclopédia livre,” 2022. https://pt.wikipedia.org/wiki/Machine_to_Machine (accessed Feb. 02, 2022).
- [17] FAQtos, “O que é o sistema GSM? - FAQtos,” 2020. <https://faqtos.pt/faq-items/o-que-e-o-sistema-gsm/> (accessed Feb. 02, 2022).
- [18] OFICINA REDAÇÃO, “O que é GSM e como funciona?,” 12/5/2013, 2013. https://www.oficinadanet.com.br/artigo/733/gsm_o_que_e_e_como_funciona (accessed Feb. 02, 2022).
- [19] A. Emerson, “Tecnologias 2G e 2,5G: TDMA, CDMA, GSM, GPRS e EDGE,” 4/6/2012, 2012. <https://www.infowester.com/2g.php#gsm> (accessed Feb. 02, 2022).
- [20] SkyView, “SSS-Operador Rede Publica,” GSM, 2022.

- [21] Wikipedia, "General Packet Radio Service – Wikipédia, a enciclopédia livre," 3/10/2020, 2020. https://pt.wikipedia.org/wiki/General_Packet_Radio_Service (accessed Feb. 02, 2022).
- [22] Juniper Networks, "GGSN Overview | Junos OS | Juniper Networks," 11/3/2021, 2021. <https://www.juniper.net/documentation/us/en/software/junos/gtp-sctp/topics/topic-map/security-gprs-ggsn-overview.html> (accessed Feb. 02, 2022).
- [23] M. del Rey, "RFC 793 - Transmission Control Protocol," 1981. <https://datatracker.ietf.org/doc/html/rfc793> (accessed Feb. 02, 2022).
- [24] T. A. e Y. N. Marina Lacerda, "CAN - Arquitetura," 2011. <https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/can/arquitetura.htm> (accessed Feb. 02, 2022).
- [25] FE-UP, "3 PROTOCOLO DE COMUNICAÇÕES CAN 3.1 SISTEMAS EM REDE vs vs vs vs SISTEMAS CENTRALIZADOS," 2020.
- [26] ISEP DEI, "Arquiteturas de Rede," 2019. <https://www.dei.isep.ipp.pt/~asc/doc/arquiteturas.html> (accessed Jun. 25, 2022).
- [27] A. Albert, "Comparison of Event-Triggered and Time-Triggered Concepts with Regard to Distributed Control Systems," *Embedded World*, pp. 235–252, 2004.
- [28] V. A. Souza, "Introdução ao CAN Vitor Amadeu Souza-Cerne Tecnologia," 2014.
- [29] 4x4brasil, "OBD - Onboard Diagnostics System," 2019. <http://www.4x4brasil.com.br/forum/jeep-wrangler-cherokee/91657-obd-onboard-diagnostics-system.html> (accessed Feb. 02, 2022).
- [30] FCA, "Node.js - Informática - Tecnologias & Programação Web - FCA," 2021. <https://www.fca.pt/pt/catalogo/informatica/tecnologias-programacao-web/node-js/> (accessed Feb. 13, 2022).
- [31] A. Alexandre, "O que é Angular?," 2018. <https://blog.algaworks.com/o-que-e-angular/> (accessed Feb. 13, 2022).
- [32] DevMedia, "Guia de Programação em C#: Aprenda C# do Básico ao Avançado," 2022. <https://www.devmedia.com.br/guia/linguagem-csharp/38152> (accessed Feb. 13, 2022).
- [33] A. Vladimir, "Leaflet - a JavaScript library for interactive maps," 2022. <https://leafletjs.com/> (accessed Feb. 13, 2022).
- [34] Wikiped, "Leaflet (software) - Wikipedia," 2022. [https://en.wikipedia.org/wiki/Leaflet_\(software\)](https://en.wikipedia.org/wiki/Leaflet_(software)) (accessed Feb. 13, 2022).
- [35] PostgreSQL, "PostgreSQL: About," 2022. <https://www.postgresql.org/about/> (accessed Feb. 13, 2022).
- [36] Wiki, "Pt:Sobre o OpenStreetMap - OpenStreetMap Wiki," 2019. https://wiki.openstreetmap.org/wiki/Pt:Sobre_o_OpenStreetMap (accessed Feb. 13, 2022).
- [37] MapQuest, "MapQuest Development and Documentation - Getting Started | MapQuest API Documentation," 2021. <https://developer.mapquest.com/documentation/> (accessed Feb. 13, 2022).
- [38] Google, "Plataforma do Google Maps | Google Developers," 2021. <https://developers.google.com/maps/documentation> (accessed Feb. 13, 2022).
- [39] Google, "Overview | Maps JavaScript API | Google Developers," 2021. https://developers.google.com/maps/documentation/javascript/overview#maps_map_simple-javascript (accessed Feb. 13, 2022).
- [40] HERE, "Location Technology | Autonomous World | About Us | HERE," 2021. <https://www.here.com/company/about-us> (accessed Feb. 13, 2022).

- [41] HERE, "Guide - HERE Geocoding & Search API - HERE Developer," 2021. https://developer.here.com/documentation/geocoding-search-api/dev_guide/index.html (accessed Feb. 13, 2022).
- [42] HERE, "HERE Base Plan | Location Services | Pricing | HERE," 2022. <https://www.here.com/pricing> (accessed Feb. 13, 2022).
- [43] DevMedia, "Google Geocoding API: como obter a localização de endereços," 2019. <https://www.devmedia.com.br/como-utilizar-a-google-geocoding-api-para-obter-enderecos/36751> (accessed Feb. 13, 2022).
- [44] Bing, "Know what businesses are nearby - Bing Maps Local Search API | Maps Blog," 2021. <https://blogs.bing.com/maps/2018-12/know-what-businesses-are-nearby-with-bing-maps-local-search-api> (accessed Feb. 13, 2022).
- [45] Microsoft, "Pricing - Bing Search API v7 | Microsoft Azure," 2020. <https://azure.microsoft.com/en-us/pricing/details/cognitive-services/search-api/> (accessed Feb. 13, 2022).
- [46] Frotcom, "Empresa | Frotcom," 2021. <https://www.frotcom.com/pt-pt/company> (accessed Feb. 13, 2022).
- [47] Frotcom, "Localização GPS e Gestão de frotas - Frotcom," 2021. <https://www.frotcom.com/pt-pt/gestao-de-frotas> (accessed Feb. 13, 2022).
- [48] W. S. PT, "Gestor de Frotas - Otimize o desempenho da sua frota — Webfleet Solutions PT," 2021. https://www.webfleet.com/pt_pt/webfleet/solutions/fleet-manager/ (accessed Feb. 13, 2022).
- [49] W. S. PT, "O que é a gestão de frotas? Webfleet Solutions PT," 2021. https://www.webfleet.com/pt_pt/webfleet/fleet-management/ (accessed Feb. 13, 2022).
- [50] V. C. PT, "Sobre nós | Verizon Connect PT," 2022. <https://www.verizonconnect.com/pt/empresa/sobre/> (accessed Feb. 13, 2022).
- [51] V. C. PT, "Verizon Connect chega a Portugal com novas soluções telemáticas para gestão de frotas - Revista Cargo," 2022. <https://revistacargo.pt/verizon-connect-chega-a-portugal-com-novas-solucoes-telematicas-para-gestao-de-frotas/> (accessed Feb. 13, 2022).
- [52] V. C. PT, "Localizador GPS para Frotas | Verizon Connect PT," 2022. <https://www.verizonconnect.com/pt/solucoes/localizador-gps/> (accessed Feb. 13, 2022).
- [53] P. A. Koen *et al.*, "Fuzzy Front End : and Techniques," *Industrial Research*, vol. pp, pp. 5–35, 1996.
- [54] Vendus, "Análise SWOT o que é - Vendus," 2022. <https://www.vendus.pt/blog/analise-swot/> (accessed Feb. 17, 2022).
- [55] P. Technology, "What is the Analytic Hierarchy Process (AHP)? | Passage Technology," 2022. <https://www.passagetechology.com/what-is-the-analytic-hierarchy-process> (accessed Feb. 17, 2022).
- [56] C. S. Marins, D. de O. Souza, and M. da S. Barros, "O USO DO MÉTODO DE ANÁLISE HIERÁRQUICA (AHP) NA TOMADA DE DECISÕES GERENCIAIS UM ESTUDO DE CASO - PDF Free Download," 2009. <https://docplayer.com.br/39331176-O-uso-do-metodo-de-analise-hierarquica-ahp-na-tomada-de-decisoes-gerenciais-um-estudo-de-caso.html> (accessed Feb. 23, 2022).
- [57] Simplicable, "29 Examples of Product Value - Simplicable," 2022. <https://simplicable.com/new/product-value> (accessed Feb. 17, 2022).
- [58] Tallyfy, "Customer Value: What it Means and How to Create It [5+ Ideas] - Tallyfy," 2021. <https://tallyfy.com/customer-value/> (accessed Feb. 17, 2022).

- [59] Investopedia, "Perceived Value Definition," 2022. <https://www.investopedia.com/terms/p/perceived-value.asp> (accessed Feb. 17, 2022).
- [60] R. Content, "Proposta de valor: o que é e como criar a proposta perfeita," 2019. <https://rockcontent.com/br/blog/proposta-de-valor/> (accessed Feb. 17, 2022).
- [61] Runrun, "Canvas online: experimente novos modelos de negócio | Runrun.it," 2019. <https://blog.runrun.it/canvas-online/> (accessed Feb. 17, 2022).
- [62] IBM, "O que é arquitetura de três camadas (tiers) - Brasil | IBM," 2020. <https://www.ibm.com/br-pt/cloud/learn/three-tier-architecture> (accessed Feb. 19, 2022).
- [63] Wikipedia, "MVC – Wikipédia, a enciclopédia livre," 2022. <https://pt.wikipedia.org/wiki/MVC> (accessed Feb. 19, 2022).
- [64] DevMedia, "Requisitos Não Funcionais e Arquitetura de Software," 2021. <https://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525> (accessed Feb. 19, 2022).
- [65] BATNA24, "Teltonika FMB920 | Localizador GPS | Compact Tracker GNSS, GSM," 2019. https://www.batna24.com/pt/p/teltonika-fmb920-localizador-gps-rmngj?gclid=CjwKCAjwIaVBhBkEiwAsr7-c89WqfISW2CPdvRvbulqqaXgOFLYnVam_jdkkuowvWS5v6r1249RcRoCOs8QAvD_BwE#specs (accessed Jun. 26, 2022).
- [66] SistemaRodar, "Configuração equipamento Teltonika FMB920," 2021. <http://suporte.sistemarodar.com.br/2021/05/configuracao-equipamento-teltonika.html> (accessed Sep. 03, 2022).
- [67] Kenzie, "O que é teste de integração e quais são os tipos de teste?," 2020. <https://kenzie.com.br/blog/teste-de-integracao/> (accessed Jun. 26, 2022).
- [68] M. Focus, "O que é teste de desempenho? | Micro Focus," 2018. <https://www.microfocus.com/pt-br/what-is/performance-testing> (accessed Jul. 16, 2022).
- [69] guro99, "How to Use JMeter for Performance & Load Testing," 2020. <https://www.guru99.com/jmeter-performance-testing.html> (accessed Jul. 16, 2022).
- [70] Amen, "Servidores virtuais Windows & Linux - Alojamento Cloud VPS," 2022. <https://www.amen.pt/server/cloud-vps.html> (accessed Sep. 03, 2022).
- [71] R. Globo, "Rede Globo > globo ciência - Da bússola ao GPS, entenda como funciona a navegação na era digital," 2016. <http://redeglobo.globo.com/globociencia/noticia/2011/10/da-bussola-ao-gps-entenda-como-funciona-navegacao-na-era-digital.html> (accessed Feb. 20, 2022).
- [72] ResearchGate, "Dead Reckoning concept for car navigation | Download Scientific Diagram," 2022. https://www.researchgate.net/figure/Dead-Reckoning-concept-for-car-navigation_fig2_322328126 (accessed Feb. 20, 2022).
- [73] DataRio, "Cabo extensão J1962 OBDII OBD2 Macho-Fêmea 150 centímetros - DRTECHPARTS - Data Rio Serviços de Informática Lda," 2020. <https://www.datario.pt/produto/cabo-extensao-j1962-obdii-obd2-macho-femea-150-centimetros/> (accessed Feb. 20, 2022).
- [74] ResearchGate, "The New Concept Development (NCD)-model. (Koen et al., 2001) | Download Scientific Diagram," 2022. https://www.researchgate.net/figure/The-New-Concept-Development-NCD-model-Koen-et-al-2001_fig12_324208591 (accessed Feb. 20, 2022).
- [75] S. Nicola, "Análise_valor Aula_2_22OUT_2019_1hora_AHP." 2019.
- [76] ResearchGate, "3-tier architecture | Download Scientific Diagram," 2020. https://www.researchgate.net/figure/3-tier-architecture_fig1_277187696 (accessed Feb. 20, 2022).

- [77] HackerNoon, "Writing a Simple MVC (Model, View, Controller) App in Vanilla Javascript | HackerNoon," 2021. <https://hackernoon.com/writing-a-simple-mvc-model-view-controller-app-in-vanilla-javascript-u65i34lx> (accessed Feb. 20, 2022).

Anexo 1 Diagramas auxiliares

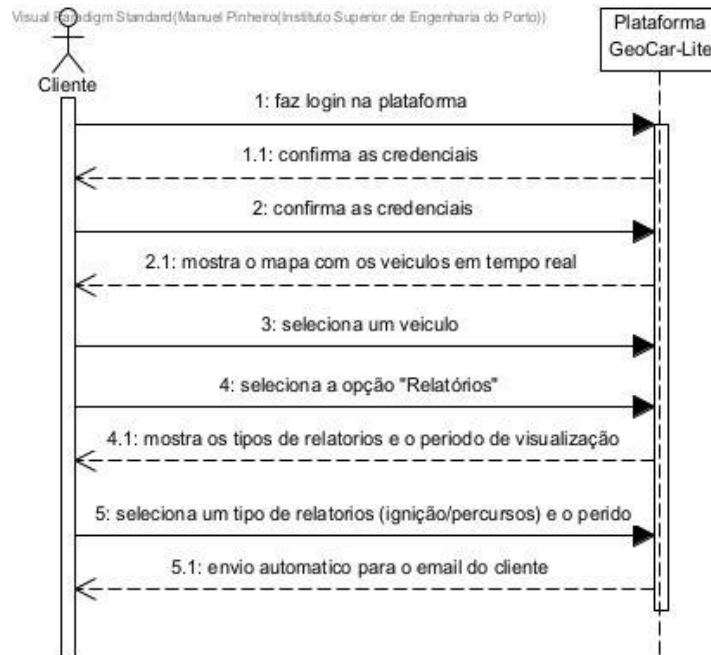


Figura 54 – UC3 Visualizar Relatório



Figura 55 – UC4 Configurar Alertas



Figura 56 – UC6 Criar Veículos, UC7 Editar Veículos e UC8 Desativar Veículos

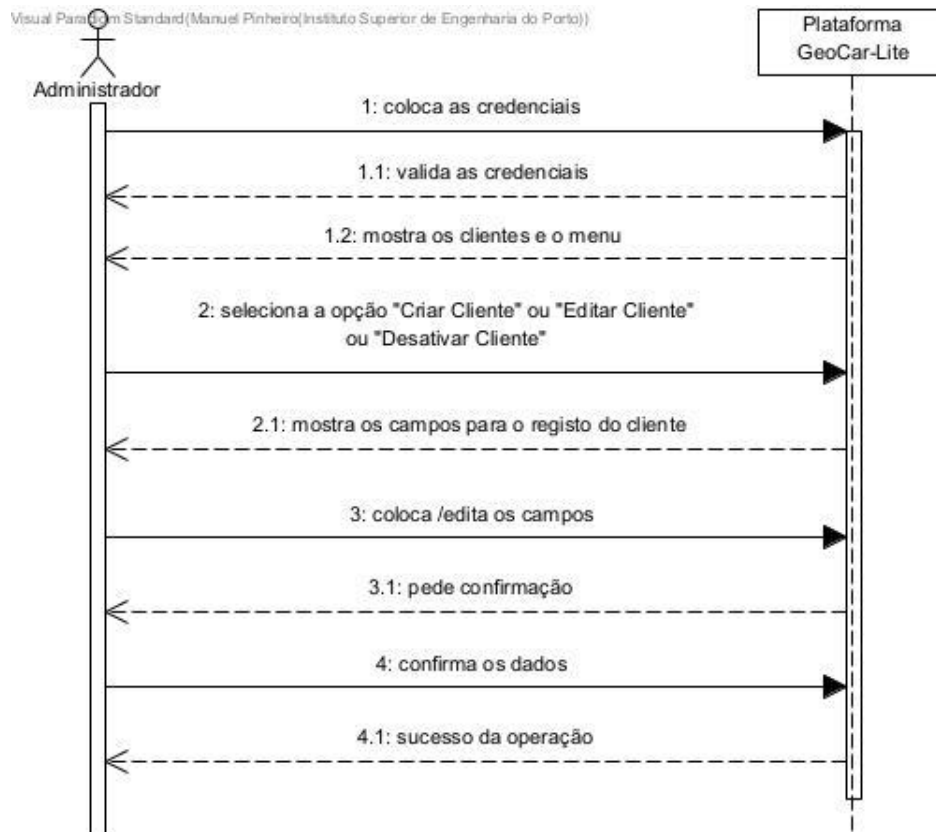


Figura 57 - UC9 Criar Clientes, UC10 Editar Clientes e UC11 Desativar Clientes

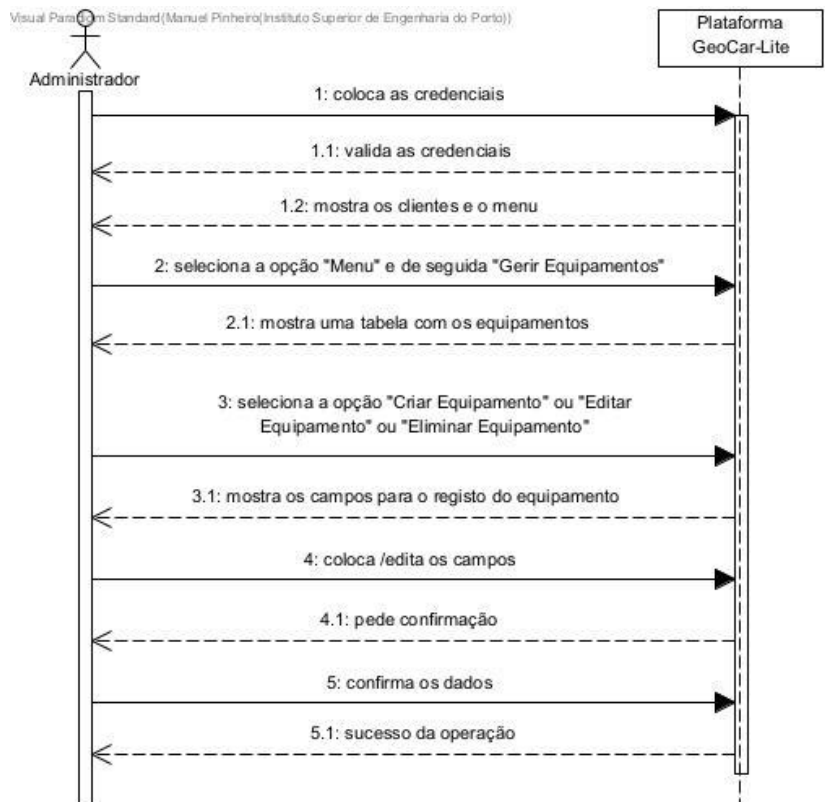


Figura 58 – UC12 Criar Equipamentos, UC13 Editar Equipamentos e UC14 Eliminar Equipamentos

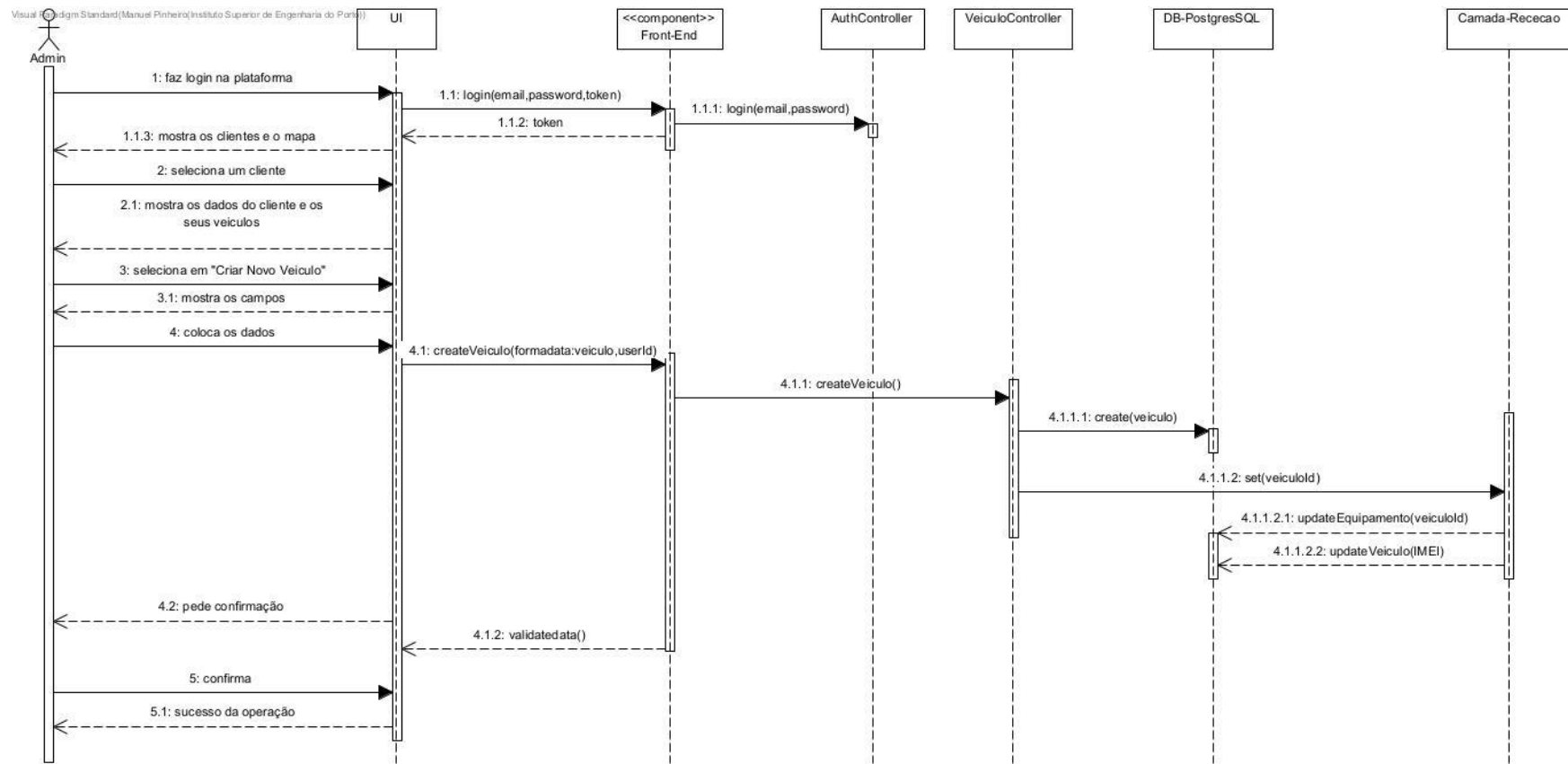


Figura 59 – Diagrama de sequência do UC6

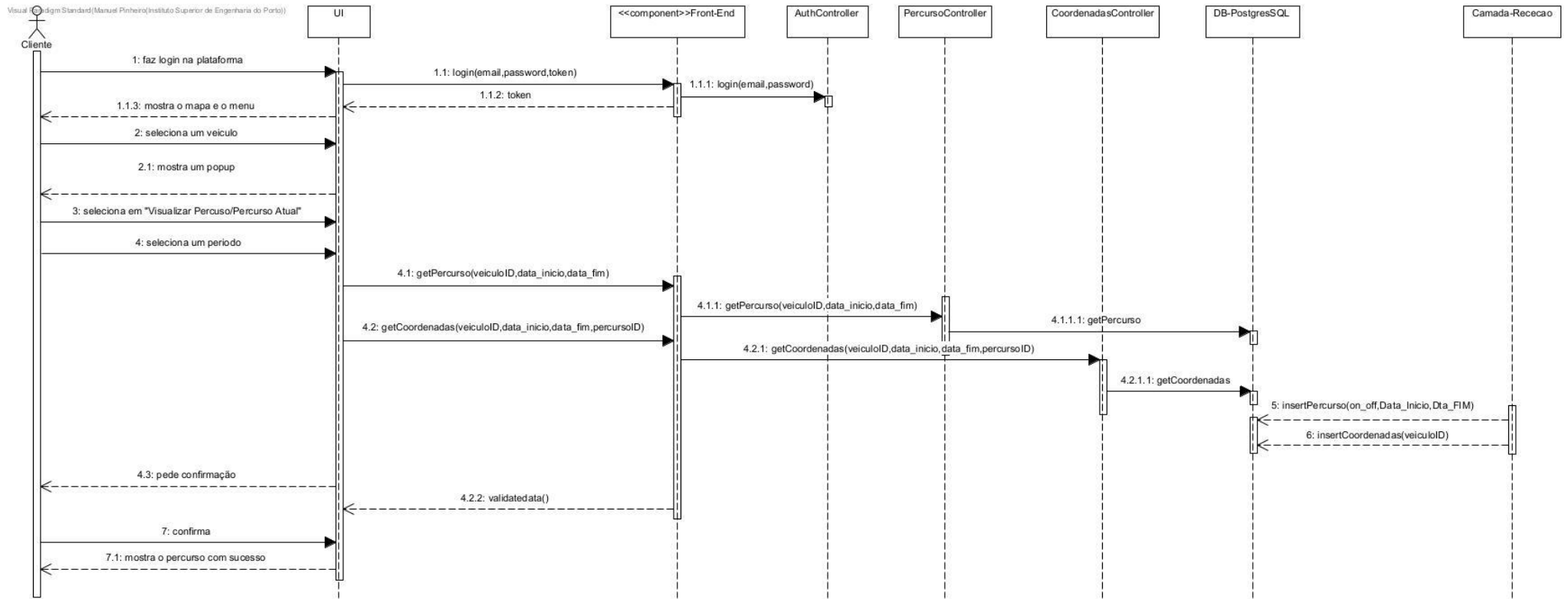


Figura 60 – Diagrama de sequência do UC2

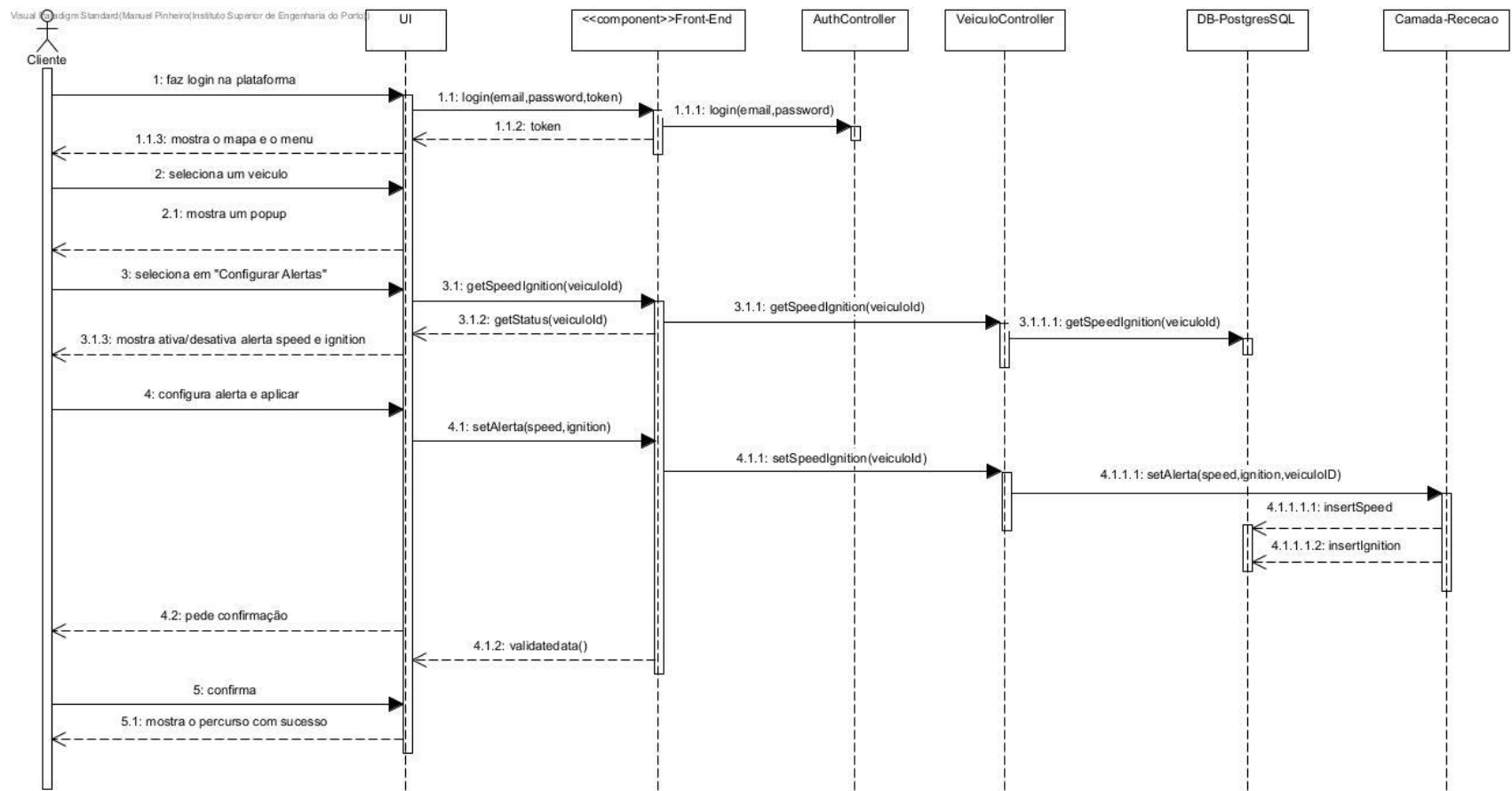
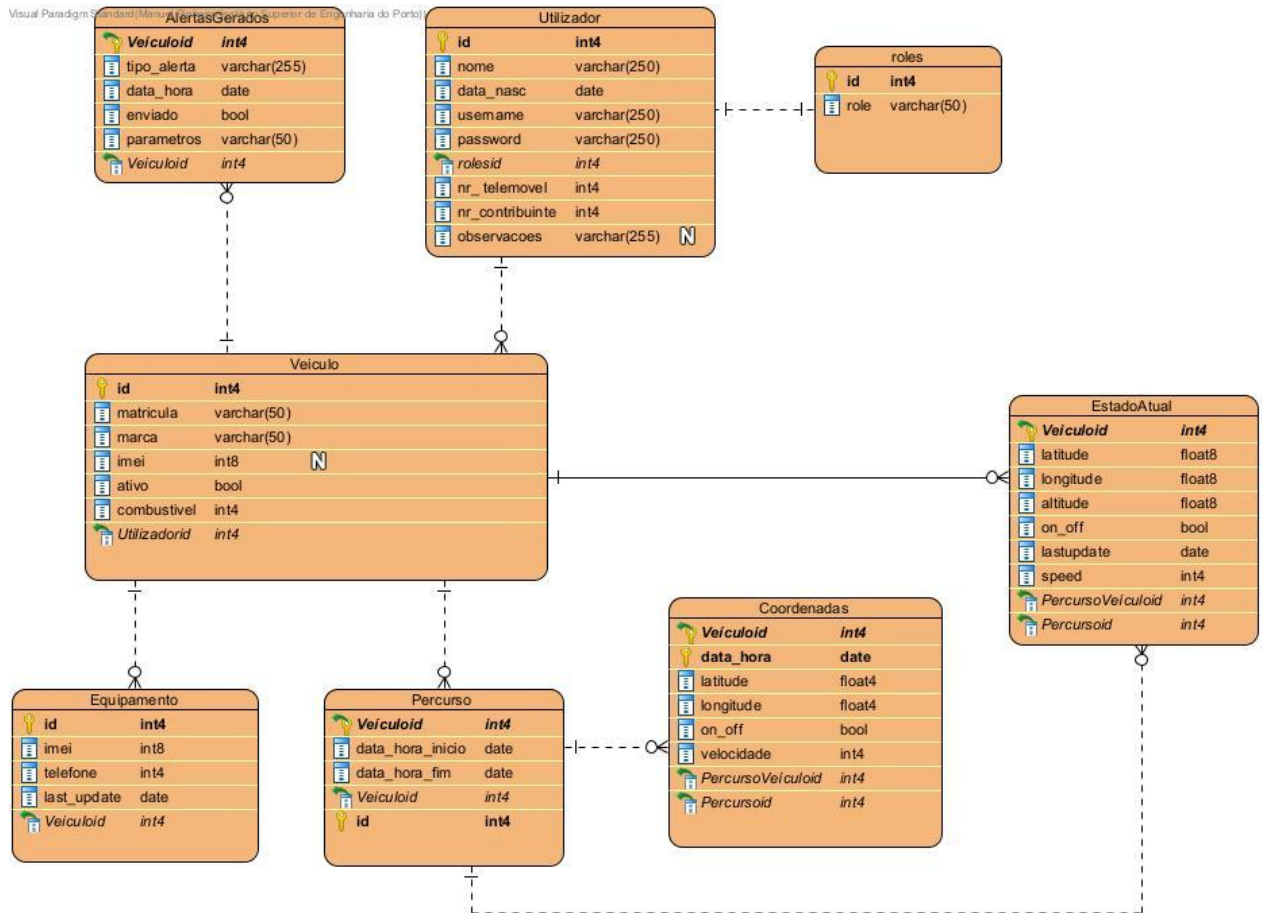
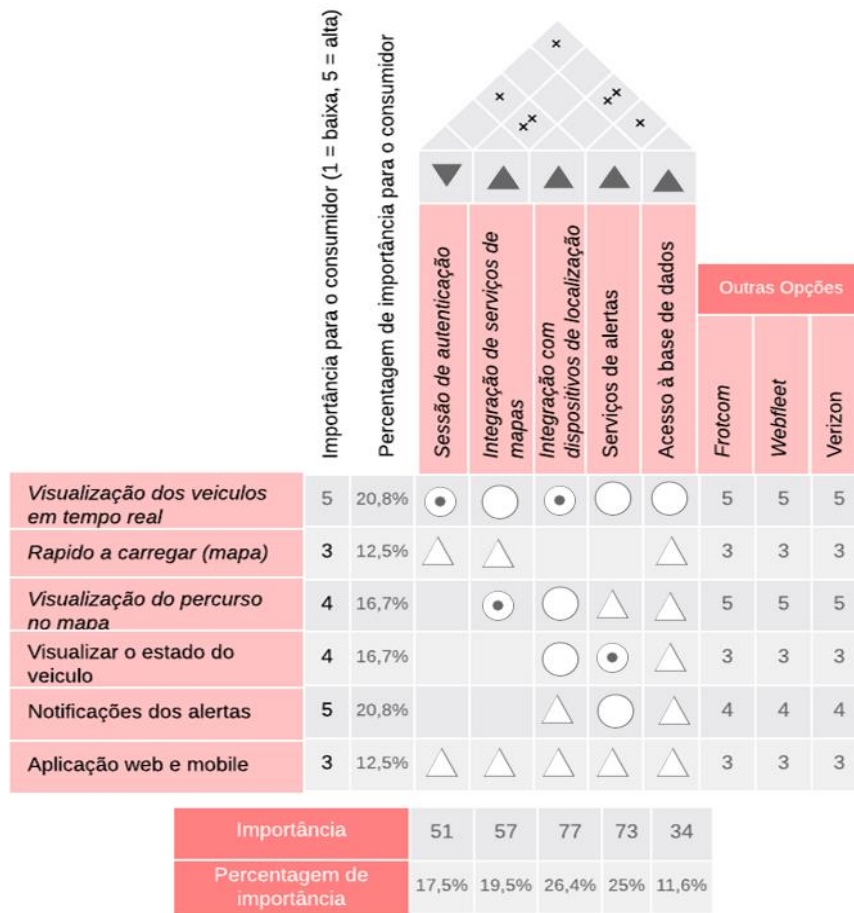


Figura 61 – Diagrama de sequência do UC4

Anexo 2 Modelo Relacional



Anexo 3 Quality Funtion Deployment



Anexo 4 Análise de usabilidade da *Geocar Lite* (administrador)

Análise de usabilidade Geocar-Lite (administrador)

A plataforma GeoCar-Lite é uma plataforma de gestão frotas que permite geo-localizar as viaturas em tempo real, bem como saber o estado (ignição e/ ou velocidade) das mesmas. Para além disso dá a conhecer ao cliente os percursos efetuados pelas suas viaturas selecionadas e também permite gerar relatórios de ignição e dos percursos efetuados.

O objetivo é analisar a opinião dos utilizadores do serviço do Geocar-Lite.

Agradecemos desde já a sua disponibilidade ao preencher o questionário.

Pre vemos que o seu preenchimento é breve, não demorando mais do que 5 minutos.

1. Está satisfeito com a interface gráfica da plataforma *GeoCar-Lite* (administrador)?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

2. Em relação ao processo de criação de clientes, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

3. Em relação ao processo de edição de clientes, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

4. Em relação ao processo de eliminação dos clientes, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

5. Em relação ao processo de criação de equipamentos, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

6. Em relação ao processo de edição de equipamentos, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

7. Em relação ao processo de eliminação dos equipamentos, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

8. Em relação ao processo de criação de veículos, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

9. Em relação ao processo de edição de veículos, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

10. Em relação ao processo de eliminação dos veículos, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

11. Está satisfeito quanto ao tempo que levou a completar as tarefas da plataforma?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

12. Está satisfeito com a fluidez da plataforma do administrador?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

13. Qual é o seu grau de satisfação quanto à utilização do *Geocar-Lite* da parte do administrador?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

14. Recomendaria o uso desta a plataforma a alguém?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

Anexo 5 Análise de usabilidade da *Geocar Lite* (cliente)

Análise da Usabilidade do GeoCar-Lite (cliente)

A plataforma GeoCar-Lite é uma plataforma de gestão frotas, que permite geo-localizar as viaturas em tempo real, bem como saber o estado (ignição e/ ou velocidade) das mesmas. Para além disso dá a conhecer ao cliente os percursos efetuados pelas suas viaturas selecionadas e também permite gerar relatórios de ignição e dos percursos efetuados.

O objetivo é analisar a opinião dos utilizadores do serviço do GeoCar-Lite.

Agradecemos desde já a sua disponibilidade ao preencher o questionário. Prevemos que o seu preenchimento é breve, não demorando mais do que 5 minutos.

1. Está satisfeito com a interface gráfica da plataforma *GeoCar-Lite* ?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

2. Em relação à visualização dos veículos em tempo real, está satisfeito quanto aos dados dos veículos (informações) apresentados no mapa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante Satisfatório

3. Em relação ao processo de visualização dos percursos, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante Satisfatório

4. Em relação ao processo de notificações de alertas, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

5. Em relação ao processo de visualização de relatórios, está satisfeito quanto à facilidade em completar a tarefa?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

6. Está satisfeito quanto ao tempo que levou a completar as tarefas da plataforma?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

7. Qual é o seu grau de satisfação quanto à utilização do *Geocar-Lite* da parte do cliente?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório

8. Recomendaria o uso desta a plataforma a alguém?

Marcar apenas uma oval.

	1	2	3	4	5	
Insatisfatório	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Bastante satisfatório