



Centro de Controlo para Veículos Eléctricos

TIAGO MANUEL SANTOS MEIRELES

Outubro de 2021

Centro de Controlo para Veículos Elétricos

Tiago Manuel Santos Meireles

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Filipe Pacheco

Supervisor: Carlos Almeida

Júri:

Presidente:

Piedade Carvalho, Docente, DEI/ISEP

Vogais:

Filipe Pacheco, Docente, DEI/ISEP

Isabel Praça, Docente, DEI/ISEP

Porto, Outubro 2021

Dedicatória

Dedico o trabalho realizado neste projeto ao Instituto Superior de Engenharia de Porto, uma vez que, ao longo destes últimos 6 anos, forneceu-me a formação académica necessária para que eu possa conseguir abraçar novos desafios.

À EVIO, e principalmente ao Carlos Almeida, por acreditar nas minhas aptidões para a realização deste projeto e por todo o apoio e ajuda prestada quando foi necessária.

Ao meu orientador, por todo o auxílio fornecido ao longo do desenvolvimento do projeto.

Por último, também dedico este trabalho à minha família e amigos, uma vez que estes foram fundamentais ao longo desta etapa.

Resumo

Atualmente, cada vez mais pessoas decidem adaptar os seus hábitos de consumo para serem mais ecológicos, uma vez que o impacto da quantidade de poluição que é gerada e libertada diariamente para a atmosfera é cada vez mais visível no mundo e a causa de múltiplos incidentes climáticos. Uma das mudanças que pode contribuir para a diminuição da libertação de gases tóxicos para a atmosfera é a substituição de veículos de combustão interna por veículos elétricos sem emissões.

Deste modo, cada vez mais o mercado avança no sentido de incentivar esta mudança, focando-se no desenvolvimento de soluções que complementem e tornem essa transição mais fácil, tais como a criação, em Portugal, de uma rede pública de postos de carregamento elétricos ou a criação de carregadores, de dimensão mais pequena, com o objetivo de serem instalados numa habitação e permitirem o carregamento de viaturas elétricas a partir de casa.

A EVIO é uma empresa que se insere no mercado da mobilidade elétrica e, como tal, pretende simplificar esta transição.

Assim, a EVIO pretende fornecer aos seus clientes aplicações que permitam oferecer mais controlo sobre os seus veículos elétricos e a monitorização da sua utilização. Deste modo, identificou-se a necessidade de desenvolver soluções que integrassem com os veículos elétricos dos utilizadores, de modo a aceder aos dados gerados pelo veículo, com o objetivo de criar novos produtos e desenvolver e melhorar funcionalidades já existentes.

Para tal, é necessário desenvolver uma plataforma que permita aos seus utilizadores registar, gerir e monitorizar remotamente as suas viaturas elétricas. Para tal, a plataforma estabelece comunicação com os veículos, sendo possível assim recolher dados da sua utilização. Através da comunicação estabelecida, a plataforma permite a execução de comandos à distância, a apresentação de dados estatísticos relevantes e a notificação do utilizador na deteção de situações anómalas, identificadas através da análise dos dados coletados.

Palavras-chave: Veículo elétrico, poluição, mobilidade elétrica, monitorização, EVIO

Abstract

Nowadays, more and more people decide to adapt their consumption habits to be more ecological, since the impact of the amount of pollution that is generated and released into the atmosphere daily is becoming more increasingly visible in the world and it's the cause of multiple weather incidents. One of the changes that can contribute to the reduction of toxic gases released into the atmosphere is the replacement of internal combustion vehicles with emission-free electric vehicles.

Thus, the market is increasingly moving towards encouraging this change, focusing on the development of solutions that complement and make this transition easier, such as the creation, in Portugal, of a public network of electric charging stations or the creation of chargers, with a smaller dimension, with the objective of being installed in a house and allowing the charging of electric vehicles from home.

EVIO is a company that is part of the electric mobility market and, as such, intends to simplify this transition.

Thereby, EVIO intends to provide its customers with applications that allow them to have more control over their electric vehicles and monitoring their use. Thus, it was identified the need to develop solutions that integrate with the users' electric vehicles, to access the data generated by the vehicle, with the objective of creating new products and developing and improving existing functionalities.

So, it's necessary to develop a platform that allows its users to register, manage and monitor their electric vehicles remotely. To this end, the platform establishes communication with the vehicles, making it possible to collect data of their use. Through the established communication, the platform allows the execution of remote commands, the viewing of relevant statistical data and the notification of the user when detected anomalous situations, identified through the analysis of collected data.

Keywords: Electric vehicle, pollution, electric mobility, monitoring, EVIO

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Apresentação do projeto	2
1.3	Descrição do Problema	3
1.4	Contributos do projeto	4
1.5	Estrutura do documento	5
2	Estado da Arte	7
2.1	Mobilidade Elétrica	7
2.1.1	Veículo Elétrico	7
2.1.2	Comercializadores para a mobilidade elétrica (CEME)	8
2.2	Sistema de Infoentretenimento	9
2.3	Aplicações existentes para acesso remoto a veículos	10
2.3.1	ConnectedDrive	10
2.3.2	Tesla App	12
2.3.3	We Connect Go	14
2.3.4	My Renault	15
2.3.5	NissanConnect	16
3	Contexto e Análise de Valor	19
3.1	Análise de valor	19
3.1.1	New Concept Development Model (NCD)	19
3.1.2	Proposta de Valor	20
3.1.3	Priorização dos mecanismos de comunicação com veículos	26
3.2	Análise de arquiteturas da solução	32
3.2.1	Arquitetura monolítica	32
3.2.2	Arquitetura baseada em microsserviços	34
3.3	Interoperabilidade entre serviços	36
3.4	Visão da solução	37
4	Ambiente de Trabalho	39
4.1	Gestão de projeto	39
4.1.1	Planeamento	39
4.1.2	Reuniões de acompanhamento	39
4.2	Tecnologias utilizadas	40
4.2.1	Desenvolvimento do portal <i>web</i> do Centro de Controlo para Veículos Elétricos	41
4.2.2	Desenvolvimento de <i>software</i> baseado numa arquitetura de microsserviços	43
4.2.3	Desenvolvimento da camada de abstração	45

4.2.4	Criação de uma base de dados não relacional	45
4.3	Ferramentas de apoio	45
4.4	Metodologia de investigação.....	46
5	Descrição técnica	49
5.1	Engenharia de requisitos	49
5.1.1	Atores	49
5.1.2	Requisitos funcionais	50
5.2	Análise	53
5.2.1	Requisitos funcionais	53
5.2.2	Requisitos não funcionais	64
5.3	Modelação	67
5.3.1	Modelo de Domínio.....	67
5.3.2	Infraestrutura	68
5.4	Estratégia de desenvolvimento.....	71
5.5	Método de avaliação	72
6	Desenvolvimento da solução	75
6.1	Desenvolvimento dos serviços	75
6.1.1	Centro de Controlo.....	75
6.1.2	Serviço de integração Tesla.....	78
6.1.3	Serviço de simulação Tesla	80
6.2	Desenvolvimento do portal <i>web</i>	81
6.2.1	Registo de utilizador	82
6.2.2	Autenticação de utilizador (<i>Login</i>)	83
6.2.3	Página Principal	85
6.2.4	Gestão de veículos	86
6.2.5	Monitorização de veículos.....	88
6.2.6	Estatísticas	91
6.2.7	Alertas	94
6.3	Implantação da plataforma	95
7	Avaliação.....	99
7.1	Quality Evaluation Framework (QEF)	99
7.1.1	Dimensão Funcionalidade	100
7.1.2	Dimensão Adaptabilidade	113
7.1.3	Dimensão Usabilidade	117
8	Conclusão.....	127
8.1	Objetivos realizados	127
8.2	Limitações do projeto	129
8.3	Contributos do projeto	129
8.4	Trabalho futuro.....	130

9	Referências.....	133
---	------------------	-----

Lista de Figuras

Figura 1 - Interface da aplicação ConnectedDrive	11
Figura 2 - Interface da aplicação Tesla	12
Figura 3 - Interface da aplicação We Connect Go	15
Figura 4 - Interface da aplicação My Renault.....	16
Figura 5 - Interface da aplicação NissanConnect	17
Figura 6 - Modelo <i>Canvas</i> da proposta de valor	23
Figura 7 - <i>House of Quality</i> (QFD).....	25
Figura 8 - Arquitetura monolítica (Gnatyk, s.d.)	33
Figura 9 - Arquitetura de microsserviços (Gnatyk, s.d.).....	34
Figura 10 – Comportamento das arquiteturas monolítica e de microsserviços ao longo do desenvolvimento de uma solução (Fowler, s.d.)	35
Figura 11 - Diagrama de Gantt para o CCVE	39
Figura 12 - Atores do sistema	50
Figura 13 - Diagrama de Casos de Uso.....	52
Figura 14 - SSD de registo de utilizador	54
Figura 15 - SSD de <i>login</i> de utilizador	54
Figura 16 - SSD de opção de pagamento para retificação do estado da conta	55
Figura 17 - SSD de mudança de plano de subscrição contratado	56
Figura 18 - SSD de registo de veículo na plataforma	57
Figura 19 - SSD de apresentação de informação de veículo	57
Figura 20 - SSD de apresentação dos dados recolhidos do veículo	58
Figura 21 - Apresentação de estatísticas baseadas nos dados recolhidos	59
Figura 22 - SSD de execução de comandos sobre o veículo	60
Figura 23 - SSD do mecanismo de recolha dos dados do veículo	61
Figura 24 - SSD do mecanismo de envio de alertas inteligentes	62
Figura 25 - SSD da configuração de alertas.....	63
Figura 26 - SSD do mecanismo de envio de alertas configurados pelo utilizador	64
Figura 27 - Modelo de Domínio	67
Figura 28 - Vista lógica da infraestrutura do sistema	69
Figura 29 - Diagrama de implantação do sistema.....	71
Figura 30 – Componente de registo de utilizador no portal <i>web</i>	83
Figura 31 - Fluxograma para o registo de um novo utilizador	83
Figura 32 - Componente de autenticação de utilizador no portal <i>web</i>	84
Figura 33 - Fluxograma da autenticação de um utilizador no sistema	85
Figura 34 – Página principal do portal	86
Figura 35 - Componente que disponibiliza operações para a gestão de veículos do utilizador	86
Figura 36 - Formulário de registo de um novo veículo	87
Figura 37 - Fluxograma para o registo de um novo veículo no portal	88
Figura 38 - Ecrã de monitorização de veículos.....	89
Figura 39 - Componente de monitorização de uma viatura em carregamento	89

Figura 40 - Fluxograma de execução de um comando numa viatura	90
Figura 41 - Execução de um comando de desbloqueio de portas numa viatura.....	91
Figura 42 - Ecrã das estatísticas	91
Figura 43 - Representação gráfica da média de kWh carregados por minuto para as sessões efetuadas pelo utilizador	92
Figura 44 - Curva de carregamento típica para um carregamento rápido de veículo (EV Life, s.d.).....	93
Figura 45 - Fluxograma da interação entre o utilizador e as estatísticas do portal.....	94
Figura 46 – Ecrã das funcionalidades de alertas	94
Figura 47 - Fluxograma para alertas no portal <i>web</i>	95

Lista de Tabelas

Tabela 1 - Veículos elétricos mais vendidos em 2019 na Europa	10
Tabela 2 - Benefícios e sacrifícios para o cliente.....	21
Tabela 3 - Escala fundamental	26
Tabela 4 - Matriz de comparação	27
Tabela 5 - Somatório dos critérios da matriz de comparação	27
Tabela 6 - Matriz de comparação normalizada.....	27
Tabela 7 - Peso dos critérios	28
Tabela 8 – Valores de IR para matrizes quadradas de ordem n	29
Tabela 9 - Matriz de comparação normalizada do critério “Comandos”	29
Tabela 10 - Matriz de comparação normalizada do critério “Conectividade”	30
Tabela 11 - Matriz de comparação normalizada do critério “Acesso a dados”	31
Tabela 12 - Matriz de comparação normalizada do critério "Performance"	31
Tabela 13 - Reuniões de acompanhamento da EVIO.....	40
Tabela 14 - Descrição das entidades apresentadas no Modelo de Domínio	67
Tabela 15 – Legenda dos diferentes comandos do <i>docker-compose</i>	97
Tabela 16 - Dimensões e fatores presentes no QEF	99
Tabela 17 - Requisito FF01	100
Tabela 18 - Requisito FF02	101
Tabela 19 - Requisito FF03	101
Tabela 20 - Requisito FF04	102
Tabela 21 - Requisito FE01	103
Tabela 22 - Requisito FE02	103
Tabela 23 - Requisito FE03	104
Tabela 24 - Requisito FE04	104
Tabela 25 - Requisito FE05	105
Tabela 26 - Requisito FE06	105
Tabela 27 - Requisito FE07	106
Tabela 28 - Requisito FE08	106
Tabela 29 - Requisito FUI01	107
Tabela 30 - Requisito FUI02	107
Tabela 31 - Requisito FUI03	108
Tabela 32 - Requisito FUI04	109
Tabela 33 - Requisito FUI05	109
Tabela 34 - Requisito FUI06	109
Tabela 35 - Requisito FUI07	110
Tabela 36 - Requisito FUI08	111
Tabela 37 - Requisito FUI09	111
Tabela 38 - Requisito FA01.....	112
Tabela 39 - Requisito FA02.....	112
Tabela 40 - Requisito FA03.....	113

Tabela 41 - Requisito AF01.....	114
Tabela 42 - Requisito AM01	114
Tabela 43 - Requisito AM02	115
Tabela 44 - Requisito AV01	116
Tabela 45 - Requisito AV02	116
Tabela 46 - Requisito AV02	117
Tabela 47 - Requisito UPN01.....	118
Tabela 48 - Tempo de execução dos pedidos realizados para obter os dados de veículo do utilizador	118
Tabela 49 - Requisito UPN02.....	119
Tabela 50 - Requisito UMN01	120
Tabela 51 - Requisito UMN02	120
Tabela 52 - Requisito UCQ01	121
Tabela 53 - Escala de avaliação utilizada no questionário	122
Tabela 54 – Resultados do questionário para o requisito UCQ01	122
Tabela 55 - Requisito UCQ02	123
Tabela 56 - Resultados do questionário para o requisito UCQ02	123
Tabela 57 - Requisito UCQ03	124
Tabela 58 - Resultados do questionário para o requisito UCQ03	124
Tabela 59 - Requisito UIN01.....	125
Tabela 60 - Requisito UIN02.....	125
Tabela 61 - Requisito UIN03.....	125
Tabela 62 - Objetivos propostos e a respetiva percentagem de realização	127

Lista de Código

Código 1 - Exemplo de um objeto JSON utilizador para registar uma viatura na plataforma ...	76
Código 2 - Exemplo de um objeto JSON que contém a informação de um alerta.....	78
Código 3 - Exemplo de um objeto JSON obtido como resposta a um pedido de atualização de dados da viatura.....	79
Código 4 - Exemplo de resposta ao pedido de autenticação.....	80
Código 5 - Exemplo de resposta à execução de um comando num veículo.....	81
Código 6 - <i>Dockerfile</i> desenvolvido para o portal <i>web</i>	96
Código 7 - Excerto responsável por executar o portal <i>web</i> , o Centro de Controlo e a respetiva base de dados	97

Acrónimos e Símbolos

Lista de Acrónimos

NCD	<i>New Concept Development</i>
PaaS	<i>Product-as-a-Service</i>
B2B	<i>Business-to-Business</i>
QFD	<i>Quality Function Deployment</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
API	<i>Application Programming Interface</i>
UML	<i>Unified Modelling Language</i>
QEF	<i>Quantitative Evaluation Framework</i>
CEME	Comercializador de Energia para a Mobilidade Elétrica
HUD	<i>Head-up display</i>
HDMI	<i>High-Definition Multimedia Interface</i>
USB	<i>Universal Serial Bus</i>
SSD	<i>System Sequence Diagram</i>
AHP	<i>Analytic Hierarchy Process</i>
IC	Índice de Consistência
RC	Razão de Consistência
CCVE	Centro de Controlo para Veículos Elétricos
UI	<i>User Interface</i>

RGPD	Regulamento Geral sobre a Proteção de Dados
FF	<i>Functionality Functional</i>
FE	<i>Functionality Features</i>
FUI	<i>Functionality User Interaction</i>
FA	<i>Functionality Account</i>
AF	<i>Adaptability Features</i>
AM	<i>Adaptability Maintenance</i>
AV	<i>Adaptability Versatility</i>
UPN	<i>Usability Portal Navigation</i>
UMN	<i>Usability Map Navigation</i>
UCQ	<i>Usability Content Quality</i>
UIN	<i>Usability Integrity</i>

1 Introdução

O primeiro capítulo deste documento tem como objetivo enquadrar o projeto realizado, incluindo uma apresentação do mesmo, a descrição do problema e os contributos resultantes da elaboração da solução. Por fim, é também apresentada a estrutura deste documento, juntamente com uma breve descrição do conteúdo abordado em cada capítulo.

1.1 Enquadramento

Este documento foi realizado no âmbito da unidade curricular de Tese/Dissertação/Estágio (TMDEI), do Mestrado em Sistemas Gráficos e Multimédia do Instituto Superior de Engenharia do Porto (ISEP), e descreve o projeto desenvolvido na empresa EVIO, que tem o objetivo de desenvolver uma plataforma para a gestão e monitorização de veículos elétricos, através da utilização dos dados recolhidos diretamente das viaturas.

A EVIO é uma empresa que fornece serviços e soluções na área de carregamento de veículos elétricos para empresas, espaços privados e utilizadores de viaturas elétricas. Assim, esta tem como principal objetivo o desenvolvimento de soluções ambientais sustentáveis, com o foco na mobilidade elétrica, de modo a auxiliar na rentabilização de postos de carregamento para empresas e espaços privados, fornecendo aos utilizadores de veículos elétricos maior diversidade no carregamento elétrico das suas viaturas. Um objetivo secundário da empresa é informar os utilizadores da aplicação EVIO das consequências financeiras e ambientais que estão associadas à utilização de veículos elétricos.

De modo a desenvolver novas funcionalidades e melhorar os serviços que a empresa presta atualmente aos seus clientes, é necessário realizar a integração com veículos elétricos, de modo a ser possível aceder aos dados destas viaturas e disponibilizar estes dados aos clientes numa solução. Para tal, tornou-se necessário o desenvolvimento de uma plataforma de monitorização que permita a um cliente o registo e gestão dos seus veículos, o processamento dos dados recolhidos de cada veículo e a visualização simples e eficaz desses dados. Ao longo deste documento, esta solução será denominada de Centro de Controlo para Veículos Elétricos (CCVE).

1.2 Apresentação do projeto

Este projeto consiste no desenvolvimento de uma plataforma que irá permitir a comunicação com o *software* disponibilizado pelas marcas de veículos, de forma a ser possível a comunicação com veículos elétricos, de modo a aceder aos dados e interagir com estas viaturas remotamente. De forma a facilitar a integração com futuras marcas de veículos, deverá ser adotada uma arquitetura baseada em microsserviços e será necessária a existência de uma camada de abstração, responsável pelo processamento dos dados a serem recebidos pelos veículos de diversas marcas automóveis.

Esta plataforma também deverá disponibilizar um portal *web* que permita a um utilizador registar veículos na plataforma e gerir remotamente uma frota de viaturas, disponibilizando funcionalidades que permitam monitorizar a sua posição geográfica, estado de carregamento, percentagem de bateria e executar comandos que controlam remotamente o veículo. De modo a auxiliar na deteção de falhas ou na manutenção das viaturas, deverão ser desenvolvidos algoritmos baseados na análise dos dados provenientes dos veículos, notificando os utilizadores das conclusões obtidas.

Após o desenvolvimento da plataforma, deverá ser possível desenvolver novas funcionalidades e melhorar os serviços já existentes para os utilizadores, sendo o desenvolvimento dessas novas alterações baseadas no processamento dos dados recolhidos das viaturas que se registaram previamente na plataforma.

1.3 Descrição do Problema

Tal como foi descrito na secção 1.1, a EVIO oferece soluções na área da mobilidade elétrica, especificamente, na área do carregamento de veículos elétricos, com o objetivo de fornecer ao utilizador soluções que inovem o processo atual de carregamento de viaturas elétricas e que inovem e promovam o uso sustentável e informado de energia.

Atualmente, os dados que são apresentados aos clientes estão relacionados com as operações que estes realizam através da aplicação EVIO, tendo em consideração o posto de carregamento utilizado no carregamento do veículo elétrico, a data e hora em que ocorreu e o tipo de veículo utilizado.

De modo a desenvolver novas funcionalidades e melhorar os serviços que presta atualmente aos seus clientes, identificou-se a necessidade de realizar a integração com veículos elétricos, de modo a ser possível aceder aos dados destas viaturas diretamente e disponibilizar estes dados aos clientes numa aplicação, para que estes possam gerir e monitorizar os seus veículos.

Assim, o acesso a estes dados permitirá apresentar aos utilizadores informações mais precisas e específicas sobre os seus veículos, uma vez que ao estabelecer a comunicação com a viatura, será possível coletar e processar os dados gerados por essa viatura para calcular métricas que tenham em consideração o estado atual do veículo. Através desta comunicação também deverá ser possível executar comandos remotos na viatura (caso esta opção seja suportada) e a identificação de situações potencialmente anómalas ao correto funcionamento do veículo, as quais são notificadas ao utilizador.

Uma vez que a manutenção e estado de uma viatura depende do seu proprietário, através da coleção dos dados será possível apresentar informação mais realista e que tem em consideração o estado atual da viatura. Tal permitirá fornecer um serviço mais preciso e de maior qualidade para os seus clientes.

Desta forma, será possível fornecer aos utilizadores da EVIO uma solução que, através da recolha das métricas de um veículo, complemente as restantes aplicações da EVIO. Esta irá centralizar, em apenas uma aplicação, o acesso remoto de um utilizador aos seus veículos, deixando de ser necessário o uso das múltiplas aplicações, fornecidas pelas diferentes marcas automóveis, para aceder remotamente aos veículos.

Atualmente, não é possível aceder aos dados de um veículo diretamente, uma vez que existem normas de segurança que garantem a privacidade dos dados do utilizador e impedem o controlo remoto do veículo, sendo que é da responsabilidade da marca fabricante do automóvel fornecer um ponto de acesso destinado a outras entidades autorizadas. Deste modo, algumas empresas automóveis disponibilizam uma interface que expõe, através de serviços, o controlo das funcionalidades existentes na interação com as suas viaturas. No entanto, dependendo do tipo de marca automóvel, o processo de autenticação e comunicação com os respetivos serviços pode variar, dependendo dos requisitos de cada marca automóvel.

1.4 Contributos do projeto

Dado que a EVIO tem um *standard* muito elevado nos produtos que disponibiliza para o público e tem como principal objetivo incentivar a utilização sustentável de energia no carregamento de veículos elétricos, continua a trabalhar no desenvolvimento de soluções que permitam fornecer uma escolha mais sustentável às soluções existentes atualmente.

Deste modo, o resultado esperado deste projeto é disponibilizar aos clientes da EVIO uma plataforma que contribui para:

- A integração com o *software* atualmente disponibilizado pelas marcas de veículos elétricos e que permite a recolha e registo dos dados gerados durante utilização do veículo, tais como a percentagem de bateria, a localização e valores associados à manutenção do veículo. Esta integração, dependendo do tipo de veículo, também pode permitir a execução de comandos remotos, tais como abrir ou fechar a viatura.
- Permitir que o *software* da EVIO, já existente, possa ser facilmente integrado com este, de modo que a informação disponibilizada pela utilização do veículo possa ser compilada no desenvolvimento de novas funcionalidades, que visam a utilização mais sustentável do meio de transporte, e complementar funcionalidades já existentes.
- Permitir, através de um portal *web*, gerir e monitorizar os veículos registados na plataforma por um utilizador, de modo a ser possível aceder remotamente aos dados obtidos de viaturas de diversas marcas numa solução centralizada. Esta monitorização deve ser assistida por algoritmos de análise dos dados recolhidos e que são responsáveis por despoletar alertas para o utilizador. Estes mecanismos podem ser

automáticos ou configuráveis pelo utilizador. Este também deve possibilitar a execução remota de comandos nas viaturas, caso estas suportem essas funcionalidades.

1.5 Estrutura do documento

Este documento está dividido nos seguintes capítulos: Introdução, Estado da Arte, Contexto e Análise de Valor, Ambiente de Trabalho, Descrição Técnica, Desenvolvimento da solução, Avaliação e Conclusão.

No primeiro capítulo, estão presentes o enquadramento, a descrição do projeto e do problema, sendo de seguida apresentados os contributos que advêm da realização do projeto. Por último, é apresentada a respetiva estrutura do relatório.

No segundo capítulo, é apresentado ao leitor o Estado da Arte realizado para enquadrar este no tema do projeto e para o desenvolvimento do documento. Nesta secção, encontra-se uma introdução ao contexto da mobilidade elétrica em Portugal, bem como uma análise das funcionalidades das principais aplicações das marcas automóveis que são utilizadas na comunicação remota com veículos.

No terceiro capítulo, está presente a análise de valor, sendo apresentado ao leitor o processo de desenvolvimento do projeto, a proposta de valor do produto e o processo de decisão adotado para priorizar as soluções atuais que fornecem mais valor aos clientes e que define a prioridade de integração para o projeto. Neste capítulo, encontra-se também presente a visão da solução para o problema proposto.

No quarto capítulo, estão incluídos os temas relacionados com o planeamento, gestão do projeto e a descrição das tecnologias e ferramentas de apoio utilizadas na sua elaboração e na redação deste documento. Neste capítulo, também é abordada a metodologia de investigação utilizada na procura de informação científica.

No quinto capítulo, é apresentado ao leitor a análise realizada ao problema, na qual são apresentados os requisitos funcionais e não funcionais identificados, a descrição desses requisitos e os respetivos diagramas de sequência que contêm as comunicações entre o utilizador e o sistema. Neste capítulo, são também apresentados o modelo de domínio, a vista

lógica e a vista de implantação. Por último, este também descreve os métodos utilizados para a avaliação do projeto.

No sexto capítulo, é descrito o desenvolvimento para cada um dos diferentes componentes que compõem a solução implementada, descrevendo em detalhe o processo de implementação para os requisitos funcionais identificados na análise e o processo de implantação.

No sétimo capítulo, é apresentada ao leitor a descrição detalhada do método de avaliação utilizado para determinar a percentagem de conclusão da solução e as métricas definidas para os requisitos implementados estão de acordo com os requisitos identificados na análise do projeto.

No oitavo capítulo do documento, são apresentadas ao leitor as conclusões finais do documento. Neste capítulo, são elaboradas as comparações finais para o estado de conclusão dos objetivos estabelecidos no início do desenvolvimento do projeto, as limitações encontradas durante o desenvolvimento da solução, os contributos do produto final e a descrição das tarefas que compõem o trabalho a ser realizado no futuro.

2 Estado da Arte

Neste capítulo do documento é apresentado o estado da arte do tema em análise. Deste modo, são apresentados ao leitor os temas dos veículos elétricos e da mobilidade elétrica em Portugal, bem como as principais soluções disponibilizadas atualmente pelas marcas automóveis para permitir a comunicação remota com veículos.

2.1 Mobilidade Elétrica

2.1.1 Veículo Elétrico

O início do desenvolvimento do motor elétrico remonta a 1829, onde o húngaro Ányos Jedlik foi responsável pela criação do primeiro motor elétrico. Em 1835, Thomas Davenport foi responsável pelo desenvolvimento do primeiro veículo elétrico (EDP Comercial, 2020).

Um veículo elétrico é um tipo de meio de transporte alimentado por energia elétrica armazenada em baterias e propulsionado por um motor elétrico, não sendo assim responsável pela emissão de CO₂ durante a sua utilização. São apenas considerados elétricos, os automóveis que dependem a 100% de eletricidade para a sua deslocação, sem a assistência de outros motores de combustão (EDP Comercial, 2020). Caso uma viatura disponha de um motor elétrico e de um motor de combustão, então esta é denominada de veículo híbrido. Atualmente, existem dois tipos de automóveis híbridos:

- **Híbrido:** Para este tipo de veículos, o motor elétrico carrega exclusivamente através da energia regenerada pela viatura durante a utilização do motor de combustão. Dependendo do tipo de viatura, a regeneração das baterias é obtida através da utilização normal da viatura, sendo o valor de regeneração superior em travagens a acelerações. Deste modo, uma aceleração extremamente rápida com um motor de combustão causa um grande consumo de combustível, o que leva a uma maior produção de CO₂, quando comparada com uma deslocação em velocidade constante (EDP Comercial, 2020).

- **Híbrido *plug-in*:** Para este tipo de viaturas, o motor de combustão funciona de forma independente do motor elétrico, sendo o motor elétrico utilizado em deslocações de curta distância, principalmente em ambiente urbano. De modo a utilizar o motor elétrico, é necessário carregar as baterias do automóvel num posto de carregamento, no qual um utilizador conecta o cabo de carregamento ao adaptador *plug-in* da viatura para iniciar a carga (EDP Comercial, 2020).

Tendo em consideração os veículos elétricos e híbridos *plug-in*, estes podem ser carregados numa habitação através de um posto de carregamento residencial, também conhecido como *wallbox*, ou então através da utilização das tomadas convencionais já existentes. Para tal, é necessário que a energia fornecida à habitação seja elevada o suficiente para suportar o carregamento de viaturas sem afetar a infraestrutura elétrica existente. Para além da existência de opções de carregamento pessoais, também existem redes públicas e privadas que disponibilizam postos de carregamento para utilização. Os utilizadores de viaturas elétricas podem usar estes postos através da utilização de um cartão fornecido pelo Comercializador de Energia para a Mobilidade Elétrica (CEME) que seja válido nessa rede (EDP Comercial, 2020).

2.1.2 Comercializadores para a mobilidade elétrica (CEME)

Um CEME é uma entidade detentora de um registo que permite a atividade de comercialização de energia para a mobilidade elétrica em Portugal. Esta atividade consiste na compra e venda de energia elétrica no mercado retalhista em Portugal (O que é um CEME?, s.d.).

De modo a ter acesso à rede de postos de carregamento desse CEME, públicos ou privados, que podem ser utilizados para o carregamento de viaturas é necessário requisitar um cartão CEME, podendo este ter um formato físico ou digital.

Atualmente, a MOBI.E é a entidade em Portugal que é exclusivamente responsável pela distribuição e venda de energia em postos de carregamento público, sendo assim necessária uma parceria entre os operadores dos postos e esta entidade para que estes postos possam ser utilizados (MOBI.E, s.d.).

As redes de postos privados, tal como a rede da Tesla, são responsáveis pela manutenção dos seus postos de carregamento, sendo também responsáveis pela gestão de acesso dos seus utilizadores a esses postos.

2.2 Sistema de Infoentretenimento

O sistema de infoentretenimento de um veículo é composto pela combinação dos diferentes sistemas que fornecem entretenimento e informação ao condutor e aos passageiros da viatura. Estes sistemas fornecem essa informação através da utilização de interfaces de áudio e de vídeo, ecrãs táteis, botões e comandos de voz. Estes têm como objetivo oferecer uma experiência de condução mais segura, produtiva e informada (Kozyra, 2019) .

Desde modo, nos modelos mais recentes, este sistema é composto por diversos tipos de tecnologias que comunicam entre si para partilhar e apresentar informação, tais como (Kozyra, 2019):

- **HUD (*Head-up display*):** Ecrã transparente que apresenta ao condutor de um veículo informação, sem ser necessário perder a concentração na estrada;
- **Ecrã Tátil:** Ecrã que fornece dados e informação complexa em formato de gráficos dinâmicos ao condutor. Permite a interação do utilizador com o sistema;
- **Emparelhamento com telemóvel:** Os automóveis têm funcionalidades que permitem a integração de um telemóvel com o sistema de infoentretenimento;
- **Suporte multimédia:** Dependendo do modelo do veículo, estes podem suportar *Bluetooth*, cabo *High-Definition Multimedia Interface* (HDMI) e cabo *Universal Serial Bus* (USB) para partilha e apresentação de material multimédia no veículo;
- **Funcionalidades avançadas de assistência:** Dependendo do modelo da viatura, podem ser fornecidas funcionalidades que auxiliam os condutores na sua condução, tais como a assistência no estacionamento, controlo climático e controlo por voz.

Assim, o sistema infoentretenimento é responsável pelo controlo dos dados gerados pelo veículo, derivados da sua utilização. Através deste sistema, é possível partilhar essa informação com a marca fabricante do automóvel, sendo esta responsável pelo seu registo para, mais tarde, poder ser utilizada noutras soluções que beneficiem destes valores. Através deste sistema, é também possível estabelecer a comunicação da marca fabricante com a viatura, sendo possível a execução de comandos remotos.

2.3 Aplicações existentes para acesso remoto a veículos

Diversas marcas automóveis disponibilizam aplicações que permitem aos seus clientes aceder, de forma remota, aos seus veículos. Nesta secção do documento vão ser analisadas cinco dessas aplicações. De modo a determinar que marcas devem ser consideradas no levantamento de informação, foram identificados os modelos elétricos que tiveram mais unidades vendidas no ano de 2019, na Europa (Demandt, s.d.) . Esta informação é apresentada na Tabela 1.

Tabela 1 - Veículos elétricos mais vendidos em 2019 na Europa

Modelo	Unidades vendidas na Europa
Tesla Model 3	95.168
Renault Zoe	45.129
Nissan Leaf	31.792
Volkswagen e-Golf	28.710
BMW i3	23.882

Assim, nas seguintes secções do documento vão ser apresentadas em detalhes as principais soluções desenvolvidas pelas marcas de automóveis identificadas na Tabela 1 - Tesla, Renault, Nissan, Volkswagen, BMW - para comunicação remota com os veículos.

2.3.1 ConnectedDrive

Esta é a solução que a BMW disponibiliza aos seus clientes, de modo a ser possível aceder e controlar um veículo à distância.

De modo a garantir este serviço, a BMW realiza o processamento das seguintes métricas que recolhe das viaturas (BMW, 2018):

- Dados do estado do veículo (quilometragem, voltagem da bateria, estado das portas, etc.);
- Dados da posição e movimentação (tempo, posição, velocidade, etc.);
- Dados de manutenção do veículo (próxima data de manutenção, nível do óleo, estado dos travões, etc.);

- Dados dinâmicos do tráfego (engarrafamento, sinalização, parques de estacionamento, etc.);
- Informação ambiental (temperatura, ocorrência de chuva, etc.);
- Informação pessoal do utilizador (notícias, email, fornecedores de áudio, etc.);
- Informações dos sensores do veículo (radar, ultrassom, gestos, fala, etc.).

Deste modo, este serviço disponibiliza as seguintes funcionalidades remotas (BMW, s.d.):

- Operação à distância através do dispositivo móvel;
- Trancar e destrancar portas, procurar o veículo e regular a temperatura;
- Transmissão direta de destinos para a pesquisa local do veículo;
- Visualização remota das imediações atuais do veículo;
- Perspetiva e *zoom* livre sobre o veículo.

Dependo do modelo do veículo, do ano em que foi construído e dos pacotes extras disponíveis para a viatura, algumas funcionalidades acima referidas podem não estar disponíveis. Na Figura 1, está presente a interface da aplicação e algumas das funcionalidades acima mencionadas.

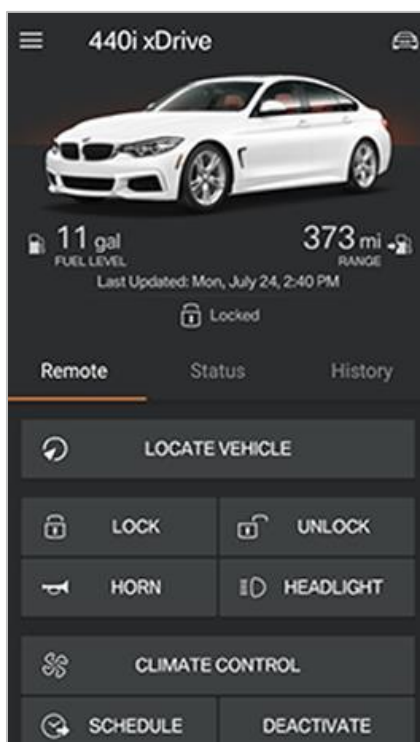


Figura 1 - Interface da aplicação ConnectedDrive

2.3.2 Tesla App

A Tesla é uma empresa norte americana que opera nas áreas da construção automóvel e armazenamento de energia. É responsável pelo desenvolvimento, produção e venda de veículos elétricos, postos de carregamento, componentes para motores e transmissões para automóveis elétricos.

De modo a permitir que os seus clientes monitorizem e controlem remotamente os seus veículos, a Tesla criou uma aplicação móvel com as seguintes funcionalidades (Tesla, s.d.):

- Destranque do veículo e condução sem chave;
- Assistência técnica, permitindo o agendamento de novas marcações de manutenção e visualizar o histórico das que já ocorreram;
- Controlo da climatização do veículo;
- Controlo do carregamento do veículo;
- Localização via GPS;
- Modo de serviço de estacionamento.

Na Figura 2, está presente a página inicial da aplicação, onde estão presentes algumas das funcionalidades acima referidas.



Figura 2 - Interface da aplicação Tesla

De modo a analisar o comportamento do serviço da Tesla, foi consultada a documentação não oficial de bibliotecas de terceiros que foram previamente desenvolvidas. Estas bibliotecas, denominadas de Tesla JSON API (Unofficial) (Dorr, s.d.) , TeslaJS (TeslaJS, s.d.) e node-tesla-api (node-tesla-api, s.d.) serviram de exemplo, ao longo do desenvolvimento do projeto, de forma a aproximar o produto final à solução original da Tesla.

Deste modo, nos próximos subcapítulos são descritas as principais funcionalidades presentes na documentação não oficial. Para tal, de modo a simplificar a apresentação desta informação, as funcionalidades podem ser agrupadas nos seguintes segmentos:

- Autenticação;
- Dados;
- Comandos.

2.3.2.1 Autenticação

A Tesla utiliza um sistema de autenticação para validar a identidade de quem tenta interagir com o seu sistema. Deste modo, um utilizador válido ao autenticar-se no sistema, obtém um *token* de autenticação. Cada interação com o sistema necessita de ter o *token* de autenticação obtido inicialmente, de modo a garantir a segurança e autenticidade do pedido.

2.3.2.2 Dados

A Tesla disponibiliza múltiplos comandos para obter a informação dos veículos associados a um utilizador. Estes permitem obter:

- Dados sobre o estado de carregamento e configurações da viatura;
- Dados sobre o sistema de controlo de clima do veículo;
- Dados sobre a condução e posição da viatura;
- Dados sobre o estado físico do veículo.

Através destes pedidos é possível obter diversa informação sobre uma viatura, tais como a percentagem da bateria, o estado de carregamento, a sua localização geográfica, o estado da temperatura no seu interior e exterior, percentagem de carregamento e configurações gerais.

2.3.2.3 Comandos

A Tesla permite a interação remota com o veículo através de comandos que podem ser executados na aplicação. A seguir encontram-se listados alguns desses comandos:

- Início remoto da viatura;
- Desbloqueio das portas;
- Desbloqueio da mala;

- Controlar o carregamento do veículo, como iniciar/terminar carregamentos ou desligar o cabo da porta de carregamento;
- Controlar o sistema de climatização.

2.3.3 We Connect Go

A aplicação “We Connect Go” é a solução desenvolvida pela Volkswagen para permitir a um utilizador, através de um dispositivo móvel, conectar-se, via *Bluetooth*, à sua viatura, permitindo que aceda a informação e funcionalidades à distância (Volkswagen, s.d.).

Deste modo, a aplicação permite (Volkswagen, s.d.):

- Aceder às informações da viatura;
- Aceder ao histórico do registo de viagens e monitorização do depósito/bateria;
- Requisitar assistência em viagem;
- Utilizar serviços e marcar sessões de manutenção;
- Utilizar funcionalidades de assistência após o estacionamento, em que a aplicação apresenta ao utilizador a localização mais recente do seu veículo, o caminho mais rápido para este e regista o tempo de estacionamento;
- Selecionar desafios, nos quais um utilizador recebe pontos e troféus de acordo com o seu modo de condução, pelas visitas à manutenção e pela utilização da aplicação.

Na Figura 3, está presente a interface da aplicação, onde estão visíveis algumas das funcionalidades acima descritas. Nesta imagem também está presente um dispositivo (DataPlug) que pode ser adicionado a um veículo para permitir a conexão remota à viatura e estabelecer comunicação com a aplicação.

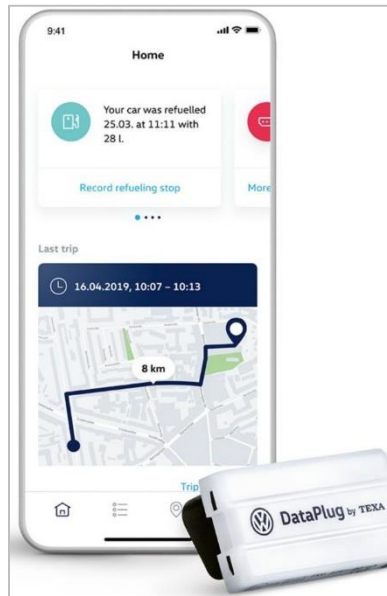


Figura 3 - Interface da aplicação We Connect Go

2.3.4 My Renault

A aplicação “My Renault” permite a sincronização dos serviços remotos disponibilizados pela Renault com a viatura de um utilizador, sendo que esta varia de acordo com o modelo do veículo. Deste modo, o utilizador, através da aplicação, consegue (Renault, s.d.):

- Aceder à bateria e estado do carregamento do veículo;
- Programar carregamentos;
- Aceder ao controlo da climatização do veículo;
- Programar a climatização do veículo;
- Encontrar postos de carregamento disponíveis na sua proximidade;
- Planear rotas, identificando postos de carregamento disponíveis para utilização na rota.

Na Figura 4, é apresentada a interface principal da aplicação, onde são visíveis as principais funcionalidades que esta disponibiliza ao utilizador.



Figura 4 - Interface da aplicação My Renault

2.3.5 NissanConnect

A aplicação "NissanConnect" é a solução apresentada aos proprietários de veículos da marca Nissan que desejam integrar a sua viatura com os serviços remotos disponibilizados pela marca. Deste modo, e dependendo do modelo da viatura, a aplicação permite ao utilizador (Nissan, s.d.):

- Localizar o veículo;
- Aceder ao histórico de destinos visitados pelo utilizador;
- Executar comandos remotos, tais como a abertura sem chave ou o início remoto da viatura;
- Utilizar comandos de voz.

Na Figura 5, é visível a interface principal da aplicação.

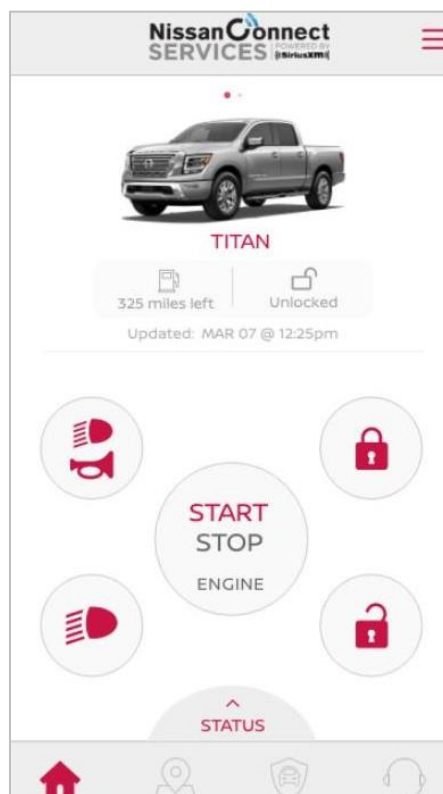


Figura 5 - Interface da aplicação NissanConnect

3 Contexto e Análise de Valor

Neste capítulo do documento, é apresentada uma análise detalhada da oportunidade gerada pela existência do problema apresentado na secção 1.3. Este capítulo também apresenta a análise de valor realizada durante o desenvolvimento do projeto e, por fim, a visão da solução proposta como resolução do problema.

3.1 Análise de valor

A análise de valor permite aplicar um conjunto de processos sistematizados de análise e de avaliação, com o objetivo de determinar o valor do projeto a ser desenvolvido e de que modo o custo pode ser reduzido, sem sacrificar a qualidade do produto final.

De modo a efetuar esta análise de valor e aplicá-la ao Centro de Controlo para Veículos Elétricos, utilizou-se o modelo NCD (*New Concept Development Model*), desenvolvido por Peter Koen (Koen, et al., 2001).

3.1.1 New Concept Development Model (NCD)

Tendo em consideração o modelo NCD, foram identificados os seguintes elementos envolvidos no processo de desenvolvimento do negócio associado ao projeto:

- **Identificação de Oportunidade** (*Opportunity Identification*): A oportunidade surgiu quando foi identificada na EVIO a necessidade de ter acesso a dados provenientes dos veículos elétricos.
- **Análise de Oportunidade** (*Opportunity Analysis*): Tendo em conta as funcionalidades que são disponibilizadas atualmente pelos produtos da EVIO aos clientes, averiguou-se a possibilidade de aceder aos dados resultantes da utilização das viaturas destes, de modo a desenvolver novas funcionalidades para melhorar os serviços prestados atualmente pela empresa.
- **Génese de Ideias** (*Idea Genesis*): Gerir e monitorizar um conjunto de viaturas elétricas, através do acesso aos dados de cada viatura, independentemente da marca automóvel.

Analisar os dados relacionados com a utilização da bateria, de modo a identificar degradação desta e garantir o acesso aos dados de diversas viaturas, independentemente da marca, em apenas uma solução.

- **Seleção da Ideia (*Idea Selection*):** Desenvolvimento de uma plataforma que permita a um utilizador registar os seus veículos e onde é apresentada informação sobre as viaturas, enviadas notificações baseadas nos dados recolhidos e permitir a utilização de comandos remotos sobre esses veículos. Esta também deve fornecer dados estatísticos e alertas baseados na análise das métricas recolhidas. Esta plataforma deve centralizar o acesso a essa informação, dividindo a solução em dois componentes, sendo o primeiro responsável pela integração dos veículos e o segundo pela interação com o utilizador e apresentação de informação.
- **Conceito e Desenvolvimento de Tecnologia (*Concept and Technology Development*):** De modo a inovar o processo envolvente na visualização aos dados dos veículos e monitorização de viaturas, desenvolveu-se a ideia da criação de uma plataforma que permita a integração com o *software* disponibilizado por diversas marcas de automóveis. Através desta integração é possível aceder aos dados e comandos de uma determinada viatura, recolhendo essa informação para cada veículo e centralizando a solução numa única plataforma. Para além disso, juntamente com o conhecimento adquirido pela empresa EVIO e através do processamento das métricas recolhidas, podem ser desenvolvidas novas funcionalidades, com o foco na mobilidade sustentável e na identificação de ações preventivas nas viaturas. Deste modo, através da solução a implementar, é possível centralizar apenas numa ferramenta a gestão e monitorização de múltiplos veículos elétricos.

3.1.2 Proposta de Valor

O Centro de Controlo para Veículos Elétricos é um *software* que permite aos utilizadores monitorizar, de forma remota, os dados provenientes dos seus veículos, tais como a localização geográfica, consumos relacionados com a deslocação do carro e utilização da bateria e, dependendo das funcionalidades suportadas pelo automóvel, pode também permitir a execução de comandos.

Assim sendo, o modelo de negócio do CCVE enquadra-se como um *Product-as-a-Service* (PaaS), uma vez que este é um produto disponibilizado para todos, mas apenas acessível através de um

modelo de subscrição, no qual os utilizadores têm de ter uma subscrição ativa e válida para utilizar o serviço.

Deste modo, o modelo de negócio do CCVE enquadra-se num modelo *Business-to-Business* (B2B) e também *Business-to-Consumer* (B2C). O público-alvo do serviço é dividido em dois segmentos de utilizadores: individuais e empresas, existindo diferentes modelos de subscrição para cada uma das diferentes entidades. Para os utilizadores individuais, o uso da plataforma permite o registo e visualização dos dados dos veículos registados, apresentação de estatísticas baseadas no processamento destes dados e criação de alertas, sendo que as funcionalidades disponibilizadas podem variar de acordo com o modelo de subscrição contratado. Por outro lado, para empresas, a plataforma oferece uma solução de gestão de frota, disponibilizando avisos preventivos baseados no processamento dos dados recolhidos dos veículos, localização geográfica e, para casos em que o veículo permita, a execução remota de comandos.

De modo a definir o valor gerado por este *software* para os seus utilizadores, estão presentes na Tabela 2 os benefícios e sacrifícios, de acordo com a definição de perceção de valor por Lapierre (Lapierre, 2000).

Tabela 2 - Benefícios e sacrifícios para o cliente

	Produto	Serviço	Relação
Benefício	Qualidade do produto	Flexibilidade	Segurança
	Desempenho	Segurança	Confiança
	Competência técnica		
Sacrifício	Preço	Preço	Utilização

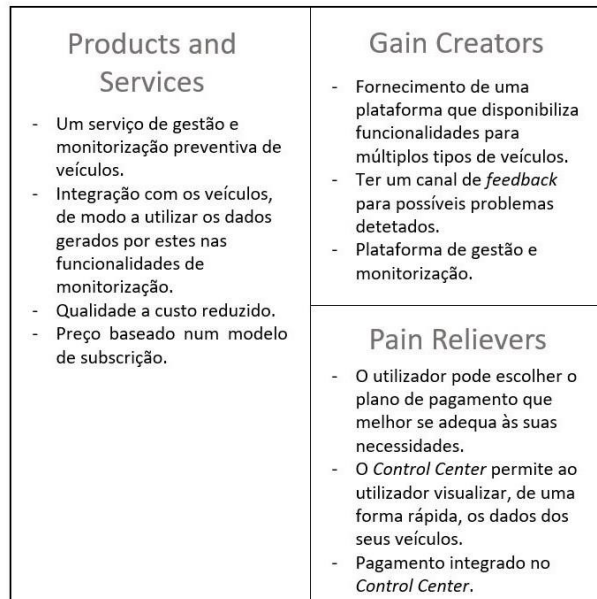
Assim, os benefícios e sacrifícios apresentados na Tabela 2 são justificados da seguinte forma:

- **Benefícios do produto:**
 - **Qualidade do produto:** O cliente pode confiar no produto que está a utilizar e nos dados que são apresentados.
 - **Desempenho:** Nota-se uma preferência, por parte dos clientes, de uma plataforma que se desempenha de forma igual, independentemente do número de veículos por utilizado.

- **Competência Técnica:** O produto que é apresentado ao cliente e as funcionalidades que são disponibilizadas nesse produto foram desenvolvidas por especialistas na área da mobilidade elétrica.
- **Benefícios do serviço:**
 - **Flexibilidade:** O serviço disponibilizado ao cliente está adaptado e disponível para ser utilizado em diversos dispositivos. O *feedback* que o cliente decide reportar também é considerado no melhoramento do serviço apresentado.
 - **Segurança:** Os dados que são coletados dos veículos dos clientes, os dados de pagamento, de autenticação e informação pessoal fornecida são protegidos e, de acordo com as leis de RGPD, não permitem a identificação dos seus proprietários.
- **Benefícios da relação:**
 - **Segurança:** O produto assegura a segurança dos dados do cliente.
 - **Confiança:** Os clientes acreditam que os dados que são fornecidos pelo produto são precisos e correspondem à verdade.
- **Sacrifícios do produto:**
 - **Preço:** O preço associado ao desenvolvimento do produto, a utilização de serviços que têm custos monetários e a criação de parcerias com as marcas de veículos elétricos, tem um custo associado e que tem de ser gerido.
- **Sacrifícios do serviço:**
 - **Preço:** Uma vez que o produto é apresentado aos seus clientes como um serviço, existem custos associados ao alojamento e manutenção deste serviço, de modo a garantir a disponibilidade constante e o correto funcionamento desta, independentemente do número de utilizadores.
- **Sacrifícios da relação:**
 - **Utilização:** A continuação da utilização do produto por parte dos clientes está dependente do pagamento do plano de subscrição associado à conta do cliente.

De modo a complementar a informação presente na tabela de benefícios e sacrifícios, foi também aferida, de uma forma mais detalhada, a relação entre o produto, o cliente e os respetivos benéficos (*Gains*) e sacrifícios (*Pains*) resultantes. Na Figura 6, está presente o modelo *Canvas* para a proposta de valor.

Produto



Cliente



Figura 6 - Modelo *Canvas* da proposta de valor

De modo a garantir que o produto vai de encontro às necessidades dos clientes analisadas na secção 3.1 deste documento, e com objetivo de garantir que o desenvolvimento deste tem constantemente estas necessidades em consideração, foi aplicada a técnica *Quality Function Deployment* (QFD).

A técnica QFD consiste, numa primeira fase, na tradução das necessidades e desejos dos clientes em requisitos funcionais, de modo que a relação estabelecida entre cada um destes elementos seja considerada, numa segunda fase, no desenvolvimento do produto.

Na Figura 7, é apresentado o diagrama *House of Quality*, utilizado para representar visualmente a aplicação da técnica QFD.

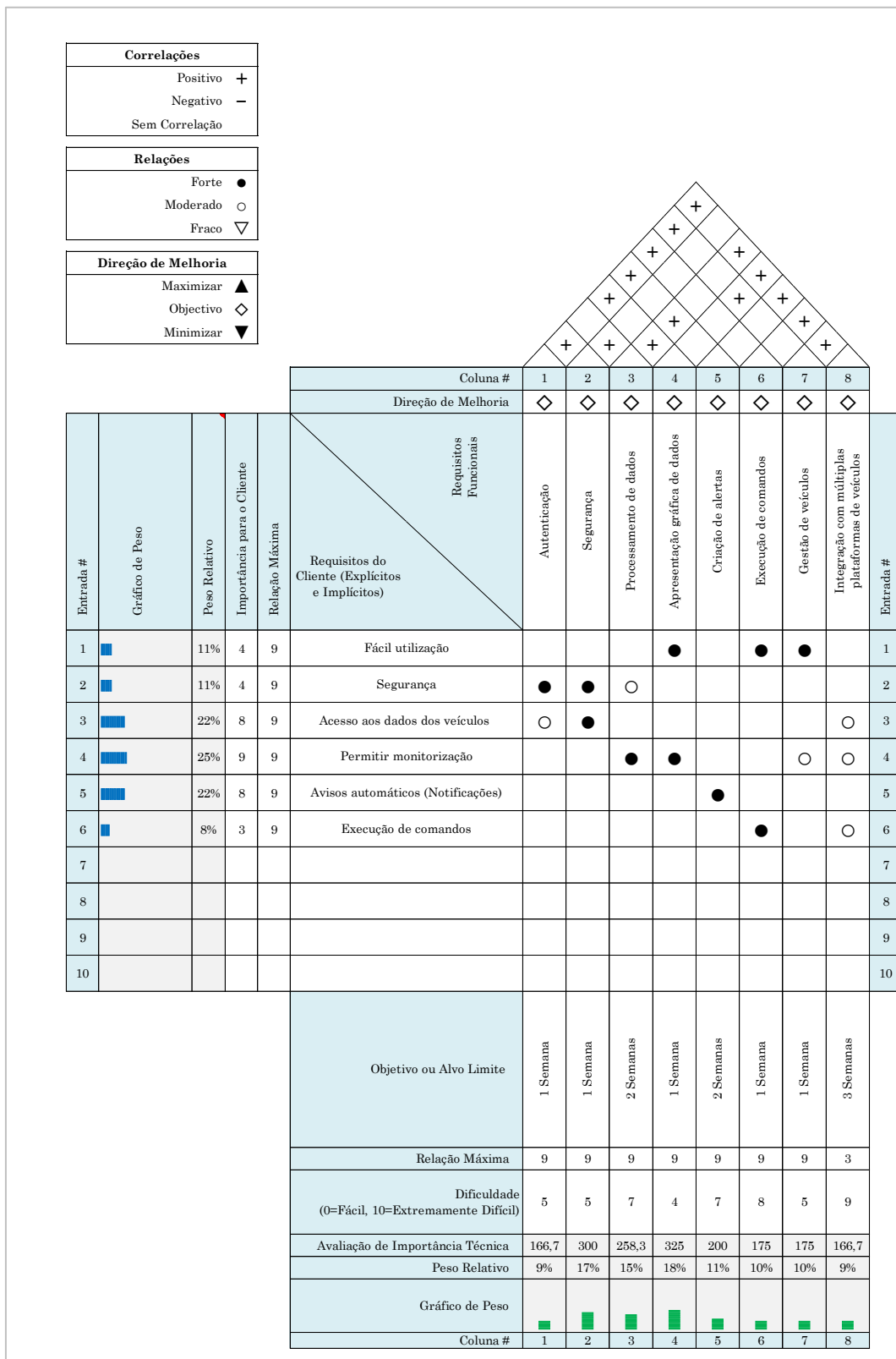


Figura 7 - House of Quality (QFD)

3.1.3 Priorização dos mecanismos de comunicação com veículos

Para que seja possível iniciar o processo de visão da solução, é necessário priorizar as tecnologias cuja integração com a plataforma permite fornecer mais valor ao cliente. Este valor é determinado através do desenvolvimento de novas funcionalidades que vão ao encontro das necessidades dos utilizadores. Para tal, foi utilizado o Método Análise Hierárquica (AHP – *Analytic Hierarchy Process*) como método de auxílio na tomada desta decisão, tendo em consideração as tecnologias apresentadas na secção 2.3 deste documento. Este método permite derivar o rácio de escalas entre comparações, sendo que estas comparações podem ser obtidas através de medições reais ou através de uma escala fundamentada que reflita a força relativa entre a medição e a dependência entre os elementos da sua estrutura (Saaty, 1987). Deste modo, é possível dividir o problema da decisão em diversos níveis hierárquicos, o que facilita a sua compreensão e avaliação.

A aplicação deste método é composta por diversas fases. Na primeira fase, a decisão a tomar é definida e decomposta numa hierarquia constituída, pelo mínimo, por um objetivo, critérios e alternativas. Assim, os fatores a considerar para análise são a execução de comandos remotos, conectividade com o veículo, acesso a dados e performance durante o processamento desses dados. Os valores atribuídos às prioridades são baseados nas relações presentes na Tabela 3 (Saaty, 1987).

Tabela 3 - Escala fundamental

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 à 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

Numa segunda fase, é necessário desenvolver a matriz de comparação, presente na Tabela 4. Esta matriz permite apresentar as prioridades definidas para os fatores acima descritos.

Tabela 4 - Matriz de comparação

Critério	Comandos	Conectividade	Acesso a dados	Performance
Comandos	1	4	1/5	1/3
Conectividade	1/4	1	1/7	1/5
Acesso a dados	5	7	1	3
Performance	3	5	1/3	1

É então depois calculada a prioridade relativa de cada critério, presente na Tabela 5.

Tabela 5 - Somatório dos critérios da matriz de comparação

Critério	Comandos	Conectividade	Acesso a dados	Performance
Somatório	9.25	17	1.68	4.50

Em seguida, a matriz de comparação acima referida é normalizada, sendo a matriz normalizada apresentada na Tabela 6.

Tabela 6 - Matriz de comparação normalizada

Critério	Comandos	Conectividade	Acesso a dados	Performance
Comandos	0.11	0.23	0.12	0.07
Conectividade	0.03	0.06	0.08	0.05
Acesso a dados	0.54	0.41	0.60	0.66
Performance	0.32	0.30	0.20	0.22

Após obter a matriz de comparação normalizada, é então calculada a média aritmética dos valores de cada linha da matriz, obtendo como resultado o peso de cada critério (Tabela 7).

Tabela 7 - Peso dos critérios

Critério	Peso
Comandos	0.13
Conectividade	0.06
Acesso a dados	0.55
Performance	0.26

O próximo passo é o cálculo da Razão de Consistência (RC) para verificar a consistência dos dados. Para tal, é necessário calcular o vetor próprio, sendo posteriormente calculado o valor próprio. Na fórmula abaixo apresentada é calculado o vetor próprio, através da multiplicação da matriz normalizada pela matriz dos pesos por critérios, calculado no passo anterior.

$$\begin{bmatrix} 1 & 4 & 1/5 & 1/3 \\ 1/4 & 1 & 1/7 & 1/5 \\ 5 & 7 & 1 & 3 \\ 3 & 5 & 1/3 & 1 \end{bmatrix} \times \begin{bmatrix} 0.13 \\ 0.06 \\ 0.55 \\ 0.26 \end{bmatrix} = \begin{bmatrix} 0.55 \\ 0.22 \\ 2.38 \\ 1.11 \end{bmatrix}$$

Com o resultado obtido, é calculado o valor próprio:

$$(0.55 / 0.13 + 0.22 / 0.06 + 2.38 / 0.55 + 1.11 / 0.26) / 4 = 4.18$$

Com o cálculo do valor próprio, é possível calcular o Índice de Consistência (IC). Para tal, foi aplicada a fórmula:

$$IC = \frac{\lambda_{max} - n}{n - 1}$$

O valor de n é um valor médio de ICs derivado de uma amostra de matrizes recíprocas aleatoriamente geradas, dependendo da ordem da matriz utilizada. Na Tabela 8, estão presentes os valores previamente definidos como amostra para matrizes quadradas (Saaty, 1987).

Tabela 8 – Valores de IR para matrizes quadradas de ordem n

1	2	3	4	5	6	7	8	9	10
0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Os valores presentes na Tabela 8 são utilizados para obter a Razão de Consistência (RC). Este valor permite verificar a existência de emparelhamento dos dados de entrada, indicando se estes são confiáveis. Este valor é obtido pela fórmula:

$$RC = \frac{IC}{IR}$$

Deste modo, a matriz utilizada é de ordem 4, sendo que o valor de IR correspondente é de 0.90. Para determinar os valores de IC e RC foram realizado os seguintes cálculos:

$$IC = (4.18 - 4) / (4 - 1) = 0.06$$

$$RC = 0.06 / 0.90 = 0.07$$

Assim, e uma vez que o valor do IC e do RC são inferiores a 0.1, é possível concluir que os valores dos critérios utilizados são consistentes e os dados são confiáveis.

De forma a analisar cada tecnologia, foram repetidos os passos anteriores para cada um dos critérios a analisar. Para tal, é necessário desenvolver a matriz de comparação por critério, tendo em consideração as tecnologias que afetam essa decisão, calcular a normalização da matriz e, através do vetor próprio obtido previamente, determinar os valores para IC e RC.

I. Critério – Comandos

Tabela 9 - Matriz de comparação normalizada do critério “Comandos”

Comandos	Tesla	BMW	Renault
Tesla	1	5	4
BMW	1/5	1	2
Renault	1/4	1/2	1

$$\begin{bmatrix} 1 & 5 & 4 \\ 1/5 & 1 & 2 \\ 1/4 & 1/2 & 1 \end{bmatrix} \times \begin{bmatrix} 0.68 \\ 0.19 \\ 0.13 \end{bmatrix} = \begin{bmatrix} 2.16 \\ 0.59 \\ 0.40 \end{bmatrix}$$

$$IC = 0.05$$

$$RC = 0.08$$

Deste modo, os valores para IC e RC são inferiores a 0.1, logo é possível concluir que os valores dos critérios utilizados são consistentes e os dados são confiáveis.

II. Critério – Conectividade

Tabela 10 - Matriz de comparação normalizada do critério “Conectividade”

Conectividade	Tesla	BMW	Renault
Tesla	1	4	3
BMW	1/4	1	2
Renault	1/3	1/2	1

$$\begin{bmatrix} 1 & 4 & 3 \\ 1/4 & 1 & 2 \\ 1/3 & 1/2 & 1 \end{bmatrix} \times \begin{bmatrix} 0.62 \\ 0.22 \\ 0.16 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 0.70 \\ 0.47 \end{bmatrix}$$

$$IC = 0.05$$

$$RC = 0.09$$

Deste modo, os valores para IC e RC são inferiores a 0.1, logo é possível concluir que os valores dos critérios utilizados são consistentes e os dados são confiáveis.

III. Critério – Acesso a dados

Tabela 11 - Matriz de comparação normalizada do critério "Acesso a dados"

Acesso a dados	Tesla	BMW	Renault
Tesla	1	5	3
BMW	1/5	1	1/3
Renault	1/3	3	1

$$\begin{bmatrix} 1 & 5 & 3 \\ 1/5 & 1 & 1/3 \\ 1/3 & 3 & 1 \end{bmatrix} \times \begin{bmatrix} 0.63 \\ 0.11 \\ 0.26 \end{bmatrix} = \begin{bmatrix} 1.95 \\ 0.32 \\ 0.79 \end{bmatrix}$$

$$IC = 0.02$$

$$RC = 0.03$$

Assim, os valores para IC e RC são inferiores a 0.1, logo é possível concluir que os valores dos critérios utilizados são consistentes e os dados são confiáveis.

IV. Critério – Performance

Tabela 12 - Matriz de comparação normalizada do critério "Performance"

Performance	Tesla	BMW	Renault
Tesla	1	4	3
BMW	1/4	1	2
Renault	1/3	1/2	1

$$\begin{bmatrix} 1 & 4 & 3 \\ 1/4 & 1 & 2 \\ 1/3 & 1/2 & 1 \end{bmatrix} \times \begin{bmatrix} 0.62 \\ 0.22 \\ 0.16 \end{bmatrix} = \begin{bmatrix} 1.98 \\ 0.70 \\ 0.47 \end{bmatrix}$$

$$IC = 0.05$$

$$RC = 0.09$$

Deste modo, os valores para IC e RC são inferiores a 0.1, logo é possível concluir que os valores dos critérios utilizados são consistentes e os dados são confiáveis.

Após terem sido construídas as matrizes de comparação para cada critério da decisão, e também calculado o respetivo vetor próprio, é calculada a prioridade composta para todas as alternativas que foram abordadas.

$$\begin{bmatrix} 0.68 & 0.62 & 0.63 & 0.62 \\ 0.19 & 0.22 & 0.11 & 0.22 \\ 0.13 & 0.16 & 0.26 & 0.16 \end{bmatrix} \times \begin{bmatrix} 0.13 \\ 0.06 \\ 0.55 \\ 0.26 \end{bmatrix} = \begin{bmatrix} 0.63 \\ 0.15 \\ 0.21 \end{bmatrix}$$

Deste modo, é possível inferir que a alternativa Tesla (0.63) é a mais indicada para a integração com a plataforma, sendo aquela cujas funcionalidades permitem acrescentar mais valor ao produto. Em seguida, devem ser exploradas as opções Renault (0.21) e BMW (0.15), respetivamente, para possível integração na plataforma.

3.2 Análise de arquiteturas da solução

Tal como foi descrito na secção 1.1, deve ser adotada uma arquitetura baseada em microsserviços no desenvolvimento da plataforma. Esta é responsável pela integração que possibilita a comunicação com uma viatura, independentemente da marca automóvel, de forma a obter os dados relativos à sua utilização.

No entanto, de modo a verificar que adoção de uma arquitetura de microsserviços é a mais indicada para o projeto, foram analisadas as características de uma arquitetura monolítica e de uma arquitetura de microsserviços.

3.2.1 Arquitetura monolítica

A arquitetura monolítica é considerada a forma tradicional de desenvolvimento de aplicações. Uma aplicação monolítica é desenvolvida como uma única e indivisível unidade, sendo este tipo de solução composto por uma base extensiva de código e falta de modularidade, tal como é visível na Figura 8. Deste modo, alterações que sejam realizadas na aplicação afetam toda a estrutura da solução desenvolvida (Gnatyk, s.d.).

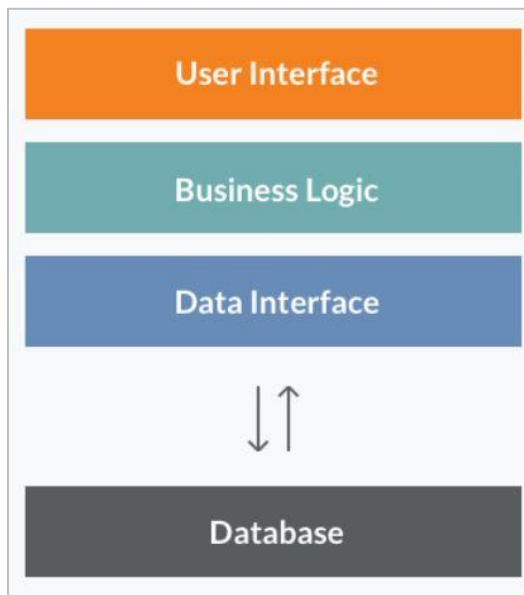


Figura 8 - Arquitetura monolítica (Gnatyk, s.d.)

Assim, foram identificados os seguintes pontos fortes e fracos de uma arquitetura monolítica (Gnatyk, s.d.).

- **Pontos Fortes**

- Redução das preocupações transversais à aplicação, uma vez que, como as funcionalidades estão centralizadas em apenas uma plataforma, as preocupações são lidadas com mais facilidade.
- Facilidade na execução de operações de teste e *debug*, uma vez que, como a solução é uma unidade, estas operações podem ser executadas muito mais rapidamente.
- Facilidade no processo de implantação, dado que aplicações monolíticas não precisam de lidar com vários processos de implantação, sendo normalmente apenas necessário lidar com um ficheiro ou um diretório.

- **Pontos Fracos**

- A escalabilidade de uma aplicação monolítica leva a um aumento da complicação do entendimento da aplicação.
- Novos desenvolvimentos provocam alterações de grandes dimensões que afetam o sistema na totalidade. Este têm de ser coordenados cuidadosamente, o que leva ao aumento da lentidão do processo de desenvolvimento.
- Não é possível escalar componentes de forma independente, apenas o sistema na totalidade.
- A integração de novas tecnologias em aplicações monolíticas é problemática, uma vez que é necessário reescrever a aplicação existente para suportar essas alterações.

Deste modo, após analisados os pontos fortes e fracos, determinou-se que a aplicação de uma abordagem monolítica não seria a mais indicada para a arquitetura do projeto, uma vez que um dos principais requisitos é a escalabilidade da solução a longo prazo, de forma a facilmente realizar novos desenvolvimentos, ou deixar de suportar funcionalidades, sem ser necessário a reestruturação total da aplicação ou a alocação de um maior período de tempo para garantir o correto funcionamento da aplicação após essas alterações.

3.2.2 Arquitetura baseada em microsserviços

Uma arquitetura baseada em microsserviços é caracterizada por permitir a entrega de aplicações complexas e de grandes dimensões, de forma rápida e consistente, uma vez que, neste tipo de arquitetura, as aplicações são estruturadas num conjunto de serviços que são (What are microservices?, s.d.):

- De fácil manutenção e testáveis;
- Baixo acoplamento entre eles;
- Independentemente implantáveis.

Na Figura 9, é apresentado um exemplo de uma arquitetura baseada em microsserviços.

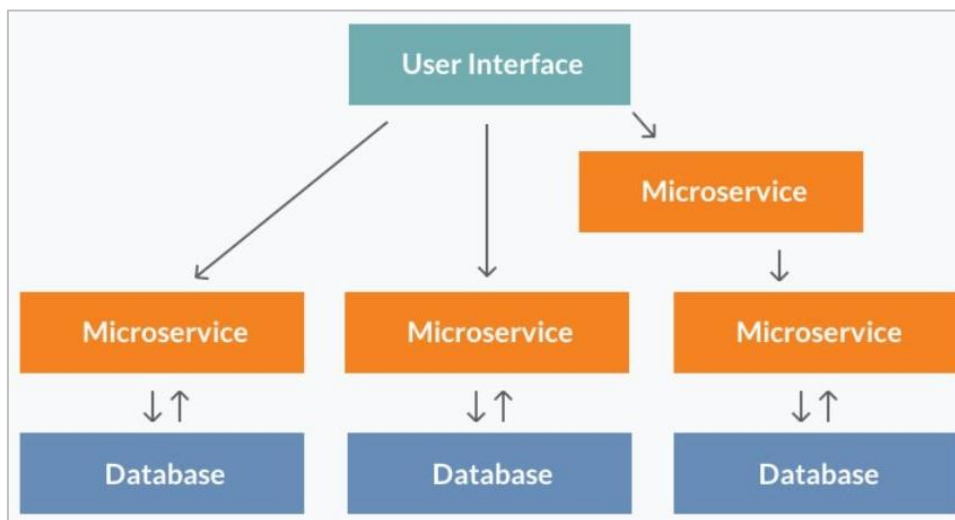


Figura 9 - Arquitetura de microsserviços (Gnatyk, s.d.)

Assim, foram identificados os seguintes pontos fortes e fracos para a arquitetura acima descrita (Gnatyk, s.d.).

- **Pontos Fortes**

- Aumento da flexibilidade da aplicação, uma vez que os serviços podem ser implantados e atualizados de forma independente entre eles, sendo que a ocorrência de uma falha num serviço específico não influencia a totalidade do funcionamento da aplicação
- A separação das funcionalidades em diferentes componentes torna a aplicação mais fácil de entender e gerir.
- Permite a fácil integração de novas funcionalidades.
- Permite a escalabilidade da solução, uma vez que cada componente pode ser escalável de forma independente, sendo o processo de desenvolvimento mais eficaz do ponto de vista temporal e monetário. Na Figura 10, é apresentado o comportamento de uma arquitetura monolítica e uma arquitetura de microsserviços ao longo do aumento da complexidade do projeto, sendo que a utilização de uma arquitetura de microsserviços permite uma melhor gestão da complexidade da aplicação ao longo de novos desenvolvimentos.

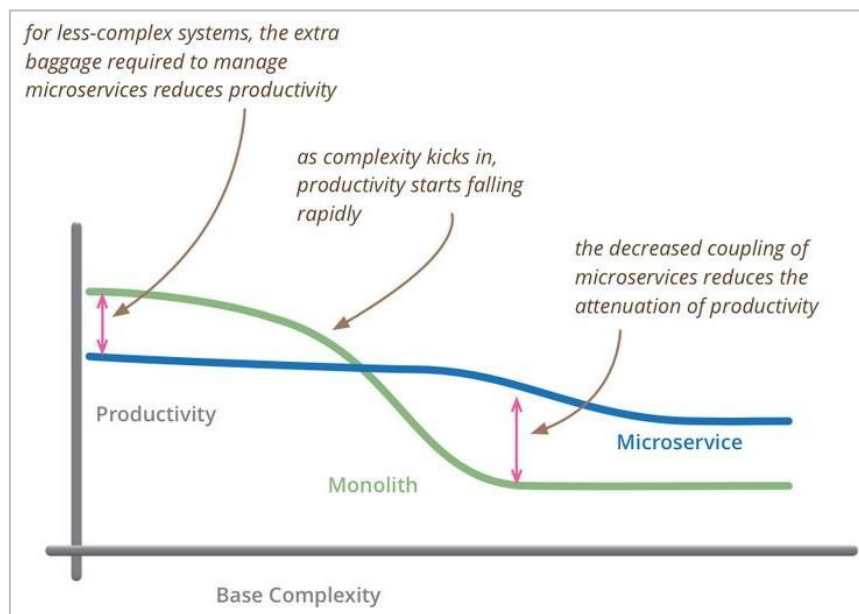


Figura 10 – Comportamento das arquiteturas monolítica e de microsserviços ao longo do desenvolvimento de uma solução (Fowler, s.d.)

- **Pontos Fracos**

- A complexidade da aplicação é alta, uma vez que a implementação de um sistema distribuído implica a gestão de um maior número de ligações entre os serviços e as bases de dados. Também aumenta a complexidade do processo de implementação, uma vez que cada serviço é gerido individualmente.
- A utilização de múltiplos serviços torna a realização de testes e *debug* mais complexa.

Deste modo, a utilização de uma arquitetura baseada em microsserviços adequa-se ao contexto do projeto, uma vez que esta permite tornar o funcionamento da aplicação, a longo prazo, o mais independente possível da comunicação com o veículo. Deste modo, caso existam alterações ou atualizações do lado do *software* das marcas automóveis que garantem o acesso aos dados dos veículos, é possível, de uma forma rápida e isolada, realizar as respetivas alterações no serviço responsável por essa integração ou até desenvolver um novo serviço para suportar essas alterações, caso seja necessário. Desta forma, caso uma versão deixe de ser suportada ou passe a existir uma nova versão, é apenas necessário realizar alterações nesse serviço específico, sendo que a aplicação continua disponível e a funcionar normalmente.

3.3 Interoperabilidade entre serviços

De modo a assegurar uma solução centralizada e consistente, é necessário que todos os dados sejam processados de forma igual. Uma vez que cada serviço é responsável pela gestão da integração com a API da marca automóvel, espera-se que os dados que são obtidos tenham um modelo de dados ou formatação diferente dos dados provenientes de outras marcas. Assim, funcionalidades de notificação e análise de dados não conseguem processar estes dados de forma coerente.

Para tal, é necessário desenvolver uma camada de abstração dos dados provenientes de serviços de integração, de modo a assegurar que os dados recolhidos são convertidos para apenas um modelo de dados específico, sendo este utilizado ao longo dos múltiplos componentes da solução.

Desta maneira, é possível assegurar que os dados que são comunicados entre a plataforma *web* e a plataforma de microsserviços se mantêm coerentes e independentes da marca automóvel. Assim, devido a esta abstração, as funcionalidades de processamento de dados e de criação de alertas também conseguem estabelecer relações entre os dados, obter resultados e apresentá-los, através de gráficos ou notificações, ao utilizador.

3.4 Visão da solução

De modo a responder ao problema descrito na secção 1.3, foi desenhada uma solução que permite integrar, de forma independente, múltiplas plataformas de diversas marcas automóveis, que permitem o acesso a dados e funcionalidades de veículos. É necessário desenvolver uma camada de abstração para o processamento dos dados recolhidos e uma plataforma que permita aceder, de uma forma centralizada e rápida, a esses dados e funcionalidades.

Para tal, o utilizador terá acesso ao seu perfil no Centro de Controlo para Veículos Elétricos, no qual pode registar os seus veículos, visualizar as informações destes e ter acesso a serviços que permitem interagir com essas viaturas. A plataforma *web* permite também aceder a dados estatísticos obtidos através de algoritmos responsáveis pela análise dos dados fornecidos pelas viaturas. A plataforma deve analisar esses mesmos dados, de modo a gerar avisos e notificações de possíveis problemas, permitindo uma monitorização preventiva dos seus automóveis.

O CCVE também permite visualizar estatísticas de um veículo desde o momento em que foi registado na plataforma, sendo que essa informação é apresentada de uma forma simplificada e intuitiva, por meio de tabelas e gráficos.

O acesso ao portal do CCVE é concedido através da contratação de um plano de subscrição selecionado do leque de ofertas disponíveis. As ofertas disponíveis dividem-se em três planos: o plano iniciante, o plano médio e o plano avançado. De acordo com o plano selecionado, algumas funcionalidades (mais avançadas) poderão estar desativadas ou limitadas. Por exemplo, o número de veículos que um utilizador pode registar na plataforma varia de acordo com o plano de subscrição selecionado, sendo que um utilizador com um plano avançado pode registar um maior número de veículos na plataforma, quando comparado ao número de veículos que os planos iniciante ou médio permitem registar.

4 Ambiente de Trabalho

Neste capítulo é apresentada a abordagem à gestão adotada no projeto, apresentado o planeamento e a metodologia de trabalho. Também são descritas, de forma breve, as tecnologias e as ferramentas de apoio utilizadas no desenvolvimento do projeto. Por fim, está também presente o método de investigação adotado ao longo do projeto.

4.1 Gestão de projeto

Neste subcapítulo é abordado o planeamento das tarefas realizadas e a metodologia de trabalho adotada durante o desenvolvimento do projeto.

4.1.1 Planeamento

Para gerir e auxiliar o desenvolvimento de projeto, adotou-se a metodologia ágil *Scrum*, de modo a organizar as tarefas a serem realizadas e utilizou-se o *Jira*, uma ferramenta usada para monitorizar o estado e a progressão dessas tarefas. O processo de planeamento foi realizado tendo em consideração a complexidade das tarefas a realizar e a duração do projeto, sendo as tarefas mais complexas divididas em tarefas mais simples. Na Figura 11, é apresentado o diagrama de Gantt elaborado para o projeto, disponível por completo no Anexo 1 do documento.

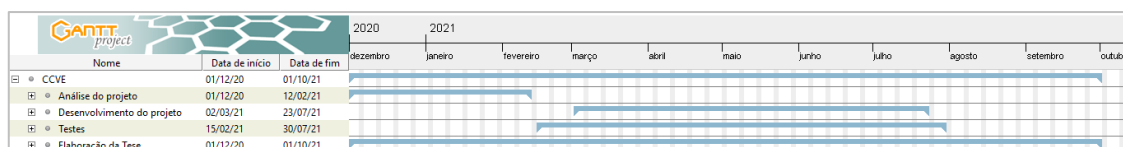


Figura 11 - Diagrama de Gantt para o CCVE

4.1.2 Reuniões de acompanhamento

No decorrer do desenvolvimento do projeto, e de modo a auxiliar o desenvolvimento deste, foram efetuadas diversas reuniões entre o aluno e os respetivos elementos da EVIO, apresentadas na Tabela 13.

Tabela 13 - Reuniões de acompanhamento da EVIO

Data da reunião	Participantes	Tema(s) debatido(s)
10/11/2020	- Tiago Meireles - Carlos Almeida (supervisor)	- Tema da tese a ser desenvolvida; - Formalização da proposta da tese.
06/01/2021	- Tiago Meireles - Carlos Almeida (supervisor)	- Acompanhamento do projeto. - Discussão dos casos de uso abordados no desenvolvimento do projeto.
29/03/2021	- Tiago Meireles - Carlos Almeida (supervisor)	- Discussão dos comentários realizados no primeiro momento de avaliação (P1). - Acompanhamento do projeto.
09/08/2021	- Tiago Meireles - Carlos Almeida (supervisor)	- Ponto de situação do desenvolvimento da plataforma. - Acompanhamento do projeto.

Para além das reuniões que ocorreram no contexto da EVIO, a comunicação entre o aluno, o orientador e supervisor manteve-se através de email, de forma a esclarecer dúvidas que surgiram durante o desenvolvimento do projeto e reportar pontos de situação.

4.2 Tecnologias utilizadas

De modo a simplificar o desenvolvimento do projeto, este foi separado em dois componentes principais. O primeiro componente é composto pelo portal do CCVE, sendo este uma plataforma *web* que disponibiliza funcionalidades de gestão e monitorização de veículos elétricos. O segundo componente é composto por um sistema que adota uma arquitetura de *microserviços* e que permite a integração com o *software* dos veículos e expõe esses dados para o CCVE.

Assim sendo, as tecnologias utilizadas para o desenvolvimento do projeto foram agrupadas pelas seguintes tarefas:

- Desenvolvimento do portal *web* do Centro de Controlo para Veículos Elétricos;
- Desenvolvimento de *software* baseado numa arquitetura de microsserviços, o que permite a integração de diversos serviços com as múltiplas plataformas de veículos que disponibilizam os dados dos veículos para análise;
- Desenvolvimento de uma camada que permite a abstração e processamento dos dados recolhidos;
- Criação de uma base de dados não relacional que permite o armazenamento dos dados recolhidos e dos dados necessários para o correto funcionamento da aplicação.

Para a realização de cada tarefa acima definida, são utilizadas uma ou mais tecnologias. Uma vez que o desenvolvimento desta solução é realizado no contexto da EVIO, esta já possui uma infraestrutura otimizada para suportar as tecnologias que são usadas atualmente nas soluções existentes da empresa.

Deste modo, e dado que a solução a ser desenvolvida expõe serviços que podem ser utilizados pelas outras aplicações da EVIO, é necessário que as tecnologias escolhidas no projeto estejam em concordância com as tecnologias predominantes nas aplicações da EVIO, sendo que as seguintes tecnologias foram consideradas no desenvolvimento do projeto:

- Node.js;
- Express;
- Docker;
- MongoDB.

Nas seguintes secções do documento, são apresentadas as tecnologias associadas a cada tarefa acima descrita, analisando brevemente a tecnologia e justificando a sua utilização na realização dessa tarefa.

4.2.1 Desenvolvimento do portal *web* do Centro de Controlo para Veículos Elétricos

Para o desenvolvimento do portal, foram analisadas três tecnologias no desenvolvimento de interfaces *web*: Vue, React e Angular.

- **Vue**
Vue é uma *framework* progressiva para o desenvolvimento de interfaces. Ao contrário de *frameworks* monolíticas, esta foi projetada para ser adaptável de forma incremental.

O foco principal desta tecnologia é a *view layer* (camada de visualização), mas permite a fácil integração de bibliotecas externas ou projetos já existentes (Introduction | Vue.js, s.d.). Desta forma, são listados abaixo os pontos positivos e negativos identificados para esta tecnologia (Freire, Medium, s.d.).

- **Pontos Fortes**

- Para além de uma biblioteca UI, disponibiliza pacotes de gestão do roteamento e estados da aplicação;
- Desenvolvimento de aplicações *web* em JavaScript ou Typescript.

- **Pontos Fracos**

- Sendo o mais recente das três tecnologias analisadas, ainda disponibiliza um número reduzido de bibliotecas externas.

- **React**

React é uma biblioteca JavaScript para a construção de interfaces *web*. Esta permite a construção de *views* (ecrãs) para cada estado da aplicação, sendo que após uma atualização de estado são renderizados os componentes associados a esse estado (React - A JavaScript library for building user interfaces, 2021). Deste modo, em seguida são apresentados os pontos positivos e negativos identificados para esta tecnologia (Benefits of React, s.d.).

- **Pontos Fortes**

- Extremamente flexível, sem a existência de uma estrutura obrigatória a ser seguida ou a utilização obrigatória de padrões arquiteturais específicos de desenvolvimento *web*;
- Utilização de componentes, o que permite a reutilização e desenvolvimento de código, facilitando a manutenção deste;
- Aplicações adaptáveis para dispositivos móveis.

- **Pontos Fracos**

- Uma vez que é extremamente flexível, pode ser utilizado no desenvolvimento de projetos pouco escaláveis.

- **Angular**

Angular é uma plataforma e *framework* utilizada no desenvolvimento de *Single-Page Applications* (SPA) através da utilização de HTML e *Typescript*. Esta tem como objetivo permitir o desenvolvimento de aplicações otimizadas para plataformas *web*, com o foco na velocidade de renderização e na performance destas para o cliente (Angular, s.d.). Em seguida são apresentados os pontos positivos e negativos identificados para esta tecnologia (Freire, Medium, s.d.).

- **Pontos Fortes**

- Disponibiliza nativamente todas as bibliotecas necessárias para o desenvolvimento de aplicações *web*;
- Utilização do padrão MVC (*Model-View-Controller*);
- Desenvolvimento de aplicações *web* em JavaScript ou Typescript;
- Simplifica a gestão dos estados dos componentes da aplicação, através da ligação bidirecional dos dados;
- Permite nativamente a gestão do roteamento da aplicação;
- Aplicação responsiva em dispositivos móveis.

- **Pontos Fracos**

- Aplicações com menor performance, quando comparada com outras tecnologias de desenvolvimento *web*.

Assim, após a análise destas tecnologias, optou-se pelo Angular para o desenvolvimento do portal *web*, uma vez que, apesar de uma redução na performance da aplicação, esta permite a implementação estruturada e escalável de uma plataforma *web*, responsiva em dispositivos móveis, que disponibilize as funcionalidades identificadas no projeto.

4.2.2 Desenvolvimento de *software* baseado numa arquitetura de microsserviços

- **Node.js**

Node.js é um *runtime* de *Javascript* que opera sobre um motor de alta performance de *Javascript* e *WebAssembly*, denominado de V8. Este utiliza eventos de forma a permitir uma execução assíncrona e foi desenhado com vista a permitir o desenvolvimento de aplicações que necessitem de suportar múltiplas ligações concorrentes (Node.js, s.d.).

Esta tecnologia foi selecionada para o desenvolvimento da rede de microsserviços que vai compor a plataforma responsável pela integração com o *software* dos veículos, gestão de utilizadores e das funcionalidades apresentadas pelo portal *web*. Assim, esta permite a criação, juntamente com Express, de servidores que disponibilizem os serviços necessários para o correto funcionamento da solução.

- **Docker**

Docker é uma plataforma aberta utilizada no desenvolvimento, compartimentação e execução de aplicações. Esta plataforma permite a separação da infraestrutura das aplicações em funcionalidades, sendo cada uma executada individualmente e independentemente de outras funcionalidades, mas permitindo a comunicação entre elas. Estas metodologias permitem o desenvolvimento e implantação mais rápida de aplicações, reduzindo o tempo de entrega de *software* (Docker, s.d.).

No contexto do projeto, esta tecnologia é utilizada na execução dos serviços que compõem a rede de microsserviços responsável pela integração com o *software* dos veículos. A utilização desta tecnologia permite a execução individual de cada serviço responsável pela integração com uma marca de veículo. Deste modo, é possível desenvolver, de forma isolada, novos serviços ou alterar serviços já existentes, sendo apenas necessário voltar a executar os serviços alterados, sem ser necessário *downtime* da aplicação e sem afetar os restantes serviços que estão a operar de forma isolada.

- **Express**

O Express é uma *framework* minimalista e flexível utilizada na criação de aplicações *web* em Node.js, que disponibiliza funcionalidades para o desenvolvimento de aplicações móveis e *web*. Através da utilização de um conjunto de métodos HTTP e de *middleware*, permite a criação de APIs robustas, de forma rápida e fácil (Express, s.d.). Esta tecnologia é utilizada na criação das APIs que, através de servidores *web* em Node.js, expõem as funcionalidades desenvolvidas para o projeto. Assim, através destas, os serviços conseguem comunicar entre si, com o portal *web* e com o *software* disponibilizado pelos fabricantes automóveis.

4.2.3 Desenvolvimento da camada de abstração

A camada de abstração é responsável por garantir a coerência da comunicação entre o portal *web* e os serviços do CCVE. Como tal, esta camada está presente no serviço que processa todos os pedidos provenientes do portal.

Deste modo, utilizou-se Node.js e Express para o desenvolvimento deste serviço e Docker para a compartimentação e execução deste.

4.2.4 Criação de uma base de dados não relacional

- **MongoDB**

O *MongoDB* é uma base de dados não relacional que organiza a informação em documentos, o que permite que a indexação dos dados e pesquisa nestes seja mais rápida, com maior flexibilidade e escalabilidade (MongoDB, s.d.).

No contexto do projeto, esta tecnologia foi utilizada para a criação de uma base de dados capaz de guardar os dados recolhidos dos veículos e dos clientes e os restantes dados necessários para garantir o correto funcionamento da aplicação. Esta foi selecionada pela rapidez com que permite que sejam realizadas pesquisas nos dados, o que é necessário para o processamento do grande volume de dados recolhidos das viaturas.

4.3 Ferramentas de apoio

Nesta secção do documento serão apresentadas as ferramentas que apoiaram o desenvolvimento do projeto e a escrita deste documento, juntamente com uma breve descrição e como foram aplicadas.

- **Visual Paradigm**

Visual Paradigm é uma aplicação que auxilia no desenvolvimento de *software*. Esta é utilizada na criação de notação UML que traduz os requisitos de um projeto, de modo a garantir que, mesmo que estes sejam implementadas por múltiplas equipas, as funcionalidades estão de acordo com os requisitos (Visual Paradigm, s.d.) inicialmente definidos.

No contexto do projeto, esta tecnologia foi utilizada na criação dos diagramas presentes ao longo deste relatório, que auxiliaram no desenvolvimento das funcionalidades do projeto.

- **Jira**

Jira é uma ferramenta que permite a monitorização dos estados das tarefas associadas ao desenvolvimento de um projeto, possibilitando às equipas que a utilizam uma melhor gestão temporal da carga de trabalho a ser realizada (Jira, s.d.).

Esta ferramenta foi utilizada para auxiliar na divisão de tarefas e na gestão do progresso da execução destas.

- **Excel**

Excel é um programa de folha de cálculo que permite analisar, manipular e visualizar dados de uma forma intuitiva e simples (Microsoft, s.d.).

Esta ferramenta foi utilizada no projeto para auxiliar nas operações de análise estatística de dados, tais como o AHP, o QFD e a validação dos resultados obtidos no final do desenvolvimento do produto.

4.4 Metodologia de investigação

Nesta secção do documento, é apresentada a metodologia de investigação adotada durante o desenvolvimento do projeto. Esta metodologia é o instrumento utilizado na investigação do problema apresentado e que está a ser alvo de estudo, com o objetivo de demonstrar a validade dos resultados obtidos.

O processo associado à metodologia de investigação tem como fase inicial a identificação das questões que devem ser consideradas para o início da pesquisa dos temas a abordar no projeto.

Numa segunda fase, são considerados os resultados que são esperados obter como resposta às questões identificadas na primeira fase e quais os métodos de pesquisa devem ser adotados para melhor atingir esses resultados. Também devem ser identificadas as evidências que podem ser demonstradas para validar a veracidade dos resultados positivos obtidos e como é que esses resultados positivos se distinguem dos resultados negativos obtidos.

Assim sendo, como base do processo de investigação do projeto, foram identificadas as seguintes questões:

- I. Qual é a melhor forma de desenvolver uma plataforma capaz de permitir a integração com múltiplos *softwares* de veículos, diferentes e isolados entre si;
- II. Qual é a melhor abordagem a ser adotada, em termos de *design*, para uma melhor comunicação entre as duas componentes principais do projeto.

Para estas questões, espera-se que os resultados obtidos sejam apresentados sobre a forma de um produto e procedimentos, quer de implementação como de *design*, a aplicar ao longo do desenvolvimento deste.

É aplicada uma metodologia de investigação denominada de *Action Research*. Esta é descrita como um conjunto de metodologias de pesquisa com o objetivo de atingir uma ação (*Action*) e uma pesquisa (*Research*) em simultâneo (Emeral Publishing, s.d.). Para tal, esta metodologia realiza esta pesquisa através da (Emeral Publishing, s.d.):

- I. Utilização de um processo de análise cíclico, que alterna entre ação e reflexão crítica;
- II. Durante os últimos ciclos do processo de análise, existe a continuação da refinação dos métodos, dados e interpretações desenvolvidas nos ciclos iniciais.

Deste modo, através da aplicação desta metodologia, é possível, de forma iterativa, determinar o processo que melhor se adequa para o resultado que desejamos obter.

De modo a validar esses resultados obtidos e determinar a necessidade de continuar a realizar ciclos de processo de análise, é realizada uma validação externa e sistemática da validade (*external validity*) desses resultados para o critério acima definido. Esta validação externa é suportada pela avaliação do grau de relação entre a ação que se queria atingir e o resultado obtido.

5 Descrição técnica

Este capítulo tem como principal objetivo descrever os requisitos funcionais e não funcionais identificados na análise do problema, a infraestrutura adotada para o desenvolvimento do projeto e o método de avaliação dos requisitos identificados.

5.1 Engenharia de requisitos

Nesta secção do documento são apresentados os atores que interagem com sistema e a análise realizada aos requisitos do projeto identificados.

5.1.1 Atores

Um ator é a representação de uma entidade responsável pela realização de certos papéis no sistema. Cada papel que é representado por um ator em UML corresponde a um papel a ser realizado por um utilizador desse sistema (Chintis, Tiwari, & Ananthamurthy, 2003). Deste modo, para o sistema desenvolvido, foram identificados os seguintes atores:

- **Utilizador não registado:** Utilizador que não tem uma conta no sistema;
- **Utilizador não autenticado:** Utilizador que já tem conta, mas que ainda não está autenticado no sistema;
- **Utilizador autenticado e inválido:** Utilizador autenticado no sistema, mas que não tem permissões para aceder aos serviços da plataforma;
- **Utilizador autenticado e válido:** Utilizador autenticado no sistema e que se encontra num estado válido, tendo permissões para aceder aos serviços da plataforma;
- **Sistema:** Responsável por despoletar eventos para os utilizadores da plataforma.

Deste modo, o papel de um utilizador no sistema varia de acordo com a existência de registo deste no sistema, do estado de autenticação e da validade da sua conta. Assim, de modo a clarificar o estado de um utilizador quando este realiza ações no sistema, o papel do utilizador foi dividido em múltiplos atores. De acordo com o seu estado no sistema, um utilizador tem

acesso a diferentes funcionalidades na plataforma. O ator Sistema é a exceção, uma vez que este não representa nenhum utilizador da plataforma.

O ator Sistema é o ator que pretende representar todas as ações que são da responsabilidade do sistema e que não são despoletadas por nenhum utilizador da aplicação, tais como processos automáticos de análise de dados ou envio de notificações.

Na Figura 12, estão presentes os tipos de atores, em notação UML, que interagem com o sistema. Esta representação é utilizada para identificar o respetivo ator nos diagramas de sequência ao longo deste capítulo.



Figura 12 - Atores do sistema

5.1.2 Requisitos funcionais

Os requisitos funcionais capturam o comportamento pretendido do sistema. Este comportamento pode ser expresso em serviços, tarefas ou funções que o sistema é responsável por executar. Estes requerimentos funcionais podem ser apresentados através de casos de uso. Um caso de uso define um conjunto de interações entre um ator e o sistema, tendo em consideração o objetivo principal dessa interação (Malan & Bredemeyer, 2001). Os casos de uso identificados podem ser representados através de um diagrama de casos de uso.

Assim, foram identificados os seguintes casos de uso para o projeto:

1. **UC1: Registo de utilizador** – O utilizador não registado, através da aplicação *web*, deve ser capaz de se registar na plataforma;
2. **UC2: Login de utilizador** – O utilizador, através da aplicação *web*, deve fornecer as suas credenciais de acesso para aceder à plataforma;
3. **UC3: Opção de pagamento para retificação do estado da conta** – O utilizador autenticado e inválido, através da aplicação *web*, deve ter acesso às opções de pagamento para retificar a sua situação na plataforma;

4. **UC4: Mudança de plano de subscrição atual** – O utilizador autenticado e válido, através da aplicação *web*, deve poder alterar o plano de subscrição contratado, sem ter de esperar que este termine;
5. **UC5: Registo do veículo na plataforma** – O utilizador autenticado e válido, através da aplicação *web*, deve poder registar o seu veículo na plataforma;
6. **UC6: Apresentação da informação do veículo** – O utilizador autenticado e válido, através da aplicação *web*, deve poder visualizar as informações do veículo registado na plataforma;
7. **UC7: Apresentação dos dados recolhidos do veículo** – O utilizador autenticado e válido, através da aplicação *web*, deve poder visualizar os dados do veículo registado na plataforma;
8. **UC8: Apresentação de estatísticas baseadas nos dados recolhidos** – O utilizador autenticado e válido, através da aplicação *web*, deve poder visualizar estatísticas dos seus veículos registados na plataforma;
9. **UC9: Execução de comandos sobre o veículo** – O utilizador autenticado e válido, através da aplicação *web*, deve permitir ao utilizador selecionar um veículo para permitir a execução de comandos;
10. **UC10: Recolha dos dados dos veículos** – O sistema deve recolher, de forma automática, os dados dos veículos registados na plataforma;
11. **UC11: Envio de alertas inteligentes** – O sistema, através da aplicação *web*, deve enviar alertas para o utilizador. Estes alertas são o resultado da aplicação de mecanismos inteligentes na análise dos dados do veículo;
12. **UC12: Configuração de alertas** – O utilizador autenticado e válido, através da aplicação *web*, deve conseguir configurar alertas para as suas viaturas;
13. **UC13: Envio de alertas configurados pelo utilizador** – O sistema, através da aplicação *web*, deve notificar o utilizador quando um alerta é despoletado. Estas notificações têm em consideração os alertas configurados pelo utilizador e são geradas pela análise de dados das viaturas.

Na Figura 13, é apresentado o diagrama UML dos casos de uso acima descritos.

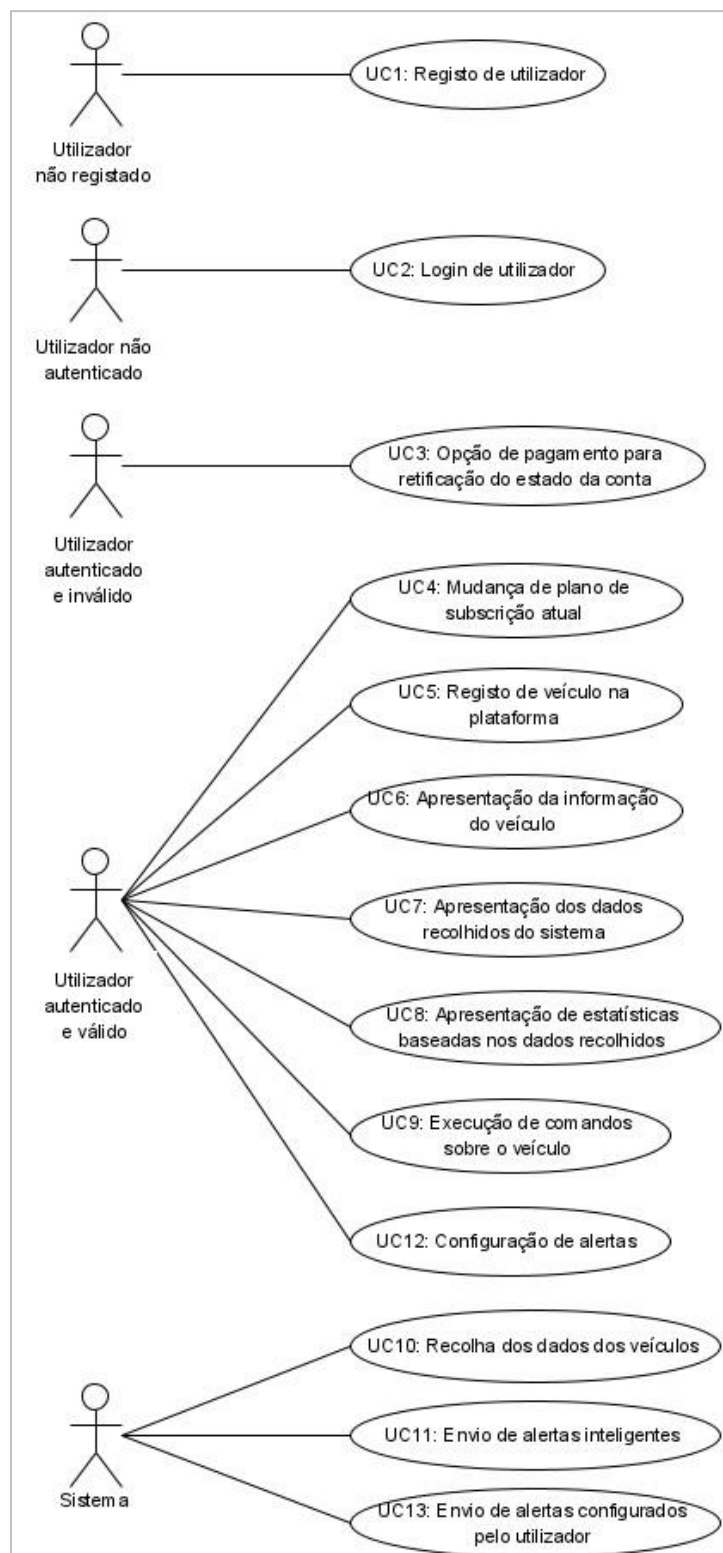


Figura 13 - Diagrama de Casos de Uso

5.2 Análise

Nesta secção do documento, é apresentada a análise realizada aos requisitos funcionais apresentados na secção 5.1.2 deste documento e os requisitos não funcionais, a análise realizada à infraestrutura do projeto e aos respetivos componentes que a integram.

Assim, de modo a avaliar as diferentes categorias nas quais os requisitos podem ser divididos, foi utilizado o modelo FURPS+. Este modelo permite a classificação de requisitos de acordo com a sua associação ao desenvolvimento de tecnologia – funcionalidade, usabilidade, fiabilidade, performance e suportabilidade. O modelo utilizado é um derivado do modelo FURPS, sendo que o “+” permite identificar restrições associadas com o desenvolvimento da plataforma. Estas podem estar relacionadas com o *design*, implementação, interface e componente física (Center of Excellence for Professional Development, s.d.).

5.2.1 Requisitos funcionais

Nesta subsecção deste documento é apresentada a análise realizada a cada caso de uso, descrevendo a funcionalidade e o modo de como esta se comporta no sistema. A interação entre um ator e o sistema é representada através de um *System Sequence Design* (SSD).

I. Registo de utilizador

Para que um utilizador tenha acesso aos serviços de gestão de veículos fornecidos pela plataforma, este tem de ter uma conta (perfil) na plataforma, de forma a identificar o utilizador no sistema. Durante o processo de registo, o utilizador tem de fornecer os dados pessoais, tais como o nome e email, selecionar o plano que melhor se adequa às suas necessidades para a utilização da plataforma e o respetivo método de pagamento. Em formato breve, o utilizador seleciona a opção de criação de conta, insere o nome, email e os dados de acesso. Em seguida, são apresentados os planos de subscrição existentes. O utilizador seleciona o plano que deseja contratar e preenche a informação requisitada pelo sistema para realizar o pagamento. Ao clicar na opção de criação de conta, este regista o utilizador num estado inválido, aguardando pela validação do pagamento. O sistema informa o utilizador do sucesso da operação. Na Figura 14, é apresentado o diagrama que descreve a interação entre o utilizador e o sistema.



Figura 14 - SSD de registo de utilizador

II. *Login* de utilizador

De modo a aceder aos serviços disponibilizados pela aplicação, o utilizador tem de estar autenticado na plataforma.

Em formato breve, o utilizador não autenticado seleciona a opção de *login* no portal *web* e insere as suas credenciais de acesso. Após esta operação, o utilizador está autenticado no sistema e, caso esteja válido, tem acesso às funcionalidades da plataforma, de acordo com a sua subscrição. Caso este se encontre num estado inválido, tem acesso às operações de retificação de estado de conta ou alteração de subscrição contratada. Na Figura 15, está presente o diagrama que descreve esta interação.

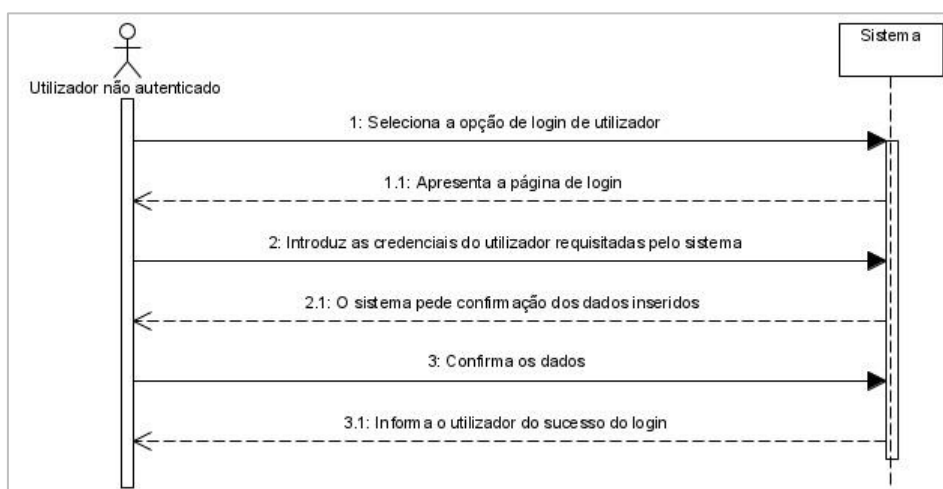


Figura 15 - SSD de *login* de utilizador

III. Opção de pagamento para retificação do estado da conta

Caso um utilizador tenha uma conta cujo pagamento não tenha sido realizado dentro do tempo estipulado no plano de subscrição contratado, este continua a conseguir realizar o *login*, mas passa a estar num estado inválido e deixa de ter acesso às funcionalidades do CCVE, sendo necessário retificar o pagamento em falta para voltar a ter acesso.

Em formato breve, o utilizador seleciona esta opção, sendo apresentado o saldo em falta. O sistema gera os dados de pagamento, apresentando-os ao utilizador, para que este possa proceder à retificação da situação. Após o pagamento, o sistema valida e atualiza o estado da conta do utilizador. Na Figura 16, é apresentada a interação entre o utilizador e o sistema para esta funcionalidade.

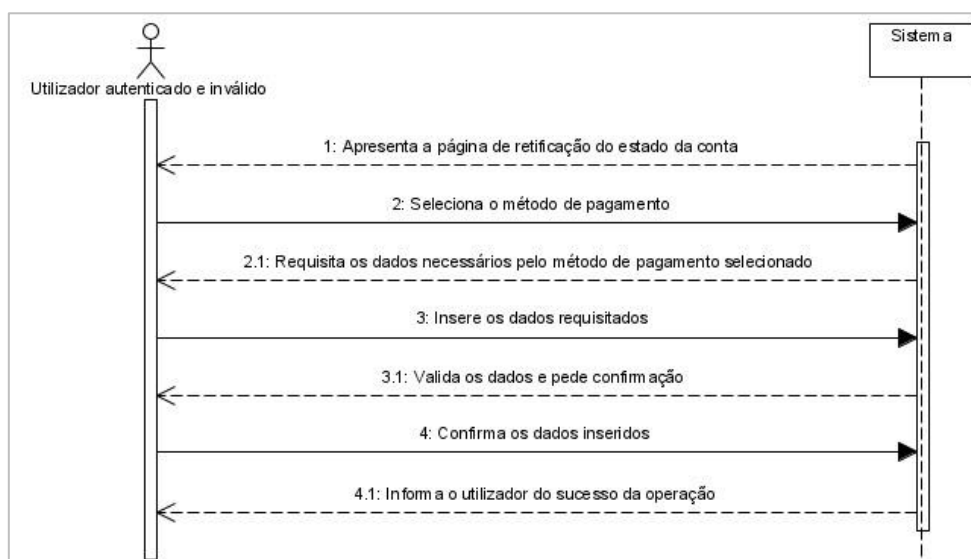


Figura 16 - SSD de opção de pagamento para retificação do estado da conta

IV. Mudança de plano de subscrição atual

Esta funcionalidade permite a um utilizador a alteração do plano de subscrição atualmente associado à sua conta.

Em formato breve, o utilizador seleciona esta opção no portal *web* e o sistema apresenta a lista de planos disponíveis para seleção (excluindo o plano contratado). Ao confirmar um novo plano, o sistema contabiliza a diferença monetária entre o plano atual e o novo plano selecionado, procedendo ao acerto necessário. Deste modo, caso

o novo plano seja mais caro, é gerada uma nova ordem de pagamento com a diferença ou, caso o valor seja inferior, é devolvida a diferença ao utilizador.

Na Figura 17, é visível a comunicação entre o utilizador e o sistema, relativamente ao processo de mudança de plano de subscrição.

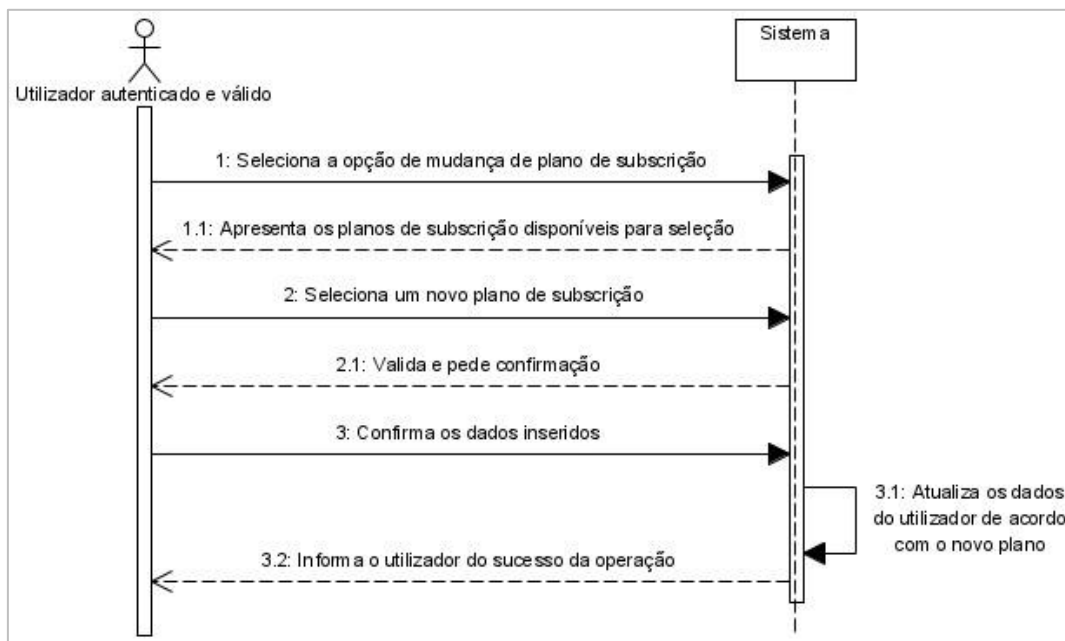


Figura 17 - SSD de mudança de plano de subscrição contratado

V. Registo do veículo na plataforma

De modo a utilizar os serviços disponibilizados pela plataforma, é necessário que o utilizador registe o seu veículo na plataforma. Para estabelecer a ligação com a viatura e iniciar a recolha dos seus dados, é necessário que a marca e/ou o modelo dessa viatura seja suportado pela plataforma.

Assim sendo, e num formato breve, este seleciona a opção de registo de viatura no portal *web*. O sistema requisita ao utilizador a marca, modelo, versão, matrícula e número de série da viatura. O utilizador insere os dados e regista a viatura. Ao terminar esta operação, o sistema valida os dados inseridos e informa o utilizador do sucesso da operação, iniciando o processo de comunicação com a viatura para aceder aos dados. Na Figura 18, é apresentado o diagrama que contém as interações entre o utilizador e o sistema acima referidas.

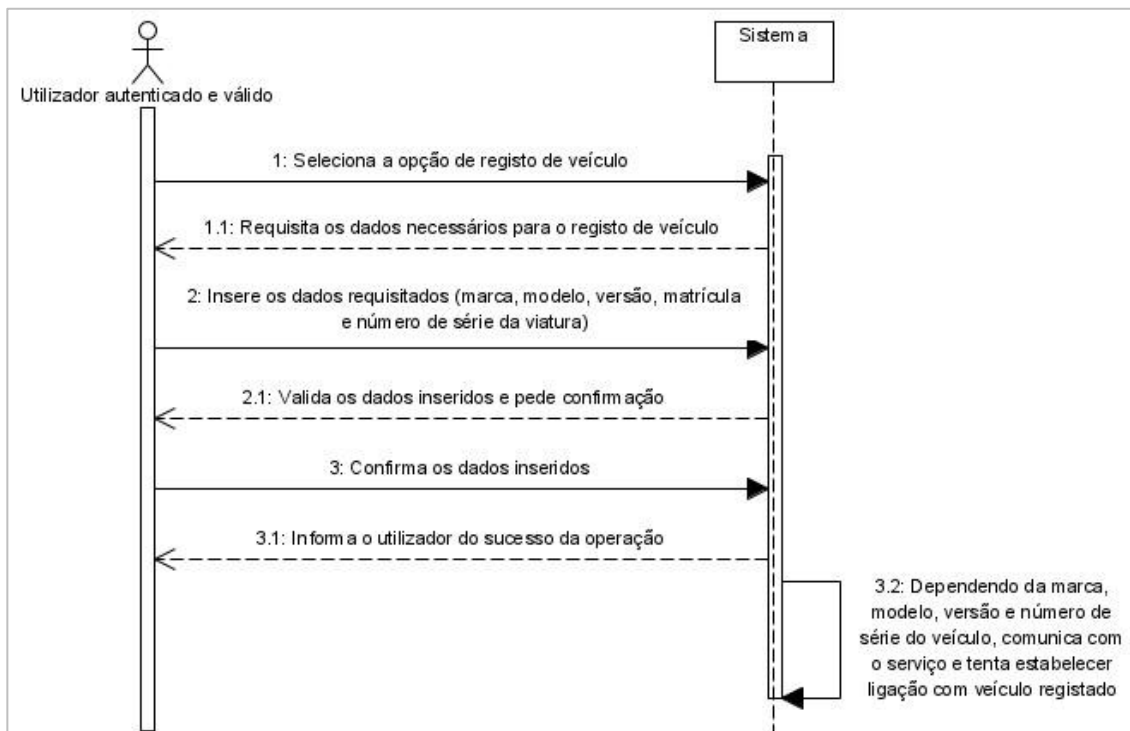


Figura 18 - SSD de registo de veículo na plataforma

VI. Apresentação da informação do veículo

Após o registo de um veículo, o utilizador pode visualizar os dados dessa viatura. Em formato breve, o utilizador seleciona a opção para visualizar a informação das viaturas que registou na plataforma. O sistema agrega os dados requisitados e apresenta-os ao utilizador. Na Figura 19, é apresentado o diagrama de sequência das interações entre o utilizador e o sistema.

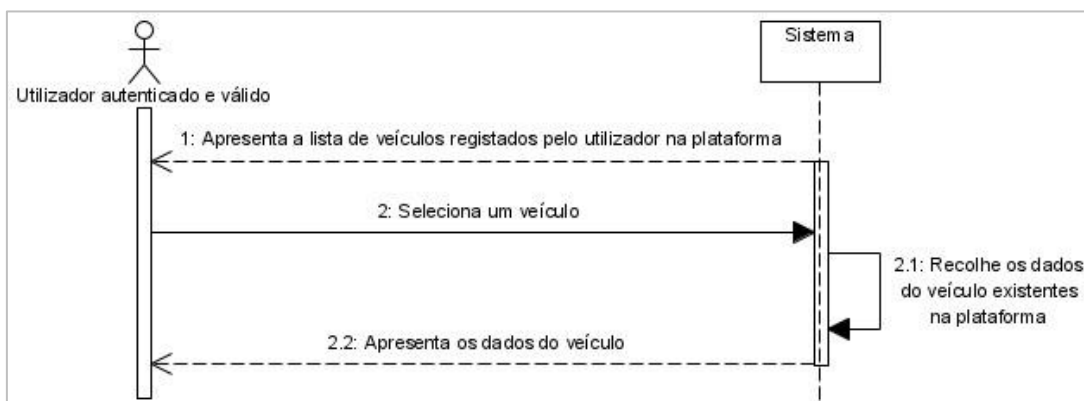


Figura 19 - SSD de apresentação de informação de veículo

VII. Apresentação dos dados recolhidos do veículo

Após o registo de um veículo, o utilizador pode monitorizar a sua viatura, com base na informação recolhida. Os dados a monitorizar estão relacionados com a percentagem de bateria, o estado de carregamento e a localização geográfica da viatura.

Em formato breve, o utilizador seleciona a opção de monitorização no portal *web*, sendo que o sistema compila os dados de todas as suas viaturas registadas e apresenta-os ao utilizador. A interação acima referida é visível na Figura 20.

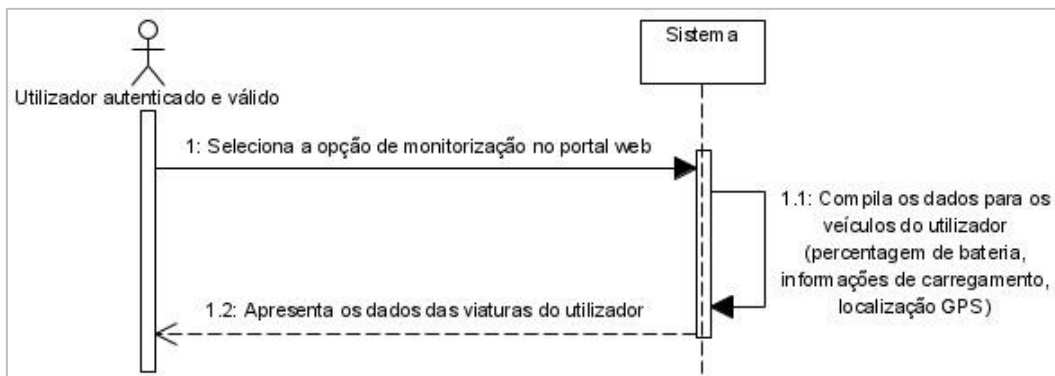


Figura 20 - SSD de apresentação dos dados recolhidos do veículo

VIII. Apresentação de estatísticas baseadas nos dados recolhidos

O utilizador pode querer aceder aos dados estatísticos relacionados com as suas viaturas e da sua atividade com estas.

Em formato breve, o utilizador requisita ao servidor os dados estatísticos da sua atividade. Por defeito, o mês atual é definido como o intervalo de tempo para as estatísticas que estão a ser solicitadas. O sistema analisa o histórico do utilizador e obtém os indicadores da sua atividade, tais como o número de sessões de carregamento realizadas pelas suas viaturas e a quantidade de kWh consumidos nos carregamentos, para o intervalo de tempo definido e apresenta-os ao utilizador.

Em seguida, o utilizador pode selecionar uma viatura para visualizar os dados estatísticos específicos a essa viatura. O sistema apresenta a lista das viaturas que este tem registadas na plataforma. O utilizador seleciona a viatura. O sistema analisa os dados recolhidos dessa viatura, para o intervalo de tempo selecionado, e processa a informação. O sistema apresenta essa informação ao utilizador através de gráficos. Na Figura 21, é apresentado o diagrama de sequência que contém a interação entre o utilizador e o sistema.

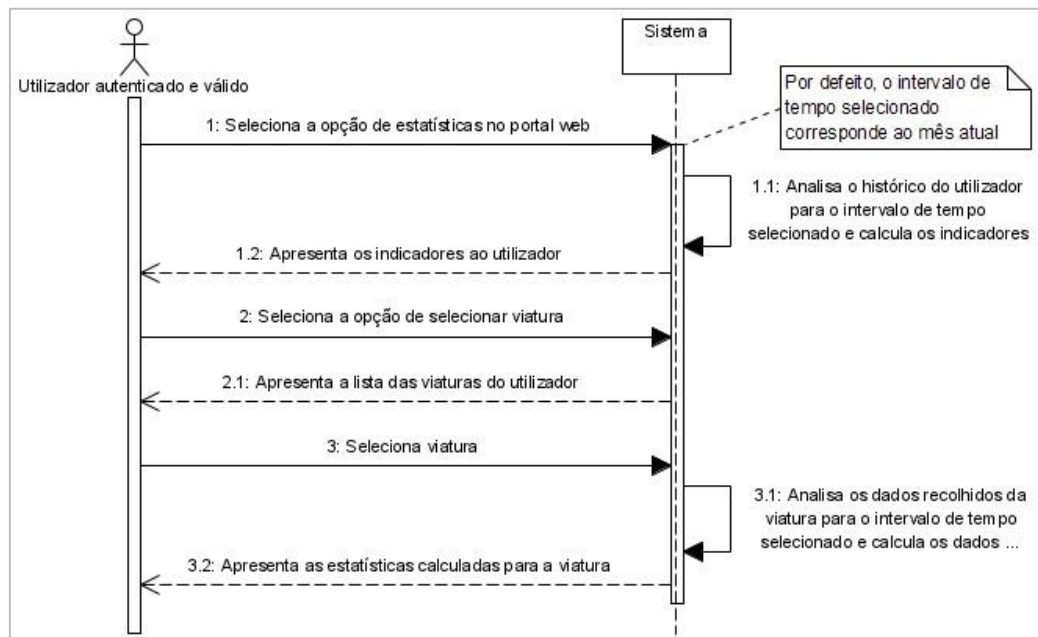


Figura 21 - Apresentação de estatísticas baseadas nos dados recolhidos

IX. Execução de comandos sobre o veículo

A qualquer altura, o utilizador pode executar comandos no seu veículo, de forma remota. Em formato breve, o sistema apresenta a lista de viaturas do utilizador, sendo que este seleciona o veículo no qual deseja executar comandos de forma remota. O sistema apresenta a lista de comandos possíveis a serem executados. O utilizador seleciona um comando, sendo este enviado para o sistema. Por sua vez, o sistema é responsável por identificar, através do veículo selecionado, qual o serviço responsável por executar este comando e reencaminha o pedido para ser processado. O sistema notifica o utilizador do sucesso da operação, tal como é visível na Figura 22.

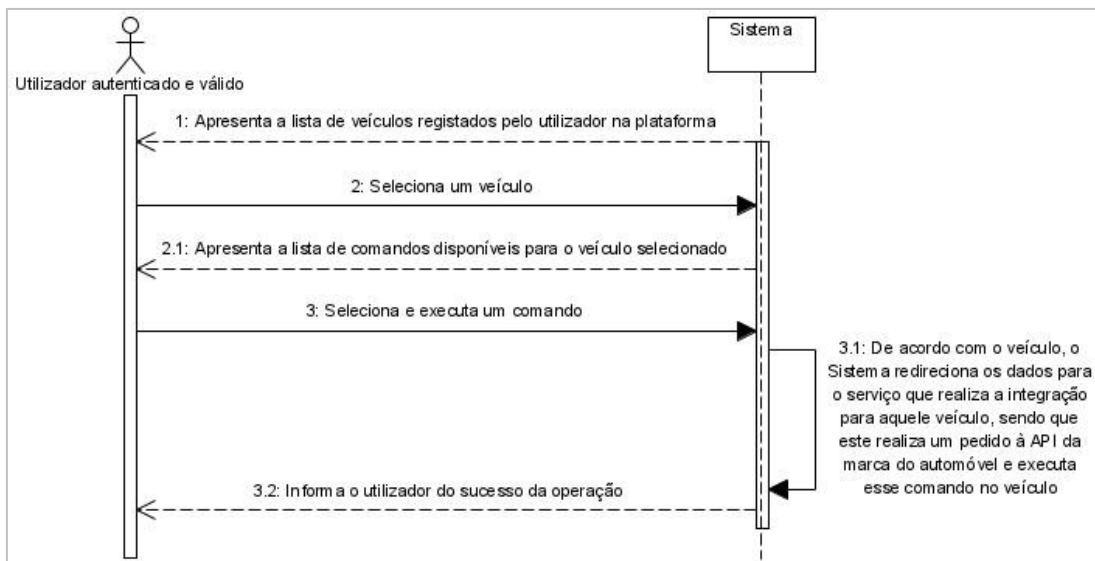


Figura 22 - SSD de execução de comandos sobre o veículo

X. Recolha dos dados dos veículos

De modo a iniciar o processo de coleção de dados de um determinado veículo, o sistema, após ter processado um pedido de registo de veículo com sucesso, despoleta o processo automático de coleção de informação gerada pela viatura.

Em formato breve, o sistema identifica, através da marca e do modelo fornecido no pedido, qual o serviço responsável por iniciar a comunicação com a viatura. Caso esta comunicação seja bem-sucedida, é iniciado o mecanismo de coleção de informação, procedendo a um processamento dos dados recebidos e o respetivo registo destes na base de dados. Quando este processo termina, ou na ocorrência de uma falha, o sistema envia para o portal *web* o aviso a ser apresentado ao utilizador. Esta interação é apresentada na Figura 23.

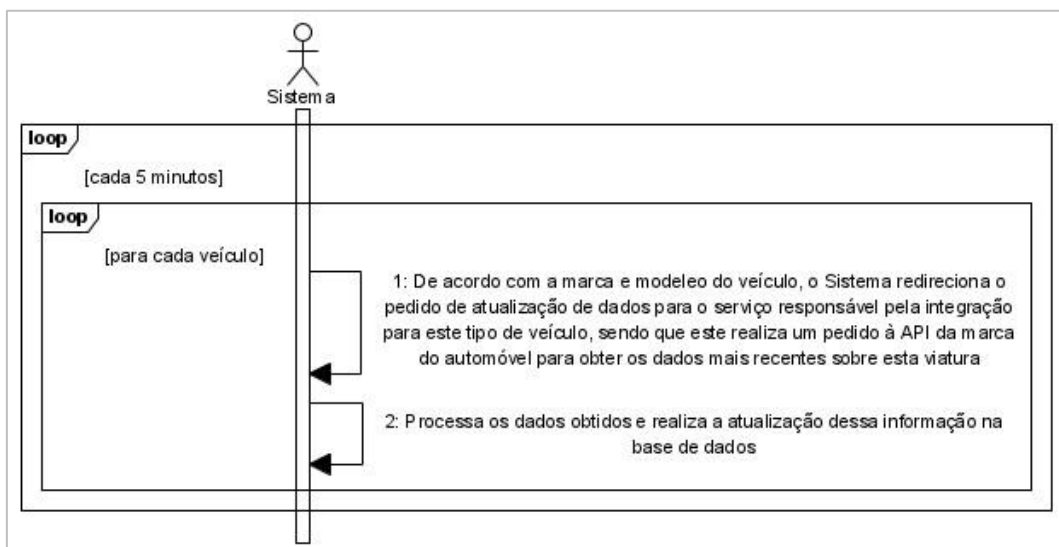


Figura 23 - SSD do mecanismo de recolha dos dados do veículo

XI. Envio de alertas inteligentes

De modo a dinamizar o processo de análise de dados e de forma a reduzir a necessidade de visualizar constantemente os dados na aplicação, o sistema tem mecanismos de análise dos dados, despoletando notificações para o utilizador. O sistema deve analisar os dados recolhidos e, através de mecanismos inteligentes, identificar possíveis situações anómalas.

Assim, foram identificados, durante o processo de análise, os seguintes mecanismos automáticos para serem desenvolvidos:

- **Mecanismo 1** - Mecanismo responsável pela análise dos consumos de energia realizados pelas viaturas e os respetivos períodos de carga, de modo a estabelecer uma relação entre os kWh carregados e o tempo de carregamento, desenvolvendo um indicador que, ao longo do tempo, é utilizado para a identificação de potencial degradação da bateria elétrica dos veículos;
- **Mecanismo 2** - Mecanismo responsável por analisar a bateria atual das viaturas, o histórico de sessões de carregamento e as tarifas de pagamento associadas aos períodos possíveis de carga. Deste modo, é possível indicar ao utilizador o período mais indicado para ser realizada a sessão de carregamento mais económica possível.

Na Figura 24, é visível o formato breve da comunicação entre o sistema e o utilizador para este requisito. Neste, o sistema inicia, de forma agendada, o processo de análise de dados dos veículos, a cada 10 minutos. Assim, para cada veículo registado na plataforma, o sistema compila os dados necessários e inicia a execução dos mecanismos existentes. Deste modo, o sistema executa o mecanismo 1 e, de acordo com o indicador obtido, notifica o utilizador da irregularidade identificada. Em seguida, o sistema executa o mecanismo 2 e, de acordo com o resultado obtido, alerta o utilizador.

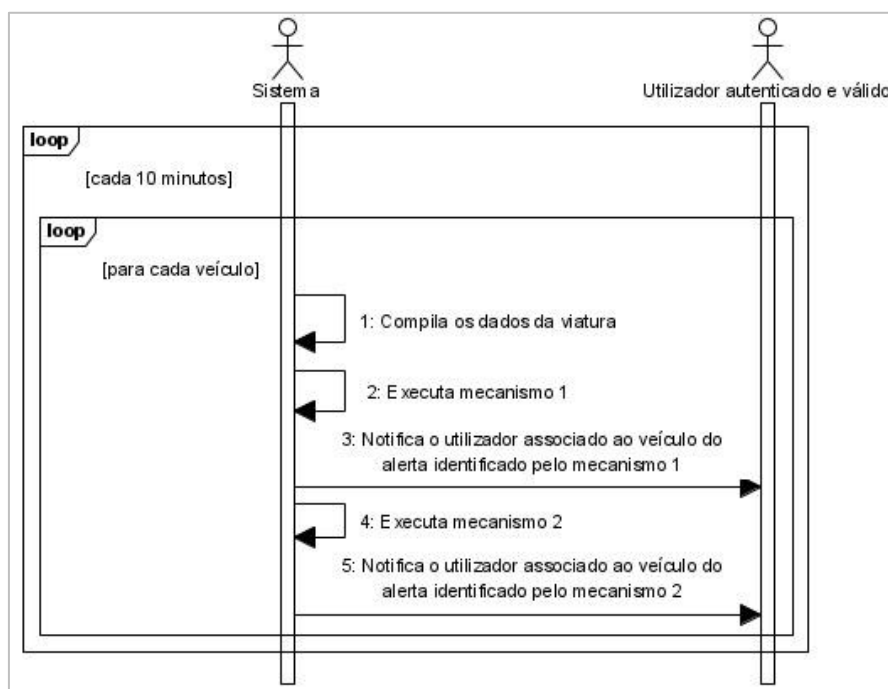


Figura 24 - SSD do mecanismo de envio de alertas inteligentes

XII. Configuração de alertas

O utilizador pode configurar os seus próprios alertas. Estes alertas são configurados de acordo com os dados recolhidos dos veículos. Para tal, o sistema disponibiliza aos utilizadores um conjunto de alertas que este pode seleccionar para as suas viaturas. Deste modo, durante o processo de análise, foi identificado o seguinte alerta para ser implementado:

- **Alerta 1** – Este alerta é baseado na percentagem de bateria de uma viatura. Ao seleccionar esta opção, é apresentado ao utilizador a lista de limites existentes. Para este tipo de alerta, a lista de limites apresentada é composta pelos valores 10%, 15%, 20%, 25% e 30%.

Em formato breve, o utilizador seleciona a opção de configurar alertas. O sistema apresenta os tipos de alertas disponíveis para configuração. O utilizador seleciona o tipo de alerta que deseja, o limite desse alerta que despoleta a notificação e o respetivo veículo a que deseja associar o alerta. O sistema valida os dados selecionados e informa o utilizador do sucesso da operação.

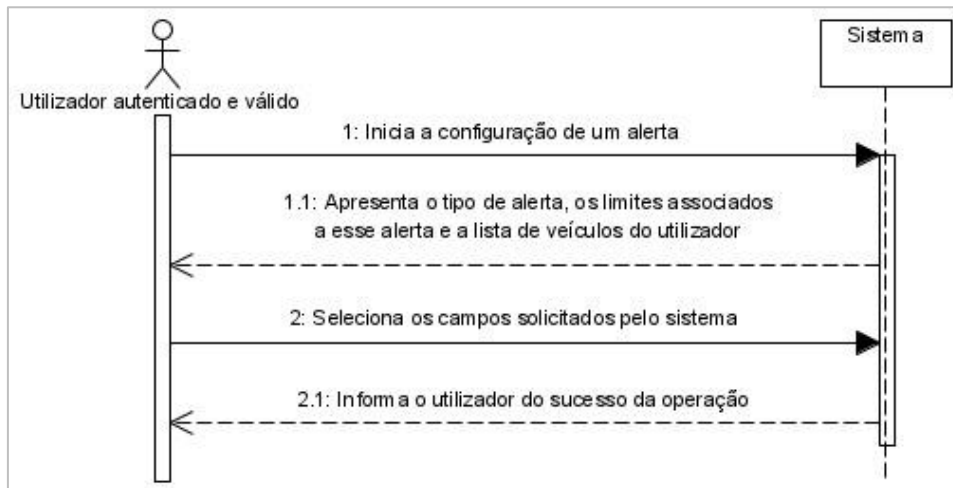


Figura 25 - SSD da configuração de alertas

XIII. Envio de alertas configurados pelo utilizador

O sistema é responsável por analisar periodicamente os alertas configurados pelo utilizador e, com base nos dados das viaturas, verificar, para cada alerta, se este se encontra numa situação válida para notificar o utilizador da ocorrência.

Em formato breve, o sistema despoleta o processo de análise dos alertas configurados pelo utilizador, no qual é determinado se o limite configurado previamente no alerta é excedido. Caso seja identificada uma situação em que o limite é excedido, o sistema notifica o utilizador da ocorrência. Na Figura 26, é apresentado o formato breve acima descrito através de um SSD.

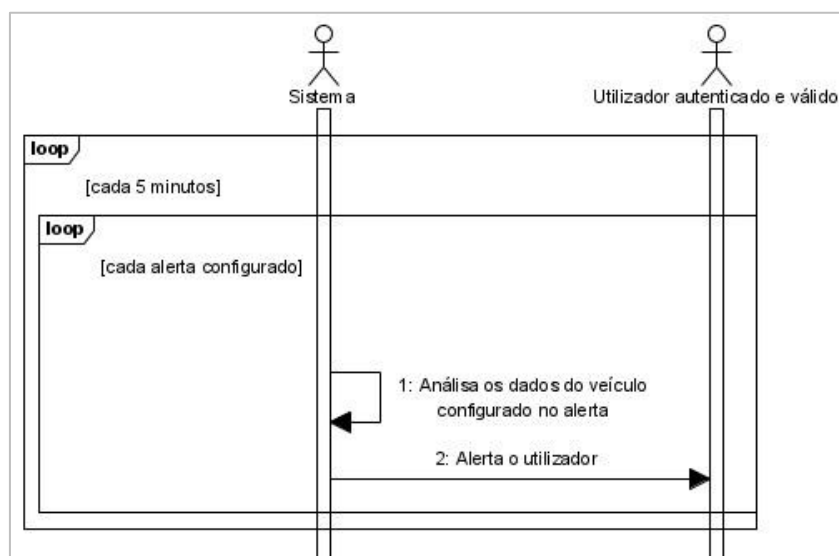


Figura 26 - SSD do mecanismo de envio de alertas configurados pelo utilizador

5.2.2 Requisitos não funcionais

Os requisitos não funcionais têm como objetivo assegurar a qualidade da solução desenvolvida. Estes avaliam os requisitos de usabilidade, fiabilidade, desempenho, suportabilidade e as restrições de *design* que estão associadas ao projeto. Assim, nos seguintes subcapítulos do documento são listados os requisitos não funcionais identificados para o CCVE.

5.2.2.1 Usabilidade

Esta categoria agrega os requisitos relacionados com a interação entre o utilizador e a interface gráfica que lhe é apresentada, tendo em consideração atributos como a acessibilidade, estética e consistência da interface (Center of Excellence for Professional Development, s.d.).

Deste modo, foram identificados os seguintes requisitos não funcionais:

- UI simples e eficiente, compatível com o browser *Chrome*;
- UI responsiva, garantindo a consistência entre plataformas móveis e *web*;
- Interface responsável por informar corretamente o utilizador de situações de erro;

5.2.2.2 Fiabilidade

Esta categoria avalia os requisitos relacionados com a disponibilidade, eficácia e recuperabilidade na ocorrência de falhas que afetem a disponibilidade da aplicação (Center of Excellence for Professional Development, s.d.).

Assim, foram identificados os seguintes requisitos:

- Os serviços fornecidos devem suportar uma grande quantidade de utilizadores em simultâneo, sem degradação de performance;
- A aplicação deve ser tolerante a falhas;
- Após a ocorrência de uma situação de falha no servidor, a aplicação deve estar imediatamente operacional, minimizando o *downtime*.

5.2.2.3 Desempenho

Esta categoria avalia os requisitos relacionados com a taxa de transferência de informação pelo sistema, tempo de resposta dos serviços e desempenho relacionado com a recuperação de falhas (Center of Excellence for Professional Development, s.d.).

Foram identificados os seguintes requisitos nesta categoria:

- Aplicação de mecanismos de desenvolvimento de *software web*, como *lazy-loading*, com o objetivo de reduzir o tempo de carregamento da plataforma e garantir tempos de resposta curtos aos utilizadores;
- O desempenho do servidor deve adaptar-se ao número de utilizadores concorrentes, de modo a evitar a degradação deste.

5.2.2.4 Suportabilidade

Nesta categoria são especificados os restantes requerimentos, tais como a compatibilidade, manutenção e escalabilidade (Center of Excellence for Professional Development, s.d.).

Para esta categoria, foram identificados os seguintes requerimentos:

- **Escalabilidade:** A aplicação tem de ser desenhada e desenvolvida de modo que novos serviços e funcionalidades possam ser integrados de forma simples;
- **Segurança:** Os mecanismos de autenticação utilizados têm de ser seguros e a comunicação entre os diferentes componentes da aplicação tem de ser segura e encriptada, recorrendo ao protocolo HTTPS;
- **Compatibilidade:** A aplicação tem de ser compatível com a versão mais recente do *browser* Google Chrome, para a plataforma móvel e *web*.

5.2.2.5 Restrições de *design*

Nesta categoria são especificadas as restrições de *design* impostas ao desenvolvimento do projeto e que devem ser consideradas na construção da infraestrutura do sistema (Center of Excellence for Professional Development, s.d.). Deste modo, foram identificadas as seguintes restrições:

- Aplicação *web*, adaptada para dispositivos móveis;
- Utilização da base de dados *MongoDB*;
- Adoção de boas práticas de desenvolvimento de *software*.

5.3 Modelação

5.3.1 Modelo de Domínio

O modelo de domínio é um diagrama que permite descrever, de uma forma visual, a estrutura de um sistema. Este diagrama apresenta os objetos de interesse do sistema, juntamente com os seus atributos e operações, bem como as relações entre eles (Visual Paradigm, s.d.). Na Figura 27, é apresentado o modelo de domínio, em notação UML, para o sistema a desenvolver.

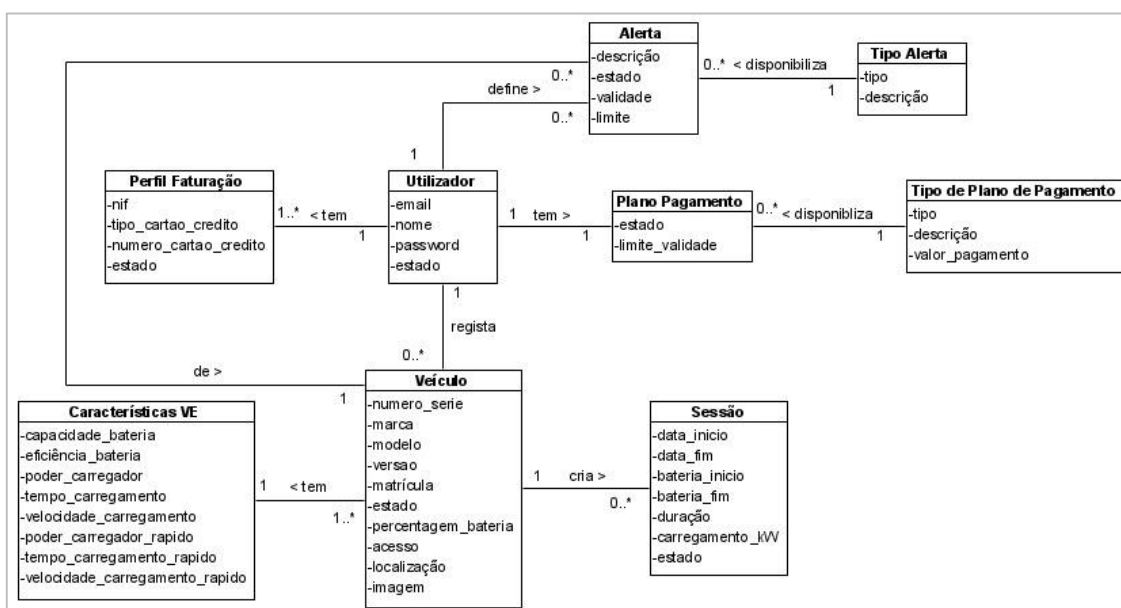


Figura 27 - Modelo de Domínio

Na Tabela 14, são descritos os objetos, e as respetivas relações entre estes, apresentados na Figura 27.

Tabela 14 - Descrição das entidades apresentadas no Modelo de Domínio

Entidade	Descrição	Relações
Utilizador	Representa um cliente da plataforma e contém os dados deste.	<ul style="list-style-type: none"> • Pode registar nenhum ou mais veículos. • Pode definir um ou mais perfis de faturação. • Tem associado um plano de pagamento. • Pode definir nenhum ou mais alertas.

Veículo	Entidade que contém os dados da viatura de um utilizador.	<ul style="list-style-type: none"> • Tem associado o conjunto de características mecânicas do veículo elétrico. • Pode criar nenhuma ou mais sessões de carregamento. • Pode estar presente em nenhum ou mais alertas.
Características VE	Contém as características mecânicas de um veículo elétrico.	<ul style="list-style-type: none"> • As mesmas características mecânicas de uma viatura elétrica podem estar presentes em um ou mais veículos registados pelo utilizador.
Sessão	Contém a informação de uma sessão de carregamento elétrico de uma viatura.	<ul style="list-style-type: none"> • Uma sessão de carregamento é associada a um veículo.
Perfil Faturação	Contém os dados de faturação associados a um utilizador.	<ul style="list-style-type: none"> • Um perfil de faturação é associado a um utilizador.
Plano Pagamento	Contém a informação do plano de pagamento ativo para um utilizador	<ul style="list-style-type: none"> • Um plano de pagamento é associado a um utilizador. • Um plano de pagamento tem um tipo de plano de pagamento associado.
Tipo de Plano de Pagamento	Contém a informação de um tipo de plano de pagamento pré-definido e disponibilizado pela plataforma.	<ul style="list-style-type: none"> • Um tipo de plano de pagamento pode estar presente em nenhum ou mais planos de pagamento.
Alerta	Contém a informação de um alerta configurado por um utilizador.	<ul style="list-style-type: none"> • Um alerta está relacionado com um utilizador. • Cada alerta corresponde a um tipo de alerta no sistema. • Referente a um ou mais veículos,
Tipo de Alerta	Contém a informação de um tipo de alerta pré-definido e disponibilizado pela plataforma.	<ul style="list-style-type: none"> • Um tipo de alerta pode estar presente em nenhum ou mais alertas definidos pelo utilizador.

5.3.2 Infraestrutura

A infraestrutura do Centro de Controlo para Veículos Elétricos tem em consideração que a aplicação tem de estar acessível, através de um *Browser*, em múltiplos tipos de dispositivos. Para tal, esta solução necessita de estar alocada num servidor, no qual as instâncias destes componentes, que estão a fornecer os serviços ao utilizador, comunicam entre si. A informação é centralizada neste servidor, sendo que este disponibiliza o portal do CCVE. Na Figura 28, é

apresentado o modelo da vista lógica para a solução, no qual é ilustrada a comunicação entre os diferentes componentes da solução.

De acordo com a Figura 28, o Portal comunica apenas com um componente, denominado de Centro de Controlo. Este componente é responsável por fornecer os dados necessários para a utilização do portal *web*. Para além disso, este também é responsável pelo processamento e encaminhamento das ações realizadas pelos utilizadores no portal para o respetivo serviço que é responsável por processar essa ação. Deste modo, a utilização destes serviços numa arquitetura de microsserviços permite que cada serviço seja responsável por uma componente de integração específica com uma API disponibilizada por uma marca automóvel. Deste modo, é possível desenvolver uma infraestrutura facilmente escalável, sendo que novas integrações podem ser rapidamente incorporadas na infraestrutura atual, sem serem necessárias alterações nos componentes já existentes.

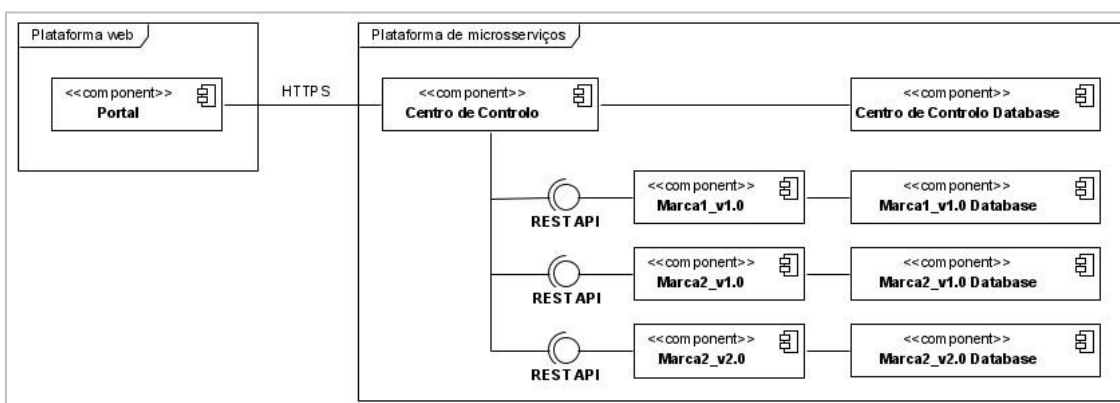


Figura 28 - Vista lógica da infraestrutura do sistema

Assim, cada serviço tem acesso à sua base de dados, onde são apenas guardados os dados relativos a esse serviço. Cada serviço é depois responsável por expor as suas funcionalidades, através de uma REST API ao Centro de Controlo, sendo este depois responsável pela distribuição das tarefas a realizar no sistema, de acordo com o tipo de marca ou veículo associado ao pedido do utilizador.

Durante o processo de investigação de tecnologias a serem utilizadas no desenvolvimento de um serviço capaz de processar múltiplos pedidos a diferentes APIs, foram consideradas duas hipóteses. A primeira hipótese consistia no desenvolvimento de uma REST API, que seria a única responsável por todas as comunicações entre as APIs das marcas de automóveis, em que o

processamento da informação recolhida estava centralizado nesse serviço. A segunda hipótese consistia na criação de uma rede de microsserviços, em que as tarefas de comunicação e processamento eram divididas por cada marca automóvel, sendo cada microsserviço responsável por estabelecer comunicação com uma API de uma marca específica e gerir o processamento dos dados. Este serviço era depois responsável por expor essa informação para o Centro de Controlo, sendo este responsável por comunicar esses dados ao portal do cliente.

Através da aplicação da metodologia de investigação descrita na secção 4.4, e após uma análise extensiva, optou-se pelo desenvolvimento de uma rede de microsserviços, uma vez que esta:

- Permite reduzir o tempo de processamento concorrente dos dados;
- Permite a redução do tempo de resposta;
- De um ponto de vista de gestão, torna mais simples as alterações ou a manutenção de um serviço específico que comunica com uma API de uma marca de automóveis, quando toda a lógica da operação se encontra centralizada em apenas um microsserviço.

Na Figura 29, é apresentado o diagrama de implantação. Este descreve o processo físico de alojamento dos componentes do sistema, e das suas interações, em componentes de *hardware*. Assim, os diversos componentes da plataforma estão hospedados num servidor Linux, sendo que o portal *web* é exposto para comunicação exterior ao servidor, de modo a ser acessível aos utilizadores, através de um *browser*.

Neste diagrama, também é possível visualizar os componentes que formam cada microsserviço da solução e a respetiva comunicação entre eles. Deste modo, o microsserviço denominado de Centro de Controlo é responsável por validar o acesso à informação na plataforma e reencaminhar os pedidos do portal *web* para os respetivos microsserviços responsáveis pelo seu processamento. Logo, a implantação deste serviço é obrigatória para o correto funcionamento da solução, no entanto, como os restantes microsserviços, como são entidades autónomas e isoladas, podem ser implantados de forma independente.

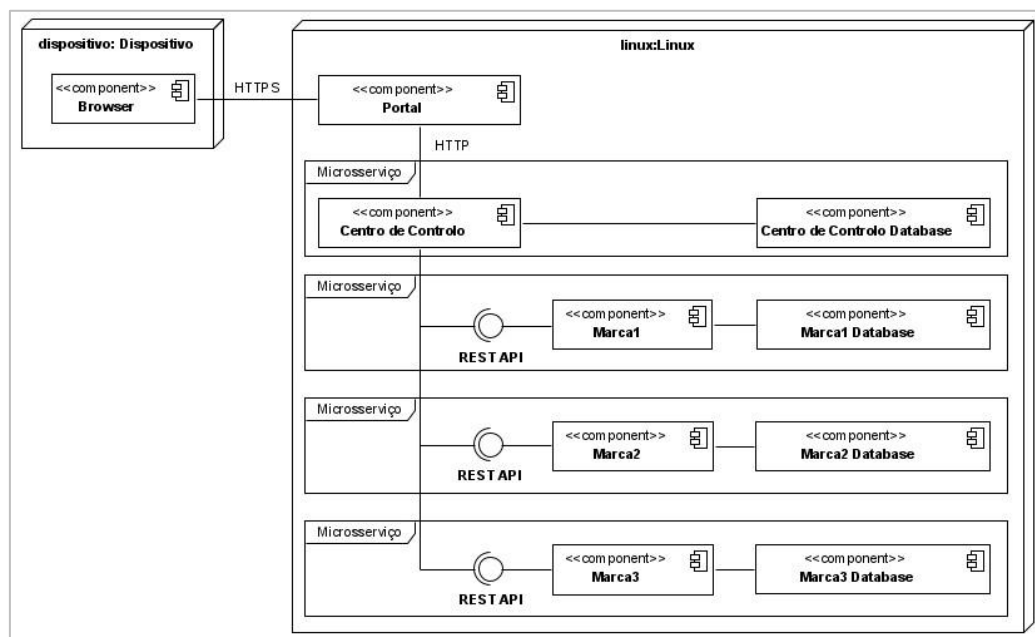


Figura 29 - Diagrama de implantação do sistema

5.4 Estratégia de desenvolvimento

O desenvolvimento do CCVE é dividido em duas fases. Numa primeira fase, é necessário configurar as definições do serviço que vai disponibilizar informação ao portal, desenvolver funcionalidades que permitam a comunicação com os microserviços, inicializar a configuração da base de dados e elaborar o modelo dos dados a serem guardados. O desenvolvimento dos microserviços está dependente da comunicação com as APIs disponibilizadas pelas marcas automóveis, de modo a garantir o acesso às informações dos veículos. Em simultâneo, o desenvolvimento inicial do portal é coordenado com os elementos responsáveis pela elaboração do *design* dos produtos da empresa, de modo a garantir que o *design* do portal e a experiência do utilizador estão de acordo com os restantes produtos.

Após a configuração do servidor, da elaboração do *design* do portal e dos serviços que expõem informação para esse portal, é então inicializada a implementação do portal. Numa primeira fase, serão utilizados dados simulados, iguais aos dados que são esperados pela integração com as marcas automóveis, enquanto não existir um serviço que estabeleça uma ligação funcional e testável com um veículo, sendo a criação desta comunicação o foco principal da primeira fase de desenvolvimento.

A partir do momento em que exista um serviço que permita a comunicação e coleção de informação diretamente de um veículo, é iniciada a segunda fase de desenvolvimento. Nesta fase, o portal passa a apresentar a informação coletada e deixam de ser utilizados dados simulados, passando o desenvolvimento do portal a ser o foco principal da segunda fase.

5.5 Método de avaliação

De modo a garantir que as funcionalidades apresentadas aos utilizadores correspondem aos requisitos que o produto deve fornecer, é necessário avaliar o grau de relação entre os requisitos identificados e as respetivas implementações no projeto.

De forma a avaliar o projeto, foi utilizada a *Quantitative Evaluation Framework* (QEF), uma vez que esta “tem como base estruturante, na sua criação, os objetivos, princípios e ações de engenharia de software (...) relativos à criação e desenvolvimento de modelos conceptuais para a avaliação de conteúdos digitais. (...) Este ambiente de avaliação, poderá ser aplicado, genericamente, no desenvolvimento de qualquer plataforma de sistemas educacionais, no sentido de validar e avaliar, durante o seu ciclo de desenvolvimento qualquer fase da sua construção, permitindo detetar e corrigir, precocemente, as falhas que eventualmente vão ocorrendo” (Escudeiro & Bidarra, 2008).

Assim, a *framework* QEF permite agrupar os requisitos funcionais e não funcionais identificados na fase de análise do projeto em dimensões, sendo que para cada dimensão, os requisitos são divididos em fatores, de acordo com o seu tipo de especificidade característica. Para cada requisito, é necessário definir também um valor, entre 0 e 10, que corresponde ao peso (importância) que o desenvolvimento desse requisito tem no contexto geral do projeto. Por último, para cada requisito é também definida a percentagem de conclusão, sendo esta responsável por indicar o grau de implementação da tarefa.

Assim, para cada requisito, é definido um conjunto de percentagens de conclusão, indicativas do grau de realização da tarefa, e a métrica de avaliação. Deste modo, de acordo com a métrica de avaliação definida e o grau de realização da tarefa, é possível determinar a respetiva percentagem de conclusão para essa tarefa.

De acordo com a *framework* QEF, a medição da qualidade é linear em relação à percentagem de conclusão dos requisitos, o que permite relacionar o cálculo da qualidade com a

percentagem de conclusão dos requisitos. Deste modo, é possível avaliar, em qualquer fase do desenvolvimento do projeto, a qualidade dos requisitos estabelecidos.

6 Desenvolvimento da solução

Este capítulo pretende explicar como foram desenvolvidos os componentes cuja descrição foi abordada nos capítulos anteriores. Deste modo, os seguintes subcapítulos descrevem o processo de desenvolvimento da rede de microsserviços e do portal *web*, componentes que integram a solução proposta na secção 3.4 deste documento.

6.1 Desenvolvimento dos serviços

Até ao término da fase de análise do projeto, não foi possível assegurar uma colaboração entre a EVIO e uma das marcas automóveis mencionadas anteriormente, pelo que a possibilidade de comunicação com veículos existentes deixou de ser uma realidade. Assim, optou-se por desenvolver um serviço que replica as funcionalidades do serviço disponibilizado pela aplicação da Tesla, simulando o comportamento de viaturas dessa marca. Este serviço tem o objetivo de permitir validar as funcionalidades que são expostas pelos serviços para o portal *web*.

Deste modo, a rede de microsserviços desenvolvida para a solução atual é composta por três serviços: o serviço principal, também denominado nas secções anteriores como Centro de Controlo, um serviço responsável pela integração com a API da Tesla e um outro serviço que disponibiliza funcionalidades para testar essa integração, simulando o comportamento esperado pela API da Tesla.

6.1.1 Centro de Controlo

O desenvolvimento do Centro de Controlo envolveu a criação de uma estrutura responsável pela centralização e abstração dos dados que são comunicados entre os diferentes componentes da plataforma, operando como um *gateway* entre o portal e os serviços. Este não só disponibiliza *endpoints* para que o portal *web* possa comunicar, como também por sua vez comunica com os *endpoints* disponibilizados pelos serviços utilizados no processamento dos pedidos dos utilizadores.

O Centro de Controlo também comunica com uma base de dados, sendo esta utilizada para registar todos os dados que não se associam apenas a um serviço específico e são genéricos para todos os utilizadores da aplicação. Nas secções seguintes, estão listados os principais *endpoints* deste servidor, juntamente com uma breve descrição do propósito de cada um.

6.1.1.1 User

O *endpoint* `/user` é utilizado para a criação de um utilizador no sistema. Este é um pedido POST, no qual é enviado o email do utilizador, o nome a ser associado ao perfil e a respetiva *password*. De modo a criar o perfil de faturação, o utilizador fornece também o número de identificação fiscal (NIF) e o plano de pagamento selecionado, juntamente com o número e o tipo do cartão utilizado para o pagamento.

6.1.1.2 Vehicle

O *endpoint* `/vehicle` é utilizado para a interação entre um utilizador e um veículo. Este é utilizado em quatro tipos de pedidos:

- GET – para obter a informação de um veículo;
- POST – para registar um novo veículo. Neste tipo de pedido devem ser incluídos a marca, o modelo e a versão da viatura, o número de série que serve para identificar o veículo na marca automóvel e a matrícula;
- PATCH – permite alterar os dados de uma viatura já existente. Neste tipo de pedidos são recebidos os novos dados para atualizar;
- DELETE – permite apagar um veículo registado.

No exemplo presente no Código 1, é apresentado um objeto JSON que contém os dados necessários para registar um novo veículo na plataforma.

```
{
  "brand": "Tesla",
  "model": "Model Y",
  "version": "Performance",
  "license_plate": "XX-XX-XX",
  "serial_number": "XXXXXXXX"
}
```

Código 1 - Exemplo de um objeto JSON utilizado para registar uma viatura na plataforma

6.1.1.3 Command

O *endpoint /command* deve ser utilizado para o envio de comandos a serem realizados numa viatura. Através de um pedido POST, este recebe um objeto JSON com número de identificação (ID) do veículo, de modo que o *endpoint* a ser utilizado é independente da marca da viatura que realiza o pedido. De modo a executar o comando, o servidor avalia, através do ID recebido, qual o serviço responsável por processar os comandos para a marca do veículo em questão e reencaminhar o pedido para o serviço adequado. Deste modo, os seguintes *endpoints* são disponibilizados pelo sistema:

- */command/door_lock* – *Endpoint* genérico utilizado para bloquear as portas de uma viatura;
- */command/door_unlock* – *Endpoint* genérico utilizado para desbloquear as portas de uma viatura;
- */command/start_air_conditioning* – *Endpoint* genérico utilizado para iniciar o ar condicionado de uma viatura;
- */command/stop_air_conditioning* – *Endpoint* genérico utilizado para parar o ar condicionado de uma viatura.

6.1.1.4 Statistics

As funcionalidades das estatísticas estão disponíveis através de dois *endpoints* que aguardam pedidos do tipo POST. O *endpoint /getUserStatistics* permite obter os dados estatísticos genéricos das ações realizadas pelo utilizador, no sistema, ao longo de um determinado período. Este analisa os dados recolhidos dos veículos, de modo a determinar o número de sessões de carregamento efetuadas, a quantidade de energia consumida nesses carregamentos e o tempo médio por carregamento. O *endpoint /getVehicleStatistics* é utilizado para obter os dados estatísticos relacionados com a utilização de um veículo.

6.1.1.5 Alert

O *endpoint /alert* deve ser utilizado para configurar os alertas de um veículo específico, dos quais um utilizador deseja ser notificado. Através de um pedido POST, é enviado um objeto JSON que contém o tipo de alerta a configurar, o “vehicleId” do veículo ao qual será associado o alerta e o respetivo limite que é responsável por despoletar a notificação para o utilizador. O atributo “vehicleId” é o número de identificação da viatura no sistema.

Deste modo, o servidor analisa periodicamente os dados recebidos do veículo selecionado e despoleta o alerta caso o valor limite configurado seja atingido. No exemplo do Código 2, é apresentado o modelo de dados de um alerta.

```
{
  "type": "BATTERY_PERCENTAGE",
  "typeDescription": "Based on battery percentage",
  "alertLimit": 15,
  "valid": true,
  "userId": "987654321",
  "vehicleId": "123456789"
}
```

Código 2 - Exemplo de um objeto JSON que contém a informação de um alerta

6.1.2 Serviço de integração Tesla

O desenvolvimento deste serviço passou por desenvolver uma estrutura que centraliza as funcionalidades e dados relativos a veículos da marca Tesla.

O desenvolvimento da estrutura deste serviço realizou-se de acordo com documentação não oficial disponível em bibliotecas que foram desenvolvidas previamente e que estabeleceram comunicação com os serviços disponibilizados pela Tesla, tal como é referido no capítulo 2.3.2.

Para permitir o acesso às suas funcionalidades e à informação registada na base de dados, este disponibiliza uma API para comunicar com o Centro de Controlo. Nas secções seguintes encontram-se listados os *endpoints* da API, juntamente com uma breve descrição da funcionalidade de cada um.

6.1.2.1 Authentication

O *endpoint /authentication* de ser utilizado para autenticar um utilizador no sistema da marca automóvel. Este *endpoint* requer um pedido POST, no qual é enviado um objeto JSON com as credenciais de autenticação do utilizador. Este pedido, se concluído com sucesso, devolve o *token* de autenticação necessário para realizar todos os restantes pedidos na API da Tesla, uma vez que é através deste *token* que esta identifica o utilizador que está a realizar o pedido no sistema e retorna os seus dados.

6.1.2.2 Vehicle

O endpoint `/data` é utilizado para obter a informação do estado atual de uma viatura. Este recebe um pedido POST que rem agregado um objeto JSON que contém o “`userId`” do utilizador que realiza o pedido e o “`vehicleId`” do veículo cujos dados deseja consultar. O atributo “`userId`” é o número de identificação do utilizador no sistema e o “`vehicleId`” o número de identificação da viatura no sistema. Na construção do pedido para a API da marca automóvel, é necessário adicionar o *token* do utilizador obtido na fase de autenticação, senão o pedido é recusado.

No caso de o processo despoletado pelo pedido concluir com sucesso, o dispositivo que enviou o pedido POST recebe um objeto JSON com os dados atuais da viatura, tal como é ilustrado no Código 3.

```
{
  "drive_state": {
    "latitude": 41.17264250337961,
    "longitude": -8.680122860023275,
    "speed": 0
  },
  "climate_state": {
    "is_auto_conditioning_on": false,
    "is_climate_on": false,
    "remote_heater_control_enabled": false
  },
  "charge_state": {
    "battery_level": 67,
    "charging_state": "Charging",
    "minutes_to_full_charge": 36
  },
  "vehicle_state": {
    "locked": true,
    "remote_start": false,
    "remote_start_enabled": true
  }
}
```

Código 3 - Exemplo de um objeto JSON obtido como resposta a um pedido de atualização de dados da viatura

6.1.2.3 Command

O *endpoint* `/command` é utilizado para o envio de comandos a serem realizados numa viatura. Este é um pedido POST que recebe um objeto JSON com o número de identificação do veículo no sistema. Na preparação do pedido a enviar para a API da marca automóvel é necessário adicionar o *token* do utilizador obtido na fase de autenticação. Para viaturas Tesla, todos os comandos suportados pela API oficial da marca são também suportados no Centro de Controlo,

sendo que os *endpoints* partilham a mesma nomenclatura e disponibilizam as mesmas funcionalidades, previamente descritas na secção 6.1.1.3 do documento.

6.1.3 Serviço de simulação Tesla

O desenvolvimento do serviço de simulação Tesla baseou-se na criação de uma estrutura que está de acordo com a documentação não oficial disponível em bibliotecas que foram desenvolvidas previamente e que estabeleceram comunicação com os serviços disponibilizados pela Tesla, tal como é referido no capítulo 2.3.2.

Este simula o comportamento da API da marca automóvel, de forma a garantir que o seu modus operandi se mantém fiel ao componente original que este pretende imitar. Como tal, qualquer pedido a este serviço tem de conter os mesmos objetos JSON, *headers* e/ou *tokens* requeridos pelo componente original.

Os seguintes subcapítulos listam os *endpoints* que o serviço de simulação Tesla disponibiliza, descrevendo brevemente a funcionalidade de cada um.

6.1.3.1 Authentication

O *endpoint* `/oauth/token` é utilizado para obter o *token* de autenticação no sistema da Tesla. Este é um pedido POST que recebe um objeto JSON com as credenciais de autenticação do utilizador. Por sua vez, este retorna um objeto com o *token* de acesso, o seu tipo e a data de validade deste, tal como é ilustrado no exemplo presente no Código 4.

```
{
  "access_token": "abc123",
  "token_type": "Bearer",
  "expires_in": 3888000,
  "refresh_token": 321cba
}
```

Código 4 - Exemplo de resposta ao pedido de autenticação

6.1.3.2 Vehicle

O *endpoint* `/api/1/vehicles/:id` é utilizado para obter os dados atuais de um veículo. Este é um pedido GET com o respetivo "id" anexado nos parâmetros do URL. O "id" é o identificador da viatura no sistema da Tesla. A resposta ao pedido retorna um objeto JSON com os dados da viatura, demonstrado previamente no excerto de Código 3.

6.1.3.3 Command

O *endpoint* `/api/1/vehicles/:id/command/:command` é utilizado para executar comandos numa determinada viatura. Este é um pedido POST com o respetivo “id” e “command” anexado nos parâmetros do URL. O “id” permite identificar a viatura no sistema da Tesla, enquanto o parâmetro “command” corresponde ao tipo de comando que este pretende executar. Este *endpoint* retorna um objeto JSON com o resultado da execução do comando, tal como é ilustrado no exemplo presente no Código 5.

```
{
  "reason": "",
  "result": true
}
```

Código 5 - Exemplo de resposta à execução de um comando num veículo

Em seguida encontram-se listados alguns exemplos do tipo de comandos que são suportados pelo serviço:

- `/api/1/vehicles/123456789/command/door_unlock` – *Endpoint* que executa o comando de desbloqueio das portas da viatura identificada pelo número 123456789;
- `/api/1/vehicles/123456789/command/door_lock` – *Endpoint* que executa o comando de bloqueio das portas da viatura identificada pelo número 123456789;
- `/api/1/vehicles/123456789/command/auto_conditioning_start` – *Endpoint* que executa o comando de início do ar condicionado da viatura identificada pelo número 123456789;
- `/api/1/vehicles/123456789/command/auto_conditioning_stop` – *Endpoint* que executa o comando de paragem do ar condicionado da viatura identificada pelo número 123456789.

6.2 Desenvolvimento do portal *web*

O desenvolvimento do portal *web* foi executado em duas etapas. Numa primeira etapa, foram elaborados os *mockups* para o portal, de modo a assegurar que o *design* deste está em

concordância com a linguagem de *design* dos produtos já existentes da EVIO. Numa segunda etapa, processou-se então à implementação dos *mockups* previamente elaborados.

Deste modo, foi desenvolvido um portal que permite aos utilizadores gerir e monitorizar em tempo real as suas viaturas, acedendo aos dados das mesmas e permitindo o acesso remoto às funcionalidades disponibilizadas por cada veículo. Este também permite aceder a dados estatísticos, com base na análise dos dados dos automóveis e do histórico da atividade do utilizador na plataforma.

Deste modo, nas próximas secções deste subcapítulo, vão ser listadas as funcionalidades disponibilizadas pelo portal *web*, juntamente com uma descrição da operação efetuada pelo utilizador e dos componentes Angular criados para atingir o *design* adotado.

6.2.1 Registo de utilizador

A funcionalidade de registo de um utilizador presente no portal *web* permite a um cliente a criação de uma nova conta. A sua implementação foi dividida em duas etapas, sendo que, numa primeira etapa, foi necessário desenvolver o processo de registo de um utilizador, através da introdução do nome, email e *password*, garantindo o acesso ao portal. Já na segunda fase, foi necessário desenvolver um mecanismo para que o utilizador seleccionasse um plano de pagamento de acordo com as suas necessidades de utilização do portal, sendo obrigatória a introdução do tipo e número de cartão que ficará associado à conta e através do qual serão realizados pagamentos. Até este momento, apenas a primeira etapa se encontra implementada, uma vez que o processo para seleção de um serviço de faturação e de emissão de faturas ainda está a decorrer. Todas as contas criadas atualmente têm associado o plano de pagamento grátis.

Na Figura 30, está presente o componente RegisterComponent, que aloja o componente RegisterForm. Tal como o nome indica, este contém o formulário que necessita de ser preenchido para criar um novo utilizador no sistema. Ao clicar no botão “Register”, este componente valida os dados inseridos pelo utilizador, garantido que os valores inseridos são válidos e estão de acordo com o tipo de dados esperados. Caso esses valores sejam inválidos, o componente informa o utilizador.

De modo a evitar que sejam introduzidos dados inválidos no sistema, o botão “Register” apenas fica ativo quando todos os dados forem inseridos no formulário.

Figura 30 – Componente de registo de utilizador no portal *web*

Na Figura 31, é apresentado o fluxograma das interações entre o utilizador e o componente que está envolvido no processo de criação de um novo utilizador.

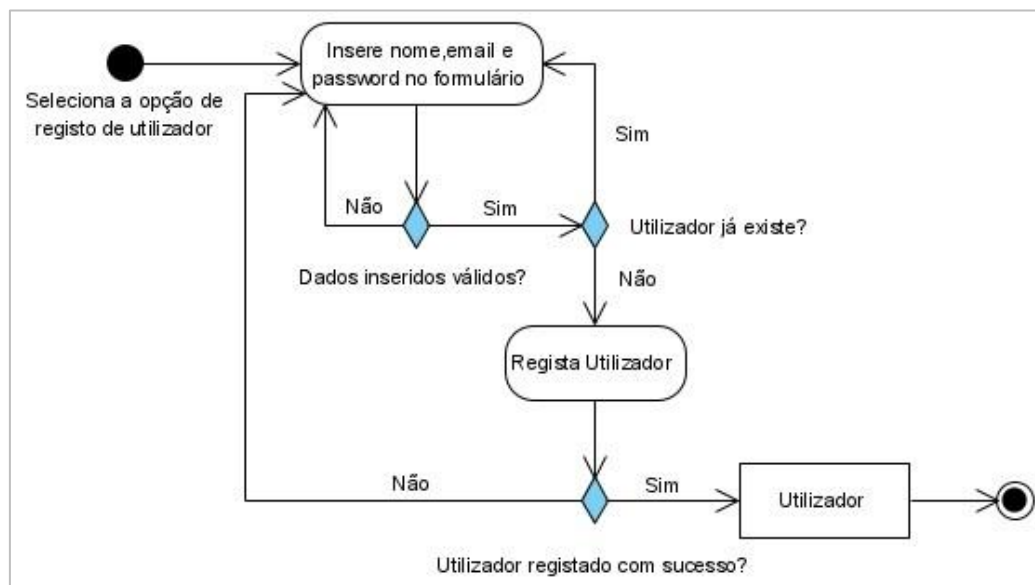


Figura 31 - Fluxograma para o registo de um novo utilizador

6.2.2 Autenticação de utilizador (*Login*)

Esta funcionalidade permite que um utilizador se autentique no sistema, através da introdução do email e da *password* utilizados no processo de criação de conta, de modo a ter acesso às

funcionalidades do portal. Após autenticado, é apresentado ao utilizador o menu principal, que contém todas as funcionalidades a que este tem acesso.

A Figura 32, ilustra o componente LoginComponent. Este componente engloba um outro componente, denominado de LoginForm, que disponibiliza o formulário a ser preenchido no processo de autenticação de um utilizador. Ao clicar no botão “Login”, este último componente valida os dados inseridos pelo utilizador, garantido que estes são válidos, e informa-o do resultado obtido. Caso esses valores sejam inválidos, o componente notifica o utilizador e requisita que sejam inseridos novos dados

De modo a evitar que sejam introduzidos dados inválidos no sistema, o botão “Login” apenas fica ativo quando todos os dados forem inseridos no formulário.

Após a autenticação do utilizador no sistema, este é redirecionado para o menu principal do portal.

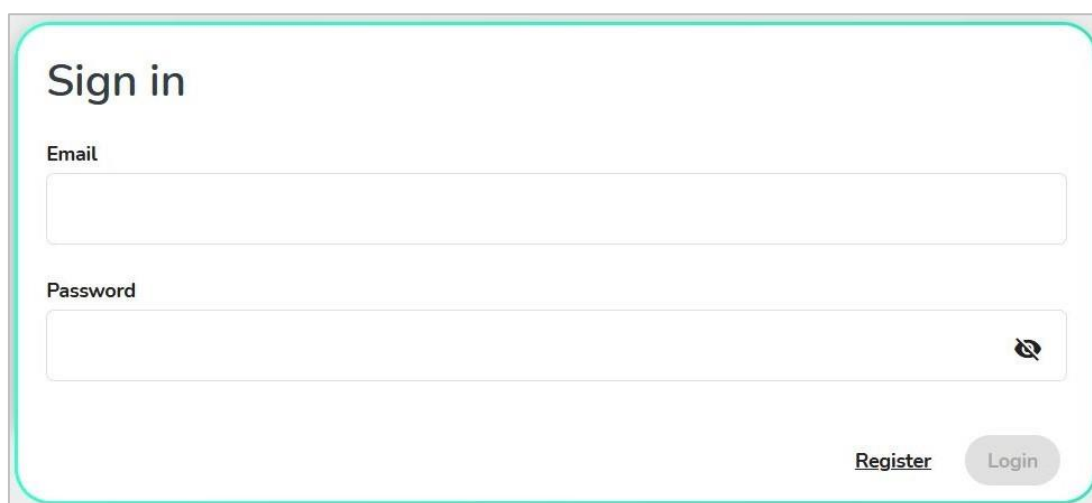
The image shows a web form titled "Sign in" enclosed in a rounded rectangular border. It features two input fields: "Email" and "Password". The "Password" field includes a small icon of an eye with a slash through it, indicating a toggle for password visibility. At the bottom right of the form, there are two buttons: a text link labeled "Register" and a button labeled "Login".

Figura 32 - Componente de autenticação de utilizador no portal *web*

Na Figura 33, é apresentado um fluxograma que contém as interações entre o utilizador e o componente responsável pela autenticação de utilizadores no portal.



Figura 33 - Fluxograma da autenticação de um utilizador no sistema

6.2.3 Página Principal

Após a autenticação de um utilizador, este é recebido com a página (*view*) principal da aplicação, uma vez que esta apresenta, de um modo centralizado, todas as funcionalidades que o portal disponibiliza ao utilizador.

Tal como ilustra a Figura 34, a página principal é composta por três componentes: HeaderComponent (assinalado na figura a amarelo), MenuComponent (assinalado na figura a azul) e FunctionalityComponent (assinalado na figura a vermelho). O HeaderComponent é responsável por apresentar o *branding* da EVIO (lado esquerdo) e indicar qual o utilizador que está atualmente autenticado no portal (lado direito). O MenuComponent aloja a barra de navegação que redireciona o utilizador para as funcionalidades que estão disponíveis e às quais ele tem acesso. Por último, o FunctionalityComponent é responsável por apresentar as funcionalidades relacionadas com a opção selecionada no MenuComponent.

De modo que o utilizador tenha logo acesso às informações dos veículos que registou na plataforma, a opção *Vehicles* na barra de navegação é selecionada por defeito quando a página inicial é apresentada.

Deste modo, dependo da opção que esta selecionada na barra de navegação existente no MenuController, são apresentadas ao utilizador no FunctionalityComponent as funcionalidades correspondentes a essa escolha.

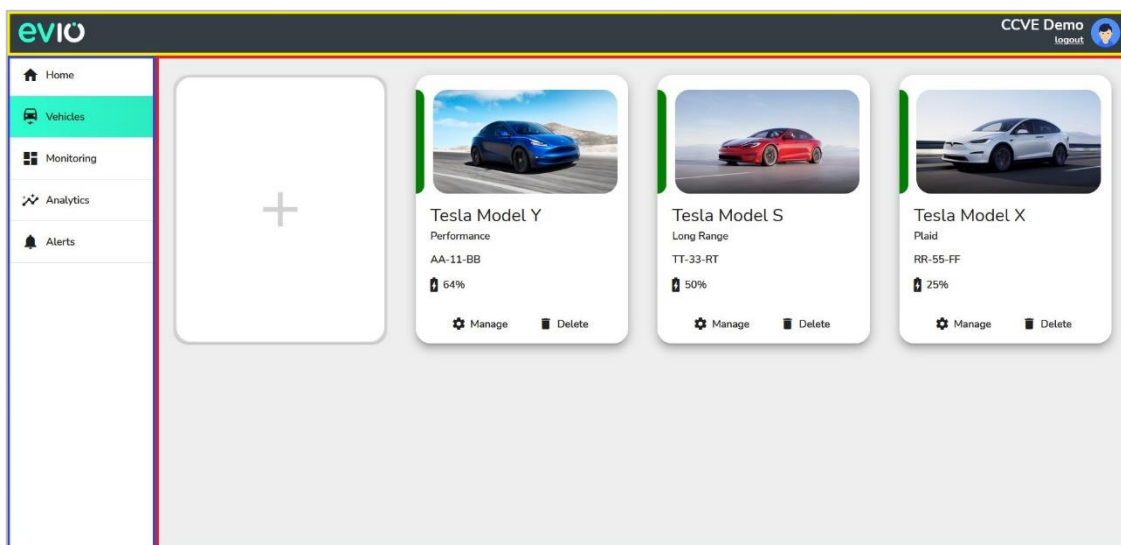


Figura 34 – Página principal do portal

6.2.4 Gestão de veículos

O componente de gestão de veículos, tal como o nome indica, disponibiliza as funcionalidades essenciais à gestão da frota de um utilizador. Para tal, o componente apresenta a lista de automóveis que o utilizador já registou previamente. Para cada viatura, é apresentado o estado da conexão da viatura com a API da respetiva marca e a percentagem de bateria atual. De modo a notificar facilmente o utilizador do estado da comunicação entre cada veículo e a API da sua respetiva marca, a barra lateral presente no componente da viatura muda de cor. Deste modo, caso a conexão seja feita com sucesso e se mantenha estável, a barra lateral assume a cor verde. Já em caso de falha, a barra apresenta a cor preta. Na Figura 35, é apresentado um exemplo do componente de gestão de veículos.

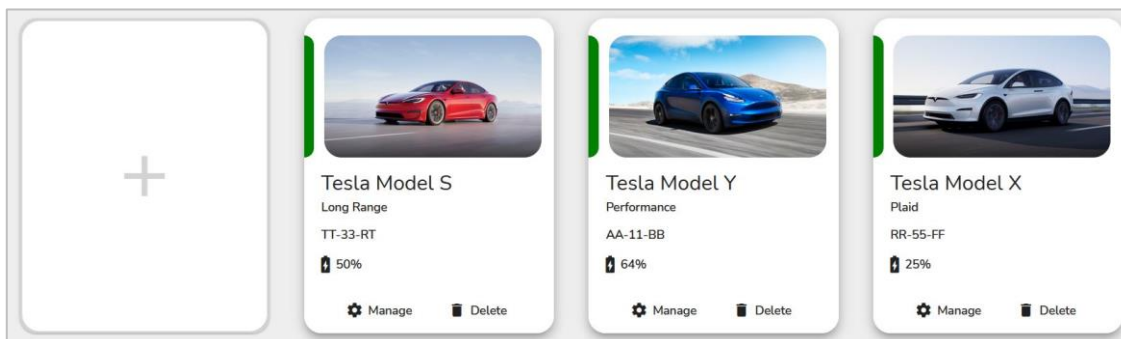


Figura 35 - Componente que disponibiliza operações para a gestão de veículos do utilizador

Os veículos registados através deste componente estão aptos para serem monitorizados na funcionalidade de *Monitoring*, desde que apresentem a cor verde na barra lateral.

6.2.4.1 Registo de veículo


O componente de registo de veículo, denominado de *AddVehicleComponent*, permite adicionar um novo automóvel à plataforma. Para tal, o utilizador terá de preencher um formulário disponibilizado por um outro componente, denominado de *AddVehicleForm*, no qual este pode selecionar a marca, modelo e versão do automóvel e introduzir a matrícula e o número de série que identificam a viatura. Na Figura 36 é apresentado um exemplo do formulário para registo de um automóvel.

Register a vehicle

Brand
Tesla

Model
Model S

Version
Plaid

	250 kWh Battery Capacity	168 Wh/km Efficiency
	16.5 kW AC Plug Power	250 kW DC Fast Plug Power
	6h30m Charge Time	23m Fast Charge Time

Cancel Register

Figura 36 - Formulário de registo de um novo veículo

Após a introdução desses dados, como é visível na Figura 36, é apresentado um resumo com as principais características para o tipo de veículo selecionado. Estas características estão relacionadas com a capacidade da bateria da viatura, os tipos de conetores que esta suporta (carregamento normal ou carregamento rápido) e a respetiva potência de cada um, juntamente com o tempo médio para cada tipo carregamento. Após registar a viatura, esta é acrescentada à lista de veículos já presente na página principal.

Na Figura 37 é apresentado o fluxograma onde são apresentadas as interações entre o utilizador e o formulário de registo de um novo veículo.

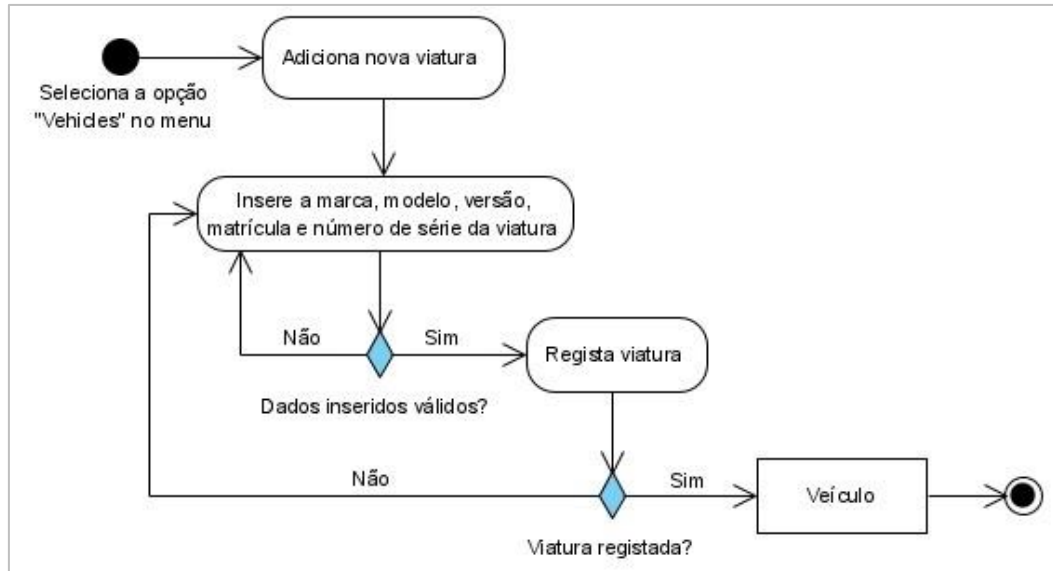


Figura 37 - Fluxograma para o registo de um novo veículo no portal

6.2.5 Monitorização de veículos

O componente de monitorização de veículos disponibiliza as funcionalidades de monitorização de viaturas e de execução de comandos. Neste ecrã, é apresentada a lista de veículos do utilizador, previamente registados, sendo apresentado, para cada viatura, o número de identificação, a percentagem de bateria atual, o estado de carregamento e um mapa com a localização geográfica atual. Esta informação é constantemente atualizada em intervalos de 2 minutos, de modo a ser possível acompanhar as mudanças quase em tempo-real.

Na Figura 38 é apresentado o ecrã de monitorização de veículos, sendo que nas próximas subsecções deste subcapítulo são apresentadas as funcionalidades a que o utilizador tem acesso

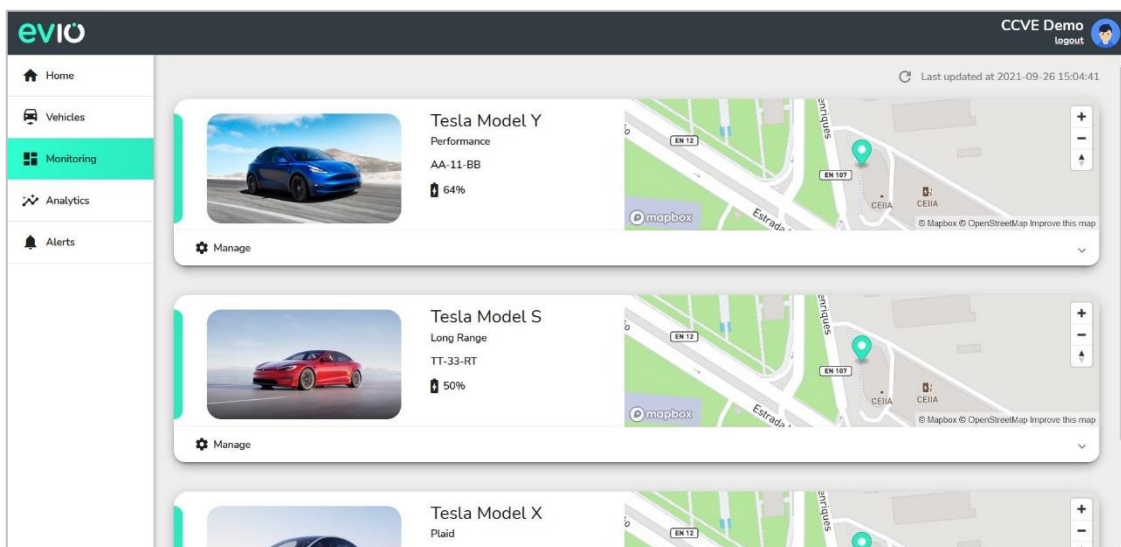


Figura 38 - Ecrã de monitorização de veículos

6.2.5.1 Carregamento de veículo

Para cada viatura é apresentado o estado de carregamento da viatura. O componente em questão indica, através de um ícone ilustrativo, se o automóvel está numa sessão de carregamento e apresenta uma estimativa, em minutos, do tempo necessário para a percentagem de bateria atingir os 100%.

Na Figura 39, é apresentado o estado do componente que apresenta os dados do veículo, no qual está a decorrer uma sessão de carregamento elétrico.

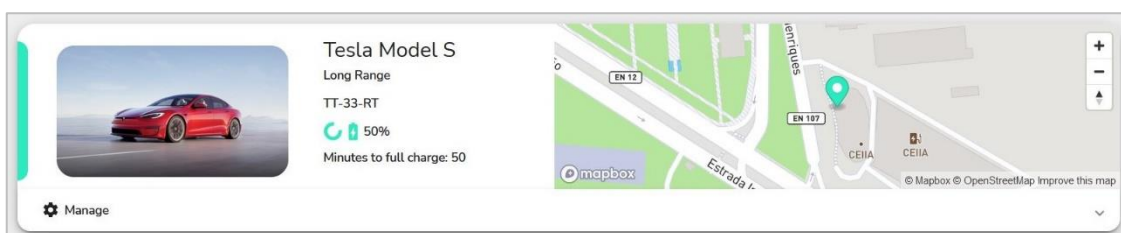


Figura 39 - Componente de monitorização de uma viatura em carregamento

6.2.5.2 Localização do veículo

Para cada viatura é apresentada a sua localização geográfica. A informação que é recebida da marca automóvel é interpretada e apresentada no formato de um mapa, de modo a comunicar mais facilmente ao utilizador a verdadeira posição do veículo.

6.2.5.3 Execução de comandos do veículo

Para cada veículo que permita a execução de comandos, são apresentados os comandos suportados pela integração realizada com as APIs disponibilizadas pelas marcas automóveis. Os comandos apresentados ao utilizador estão, por defeito, de acordo com a situação atual da viatura. Na Figura 40, é apresentado o fluxograma para o processo de execução de um comando numa viatura.

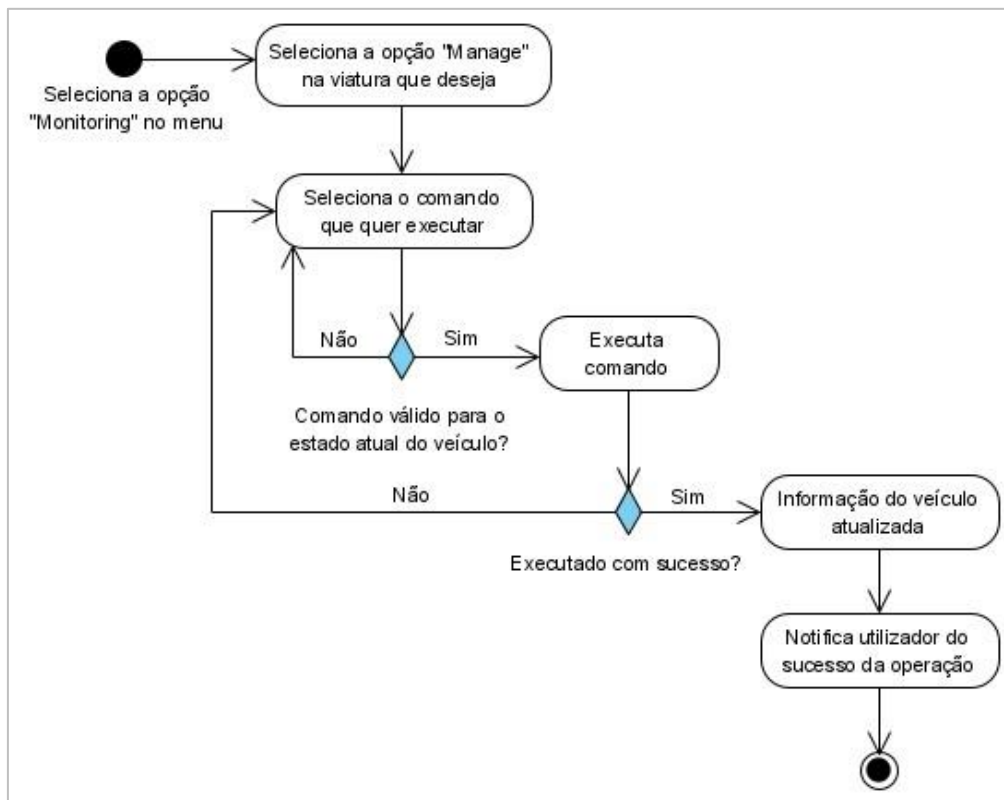


Figura 40 - Fluxograma de execução de um comando numa viatura

Após a execução de um comando, o sistema notifica o utilizador do resultado do comando executado. Um exemplo de uma notificação está visível na Figura 41, após a execução de um comando de desbloqueio de portas de um veículo.

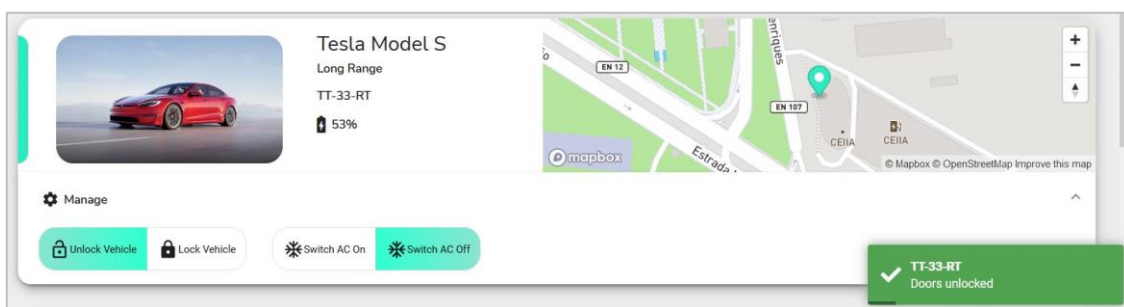


Figura 41 - Execução de um comando de desbloqueio de portas numa viatura

6.2.6 Estatísticas

O componente responsável pelas estatísticas disponibiliza as estatísticas mensais para as atividades que um utilizador realizou com as suas viaturas, tais como carregamentos e deslocações. Para tal, o componente divide os dados em dois subcomponentes, de modo a apresentar a informação de uma maneira mais apelativa e inteligível ao utilizador. Estes subcomponentes são o `userStatisticsComponent` (a azul) e `vehicleStatisticsComponent` (a vermelho) e encontram-se ilustrados na Figura 42.

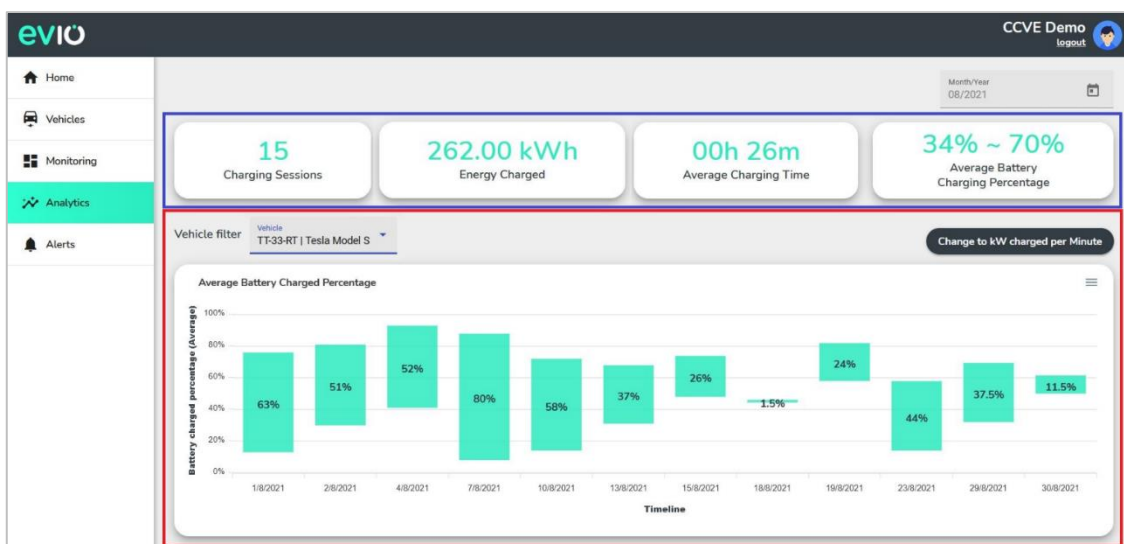


Figura 42 - Ecrã das estatísticas

No `userStatisticsComponent` são apresentados os indicadores-chave de desempenho relativos às sessões de carregamento dos veículos do utilizador. Estes indicadores são:

- **Sessões de carregamento (Charging Sessions):** Todas as sessões de carregamentos efetuadas pelos veículos do utilizador, para o intervalo de tempo selecionado;
- **Energia carregada (Energy Charged):** Quantidade de energia, em kWh, utilizada no carregamento das suas viaturas, para o intervalo de tempo selecionado;
- **Tempo médio de carregamento (Average Charging Time):** Tempo médio de carregamento das viaturas do utilizador, para o intervalo de tempo selecionado;
- **Percentagem média do carregamento da bateria (Average Battery Charging Percentage):** Intervalo médio da percentagem de bateria carregada para todas as viaturas do utilizador, para o intervalo de tempo selecionado. Por exemplo, de acordo o indicador presente na Figura 42, os veículos do utilizador carregaram, em média, durante o mês de agosto de 2021, desde os 34% até os 70% nas sessões de carregamento.

No vehicleStatisticsComponent são apresentadas as estatísticas relativas a um veículo específico do utilizador. Deste modo, são disponibilizados dois tipos de gráficos baseados na análise dos dados recolhidos da viatura. O gráfico presente na Figura 42, apresenta, para o intervalo de tempo definido, a média da percentagem de bateria carregada nas sessões de carregamento efetuadas por dia pela viatura selecionada.

Por sua vez, o gráfico presente na Figura 43 apresenta ao utilizador a média do número de kWh consumidos por minuto em cada sessão de carregamento efetuado pela viatura selecionada, para o intervalo de tempo especificado.

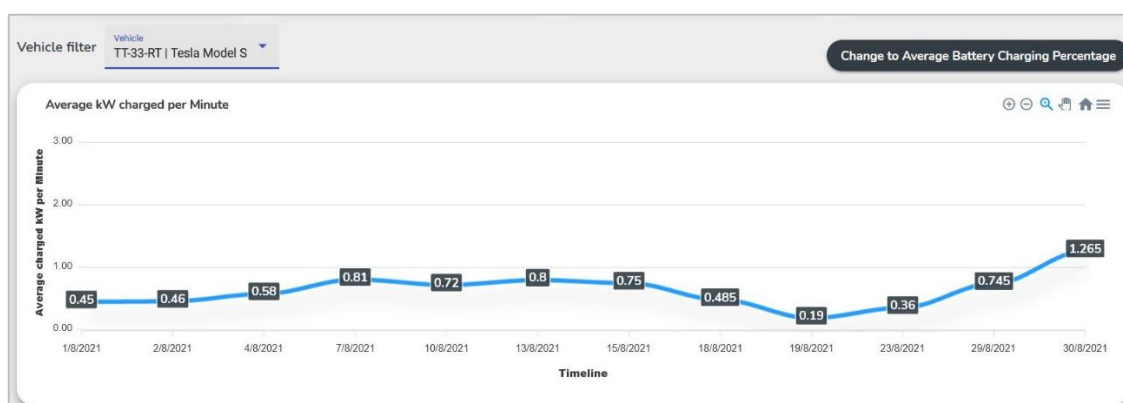


Figura 43 - Representação gráfica da média de kWh carregados por minuto para as sessões efetuadas pelo utilizador

Para expor esta informação ao utilizador foi utilizado um gráfico, uma vez que o objetivo deste é a rápida identificação da possível degradação da bateria da viatura. Para tal, é realizada uma análise a longo prazo dos valores de kWh carregados por minuto nas sessões efetuadas. O algoritmo utilizado para o cálculo dos valores de kWh tem em consideração que o carregamento rápido de uma viatura elétrica varia ao longo do tempo da sessão, uma vez que a potência fornecida para o carregamento, de forma genérica, é constante entre os 10% a 80% de bateria carregada, sendo a potência fornecida fora desse intervalo é progressivamente reduzida. Desta forma, e de modo a reduzir os períodos em que a potência fornecida não é constante, foi apenas considerado o intervalo de carregamento entre os 10% e 80% de bateria para as sessões efetuadas (EV Life, s.d.). Na Figura 44, é apresentada uma curva de carregamento genérica para um carregamento rápido de veículo.

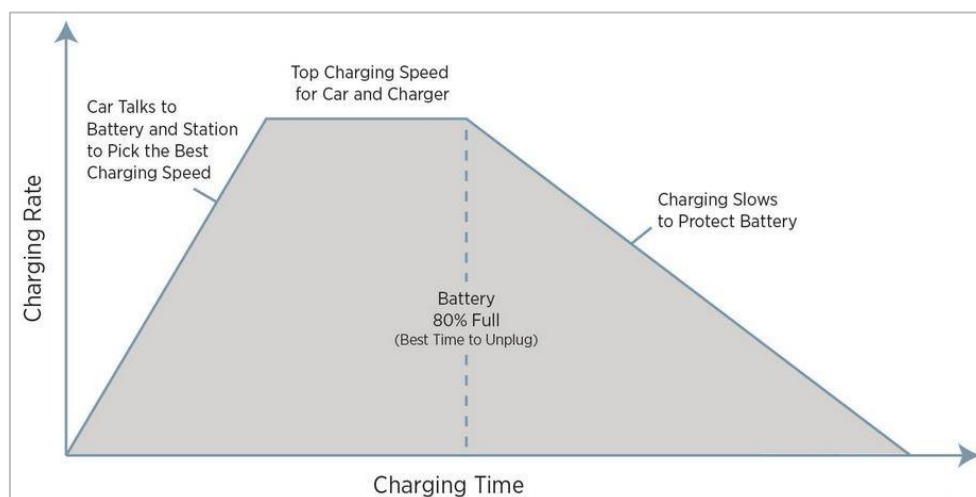


Figura 44 - Curva de carregamento típica para um carregamento rápido de veículo (EV Life, s.d.)

De acordo com a Figura 44, foi apenas considerado o intervalo de carregamento entre os 10% e 80% de bateria, uma vez que:

- Entre os 0% a 10%, o veículo está a comunicar com o posto de carregamento para determinar a melhor velocidade de carregamento.
- Após os 80%, a velocidade de carregamento abranda para proteger a bateria elétrica da viatura.

Na Figura 45, é apresentado o fluxograma das transições entre o utilizador e os componentes relacionamos com as estatísticas.

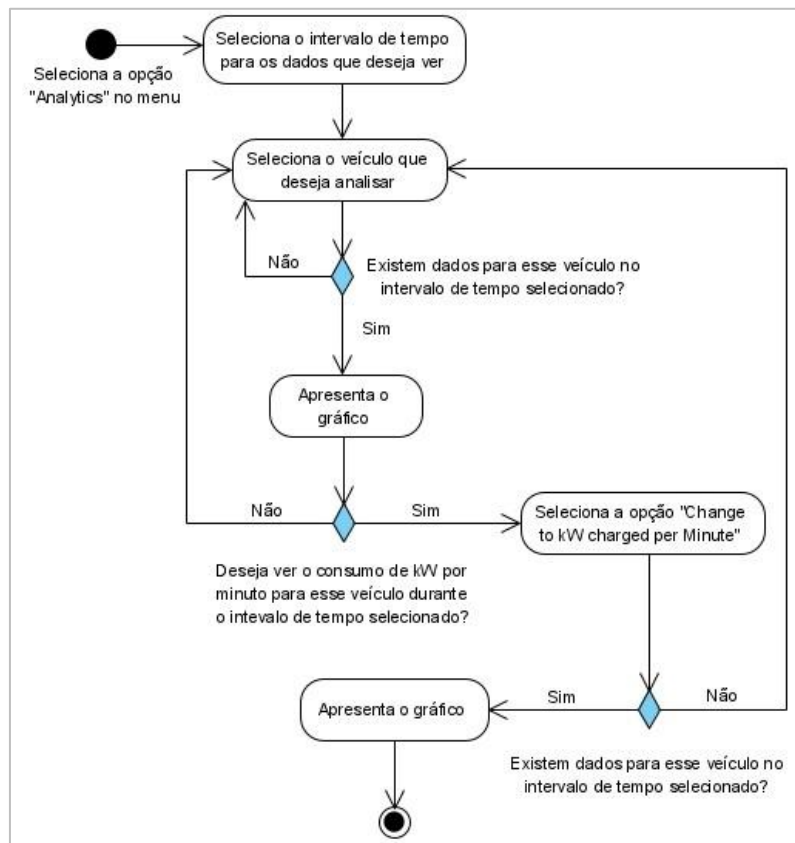


Figura 45 - Fluxograma da interação entre o utilizador e as estatísticas do portal

6.2.7 Alertas

O ecrã de alertas é composto por dois componentes principais: o CreateAlertComponent e o ViewAlertComponent.

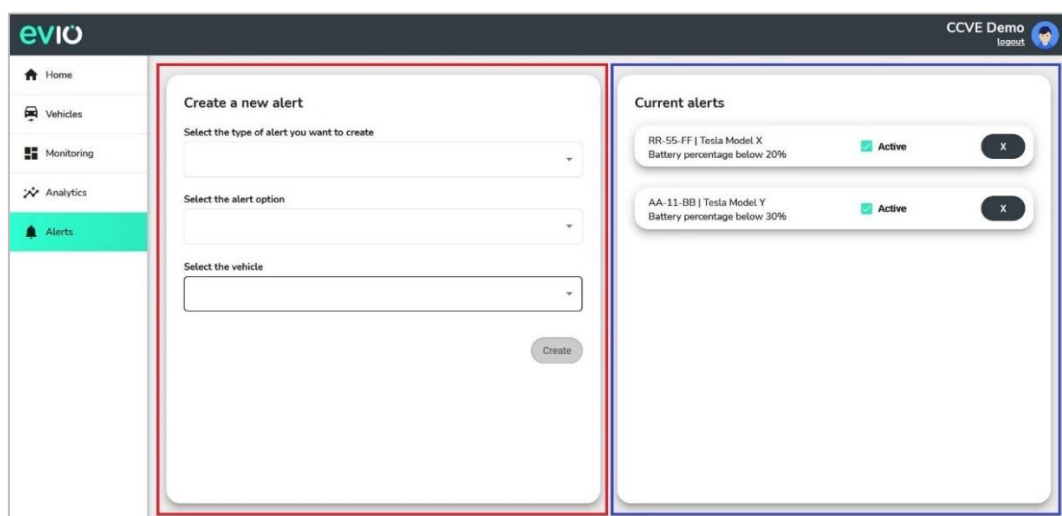


Figura 46 – Ecrã das funcionalidades de alertas

De acordo com a Figura 46, o componente CreateAlertComponent (a vermelho) permite a um utilizador configurar alertas para um determinado veículo, de modo a emitir notificações baseadas nos dados recolhidos. Para tal, é apresentado ao utilizador um formulário, denominado de AlertForm, no qual este selecciona o tipo de alerta que deseja configurar, o limite responsável por despoletar esse tipo de alerta e a viatura a que este deve ficar associado.

Por outro lado, o ViewAlertComponent (a azul) apresenta a lista de alertas que foram configurados pelo utilizador no sistema. O utilizador também pode desativar ou eliminar um alerta que tenha configurado previamente, tal como é visível no fluxograma presente na Figura 47.

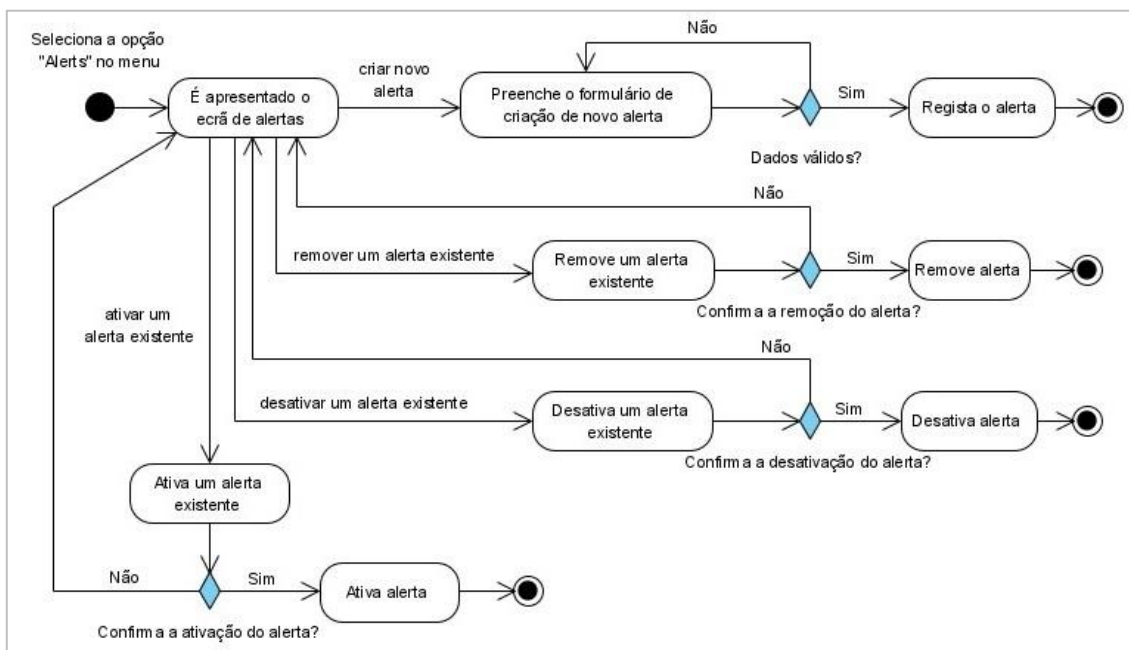


Figura 47 - Fluxograma para alertas no portal web

6.3 Implantação da plataforma

De acordo com o modelo de implantação apresentado no capítulo 5.3.2, os serviços desenvolvidos são autónomos e devem operar independentemente dos outros serviços e do sistema no qual estão alojados. Para tal, foi adotada a tecnologia do Docker, uma vez que esta plataforma de compartimentação permite o encapsulamento dos serviços desenvolvidos em *containers*.

Assim, um *container* é um componente que contém o código, as dependências da aplicação e as bibliotecas do sistema operativo necessárias para que este serviço seja executado com sucesso, independentemente da infraestrutura em que está hospedado (IBM, s.d.).

Deste modo, cada serviço desenvolvido tem associado um ficheiro *Dockerfile*. Este documento contém os comandos necessários para realizar a construção de uma imagem desse serviço. Assim, uma imagem em execução é denominada de *container* (Docker, s.d.).

```
FROM node:14.17.5      (1)
WORKDIR /src/app       (2)
COPY . /src/app        (3)
RUN npm install        (4)
EXPOSE 8080            (5)
```

Código 6 - *Dockerfile* desenvolvido para o portal *web*

No excerto de Código 6, é apresentado o *Dockerfile* utilizado para construir a imagem do portal *web*. Através da execução dos comandos presentes no ficheiro, a tecnologia cria uma imagem local a partir da imagem Docker `node:14.17.5` (1) disponibilizada *online*, configura o diretório a utilizar para `/src/app` (2), copia os ficheiros do diretório atual (raiz do projeto) para a localização `/src/app` (3), instala as dependências necessárias (4) e especifica que a porta 8080 deve ser utilizada pelo *container* para comunicação (5).

De modo a construir uma aplicação que é composta por múltiplos *containers* que estão alojados no mesmo servidor, foi criado um ficheiro YAML denominado de *docker-compose*. Este documento indica os serviços que devem ser incluídos na implantação da aplicação e quais os respetivos *containers* que devem ser executados. Assim, no Código 7 é apresentado um excerto do ficheiro *docker-compose* desenvolvido para a implantação da aplicação.

```
version: "2.0"
network:
  ccve_network:          (1)
services:
  ccve_web_portal:      (2)
    image: ccve_web_portal
    container_name: ccve_web_portal
    build: "web_portal/" (3)
    networks:
      - ccve_network    (4)
    restart: always
    ports:
      - 8080:8080      (5)
    command: npm run start
```

```

ccve_control_center:                               (6)
  image: ccve_control_center
  container_name: ccve_control_center
  build: "control_center/"
  networks:
    - ccve_network
  restart: always
  depends_on:
    - ccve_control_center_db                       (7)
  ports:
    - 4000:4000
  command: npm run start
ccve_control_center_db:                             (8)
  image: mongo
  container_name: ccve_control_center_db
  build: "control_center_db/"
  networks:
    - ccve_network
  restart: always
  ports:
    - 27017:27017
  command: npm run start

```

Código 7 - Excerto responsável por executar o portal *web*, o Centro de Controlo e a respetiva base de dados

De acordo com a informação presente no excerto Código 7, a seguinte tabela descreve os comandos realizados pela execução do ficheiro.

Tabela 15 – Legenda dos diferentes comandos do *docker-compose*

1	O campo <i>ccve_network</i> representa a rede criada para a comunicação interna entre os componentes da aplicação.
2	Configura um <i>container</i> para disponibilizar o portal <i>web</i> , através dos comandos presentes no serviço <i>ccve_portal_web</i> .
3	Indica a pasta que contém os ficheiros necessários para realizar a construção do <i>container</i> . Por defeito, é utilizado o <i>Dockerfile</i> previamente criado para este serviço.
4	O <i>container</i> é adicionado à rede criada para a plataforma. Apenas os componentes que estão presentes na mesma rede é que podem comunicar entre si.
5	Indica o mapeamento para a comunicação entre as portas da infraestrutura e as portas do <i>container</i> .
6	Configura um <i>container</i> para disponibilizar o Centro de Controlo, através dos comandos presentes no serviço <i>ccve_control_center</i> .
7	Indica que a construção deste <i>container</i> está dependente da construção prévia do <i>container</i> que disponibiliza a base de dados para este serviço.
8	Configura um <i>container</i> que disponibiliza a base de dados para o Centro de Controlo, através dos comandos presentes no serviço <i>ccve_control_center_db</i> .

Deste modo, através da execução do ficheiro *docker-compose*, a plataforma foi implantada num servidor Linux local, de modo que as funcionalidades disponibilizadas possam ser testadas.

7 Avaliação

Este capítulo pretende apresentar os mecanismos de avaliação utilizados para a avaliação da plataforma desenvolvida, de modo a determinar o grau de conclusão das funcionalidades propostas e garantir que estas correspondem às funcionalidades esperadas.

7.1 Quality Evaluation Framework (QEF)

De modo a assistir o processo de avaliação das funcionalidades da solução desenvolvida, foi aplicada a *Quality Evaluation Framework* (QEF). Assim, após o término da análise da solução, foi elaborado um documento que tem como objetivo avaliar, em qualquer fase de desenvolvimento do projeto, a qualidade dos requisitos funcionais e não funcionais identificados na análise do problema identificado. A medição de qualidade está linearmente relacionada com a percentagem de conclusão desses requisitos, o que permite determinar o grau de completude da totalidade da solução, com base em métricas de realização previamente estabelecidas.

Deste modo, os requisitos identificados são agrupados em fatores, que por sua vez são agrupados em dimensões. Foram consideradas três dimensões na elaboração do documento: Funcionalidade (*Functionality*), Adaptabilidade (*Adaptability*) e Usabilidade (*Usability*). Para cada uma das dimensões identificadas, foram identificados os respetivos fatores, presentes na Tabela 16.

Tabela 16 - Dimensões e fatores presentes no QEF

Dimensões (<i>Dimensions</i>)	Fator (<i>Factor</i>)
Funcionalidade (<i>Functionality</i>)	- Funcional (<i>Functional</i>) - Funcionalidades (<i>Features</i>) - Interação com utilizador (<i>User Interaction</i>) - Conta (<i>Account</i>)
Adaptabilidade (<i>Adaptability</i>)	- Funcionalidades (<i>Features</i>) - Manutenção (<i>Maintenance</i>) - Versatilidade (<i>Versatility</i>)
Usabilidade (<i>Usability</i>)	- Navegação do portal (<i>Portal Navigation</i>) - Navegação do mapa (<i>Map Navigation</i>) - Qualidade do conteúdo (<i>Content Quality</i>)

Assim, no Anexo 2 é apresentado o documento completo elaborado para auxiliar o processo de avaliação. De acordo com a informação presente no documento, foi possível determinar que o projeto tem um grau de completude geral de 94%, calculado através da avaliação dos requisitos definidos inicialmente para a implementação da solução. Nos seguintes subcapítulos são apresentados os principais requisitos e as respectivas métricas adotadas para avaliar o grau de conclusão de cada tarefa.

7.1.1 Dimensão Funcionalidade

Nesta dimensão estão agrupados os requisitos necessários para a comunicação entre os diferentes componentes da solução e as respectivas funcionalidades disponibilizadas pela plataforma.

7.1.1.1 Fator Funcional

Nas seguintes secções serão abordados os requisitos do fator Funcional, as suas percentagens de conclusão e as respectivas métricas adotadas para a avaliação de cada tarefa, e ainda uma explicação justificativa caso o requisito não tenha sido integrado a 100%. Optou-se pela utilização da sigla FF, indicativa da dimensão (*Functionality*) e fator (*Functional*) em que estes requisitos se enquadram.

FF01.

Tabela 17 - Requisito FF01

Requisito	Percentagem de conclusão (%)
FF01 – O servidor CCVE conecta-se com as APIs disponibilizadas pelas marcas automóveis que gerem as viaturas	50

Métrica: O servidor deve permitir a ligação entre os serviços e as APIs de veículos para coletar informação.

Percentagem de conclusão:

- 0% - O servidor não permite a comunicação entre os serviços internos e as APIs externas das viaturas;

- 50% - O servidor apenas permite algumas ligações entre os serviços internos e as APIs externas das viaturas;
- 100% - O servidor permite todas as ligações entre os serviços internos e as APIs externas das viaturas.

De acordo com a métrica estabelecida, a percentagem de conclusão deste requisito é de 50%, uma vez que foram desenvolvidos os serviços necessários para estabelecer comunicação com uma API baseada na da Tesla, mas não foi possível estabelecer comunicação com a API original da marca, não sendo possível desenvolver e testar a interação direta com veículos da Tesla.

FF02.

Tabela 18 - Requisito FF02

Requisito	Percentagem de conclusão (%)
FF02 – O portal <i>web</i> conecta-se ao servidor CCVE	100

Métrica: O portal *web* precisa de se conectar ao servidor CCVE de modo a estar disponível para utilização.

Percentagem de conclusão:

- 0% - O portal *web* não se consegue conectar ao servidor CCVE;
- 100% - O portal *web* consegue-se conectar ao servidor CCVE.

FF03.

Tabela 19 - Requisito FF03

Requisito	Percentagem de conclusão (%)
FF03 – Os serviços do CCVE são baseados numa arquitetura de microsserviços	100

Métrica: A solução segue uma arquitetura de microsserviços, com múltiplos serviços que comunicam entre si e com APIs externas.

Percentagem de conclusão:

- 0% - Os serviços implementados não seguem uma arquitetura de microsserviços;
- 100% - Os serviços implementados seguem uma arquitetura de microsserviços.

FF04.

Tabela 20 - Requisito FF04

Requisito	Percentagem de conclusão (%)
FF04 – O servidor CCVE guarda a informação dos veículos	100

Métrica: O servidor CCVE tem de processar e guardar a informação obtida dos veículos dos utilizadores.

Percentagem de conclusão:

- 0% - Os serviços de integração implementados não guardam as informações obtidas dos veículos;
- 50% - Apenas alguns serviços de integração implementados não guardam a informação obtida dos veículos;
- 100% - Todos os serviços de integração implementados guardam a informação obtida dos veículos.

7.1.1.2 Fator Funcionalidades

Nos seguintes segmentos do documento são apresentados os requisitos definidos no fator Funcionalidades, seguindo o mesmo formato que o Fator Funcional. Optou-se pela utilização da sigla FE, indicativa da dimensão (*Functionality*) e fator (*Features*) em que estes requisitos se enquadram. Uma vez que a sigla FF já está a referir um conjunto diferente de requisitos, selecionou-se o caractere “E” do fator.

FE01.

Tabela 21 - Requisito FE01

Requisito	Percentagem de conclusão (%)
FE01 – O portal <i>web</i> está disponível em vários <i>browsers</i> (Google Chrome, Mozilla Firefox, Microsoft Edge).	100

Métrica: O portal tem de estar acessível em múltiplos navegadores (Google Chrome, Mozilla Firefox, Microsoft Edge).

Percentagem de conclusão:

- 0% - O portal não está disponível em nenhuma das plataformas;
- 50% - O portal está apenas disponível no Google Chrome;
- 100% - O portal está disponível em todas as plataformas listadas.

FE02.

Tabela 22 - Requisito FE02

Requisito	Percentagem de conclusão (%)
FE02 – O portal <i>web</i> utiliza mapas para apresentar a localização geográfica da viatura	100

Métrica: O portal necessita de utilizar uma representação visual de um mapa para apresentar a localização geográfica da viatura.

Percentagem de conclusão:

- 0% - A localização geográfica dos veículos de um utilizador não é apresentada no formato de um mapa;
- 100% - Cada veículo apresenta um mapa onde o utilizador pode verificar a localização geográfica da viatura.

FE03.

Tabela 23 - Requisito FE03

Requisito	Percentagem de conclusão (%)
FE03 – O portal <i>web</i> apresenta o estado de comunicação da viatura	100

Métrica: O portal apresenta o estado de comunicação dos veículos com as APIs das marcas automóveis (verde para *online*; preto para *offline*).

Percentagem de conclusão:

- 0% - O indicador do estado de comunicação não está visível para os veículos;
- 50% - O indicador do estado de comunicação está implementado, mas não está disponível para todos os automóveis registados no portal;
- 100% - O portal apresenta o estado de comunicação de todos os veículos.

FE04.

Tabela 24 - Requisito FE04

Requisito	Percentagem de conclusão (%)
FE04 – O sistema utiliza mecanismos automáticos inteligentes de análise de dados para notificar o utilizador de problema relacionados com as suas viaturas	0

Métrica: O sistema apresenta mecanismos automáticos inteligentes que analisam os dados recolhidos das viaturas e notificam o utilizador de potenciais problemas.

Percentagem de conclusão:

- 0% - O sistema não contém mecanismos inteligentes automáticos que analisam os dados dos veículos;
- 50% - Apenas um mecanismo de análise inteligente de dados está implementado;
- 100% - Ambos os mecanismos estão implementados e o utilizador é notificado dos avisos.

De acordo com a métrica estabelecida, a percentagem de conclusão deste requisito do projeto é de 0%, uma vez que, até ao momento, os dados recolhidos das viaturas ainda estão a ser analisados, de modo que seja possível desenvolver algoritmos fiáveis que permitam estabelecer conclusões, com um nível de certeza aceitável, sobre possíveis problemas associados às viaturas.

FE05.

Tabela 25 - Requisito FE05

Requisito	Percentagem de conclusão (%)
FE05 – O sistema analisa os alertas configurados pelo utilizador e notifica o utilizador de problemas relacionados com as suas viaturas	100

Métrica: O sistema apresenta mecanismos automáticos que processam os alertas configurados pelos utilizadores.

Percentagem de conclusão:

- 0% - Não implementado;
- 50% - Os mecanismos estão implementados, mas o utilizador não é notificado dos possíveis problemas com os veículos;
- 100% - Os mecanismos estão implementados e o utilizador é notificado dos possíveis problemas com os veículos.

FE06.

Tabela 26 - Requisito FE06

Requisito	Percentagem de conclusão (%)
FE06 – O sistema processa o histórico do utilizador e apresenta estatísticas relacionadas com as suas viaturas	100

Métrica: O sistema é responsável por apresentar ao utilizador estatísticas relacionadas com as ações que este realizou com as suas viaturas registadas.

Percentagem de conclusão:

- 0% - Não implementado;

- 50% - O sistema apresenta as estatísticas para o utilizador, mas alguma informação pode estar em falta ou incorreta;
- 100% - O sistema apresenta as estatísticas relacionadas com as atividades do utilizador com as suas viaturas.

FE07.

Tabela 27 - Requisito FE07

Requisito	Percentagem de conclusão (%)
FE07 – O sistema processa o histórico do utilizador e apresenta estatísticas relacionadas com uma viatura específica	100

Métrica: O sistema é responsável por apresentar ao utilizador estatísticas relacionadas com as sessões de carregamento de uma viatura específica. Esta informação deve ser apresentada em gráficos.

Percentagem de conclusão:

- 0% - Estatísticas não implementadas;
- 50% - O sistema apresenta as estatísticas para uma viatura específica num gráfico, mas alguma informação está em falta ou incorreta;
- 100% - O sistema apresenta as estatísticas para uma viatura específica num gráfico.

FE08.

Tabela 28 - Requisito FE08

Requisito	Percentagem de conclusão (%)
FE08 – O <i>design</i> do portal <i>web</i> necessita de seguir as <i>guidelines</i> de outros produtos da EVIO	100

Métrica: O *design* do portal *web* está em concordância com o *design* de outros produtos da EVIO.

Percentagem de conclusão:

- 0% - A linguagem de *design* da EVIO não foi seguida no desenvolvimento do *design*;

- 100% - A linguagem de *design* da EVIO foi seguida no desenvolvimento do *design*.

7.1.1.3 Fator Interação com utilizador

Nas seguintes secções são apresentados os requisitos relacionados com a interação do utilizador, seguindo o mesmo formato utilizado para os fatores acima apresentados. Optou-se pela utilização da sigla FUI, indicativa da dimensão (*Functionality*) e fator (*User Interaction*) em que estes requisitos se enquadram.

FUI01.

Tabela 29 - Requisito FUI01

Requisito	Percentagem de conclusão (%)
FUI01 – A língua utilizada no portal <i>web</i> é inglês	100

Métrica: A língua do portal *web* por defeito é o inglês.

Percentagem de conclusão:

- 0% - Os componentes e mensagens apresentadas ao utilizador não estão em inglês;
- 50% - Apenas alguns componentes e mensagens apresentadas ao utilizador estão em inglês;
- 100% - Todos os componentes e mensagens apresentadas ao utilizador estão em inglês.

FUI02.

Tabela 30 - Requisito FUI02

Requisito	Percentagem de conclusão (%)
FUI02 – O utilizador pode criar uma conta com um plano de pagamento associado	50

Métrica: O utilizador pode criar uma conta na aplicação, com um plano de pagamento válido associado.

Percentagem de conclusão:

- 0% - O utilizador não consegue criar uma conta.
- 50% - O utilizador consegue criar uma conta, mas não consegue associar um plano de pagamento válido, à exceção do plano gratuito.
- 100% - O utilizador consegue criar uma conta, e escolher um plano de pagamento adequado às suas necessidades.

De acordo com a métrica de validação apresentada, a percentagem de conclusão deste requisito do projeto é de 50%, uma vez que um utilizador consegue criar uma conta válida com os dados inseridos na plataforma, mas não consegue associar um plano de pagamento a essa conta. A implementação da funcionalidade de associação de um plano de pagamento a um perfil de utilizador não foi implementada, uma vez que o processo de seleção de um parceiro que tenha a responsabilidade de fornecedor para realizar a faturação, e a respetiva de emissão de faturas, ainda está em progresso. Assim, para cada conta criada atualmente, é associado o plano de pagamento grátis.

FUI03.

Tabela 31 - Requisito FUI03

Requisito	Percentagem de conclusão (%)
FUI03 – O utilizador pode registar um novo veículo	100

Métrica: O utilizador pode registar um novo veículo na plataforma (marca, modelo, versão, matrícula e número de série).

Percentagem de conclusão:

- 0% - O utilizador não consegue registar uma nova viatura;
- 50% - O utilizador consegue registar uma viatura, mas alguma informação do veículo não fica guardada;
- 100% - O utilizador consegue registar um veículo válido com sucesso.

FUI04.

Tabela 32 - Requisito FUI04

Requisito	Percentagem de conclusão (%)
FUI04 – O utilizador pode eliminar um veículo	100

Métrica: O utilizador consegue eliminar um veículo que tenha registado previamente.

Percentagem de conclusão:

- 0% - O utilizador não consegue eliminar um veículo;
- 100% - O utilizador consegue eliminar um veículo.

FUI05.

Tabela 33 - Requisito FUI05

Requisito	Percentagem de conclusão (%)
FUI05 – O utilizador consegue monitorizar a localização de um veículo	100

Métrica: O portal permite que um utilizador consiga monitorizar a localização geográfica das suas viaturas.

Percentagem de conclusão:

- 0% - O portal não apresenta a localização das viaturas;
- 100% - O utilizador consegue monitorizar a localização das suas viaturas no portal.

FUI06.

Tabela 34 - Requisito FUI06

Requisito	Percentagem de conclusão (%)
FUI06 – O utilizador consegue monitorizar o estado de carregamento de um veículo	100

Métrica: O portal permite que um utilizador monitorize o estado de carregamento das suas viaturas.

Percentagem de conclusão:

- 0% - O portal não apresenta o estado de carregamento das viaturas;
- 50% - O portal apenas apresenta o estado de carregamento para algumas das viaturas;
- 100% - O portal apresenta o estado de carregamento para todos os veículos registados pelo utilizador.

FUI07.

Tabela 35 - Requisito FUI07

Requisito	Percentagem de conclusão (%)
FUI07 – O utilizador consegue remotamente executar um comando de bloqueio/desbloqueio de viatura	100

Métrica: O portal permite a um utilizador executar comandos de bloqueio e desbloqueio para uma viatura selecionada.

Percentagem de conclusão:

- 0% - O portal não apresenta os comandos disponíveis para o veículo;
- 25% - O portal apresenta os comandos disponíveis para o veículo, mas não permite a execução destes;
- 50% - O utilizador apenas consegue executar com sucesso um dos comandos disponíveis;
- 75% - O utilizador consegue executar os comandos, mas são detetadas falhas;
- 100% - O portal permite a execução de todos os comandos para uma viatura com sucesso.

FUI08.

Tabela 36 - Requisito FUI08

Requisito	Percentagem de conclusão (%)
FUI08 – O utilizador consegue remotamente executar um comando de ativação/desativação da opção de ar condicionado da viatura	100

Métrica: O portal permite a um utilizador executar comandos de ativação e desativação da opção de ar condicionado para uma viatura selecionada.

Percentagem de conclusão:

- 0% - O portal não apresenta os comandos disponíveis para o veículo;
- 25% - O portal apresenta os comandos disponíveis para o veículo, mas não permite a execução destes;
- 50% - O utilizador apenas consegue executar com sucesso um dos comandos disponíveis;
- 75% - O utilizador consegue executar os comandos, mas são detetadas falhas;
- 100% - O portal permite a execução de todos os comandos para uma viatura com sucesso.

FUI09.

Tabela 37 - Requisito FUI09

Requisito	Percentagem de conclusão (%)
FUI09 – O utilizador consegue configurar alertas para notificações	100

Métrica: O portal permite a um utilizador configurar alertas de notificações baseadas nos dados recolhidos dos veículos do utilizador.

Percentagem de conclusão:

- 0% - O utilizador não consegue configurar alertas.
- 50% - O utilizador consegue configurar alertas, mas são detetadas falhas no registo destes.

- 100% - O portal permite a configuração de alertas com sucesso.

7.1.1.4 Fator Conta

Nas próximas secções são apresentados os requisitos associados à gestão de conta e perfil do utilizador, seguindo um formato semelhante àquele utilizado para apresentar os outros fatores. Optou-se pela utilização da sigla FA, indicativa da dimensão (*Functionality*) e fator (*Account*) em que estes requisitos se enquadram.

FA01.

Tabela 38 - Requisito FA01

Requisito	Percentagem de conclusão (%)
FA01 – O utilizador tem acesso ao seu perfil	100

Métrica: Um utilizador tem acesso ao perfil da sua conta.

Percentagem de conclusão:

- 0% - O utilizador não consegue aceder ao perfil da sua conta;
- 50% - O utilizador consegue aceder ao perfil da sua conta, mas são detetadas falhas na informação apresentada;
- 100% - O utilizador não consegue aceder ao perfil da sua conta com sucesso.

FA02.

Tabela 39 - Requisito FA02

Requisito	Percentagem de conclusão (%)
FA02 – O utilizador consegue alterar a informação apresentada no seu perfil	100

Métrica: O utilizador consegue alterar com sucesso a informação apresentada no seu perfil.

Percentagem de conclusão:

- 0% - Os dados do utilizador não são atualizados com sucesso;

- 50% - Os dados do utilizador são atualizados, mas existem falhas no processo de atualização de dados;
- 100% - Os dados do utilizador são atualizados com sucesso.

FA03.

Tabela 40 - Requisito FA03

Requisito	Percentagem de conclusão (%)
FA03 – O utilizador consegue alterar o plano de pagamento associado à sua conta	0

Métrica: O utilizador consegue alterar com sucesso a informação apresentada no seu perfil.

Percentagem de conclusão:

- 0% - O utilizador não consegue atualizar o plano de pagamento atual;
- 50% - O utilizador consegue alterar o plano de pagamento atual, mas com falhas;
- 75% - O plano de pagamento associado ao utilizador é alterado, mas o sistema não realiza a necessária atualização do balanço monetário;
- 100% - A nova informação é atualizada e apresentada com sucesso.

De acordo com a métrica de validação apresentada, a percentagem de conclusão deste requisito é de 0%, uma vez que a implementação desta funcionalidade não foi realizada. Tal como referido no FUI02, todas as contas atualmente existentes têm associado um plano de pagamento grátis, sendo que o desenvolvimento desta funcionalidade foi adiado para ser realizado em concordância com a funcionalidade de associação de um plano de pagamento à conta de um utilizador, aquando da sua criação.

7.1.2 Dimensão Adaptabilidade

Nesta dimensão estão agrupados os requisitos responsáveis por assegurar a escalabilidade da infraestrutura da solução e a flexibilidade dos componentes que a compõem.

7.1.2.1 Fator Funcionalidades

No fator Funcionalidades é apresentado o requisito relacionado com a apresentação de mensagens geradas pela plataforma ao utilizador. Optou-se pela utilização da sigla AF, indicativa da dimensão (*Adaptability*) e fator (*Features*) em que estes requisitos se enquadram.

AF01.

Tabela 41 - Requisito AF01

Requisito	Percentagem de conclusão (%)
AF01 – O sistema apresenta ao utilizador mensagens uteis após a execução de ações	100

Métrica: O portal *web* apresenta o utilizador com mensagens úteis quando este interage com o sistema.

Percentagem de conclusão:

- 0% - O sistema não apresenta ao utilizador uma mensagem após a execução de uma ação;
- 100% - O sistema apresenta ao utilizador uma mensagem após a execução de uma ação.

7.1.2.2 Fator Manutenção

Neste fator encontram-se agrupados os requisitos que estão relacionados com a manutenção da plataforma a longo prazo. Optou-se pela utilização da sigla AM, indicativa da dimensão (*Adaptability*) e fator (*Maintenance*) em que estes requisitos se enquadram.

AM01.

Tabela 42 - Requisito AM01

Requisito	Percentagem de conclusão (%)
AM01 – O servidor CCVE apresenta um conjunto de <i>interfaces</i> e <i>endpoints</i> , permitindo a possibilidade de serem facilmente adicionados novos serviços	100

Métrica: A aplicação é robusta e facilmente suporta a integração de novos serviços.

Percentagem de conclusão:

- 0% - Os serviços não são desenvolvidos com *endpoints* que permitem a fácil comunicação com outros serviços da plataforma;
- 100% - Os serviços são desenvolvidos com *endpoints* que permitem a fácil comunicação com outros serviços da plataforma.

AM02.

Tabela 43 - Requisito AM02

Requisito	Percentagem de conclusão (%)
AM02 – O número de instâncias de um serviço operacional específico precisa de ser adaptado de acordo com o número de pedidos (<i>Auto Scaling</i>)	50

Métrica: O número de serviços operacionais precisa de se adaptar de acordo com o número de pedidos a processar, de modo a manter um nível de performance elevado.

Percentagem de conclusão:

- 0% - Não implementado;
- 50% - O número de instâncias de um serviço operacional está limitado a um número específico, independentemente do número de pedidos ou utilizadores;
- 100% - O sistema é completamente escalável.

De acordo com a métrica de validação apresentada, a percentagem de conclusão deste requisito é de 50%, uma vez que a implementação desta funcionalidade ficou incompleta. Deste modo, é possível definir um número de instâncias em execução de um serviço e a infraestrutura atual acompanha essa alteração, mas a alteração do número de instâncias em execução não está automatizada para aumentar ou diminuir de acordo com o número de pedidos a serem processados.

7.1.2.3 Fator Versatilidade

Neste fator são agrupados os requisitos que estão relacionados com a adaptabilidade da solução de acordo com o seu utilizador e os dispositivos que este utiliza para aceder à plataforma. Optou-se pela utilização da sigla AV, indicativa da dimensão (*Adaptability*) e fator (*Versatility*) em que estes requisitos se enquadram.

AV01.

Tabela 44 - Requisito AV01

Requisito	Percentagem de conclusão (%)
AV01 – A informação apresentada no portal <i>web</i> é exclusiva ao utilizador	100

Métrica: A informação apresentada no portal *web* está relacionada com o utilizador autenticado no sistema.

Percentagem de conclusão:

- 0% - A informação apresentada não está relacionada com o utilizador;
- 100% - A informação apresentada está apenas relacionada com o utilizador.

AV02.

Tabela 45 - Requisito AV02

Requisito	Percentagem de conclusão (%)
AV02 – O portal <i>web</i> ajusta-se à resolução do ecrã	100

Métrica: O portal *web* ajusta o seu *layout* à resolução do ecrã.

Percentagem de conclusão:

- 0% - Não implementado;
- 50% - Apenas alguns dos componentes ajustam-se à resolução do ecrã;
- 100% - Todos os componentes ajustam-se à resolução do ecrã.

De modo a validar este requisito, foram testadas as resoluções de ecrã 1920x1080, 1366x768 e 1536x864. Foram selecionadas estas, uma vez que estas são as três resoluções mais utilizadas mundialmente para setembro de 2021 (Desktop Screen Resolution Stats Worldwide | Statcounter Global Stats, 2021). Durante os testes, verificou-se que os componentes do portal CCVE se adaptaram às resoluções acima definidas.

AV03.

Tabela 46 - Requisito AV02

Requisito	Percentagem de conclusão (%)
AV03 – O portal <i>web</i> suporta uma versão móvel	50

Métrica: O portal *web* suporta uma versão para utilizadores com dispositivos móveis.

Percentagem de conclusão:

- 0% - Não implementado;
- 50% - Apenas alguns dos componentes ajustam-se à versão móvel;
- 100% - Todos os componentes ajustam-se à versão móvel.

De acordo com a métrica estabelecida, a percentagem de conclusão deste requisito é de 50%, uma vez que alguns componentes do portal ainda não estão adaptados para uma versão móvel, exibindo alguns problemas na formatação dos gráficos e apresentando alguns componentes incorretamente sobrepostos em cima de outros.

7.1.3 Dimensão Usabilidade

Nesta dimensão estão agrupados os requisitos responsáveis por avaliar a facilidade com que o utilizador interage com as funcionalidades disponibilizadas no portal.

7.1.3.1 Fator Navegação do Portal

No fator Navegação do Portal são listados os requisitos relacionados com a interação entre o utilizador e as funcionalidades disponibilizadas pelo portal *web*. Optou-se pela utilização da

sigla UPN, indicativa da dimensão (*Usability*) e fator (*Portal Navigation*) em que estes requisitos se enquadram.

UPN01.

Tabela 47 - Requisito UPN01

Requisito	Percentagem de conclusão (%)
UPN01 – A interface do portal <i>web</i> é simples e rápida, sem abrandar quando o número de utilizadores é alto	100

Métrica: O portal *web* deve ser rápido, de simples utilização e alternar rapidamente entre as funcionalidades, independentemente do número de utilizadores.

Percentagem de conclusão:

- 0% - Um pedido demora mais do que 5 segundos a processar;
- 50% - Um pedido demora entre 1 a 3 segundos para ser processado;
- 100% - O pedido demora menos de 1 segundo a ser processado.

De modo a avaliar o grau de conclusão deste requisito foram realizados 20 pedidos ao servidor, ao longo de quatro períodos definidos antecipadamente. O pedido realizado é efetuado após a autenticação do utilizador na aplicação e solicita que sejam retornados os veículos registados pelo utilizador para serem apresentados no portal.

Os períodos foram escolhidos com o objetivo de realizar os testes em alturas de maior utilização da plataforma pelos clientes. Deste modo, foi definido um período de testes no início de manhã (08:00), no início da hora de almoço (13:00h), na conclusão da hora de almoço (14:00h) e na conclusão do horário de trabalho (17:00h).

Tabela 48 - Tempo de execução dos pedidos realizados para obter os dados de veículo do utilizador

08:00h	13:00h	14:00h	17:00h
76ms	86ms	81ms	87ms
72ms	85ms	84ms	90ms
84ms	81ms	106ms	79ms
85ms	100ms	100ms	87ms
83ms	90ms	83ms	88ms

Assim, através do cálculo da média para os valores obtidos, foi possível determinar que o tempo médio de resposta ao pedido analisado foi de 86 ms. Deste modo, foi possível determinar que a percentagem de conclusão para este requisito foi de 100%.

UPN02.

Tabela 49 - Requisito UPN02

Requisito	Percentagem de conclusão (%)
UPN02 – Os serviços em execução não têm erros, e que erros inesperados devem ser tratados.	100

Métrica: Erros inesperados têm de ser apresentados ao utilizador com uma mensagem clara sobre o erro obtido.

Percentagem de conclusão:

- 0% - Nenhuma operação disponibiliza uma mensagem ao utilizador quando um erro é detetado;
- 50% - Apenas algumas operações disponibilizam uma mensagem ao utilizador quando um erro é detetado;
- 100% - Todas as operações disponibilizam uma mensagem ao utilizador quando um erro é detetado.

7.1.3.2 Fator Navegação do Mapa

No fator Navegação do Mapa são agrupados os requisitos relacionados com a interação do utilizador com as funcionalidades relacionadas. Optou-se pela utilização da sigla UMN, indicativa da dimensão (*Usability*) e fator (*Map Navigation*) em que estes requisitos se enquadram.

UMN01.

Tabela 50 - Requisito UMN01

Requisito	Percentagem de conclusão (%)
UMN01 – Fácil acesso a mapas em tempo real (Mapbox)	100

Métrica: As funcionalidades dos mapas estão facilmente disponíveis no CCVE e a localização atual do veículo é ilustrada no mapa.

Percentagem de conclusão:

- 0% - O mapa não é carregado;
- 50% - O mapa é carregado, mas a localização da viatura está em falta;
- 100% - O mapa é carregado e ilustra a localização da viatura.

UMN02.

Tabela 51 - Requisito UMN02

Requisito	Percentagem de conclusão (%)
UMN02 – A localização do avatar da viatura é fácil de entender	100

Métrica: A localização de um veículo no mapa é facilmente identificável.

Percentagem de conclusão:

- 0% - A localização da viatura não está clara no mapa;
- 100% - A localização da viatura no mapa está representada com um avatar facilmente identificável.

7.1.3.3 Fator Qualidade do conteúdo

No fator Qualidade do conteúdo são apresentados os requisitos associados ao conteúdo que é exposto no portal e os requisitos das mensagens que são exibidas ao utilizador. Optou-se pela utilização da sigla UCQ, indicativa da dimensão (*Usability*) e fator (*Content Quality*) em que estes requisitos se enquadram.

De modo a medir a qualidade deste fator, foi elaborado um questionário que tem como objetivo avaliar as mensagens que são apresentadas pelo portal e se o conteúdo destas é claro e/ou se é ofensivo. Assim, este questionário foi respondido, de forma anónima, por elementos da equipa da EVIO, tendo sido obtido um total de seis respostas. Esta equipa é responsável pelo desenvolvimento e manutenção dos produtos da EVIO, sendo composta por desenvolvedores de aplicações android, iOS, *web* e dos serviços que compõem a estrutura utilizada por estas para disponibilizarem as funcionalidades aos clientes. O tamanho reduzido da amostra deve-se ao facto de o produto se encontrar ainda em fase de desenvolvimento, não podendo este ser divulgado fora do contexto da EVIO. No Anexo 3, é apresentado na totalidade o questionário realizado.

Assim, ao longo dos próximos requisitos vão ser apresentados os resultados obtidos.

UCQ01.

Tabela 52 - Requisito UCQ01

Requisito	Percentagem de conclusão (%)
UCQ01 – Todas as mensagens são de fácil entendimento	100

Métrica: Todas as mensagens devem ter o espaçamento e tamanho apropriado.

Percentagem de conclusão:

- 0% - 0%-20% como resultado obtido às respostas do questionário;
- 25% - 20%-40% como resultado obtido às respostas do questionário;
- 50% - 40%-60% como resultado obtido às respostas do questionário;
- 75% - 60%-80% como resultado obtido às respostas do questionário;
- 100% - 80%-100% como resultado obtido às respostas do questionário.

De modo a avaliar este requisito, foi pedido no questionário para indicar, através de uma escala de Likert, numerada de 1 a 5, a clareza das mensagens que são apresentadas ao utilizador durante a utilização do portal. Na Tabela 53, é apresentada a escalada definida.

Tabela 53 - Escala de avaliação utilizada no questionário

Escala	Descrição
1	Do not understand (Não percebi)
2	Somewhat understandable (Pouco compreensível)
3	Neutral (Neutro)
4	Understand well (Percebi bem)
5	Easy to understand (Fácil de perceber)

Na Tabela 54, estão presentes os resultados obtidos para cada mensagem.

Tabela 54 – Resultados do questionário para o requisito UCQ01

Mensagem	Média das respostas obtidas (escala 1 a 5)
Login Failed	4.17
Login Successful	5
Vehicle created	5
Vehicle creation failed	5
Door unlocked	4.83
Door locked	4.83
Door already locked	4.5
Door already unlocked	4.5
Air conditioning on	4.83
Air conditioning off	4.83
Air conditioning already on	4.5
Air conditioning already off	4.5
Alert created	4.67
Alert creation failed	4.67

Em seguida, de modo a definir o indicador para avaliar a qualidade das mensagens apresentadas pelo portal, foi calculada a média total e a respetiva percentagem. Deste modo, a média de todas as respostas obtidas é de 4.7, o que, dividindo por 5 (valor máximo da escala), corresponde a 94%.

Desta forma, foi possível determinar que a percentagem de conclusão para este requisito é de 100%. No entanto, de acordo com os resultados obtidos na Tabela 54, pode ser necessário rever a mensagem “Login Failed”, uma vez que esta pode não estar clara para alguns utilizadores.

UCQ02.

Tabela 55 - Requisito UCQ02

Requisito	Percentagem de conclusão (%)
UCQ02 – Os textos são simples e concisos	100

Métrica: O utilizador deve conseguir ler e entender o texto presente no portal *web*.

Percentagem de conclusão:

- 0% - 0%-20% como resultado obtido às respostas do questionário;
- 25% - 20%-40% como resultado obtido às respostas do questionário;
- 50% - 40%-60% como resultado obtido às respostas do questionário;
- 75% - 60%-80% como resultado obtido às respostas do questionário;
- 100% - 80%-100% como resultado obtido às respostas do questionário.

De modo a avaliar este requisito, foi pedido no questionário para indicar, através da utilização da escala mencionada na Tabela 53, a facilidade com que as mensagens apresentadas pelo CCVE são interpretadas. Na Tabela 56, são apresentados os resultados obtidos para esta questão.

Tabela 56 - Resultados do questionário para o requisito UCQ02

Escala	Quantidade de respostas
5	4
4	2
3	0
2	0
1	0

Deste modo, e de acordo com a Tabela 56, a média dos resultados obtidos é de 4.67, o que, dividindo por 5 (valor máximo da escala), corresponde a 93.4%. Deste modo, de acordo com os valores definidos para a percentagem de conclusão, este requisito foi concluído a 100%.

UCQ03.

Tabela 57 - Requisito UCQ03

Requisito	Percentagem de conclusão (%)
UCQ03 – Nenhuma mensagem deve ter conteúdo ofensivo	100

Métrica: Nenhuma mensagem tem conteúdo ofensivo.

Percentagem de conclusão:

- 0% - Algumas mensagens apresentam conteúdo ofensivo;
- 100% - Nenhuma mensagem apresenta conteúdo ofensivo.

De modo a avaliar este requisito, foi pedido no questionário para indicar se nas mensagens apresentadas pelo CCVE está presente conteúdo ofensivo. Na Tabela 58, estão presentes os resultados obtidos para esta questão.

Tabela 58 - Resultados do questionário para o requisito UCQ03

Opções	Quantidade de respostas
Sim. Conteúdo ofensivo é apresentado nas mensagens.	0
Não. As mensagens apresentadas não têm conteúdo ofensivo.	6

Assim, de acordo com a Tabela 58, a totalidade das respostas concorda que as mensagens apresentadas não contêm conteúdo ofensivo. Deste modo, e de acordo com as percentagens de conclusão definidas para a métrica estabelecida, este requisito foi concluído a 100%.

7.1.3.4 Fator Integridade

No fator Integridade são apresentados os requisitos associados à integridade e consistência dos dados que são tratados e guardados na plataforma. Optou-se pela utilização da sigla UIN, indicativa da dimensão (*Usability*) e fator (*Integrity*) em que estes requisitos se enquadram.

UIN01.

Tabela 59 - Requisito UIN01

Requisito	Percentagem de conclusão (%)
UIN01 – Consistência dos dados processados pelo servidor e pelo portal	100

Métrica: Os dados devem manter a sua consistência no servidor e no portal quando são apresentados.

Percentagem de conclusão:

- 0% - A consistência dos dados não é mantida;
- 100% - A consistência dos dados é mantida.

UIN02.

Tabela 60 - Requisito UIN02

Requisito	Percentagem de conclusão (%)
UIN02 – Sincronização da informação dos veículos do utilizador	100

Métrica: A informação do veículo no servidor deve estar sincronizada com a informação atual da viatura, sendo que os dados deverão ter um prazo máximo de validade de 5 minutos.

Percentagem de conclusão:

- 0% - Os dados não estão sincronizados;
- 100% - Os dados estão sincronizados.

UIN03.

Tabela 61 - Requisito UIN03

Requisito	Percentagem de conclusão (%)
UIN03 – A informação do utilizador é protegida de acessos não autorizados	100

Métrica: A informação fornecida pelo utilizador é protegida de acessos não autorizados a não permitem a identificação deste, de acordo com as regras RGPD para a proteção de informação pessoal.

Percentagem de conclusão:

- 0% - O acesso à informação do utilizador não é verificado.
- 50% - O acesso à informação do utilizador é verificado, mas esta permite a identificação do utilizador.
- 100% - O acesso à informação do utilizador é verificado e não pode ser utilizada para identificar o utilizador.

Em conclusão, foi assim possível determinar que foram concluídos com sucesso 94% dos requisitos inicialmente definidos para a implementação da solução, e que a maioria dos requisitos desenvolvidos estão de acordo com os requisitos identificados durante a análise do problema.

8 Conclusão

Este capítulo visa apresentar as conclusões finais do projeto, refletindo sobre a percentagem de realização dos objetivos propostos inicialmente, as limitações associadas ao desenvolvimento do projeto e as alterações que resultaram destas, o contributo desta ferramenta para a EVIO e propostas para trabalho futuro.

8.1 Objetivos realizados

Na Tabela 62, encontram-se apresentados os objetivos da plataforma desenvolvida, juntamente com a respetiva percentagem de realização para os objetivos estabelecidos no capítulo 1.4.

Tabela 62 - Objetivos propostos e a respetiva percentagem de realização

Objetivos principais	Percentagem de realização (%)
Gestão de múltiplas viaturas, independentemente da marca automóvel	50%
Monitorização do estado de carregamento da viatura	100%
Monitorização da localização da viatura	100%
Acesso remoto aos dados das viaturas	100%
Acesso a dados estatísticos baseados no histórico de utilização das viaturas registadas na plataforma	100%
Interação com as viaturas através do envio de comandos remotos	100%
Existência de mecanismos, automáticos ou configuráveis pelo utilizador, para auxiliar na monitorização das viaturas	50%
Objetivos secundários	
Permitir a fácil integração com <i>software</i> já existente da EVIO	100%
Facilitar a identificação de potenciais problemas relacionados com a viatura	25%
Eliminar a necessidade de utilizar diferentes aplicações para gerir diversas marcas de veículos	50%

Assim, dos objetivos principais inicialmente definidos, dois não foram realizados na sua íntegra devido a restrições temporais, uma vez que foi necessário alocar tempo para o desenvolvimento do serviço de simulação da Tesla, o que não tinha sido inicialmente previsto.

No caso do primeiro objetivo apresentado, não foi possível realizar a integração com serviços disponibilizados por marcas automóveis, de modo que não foi possível validar, na totalidade, a monitorização de diversas viaturas, independentemente da marca, ou testar as funcionalidades desenvolvidas num ambiente real, com um veículo verdadeiro. Outro objetivo que também não pode ser completado está relacionado com a existência de mecanismos que auxiliem na monitorização das viaturas registadas na plataforma, uma vez que não foi possível concluir o desenvolvimento de algoritmos automáticos que analisem os dados recolhidos e construam indicadores utilizados para despoletar alertas para os utilizadores.

Em relação aos objetivos secundários propostos para o projeto, dois deles não foram completamente realizados. A identificação de potenciais problemas relacionados com as viaturas registadas na plataforma ficou incompleta, não tendo sido possível, durante o tempo para a implementação do projeto, desenvolver os mecanismos automáticos de deteção de erros com base no processamento dos dados recolhidos das viaturas, sendo apenas possível a configuração manual de alertas pelo utilizador para uma viatura. O outro objetivo secundário cujo desenvolvimento não foi possível validar na totalidade está relacionado com a eliminação da necessidade de utilização de outras aplicações para a gestão remota das viaturas dos utilizadores, uma vez que, como não foi possível integrar com o *software* de múltiplas marcas automóveis diferentes, não é possível afirmar, neste momento, que a plataforma permite a total substituição de outras soluções. No entanto, este objetivo encontra-se a 50% de realização, uma vez que a infraestrutura desenvolvida permite a integração de múltiplos *softwares* e é apenas necessário o desenvolvimento de novas integrações para se atingir o objetivo.

Em relação aos objetivos realizados na totalidade, o acesso aos dados da viatura é assegurado pelo desenvolvimento da rede de microsserviços que permite a comunicação com o *software* disponibilizado por marcas automóveis e, caso essa comunicação seja estabelecida com sucesso, permite a recolha da informação necessária da viatura. Deste modo, foi possível garantir que os dados recolhidos são apresentados ao utilizador através do portal *web* desenvolvido, permitindo que seja possível monitorizar o estado de carregamento e a localização das viaturas e, para veículos que suportem essa função, a execução de comandos remotamente. No portal é também possível aceder aos dados estatísticos da atividade do utilizador e das suas viaturas, elaborado através da análise do histórico das ações que este realiza na plataforma ou que são comunicadas pelos veículos. Por último, o desenvolvimento da plataforma foi realizado de

modo a criar e expor interfaces que, quando devidamente autenticadas, podem ser acedidas pelas restantes aplicações da EVIO.

8.2 Limitações do projeto

Devido ao grau de dificuldade associado ao projeto e da dependência de *software* disponibilizado por diferentes marcas automóveis para integração com as viaturas, foram encontradas algumas limitações ao longo do período de desenvolvimento da solução. A principal limitação foi a pandemia Covid-19 e os consecutivos períodos de confinamento que foram decretados pelo governo português para limitar a propagação do vírus, visto ter afetado a comunicação com as diversas marcas automóveis contactadas pela EVIO. Deste modo, e devido à instabilidade e incerteza causadas pela situação pandémica no mercado empresarial, as comunicações não foram bem-sucedidas e, portanto, não foi possível integrar com um sistema verdadeiro de uma marca automóvel, o que permitiria validar a solução num ambiente com dados de viaturas reais. Outro fator que também afetou o desenvolvimento do projeto foi o tempo, uma vez que foram estabelecidas datas-limite para entregas periódicas na fase de análise e para os desenvolvimentos efetuados. Ao atingir as datas-limite estabelecidas, foi necessário avançar no desenvolvimento do projeto, mesmo que funcionalidades ainda não tivessem sido implementadas na sua totalidade.

8.3 Contributos do projeto

Tal como foi referido ao longo do documento, apesar de não ter sido possível concretizar a integração com o *software* original disponibilizado por uma marca automóvel para a comunicação com as viaturas, foi possível apresentar com sucesso uma plataforma, ainda que numa fase inicial de desenvolvimento, que disponibiliza funcionalidades para gestão e monitorização de viaturas. Deste modo, este projeto contribuí para expandir o conjunto de produtos em desenvolvimento da EVIO e, através do acesso às viaturas, fornece uma plataforma que acrescenta valor à oferta que a EVIO já disponibiliza aos seus clientes. Da mesma forma, existe também a possibilidade de auxiliar as negociações com potenciais clientes (*leads*) que demonstrem interesse em integrar uma solução semelhante, sendo possível apresentar este projeto como uma prova de conceito.

8.4 Trabalho futuro

Devido ao facto de ainda existirem requisitos que não foram parcialmente ou integralmente implementados na solução, estes devem ser o foco principal no desenvolvimento posterior deste projeto. Como tal, as tarefas a realizar a curto prazo serão:

- O desenvolvimento de serviços que integram com as APIs de marcas automóveis, que são responsáveis por garantir a comunicação com as viaturas, de modo a ser possível testar as funcionalidades desenvolvidas com dados e viaturas reais;
- O desenvolvimento dos mecanismos automáticos de deteção de potenciais problemas relacionados com as viaturas que o utilizador registou na plataforma. Será necessário que os indicadores recolhidos para estes mecanismos inteligentes sejam validados e que os resultados obtidos sejam analisados durante um período de tempo considerável, de modo a evitar falsos relatórios de potenciais falhas, acabando por confundir o utilizador com informação que não é correta;
- O desenvolvimento da funcionalidade de associação de um plano de pagamento à conta de um utilizador, de modo a cobrar o utilizador pelos serviços fornecidos;
- O desenvolvimento de um centro de notificações, de forma a manter um histórico permanente das notificações que são despoletadas por mecanismos automáticos de deteção de problemas ou pela ativação de alertas que são configurados manualmente para cada viatura.

Durante o desenvolvimento da solução proposta originalmente, também foram identificados potenciais requisitos para serem desenvolvidos a longo prazo, tais como:

- A continuação da procura de novos serviços de veículos para integração, de modo a aumentar a oferta do tipo de viaturas que são suportadas na plataforma;
- Desenvolvimento de uma página informativa no portal *web* que contém a lista de viaturas que são suportadas atualmente pela plataforma;
- A análise dos dados provenientes das viaturas, de modo a desenvolver mais mecanismos inteligentes de deteção de falhas. Nesta fase, será importante realizar um questionário direcionado a utilizadores de veículos elétricos, de modo a determinar as informações das viaturas sobre as quais estes gostariam de ser alertados, identificando assim as necessidades destes;

- A análise da necessidade do desenvolvimento de uma aplicação movel para ser utilizada em paralelo com o portal *web*, sendo que esta disponibilizará apenas as funcionalidades de interação com as viaturas, de modo a ter uma plataforma mais simplificada e versátil, que permite comunicar com diversas viaturas, de diversas marcas, numa só aplicação.

9 Referências

- Angular. (s.d.). *Introduction to Angular concepts*. Obtido em 24 de janeiro de 2021, de Angular - Introduction to Angular concepts: <https://angular.io/guide/architecture>
- Benefits of React*. (s.d.). Obtido em 3 de junho de 2021, de Benefits of React: <https://brainhub.eu/library/benefits-of-react-js/>
- BMW. (novembro de 2018). *BMW ConnectedDrive Services – Information/Data Protection*. Obtido em 18 de janeiro de 2021, de BMW: https://btcapiwebapp.eu.azurewebsites.net/api/gateway/contentserver/staticcontent/Angular/gdpr/v3/?target=bmw-browser#/legal-docs-content?version=2020.11.16-2&fileName=Bmw_cd_pp_pl-en.json
- BMW. (s.d.). *Remote Services*. Obtido em 18 de janeiro de 2021, de BMW: <https://connecteddrive.bmwlaunchpad.co.uk/services/remote-services>
- Center of Excellence for Professional Development. (s.d.). *What is FURPS+?* Obtido em 28 de janeiro de 2021, de What is FURPS+?: <https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/>
- Chintis, M., Tiwari, P., & Ananthamurthy, L. (14 de março de 2003). *Creating UML Use Case Diagrams*. Obtido em 30 de janeiro de 2021, de Creating UML Use Case Diagrams: <https://www.developer.com/design/article.php/2109801/Creating-Use-Case-Diagrams.htm>
- Demandt, B. (s.d.). *European sales 2019 EV and PHEV*. Obtido em 17 de janeiro de 2021, de CARSALESBASE: <https://carsalesbase.com/european-sales-2019-ev-phev/>
- Desktop Screen Resolution Stats Worldwide | Statcounter Global Stats*. (setembro de 2021). Obtido em 4 de outubro de 2021, de statcounter: <https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide>
- Docker. (s.d.). *Docker overview*. Obtido em 25 de janeiro de 2021, de Docker: <https://docs.docker.com/get-started/overview/>
- Docker. (s.d.). *Docker overview*. Obtido em 21 de Maio de 2021, de Docker overview: <https://docs.docker.com/get-started/overview/>
- Dorr, T. (s.d.). *Tesla JSON API (Unofficial)*. Obtido em 19 de Abril de 2021, de Tesla JSON API (Unofficial): <https://tesla-api.timdorr.com/>
- EDP Comercial. (9 de setembro de 2020). *Carro elétrico: tudo o que precisa de saber*. Obtido em 15 de janeiro de 2021, de EDP: <https://www.edp.pt/particulares/content-hub/tudo-sobre-carros-eletricos/>

- Emeral Publishing. (s.d.). *How to carry out action research*. Obtido em 28 de janeiro de 2021, de Carry out action research: <https://www.emeraldgrouppublishing.com/how-to/research-methods/carry-out-action-research>
- Escudeiro, P. M., & Bidarra, J. (janeiro de 2008). Quantitative Evaluation Framework (QEF). *RISTI – Revista Ibérica de STI*, Nº 1, pp. 16-18.
- EV Life. (s.d.). *How DC Fast Charging Really Works and an Intro to Charging Curves*. Obtido em 26 de abril de 2021, de How DC Fast Charging Really Works and an Intro to Charging Curves: <https://www.chargepoint.com/blog/how-dc-fast-charging-really-works-and-intro-charging-curves/>
- Express. (s.d.). *Fast, unopinionated, minimalist web framework for Node.js*. Obtido em 25 de janeiro de 2021, de Express: <https://expressjs.com/>
- Fowler, M. (s.d.). *MicroservicePremium*. Obtido em 8 de abril de 2021, de MicroservicePremium: <https://www.martinfowler.com/bliki/MicroservicePremium.html>
- Freire, H. (s.d.). *Medium*. Obtido de React, Vue, Angular, conheça suas vantagens e desvantagens e qual é melhor para seus projetos: <https://henriquefreire.medium.com/react-vue-angular-conhe%C3%A7a-suas-vantagens-e-desvantagens-e-qual-%C3%A9-melhor-para-seus-projetos-53734bb3d37f>
- Freire, H. (s.d.). *Medium*. Obtido de React, Vue, Angular, conheça suas vantagens e desvantagens e qual é melhor para seus projetos: <https://henriquefreire.medium.com/react-vue-angular-conhe%C3%A7a-suas-vantagens-e-desvantagens-e-qual-%C3%A9-melhor-para-seus-projetos-53734bb3d37f>
- Gnatyk, R. (s.d.). *Microservices vs Monolith*. Obtido em 8 de abril de 2021, de Microservices vs Monolith: <https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/>
- IBM. (s.d.). *What is Docker?* Obtido em 10 de Maio de 2021, de What is Docker?: <https://www.ibm.com/cloud/learn/docker>
- Introduction | Vue.js*. (s.d.). Obtido em 5 de abril de 2021, de What is Vue.js?: <https://v3.vuejs.org/guide/introduction.html#what-is-vue-js>
- Jira. (s.d.). *What is Jira used for*. Obtido em 26 de janeiro de 2021, de Jira Software: <https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for>
- Koen, P., Ajamian, G., Burkart, R., Clamen, D. J., D'Amore, R., Elkins, C., . . . Wagner, K. (abril de 2001). Providing Clarity and Common Language to the Fuzzy Front End. pp. 46-55.

Kozyra, K. (24 de julho de 2019). *What is a car infotainment system?* Obtido em 16 de janeiro de 2021, de Concise Software: <https://concisesoftware.com/car-infotainment-system-guide/>

Lapierre, J. (2000). Customer-perceived value in industrial contexts. *Journal of Business & Industrial Marketing*. *Journal of Business & Industrial Marketing*, pp. 122-145.

Malan, R., & Bredemeyer, D. (2001). Architecture Resources For Enterprise Advantage. *Functional Requirements and Use Cases*, pp. 1-2. Obtido de Architecture Resources.

Microsoft. (s.d.). <https://www.microsoft.com/pt-pt/microsoft-365/excel>. Obtido em 27 de janeiro de 2021, de Microsoft Excel: <https://www.microsoft.com/pt-pt/microsoft-365/excel>

MOBI.E. (s.d.). *Porquê optar pela Mobilidade Elétrica*. Obtido em 15 de janeiro de 2021, de MOBI.E: <https://www.mobie.pt/why-electric-mobility>

MongoDB. (s.d.). *What Is MongoDB?* Obtido em 26 de janeiro de 2021, de MongoDB: <https://www.mongodb.com/what-is-mongodb>

Nissan. (s.d.). *NissanConnect Features*. Obtido em 20 de janeiro de 2021, de Nissan: <https://www.nissanusa.com/connect/features-apps.html>

Node.js. (s.d.). *About Node.js*. Obtido em 24 de janeiro de 2021, de Node.js: <https://nodejs.org/en/about/>

node-tesla-api. (s.d.). Obtido em 19 de Abril de 2021, de node-tesla-api: <https://www.npmjs.com/package/node-tesla-api>

O que é um CEME? (s.d.). Obtido em 15 de janeiro de 2021, de Poupa Energia: <https://poupaenergia.pt/faq/o-que-e-um-ceme/>

React - A JavaScript library for building user interfaces. (5 de abril de 2021). Obtido de React: <https://reactjs.org/>

Renault. (s.d.). *Renault*. Obtido em 2021 de janeiro de 20, de My Renault: <https://ie.e-guide.renault.com/eng/easy-link/MY-RENAULT>

Saaty, R. W. (1987). THE ANALYTIC HIERARCHY PROCESS - WHAT IT IS AND HOW IT IS USED. *Mathd Modelling*, 9(3-5), pp. 161-176.

Tesla. (s.d.). *Tesla*. Obtido em 19 de janeiro de 2021, de Serviço de apoio da aplicação Tesla: https://www.tesla.com/pt_PT/support/tesla-app

TeslaJS. (s.d.). Obtido em 19 de Abril de 2021, de TeslaJS: <https://www.npmjs.com/package/teslajs>

Visual Paradigm. (s.d.). *Agile project tools to help develop great software*. Obtido em 26 de janeiro de 2021, de Visual Paradigm: <https://www.visual-paradigm.com/>

Visual Paradigm. (s.d.). *UML Class Diagram*. Obtido em 12 de Abril de 2021, de UML Class Diagram: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

Volkswagen. (s.d.). *Volkswagen*. Obtido em 2021 de janeiro de 19, de We Connect Go: <https://www.volkswagen.pt/pt/conectividade-e-servicos-de-mobilidade/we-connect-go.html>

What are microservices? (s.d.). Obtido em 17 de abril de 2021, de What are microservices?: <https://microservices.io/>

Anexo 1 Planeamento



2020

2021

dezembro

janeiro

fevereiro

março

abril

maio

junho

julho

agosto

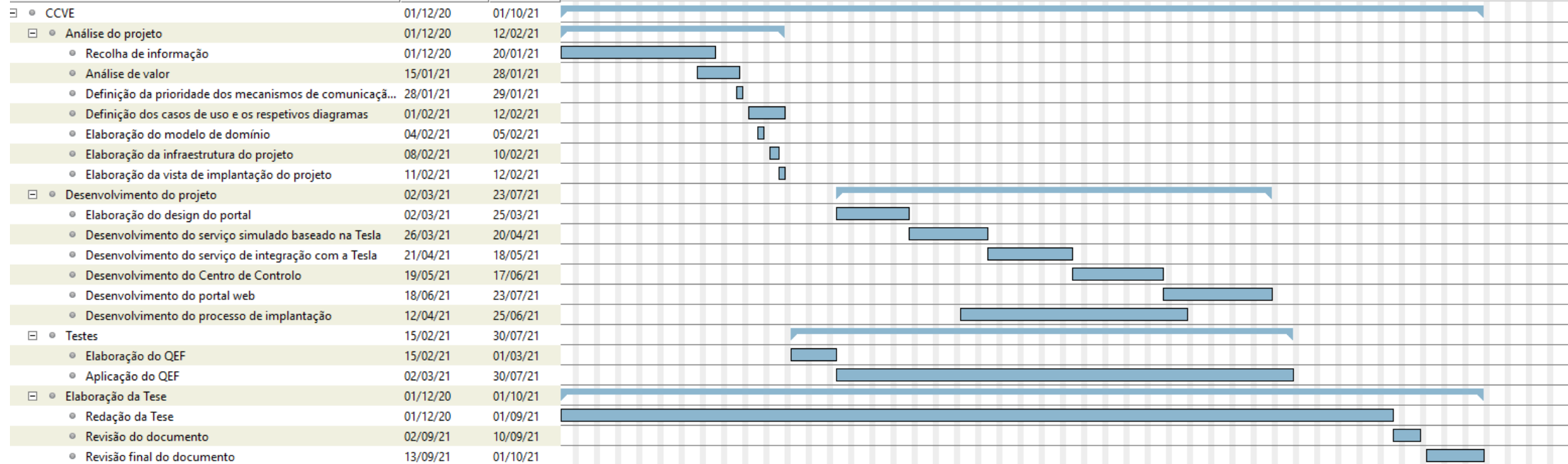
setembro

outubro

Nome

Data de início

Data de fim



Anexo 2 QEF

q	D	qi	Dimension	Qj	Wij (Factor Weight j in Dim)	Factor	rwjk (requirement weight k in Factor j) {2, 4, 6, 8, 10}	Requirement	wfk % requirement fulfillment k) [0,100]
94%	0,17	87,5	Functionality	87,50	0,17	Functional	10,00	FF01 - The CCVE server connects with the APIs made available by the car brands that manage the vehicles	50
							10,00	FF02 - The web portal connects to the CCVE's server	100
							10,00	FF03 - The CCVE's services are based in a microservices architecture	100
							10,00	FF04 - The CCVE server stores the vehicles information	100
				87,50	0,33	Features	10,00	FE01 - The web portal is available across multiple browsers (Google Chrome, Mozilla Firefox, Microsoft Edge)	100
							10,00	FE02 - The web portal uses maps to show the vehicle geographical location	100
							10,00	FE03 - The web portal shows the vehicle communication status	100
							10,00	FE04 - The system uses intelligent automatic data analysis mechanisms to notify the user about issues related to his vehicles	0
							10,00	FE05 - The system analyzes the user-configured alerts and notifies the user about issues related to his vehicles	100
							10,00	FE06 - The system processes the user history and presents statistics related to actions taken with it's vehicles	100
							10,00	FE07 - The system processes the user history and presents statistics related to a specific vehicle	100
							10,00	FE08 - The web portal design needs to follow the guidelines of other EVIO's products	100
				94,44	0,38	User Interaction	10,00	FUI01 - The web portal's language is English	100
10,00	FUI02 - The user can create a account with an associated payment plan	50							
10,00	FUI03 - The user can register a new vehicle	100							
10,00	FUI04 - The user can delete a vehicle	100							
10,00	FUI05 - The user can monitor the vehicle's location	100							
10,00	FUI06 - The user can monitor the vehicle's charging status	100							
10,00	FUI07 - The user can remotely execute a command to lock/unlock the vehicle	100							
10,00	FUI08 - The user can remotely execute a command to turn on/off the air conditioning option of the vehicle	100							
10,00	FUI09 - The user can configure alert notifications	100							

						10,00	FA01 - The user have access to their profile	100
		66,67	0,13	Account		10,00	FA02 - The user can change the information displayed in their profile	100
						10,00	FA03 - The user can change the payment plan associated to his account	0
	88,75	100,00	0,17	Features		10,00	AF01 - The system presents to the user with useful messages after performing actions	100
		81,25	0,33	Maintenance		10,00	AM01 - The CCVE server presents a set of interfaces and endpoints, allowing the possibility of easily introducing new services.	100
						6,00	AM02 - The number of instances of a specific service running needs to adapt accordingly to the number of request (Auto Scaling)	50
		90,00	0,50	Versatility		10,00	AV01 - The information displayed in the web portal is exclusive to the user	100
						10,00	AV02 - The web portal adjusts to screen resolution	100
						5,00	AV03 - The web portal supports a mobile version	50
						10,00	UPN01 - The web portal interface is simple and fast, without slowing down when the number of users is high	100
						10,00	UPN02 - The services runtime do not have errors, and unexpected errors should be handled	100
		100,00	0,20	Portal Navigation		10,00	UMN01 - Easy access to real time maps (Mapbox)	100
						8,00	UMN02 - Easy to understand where the vehicle avatar is located	100
		100,00	0,20	Map Navigation		10,00	UCQ01 - All the messages are easy to understand.	100
						8,00	UCQ02 - Texts are simple and concise.	100
						6,00	UCQ03 - No messages should contain offensive content	100
		100,00	0,30	Content Quality		10,00	UIN01 - Consistency of the data processed by the portal and server	100
						10,00	UIN02 - Synchronization of users's vehicles information	100
						10,00	UIN03 - Users data is protected from unauthorized access	100
	100			Usability				

Dimension Functionality

Factor Functional

Wfk - Fulfilment (%)

Requirement	Metric Evaluation	Wfk - Fulfilment (%)				
		0	25	50	75	100
FF01 - The CCVE server connects with the APIs made available by the car brands that manage the vehicles	The CCVE server should allow the connection between the services and the vehicles's APIs that allow the collection of information	The server does not allow the connection between internal services and external vehicle APIs	-	The server only allow some connections between internal services and external vehicle APIs	-	The server allow all communications between internal services and external vehicle APIs
FF02 - The web portal connects to the CCVE's server	The web portal needs to be connect to the CCVE's server in order to be available for use	The portal cannot connects to the CCVE server	-	-	-	The portal connects to the CCVE server
FF03 - The CCVE's services are based in a microservices architecture	The solution follows a microservices architecture, with multiple services that communicate with each other and external APIs	The services implementation do not follow a microservices architecture	-	-	-	The services implemetation follow a microservices architecture
FF04 - The CCVE server stores the vehicles information	The CCVE server must process and store the information retrieved from the user's vehicles	The implemented services do not store the information retrieved from the vehicles	-	Some of the implemented services do not store the information retrieved from the vehicles	-	All the implemented services store the information retrieved from the vehicles

Factor	Features
---------------	----------

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
FE01 - The web portal is available across multiple browsers (Google Chrome, Mozilla Firefox, Microsoft Edge)	The portal must be accessible in multiple browsers (Google Chrome, Mozilla Firefox, Microsoft Edge).	The portal is not accessible in any of the platforms	-	The portal is available only in Google Chrome	-	The portal is available in all listed platforms
FE02 - The web portal uses maps to show the vehicle geographical location	The portal needs to uses a visual representation of a map to show the vehicle's geographical location	The geographical location of the user's vehicles are not shown in map format	-	-	-	Each vehicle shows a map where the user can verify the vehicle geographical location
FE03 - The web portal shows the vehicle communication status	The portal shows the communication status of the vehicles with the vehicle brands APIs (green for online; black for offline)	The communication status indicator is not visible for the vehicles	-	The communication status indicator is implemented, but not available for all vehicles in the portal	-	The portal show the communication status of all vehicles
FE04 - The system uses intelligent automatic data analysis mechanisms to notify the user about issues related to his vehicles	The system features intelligent automatic mechanisms that analyzes the data colleted from the vehicles and notifies the user of possible issues	The system does not have intelligent automatic mechanisms that analyzes the vehicle's data	-	Only one mechanism of intelligent data analysis is implemented	-	Both mechanisms are implemented and the user is notified of vehicle's warnings
FE05 - The system analyzes the user-configured alerts and notifies the user about issues related to his vehicles	The service features automatic mechanisms that process the configured notification alerts by the user	Not implemented	-	The mechanisms are implemented, but the user is not notified of possible issues with the vehicles	-	The mechanism is implemented and the user is notified of possible issues with the vehicles
FE06 - The system processes the user history and presents statistics related to actions taken with it's vehicles	The system is responsible for presenting the user with statistics related to it's actions with the registered vehicles	Not implemented	-	The system presents the user's statistics, but some information is missing or not correct	-	The system presents the statistics related to the user's activities with his vehicles
FE07 - The system processes the user history and presents statistics related to a specific vehicle	The system is responsible for presenting the user with statistics related to charging sessions of a specfic vehicle. This information needs to be displayed in graphs	Not implemented	-	The system presents the selected vehicle statistics, but some information is missing or not correct	-	The system presents the selected vehicle statistics
FE08 - The web portal design needs to follow the guidelines of other EVIO's products	The design of the web portal follows in agreement with the design of other EVIO's solutions	EVIO's design language was not followed in the development of the design	-	-	-	EVIO's design language was followed in the development of the design

Factor	User Interaction
---------------	------------------

Wfk - Fulfilment (%)

Requirement	Metric Evaluation	0	25	50	75	100
FUI01 - The web portal's language is English	The web portal's default language is English	The components and messages presented to the user are not in English	-	Only some components and messages presented to the user are in English	-	All the components and messages presented to the user are in English
FUI02 - The user can create a account with an associated payment plan	The user can create a account in the application, with an valid payment plan associated	The user cannot create an account	-	The user can create an account, but can't associate a valid payment plan with the exception of the free plan	-	The user can create an account and choose a valid payment plan to associate
FUI03 - The user can register a new vehicle	The user can register a new vehicle in the plataform (brand, model, version, serial number, license plate)	The user cannot register a vehicle	-	The user can register a vehicle, but some vehicle information is missing	-	The user can register a valid vehicle with success
FUI04 - The user can delete a vehicle	The user can delete a vehicle that he previously registered	The user cannot delete a vehicle	-	-	-	The user can delete a vehicle
FUI05 - The user can monitor the vehicle's location	The portal allows the user to monitor the geographical location of it's vehicles	The portal do not shows the location of it's vehicles	-	-	-	The user can monitor the location of it's vehicles in the portal
FUI06 - The user can monitor the vehicle's charging status	The portal allows the user to monitor the charging status of it's vehicles	The portal does not show the charging status of the user's vehicles	-	The portal only shows the charging status of some vehicle	-	The portal shows the charging status of all the user registered vehicles
FUI07 - The user can remotely execute a command to lock/unlock the vehicle	The portal allows the user to execute commands to lock or unlock a selected vehicle	The portal does not show the available commands of a vehicle	The portal shows the available commands of a vehicle, but do not allow the execution of commands	The user can only use with success one of the available commands	The user can use both commands, but bugs are detected	The portal allows the execution of the commands in a vehicle with success
FUI08 - The user can remotely execute a command to turn on/off the air conditioning option of the vehicle	The portal allows the user to execute commands to turn on or off the air conditioning option of a selected vehicle	The portal does not show the available commands of a vehicle	The portal shows the available commands of a vehicle, but do not allow the execution of commands	The user can only use with success one of the available commands	The user can use both commands, but bugs are detected	The portal allows the execution of the commands in a vehicle with success
FUI09 - The user can configure alert notifications	The portal allows the user to configure alert notifications based on the data collects from the it's vehicles	The user cannot configure alerts	-	The user can configure alerts, but bugs are detected	-	The user can configure alerts with success

Factor	Account
---------------	---------

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
FA01 - The user have access to their profile	The user can access his account profile	The user cant access his account profile	-	The user cant access, but the profile is displayed with bugs	-	The user can access his account profile with success
FA02 - The user can change the information displayed in their profile	The user can change the information presented in his account profile	The user information fails to update	-	The user information is updated, but flaws are detected in the data update process	-	The new information is updated and displayed with success
FA03 - The user can change the payment plan associated to his account	The user can change the currently associated payment plan	The user cant change the current payment plan	-	The user can change the current payment plan, but with bugs	The payment plan associated with the user changes, but the system does not realize the necessary monetary balance update	The user can change the current payment plan with success

Dimension	Adaptability
Factor	Features

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
AF01 - The system presents to the user with useful messages after performing actions	The web portal presents the user with useful messages when he interacts with the system	The system does not presents the user with a message after completing an action	-	-	-	The system presents the user with a message after completing an action

Factor	Maintenance
---------------	-------------

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
AM01 - The CCVE server presents a set of interfaces and endpoints, allowing the possibility of easily introducing new services.	The application is robust and easily support the integration of new services	The services are not developed with endpoints that allow the easy communication with other services of the plataform	-	-	-	The services are developed with endpoints that allow the easy communication with other services of the plataform
AM02 - The number of instances of a specific service running needs to adapt accordingly to the number of request (Auto Scaling)	The number os running services needs to adapt to the number of requests to process, in order to keep the application working at peak performance	Not implemented	-	The number of instances running are limited to a specific number, regardless of the number of requests or users	-	The system is fully scalable

Factor	Versatility
---------------	-------------

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
AV01 - The information displayed in the web portal is exclusive to the user	The information displayed in the web portal is related to the user logged in the system	The information displayed is not related to the user	-	-	-	The information displayed is only related to the user
AV02 - The web portal adjusts to screen resolution	The web portal adjusts the layout to the screen resolution	Not implemented	-	Not all components are adjusted to the screen resolution	-	All components are adjusted to the screen resolution
AV03 - The web portal supports a mobile version	The web portal supports a version for mobile users	Not implemented	-	Not all components are adjusted to the mobile version	-	All components are adjusted to the mobile version

Dimension	Usability
Factor	Portal Navigation

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
UPNO1 - The web portal interface is simple and fast, without slowing down when the number of users is high	The web portal should be quick and simple to use and easily move between functionalities, independently of the number of users.	The request takes more than 5 seconds to be processed	-	The request takes between 1 to 3 seconds to be processed	-	The request takes less than 1 seconds to be processed
UPNO2 - The services runtime do not have errors, and unexpected errors should be handled	Unexpected errors must be presented to the user with a clear message about the error.	No operation provides the user with a message when a error is detected	-	Some operations provide the user with a message when a error is detected	-	All operations provide the user with a message when a error is detected

Factor	Map Navigation
---------------	----------------

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
UMNO1 - Easy access to real time maps (Mapbox)	The map functionalities are easily available in the CCVE and the current location of the vehicle is displayed in the map	The map does not load	-	The map loads, but the vehicle location is missing	-	The map presents the vehicle location
UMNO2 - Easy to understand where the vehicle avatar is located	The location of the vehicle on the map is easily identifiable	The vehicle location is not clear in the map	-	-	-	The vehicle location is represented with an easily identifiable avatar

Factor	Content Quality
---------------	-----------------

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
UCQ01 - All the messages are easy to understand.	All the messages should be appropriate in spacing and size	0%-20% as result of the questionnaire responses	20%-40% as result of the questionnaire responses	40%-60% as result of the questionnaire responses	60%-80% as result of the questionnaire responses	80%-100% as result of the questionnaire responses
UCQ02 - Texts are simple and concise.	The user should be able to read and understand the text in the portal	0%-20% as result of the questionnaire responses	20%-40% as result of the questionnaire responses	40%-60% as result of the questionnaire responses	60%-80% as result of the questionnaire responses	80%-100% as result of the questionnaire responses
UCQ03 - No messages should contain offensive content	No messages should contain offensive content	Some messages contain offensive content	-	-	-	No messages contain offensive content

Factor	Integrity
---------------	-----------

		Wfk - Fullfilment (%)				
Requirement	Metric Evaluation	0	25	50	75	100
UIN01 - Consistency of the data processed by the portal and server	The data should be consistant on the server and in the portal when displayed	The consistency of the data is not maintained	-	-	-	The consistency of the data is maintained
UIN02 - Synchronization of users's vehicles information	The vehicle information in the server should be synchronized with the current information of the vehicle, with a refresh rate of 5 minutes	Data is not synchronized	-	-	-	Data is synchronized
UIN03 - Users data is protected from unauthorized access	The information provided by the user is protected from unauthorized access and cant be used to identify him, according to the RGPD guidelines for data protection	The access to the user's information is not verified	-	The access to the information is verified, but it identifies the user	-	The access to the information is verified and it cant be used to identify the user

Anexo 3 Questionário

CCVE - Centro de Controlo para Veículos Eléctricos (Control Center for Electric Vehicles)

This questionnaire is carried out within the scope of the thesis "Control Center for Electric Vehicles", which is part of the EVIO project. This study aims to evaluate the messages that are presented to the users of the CCVE portal. Specifically, it's designed to assess the ease of understanding of those messages and whether their meaning is clear and/or if their content is offensive.

*Obrigatório

CCVE Messages

The messages are presented to the users when functionalities are performed in the application and feedback of their status is provided. These messages are used to inform the user of the success/failure of the operation that was performed.

Thus, all the messages that are presented to the users of the CCVE portal will be listed in this section.

1. Login Failed
2. Login Successful
3. Vehicle created
4. Vehicle creation failed
5. Door unlocked
6. Door locked
7. Door already locked
8. Door already unlocked
9. Air conditioning on
10. Air conditioning off
11. Air conditioning already on
12. Air conditioning already off
13. Alert created
14. Alert creation failed

Please answer the following questions.

1. Can you read and understand simple sentences written in English? *

Marcar apenas uma oval.

Yes *Pular para a pergunta 2*

No *Pular para a seção 4 (Thank you for your feedback)*

2.

*

Please answer the questions according to the following scale: 1 - Do not understand; 2 - Somewhat understandable; 3 - Neutral; 4 - Understand well; 5 - Easy to understand.

Marcar apenas uma oval por linha.

	1	2	3	4	5
Regarding the message "Login Failed", is it clear that the user could not be authenticated in the system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regarding the message "Login Successful", is it clear that the user was successfully authenticated in the system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regarding the message "Vehicle created", is it clear that a new vehicle was registered in the platform with success?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regarding the message "Vehicle creation failed", is it clear that a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

new vehicle failed to be registered in the platform?

Regarding the message "Door unlocked", is it clear that the command "Unlock Vehicle" was successfully executed?

Regarding the message "Door locked", is it clear that the command "Lock Vehicle" was successfully executed?

Regarding the message "Door already locked", is it clear that the command "Lock Vehicle" failed because the vehicle is already locked?

Regarding the message "Door already unlocked", is

Regarding the message "Door unlocked", is it clear that the command "Unlock Vehicle" was successfully executed?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regarding the message "Door locked", is it clear that the command "Lock Vehicle" was successfully executed?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regarding the message "Door already locked", is it clear that the command "Lock Vehicle" failed because the vehicle is already locked?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Regarding the message "Door already unlocked", is	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

it clear that the command "Unlock Vehicle" failed because the vehicle is already unlocked?

Regarding the message "Air conditioning on", is it clear that the command "Switch AC On" was successfully executed?

Regarding the message "Air conditioning off", is it clear that the command "Switch AC Off" was successfully executed?

Regarding the message "Air conditioning already on", is it clear that the command "Switch AC On" failed because the vehicle's AC

is already on?

Regarding the message "Air conditioning already off", is it clear that the command "Switch AC Off" failed because the vehicle's AC is already off?

Regarding the message "Alert created", is it clear that a new alert was registered in the platform successfully?

Regarding the message "Alert creation failed", is it clear that a new alert failed to be registered in the platform?

3. How easy is it to understand the messages displayed in the CCVE portal? *

Marcar apenas uma oval.

- 1 - Do not understand
- 2 - Somewhat understandable
- 3 - Neutral
- 4 - Understand well
- 5 - Easy to understand

4. Do the messages displayed in the CCVE portal have offensive content? *

Marcar apenas uma oval.

- Yes. Offensive content is displayed in some of the messages
- No. The displayed messages have no offensive content

5. If you have selected "Yes" in the previous question, please elaborate on which message did you consider offensive.

Thank you for your feedback

Please do not forget to submit the questionnaire.