



Instituto Politécnico de Leiria  
Escola Superior de Tecnologia e Gestão  
Departamento de Engenharia Informática  
Mestrado em Eng.<sup>a</sup> Informática – Computação Móvel

## PLATAFORMA REDSERVICE

BRUNO MIGUEL GOMES OLIVEIRA

Leiria, setembro de 2024





ESCOLA SUPERIOR  
DE TECNOLOGIA  
E GESTÃO

Instituto Politécnico de Leiria  
Escola Superior de Tecnologia e Gestão  
Departamento de Engenharia Informática  
Mestrado em Eng.<sup>a</sup> Informática – Computação Móvel

## PLATAFORMA REDSERVICE

BRUNO MIGUEL GOMES OLIVEIRA

Número: 2202591

Projeto de Mestrado realizado sob orientação da professora Dulce Cristina dos Santos Iria Gonçalves ([dulce.goncalves@ipleiria.pt](mailto:dulce.goncalves@ipleiria.pt)).

Leiria, setembro de 2024



## AGRADECIMENTOS

---

A conclusão deste trabalho de mestrado não teria sido possível sem o apoio e incentivo de várias pessoas e instituições, às quais gostaria de expressar minha profunda gratidão.

Primeiramente, agradeço à minha orientadora, professora Dulce Cristina dos Santos Iria Gonçalves, pela sua orientação, paciência e valiosas contribuições ao longo de todo o processo de desenvolvimento deste projeto.

Agradeço também à Leirilis, pela oportunidade de desenvolver este projeto, pela confiança depositada e a todo o apoio logístico, que permitiram que este trabalho fosse realizado com êxito.

À minha família, em especial aos meus pais, pelo suporte incondicional e por acreditarem no meu potencial.

Um agradecimento especial à minha namorada, pelo carinho, paciência e apoio contínuo. A sua presença ao meu lado foi essencial nos momentos mais desafiantes e o seu incentivo constante motivou-me a concluir este projeto.



## RESUMO

---

A Plataforma RedService nasce na necessidade da empresa Leirilis querer uniformizar a gestão oficial da sua rede de oficinas.

O presente documento tem como objetivo discriminar o desenvolvimento da Plataforma RedService, uma solução tecnológica focada na gestão de oficinas automóveis, com o objetivo de otimizar processos e melhorar a experiência dos clientes e oficinas. O projeto foi realizado para a empresa Leirilis, que identificou a necessidade de criar um sistema centralizado capaz de gerir múltiplas oficinas, facilitando desde o agendamento de serviços até o controlo dos processos internos das oficinas.

A metodologia adotada para o desenvolvimento envolveu o uso de técnicas ágeis, como Scrum e Kanban, que permitiram uma evolução contínua e incremental da plataforma. Foram utilizadas diversas tecnologias, como *ASP.NET Core*, *React.js* e *Microsoft SQL Server*, para garantir a robustez, escalabilidade e segurança da plataforma. Além disso, foi desenvolvida uma aplicação móvel híbrida e uma aplicação *web* para melhorar a acessibilidade e a interação dos utilizadores com a plataforma.

Os resultados alcançados mostraram que a Plataforma RedService é eficaz em centralizar a gestão das oficinas, permitindo um aumento da eficiência operacional, redução de custos e o melhoramento da comunicação com os clientes.

Por fim, o projeto abre portas para melhorias futuras, como a implementação de inteligência artificial para otimizar ainda mais os processos e a inclusão de novas funcionalidades voltadas para a fidelização dos clientes à rede RedService.

**Palavras-chave:** Gestão de oficinas, Plataforma RedService, Tecnologia, ASP.NET Core, React.js, SQL Server.



## ABSTRACT

---

The RedService Platform arises from the need of the company Leirilis to standardize the workshop management of its network of repair shops.

The purpose of this document is to detail the development of the RedService Platform, a technological solution focused on the management of automotive workshops, aiming to optimize processes and improve the experience for both customers and repair shops.

The project was carried out for the company Leirilis, which identified the need to create a centralized system capable of managing multiple workshops, facilitating everything from service scheduling to the control of the internal processes of the repair shops.

The methodology adopted for the development involved the use of agile techniques, such as Scrum and Kanban, which allowed for continuous and incremental evolution of the platform. Various technologies were used, such as ASP.NET Core, React.js, and Microsoft SQL Server, to ensure the platform's robustness, scalability, and security. In addition, a hybrid mobile application and a web application were developed to improve accessibility and user interaction with the platform.

The results achieved showed that the RedService Platform is effective in centralizing workshop management, allowing for an increase in operational efficiency, cost reduction, and improved communication with customers.

Finally, the project opens the door for future improvements, such as the implementation of artificial intelligence to further optimize processes and the inclusion of new features aimed at customer loyalty to the RedService network.

**Keywords:** Workshop management, RedService Platform, Technology, ASP.NET Core, React.js, SQL Server.



# ÍNDICE

---

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	vii
Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Abreviaturas	xv
1 Introdução	1
2 Estado da arte	3
2.1 Empresa Leirilis . . . . .	3
2.2 Contexto empresarial . . . . .	3
2.3 Gestão oficial . . . . .	4
2.3.1 Fluxo oficial . . . . .	4
2.4 RedService . . . . .	6
2.4.1 Plataformas concorrentes . . . . .	6
2.4.2 Conceito RedService . . . . .	8
2.4.3 Conclusão . . . . .	10
3 Arquitetura	11
3.1 Arquitetura geral da solução . . . . .	11
3.1.1 Servidores . . . . .	12
3.1.2 Dispositivos . . . . .	13
3.1.3 Conclusão . . . . .	14
4 Tecnologias	15
4.1 Application Programming Interface . . . . .	15
4.1.1 Express.js . . . . .	15
4.1.2 Spring . . . . .	16
4.1.3 WebAPI . . . . .	16
4.1.4 ASP.Net Core . . . . .	16

4.1.5	Conclusão . . . . .	17
4.2	Sistema de base de dados . . . . .	17
4.2.1	SQL - <i>Structured Query Language</i> . . . . .	17
4.2.2	<i>Firebird</i> . . . . .	18
4.2.3	MongoDB . . . . .	18
4.2.4	Conclusão . . . . .	19
4.3	Aplicação <i>Web</i> . . . . .	19
4.3.1	ASP.NET . . . . .	19
4.3.2	Vue.js . . . . .	20
4.3.3	React.js . . . . .	20
4.3.4	Conclusão . . . . .	20
4.4	Aplicação móvel . . . . .	20
4.4.1	Apache Cordova . . . . .	21
4.4.2	Xamarin . . . . .	21
4.4.3	React Native . . . . .	22
4.4.4	Flutter . . . . .	22
4.4.5	Conclusão . . . . .	22
5	Metodologia . . . . .	25
5.1	Metodologias de trabalho . . . . .	25
5.1.1	Kanban . . . . .	25
5.1.2	<i>Scrum</i> . . . . .	26
5.2	Conclusão . . . . .	27
6	Implementação . . . . .	29
6.1	Análise de requisitos . . . . .	29
6.1.1	Requisitos funcionais . . . . .	29
6.1.2	Aplicações móveis - Permissões de cliente . . . . .	29
6.1.3	Aplicações móveis - Permissões de oficina . . . . .	30
6.1.4	<i>Workshop Management System</i> . . . . .	31
6.1.5	Site RedService . . . . .	32
6.1.6	Requisitos não-funcionais . . . . .	33
6.2	Ferramentas Utilizadas . . . . .	33
6.3	Protótipos . . . . .	35
6.4	Desenvolvimento . . . . .	40
6.4.1	Base de dados . . . . .	40
6.4.2	<i>Application Programming Interface</i> . . . . .	43
6.4.3	Segurança . . . . .	44

6.4.4	Roles/Perfis . . . . .	46
6.4.5	Controladores . . . . .	47
6.4.6	Aplicação Web . . . . .	47
6.4.7	Aplicação Móvel . . . . .	51
6.5	Testes . . . . .	59
7	Conclusão	61
	Bibliografia	63
	Declaração	65



## LISTA DE FIGURAS

---

Figura 1	Diagrama de fluxo da plataforma de agendamentos . . . . .	4
Figura 2	Folha de obra . . . . .	5
Figura 3	Arquitetura geral da solução . . . . .	11
Figura 4	Excerto de quadro <i>Kanban</i> utilizado no desenvolvimento da Plataforma RedService . . . . .	26
Figura 5	Gráfico de Gantt das tarefas realizadas . . . . .	27
Figura 6	GitHub . . . . .	34
Figura 7	Quadro Kanban . . . . .	35
Figura 8	WMS - Página principal . . . . .	36
Figura 9	WMS - Agendamentos . . . . .	36
Figura 10	WMS - Utilizadores . . . . .	37
Figura 11	WMS - Formulário . . . . .	38
Figura 12	<i>Mockups</i> das aplicações móveis . . . . .	39
Figura 13	Modelo físico da Base de dados RedService . . . . .	41
Figura 14	DashBoard sobre a Plataforma RedService . . . . .	48
Figura 15	Formulário de dados de uma oficina . . . . .	48
Figura 16	Formulário de dados de um Utilizador da Plataforma . . . . .	49
Figura 17	Agendamento: Calendário . . . . .	49
Figura 18	Agendamento: Detalhes . . . . .	50
Figura 19	Agendamento: Documento . . . . .	50
Figura 20	Histórico . . . . .	51
Figura 21	Login . . . . .	52
Figura 22	Página principal . . . . .	53
Figura 23	Rede RedService . . . . .	53
Figura 24	Listagem . . . . .	54
Figura 25	Detalhes . . . . .	54
Figura 26	Notificações . . . . .	54
Figura 27	Cartão RedService . . . . .	54
Figura 28	Listagem . . . . .	55
Figura 29	Informação . . . . .	56
Figura 30	Detalhes . . . . .	56
Figura 31	Serviços . . . . .	56

LISTA DE FIGURAS

Figura 32	Definições . . . . .	56
Figura 33	Página inicial da oficina . . . . .	57
Figura 34	Agendamentos oficina . . . . .	57
Figura 35	Detalhes . . . . .	58
Figura 36	Iniciar tempos . . . . .	58
Figura 37	Lista de tempos . . . . .	58
Figura 38	Agendamentos oficina . . . . .	58

## LISTA DE TABELAS

---

Tabela 1	Análise das plataforma concorrentes à RedService . . . . .	8
Tabela 2	Plataforma <i>Web</i> . . . . .	46
Tabela 3	Aplicação <i>mobile</i> . . . . .	46
Tabela 4	Controladores . . . . .	47
Tabela 5	Estados da aplicação . . . . .	52

LISTA DE TABELAS

## LISTA DE ABREVIATURAS

---

- API     Application Programming Interface.
- ERP     Enterprise Resource Planning.
- ESTG    Escola Superior de Tecnologia e Gestão de Leiria.
- IDE     Integrated development environment.
- RGPD    Regulamento Geral de Proteção de Dados.
- SQL     Structured Query Language.
- WMS     Workshop Management System.



## INTRODUÇÃO

---

A elaboração deste documento decorre da realização de um projeto de mestrado, denominado "Plataforma RedService", para o Mestrado em Engenharia Informática – Computação Móvel, da Escola Superior de Tecnologia (ESTG) do Instituto Politécnico de Leiria.

A evolução tecnológica tem impactado diretamente diversos setores, incluindo o setor automóvel e oficial, onde a necessidade de soluções eficientes para a gestão de oficinas é crescente. No contexto deste trabalho, foi desenvolvida a Plataforma RedService, uma solução inovadora que visa centralizar e otimizar a gestão das oficinas associadas à rede RedService, proporcionando uma interface amigável tanto para as oficinas como para os clientes.

O projeto foi motivado pela necessidade da empresa Leirilil uniformizar os processos das suas oficinas e melhorar a experiência do cliente, utilizando uma plataforma tecnológica que possibilitasse agendamentos online, orçamentos e controle de serviços em tempo real.

A plataforma RedService foi desenvolvida com o objetivo de transformar a forma como as oficinas gerem as suas operações, implementando tecnologias que permitem uma gestão mais ágil, eficiente e de fácil adaptação. Neste documento serão abordados os principais objetivos da plataforma, os benefícios que ela traz para as oficinas, como a redução de custos operacionais e o aumento da produtividade e na capacidade das oficinas em atender mais clientes com maior eficiência e qualidade.

Neste documento, será abordado o desenvolvimento da Plataforma RedService, desde a sua conceção até à implementação, com a apresentação da componente tecnológica, as ferramentas utilizadas, bem como os desafios enfrentados ao longo do projeto. O trabalho está organizado da seguinte forma: no capítulo 2 é apresentado o estado da arte, onde se discute o contexto empresarial e as principais plataformas concorrentes; no capítulo 3, a arquitetura da solução é descrita em detalhe; o capítulo 4 explora as tecnologias utilizadas; no capítulo 5 são descritas as metodologias de trabalho; o capítulo 6 descreve a implementação técnica, e finalmente, as conclusões e propostas de trabalhos futuros serão apresentadas no capítulo 7.



## ESTADO DA ARTE

---

No presente capítulo vai ser descrita a empresa Leirilis, assim como o surgimento da RedService. De seguida, será apresentada a gestão oficial que é praticada nas oficinas RedService, explicando o seu fluxo. Por fim, será abordado o conceito RedService, mostrando as principais plataformas concorrentes.

### 2.1 EMPRESA LEIRILIS

A empresa foi fundada em Leiria em 1986 com o objetivo da comercialização de acessórios e peças destinadas ao mercado automóvel, no entanto, ao longo dos anos tem apostado em várias áreas de negócio dentro do ramo. Ao longo do tempo, a Leirilis, tem vindo a expandir as suas lojas a nível nacional, tendo aproximadamente 90 funcionários, distribuídos pelas 6 lojas em Portugal Continental e uma na ilha da Madeira (Leirilis, 2022).

### 2.2 CONTEXTO EMPRESARIAL

A missão da Leirilis é conseguir ser um parceiro de excelência para os seus clientes, sendo uma grande parte oficinas, mas contando também com uma cota parte de venda ao público. Numa ideia de criar novos desafios para a própria empresa e inovação, em 2016, foi desenvolvido o conceito RedService para conseguir dar aos seus parceiros, novas formas de uma gestão oficial mais eficiente e dinâmica.

A RedService é uma rede de oficinas multimarca que está estabelecida em Portugal e que visa a reparação e manutenção de automóveis. Este conceito criado pela Leirilis, tem como finalidade adaptar as oficinas as constantes mudanças no mercado permitindo que estas desenvolvam um negócio mais sustentável, competitivo e atraente, de modo a garantir os elevados níveis de notoriedade e confiança junto do consumidor final (RedService, 2022).

Para o futuro desenvolvimento da Plataforma RedService foi idealizado um sistema de gestão de oficinas, que permita através uma única plataforma centralizar toda

a informação referente aos serviços prestados aos clientes bem como otimizar os recursos humanos da empresa, como forma de obter a melhor rentabilidade da oficina, permitindo que as oficinas aumentem o volume de negócio bem como a qualidade do serviço prestado aos clientes. Com o desenvolvimento desta plataforma, também se pretende uniformizar o funcionamento das oficinas RedService.

### 2.3 GESTÃO OFICINAL

Na secção Fluxo oficial, vai ser descrito o funcionamento genérico de uma oficina, com base em diálogos com os gerentes das oficinas RedService. De seguida vai ser explicado como foi concebido o conceito da Plataforma RedService de forma a responder às necessidades e a otimizar os recursos da oficina.

#### 2.3.1 Fluxo oficial

Antes da elaboração de um conceito da Plataforma RedService, houve a necessidade de compreender a lógica de funcionamento das oficinas, para ser possível desenvolver uma plataforma o mais enquadrada no contexto real. Como descrito anteriormente, foram feitos diálogos com os gerentes de oficinas RedService, onde foi explicado o seu funcionamento e de que forma o desenvolvimento de uma plataforma seria importante para o negócio. No fim destes diálogos foi possível elaborar um diagrama de fluxo na figura 1, onde é possível observar de forma sucinta, todas as etapas do processo, desde que o cliente efetua um agendamento na oficina até o veículo sair da oficina, depois do serviço efetuado.

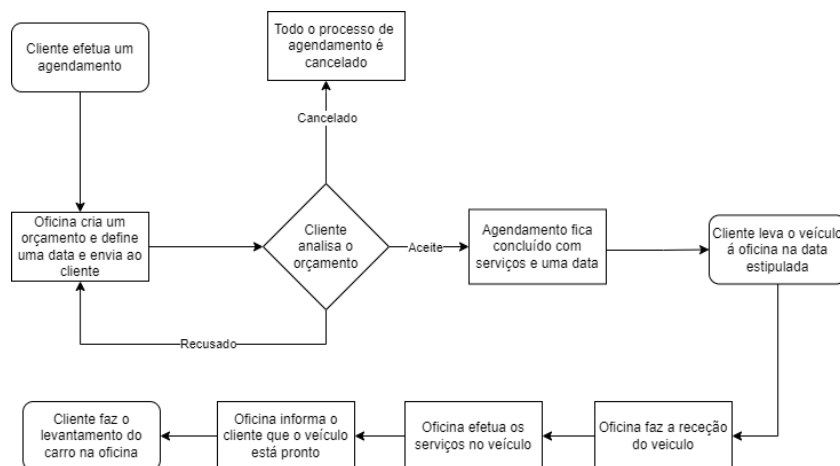


Figura 1: Diagrama de fluxo da plataforma de agendamentos



## 2.4 REDSERVICE

Aliando um dos principais valores da Leirilis (Leirilis, 2022), a inovação e as constantes mudanças que existem nos mercados, existe a necessidade constante de desenvolver novas formas de diferenciação em relação aos restantes concorrentes. Seguindo estes valores, vão ser analisadas oficinas multi-marcas a nível nacional, com o objetivo de realizar uma análise comparativa do conceito que se está a propor ser desenvolvido com a restante concorrência, para que o objetivo de diferenciação seja atingido. Será apresentado o conceito final da Plataforma RedService bem como os principais objetivos traçados na fase inicial do projeto. De seguida, será exemplificado através de uma tabela a análise feita entre a Plataforma RedService e as plataformas dos restantes concorrentes.

### 2.4.1 *Plataformas concorrentes*

Com base numa análise do mercado nacional, identificaram-se as 15 maiores redes de oficinas multimarca que operam no mesmo segmento da RedService (Consumer Guidance Institute Portugal, 2020). Para uma análise detalhada das plataformas concorrentes, serão consideradas as duas maiores redes de oficinas a nível nacional — Eco-oficina e Bosch Car Service — e também aquela que apresentou o maior crescimento em 2021, a MForce. Além disso, no que diz respeito à qualidade, serão incluídas as duas melhores oficinas de 2022, A Oficina e Norauto (Consumer Guidance Institute Portugal, 2022). Assim, a análise das redes concorrentes abrangerá as funcionalidades oferecidas por cada uma delas ao cliente final, visando um comparativo estratégico.

#### 2.4.1.1 *Eco-Oficina*

A rede **Eco-Oficina** é uma rede Oficial criada em 1996, segundo o site oficial tem mais de 200 oficinas em Portugal Continental (Eco-Oficinas, 2022). Através da análise do site oficial, não é possível efetuar um agendamento online através da sua plataforma e não é possível identificar nenhuma aplicação móvel.

#### 2.4.1.2 *Bosh Car Service*

A rede **Bosh Car Service**, segundo o site oficial conta atualmente com 140 oficinas em Portugal (Service, 2022). Através da análise do site, foi possível identificar algumas funcionalidades, como por exemplo, efetuar agendamentos e pedidos de orçamento online, localizar a rede de oficinas, verificar os serviços que tem disponível na sua rede, aconselhamento sobre veículos que podem ser utilizados pelos utilizadores e campanhas disponíveis na sua rede de oficinas.

#### 2.4.1.3 *Mforce*

A rede de oficinas **Mforce**, segundo o site oficial (Mforce, 2022), conta com mais de 80 oficinas espalhadas pelo país. Esta é uma rede de oficinas que se encontra em funcionamento em Portugal à mais de 20 anos. Esta rede oferece ao cliente uma aplicação Web, permitindo ao mesmo efetuar um agendamento online de serviços em veículos nas suas oficinas. Nesta plataforma é ainda possível encontrar conteúdos sobre os serviços prestados nas oficinas MForce, oferecendo ao cliente toda a informação necessária sobre a manutenção que é efetuada por parte dos mecânicos no serviço escolhido.

#### 2.4.1.4 *Norauto*

A rede de oficinas **Norauto** (Norauto, 2022), conta atualmente com 31 oficinas em Portugal. Com base na análise do site oficial foi possível identificar algumas funcionalidades, tais como, agendamento online, orçamentação online de pneus, localização de todas as oficinas, campanhas em vigor, informação de todos os serviços prestados, uma área de cliente, onde é possível adicionar veículos à conta e adicionar serviços externos a rede Norauto mas onde também é possível observar o histórico de manutenções feito na Rede Norauto.

#### 2.4.1.5 *A Oficina*

A rede de oficinas **A oficina**, tem atualmente 108 oficinas em Portugal. Observando o site oficial (automóvel, 2022), foi possível identificar a localização de todas as oficinas e efetuar um pedido de orçamento online.

2.4.1.6 *Quadro comparativo*

Como forma de sintetizar a informação recolhida através da análise de funcionalidade das plataformas concorrentes, foi elaborada a seguinte tabela 1.

	Eco-Oficina	Bosh Car Service	MForce	A Oficina	Norauto
Aplicação Web	x	x	x	x	x
Responsive		x	x	x	x
Aplicação Mobile					
Agendamento Online		x	x		x
Numero de oficinas	200	140	83	107	31
Localização das oficinas	x	x	x	x	x
Informação sobre os serviços		x	x		x
Campanhas em vigor		x			x
Orçamentação online		x		x	x
Área de cliente					x

Tabela 1: Análise das plataforma concorrentes à RedService

2.4.1.7 *Conclusão*

Como base na análise efetuada anteriormente, foi possível perceber o funcionamento das plataformas concorrentes e perceber as suas principais funcionalidades. Com este seguimento, será possível idealizar um conceito RedService que vá responder às necessidades que os clientes já estão habituados a obter na concorrência, mas também criar novas funcionalidades para a RedService se destacar da mesma.

2.4.2 *Conceito RedService*

Analisando o funcionamento de uma oficina RedService e dos principais concorrentes, idealizaram-se novas formas de poder efetuar um agendamento e o modo de interação com a oficina pretendida. Até então, existe a possibilidade de efetuar uma marcação numa oficina RedService através do contacto telefónico, deslocação até à oficina pretendida ou através do site *www.redservice.pt*, com o preenchimento de um formulário. Toda a gestão posterior a este contacto por parte do cliente é efetuada pela oficina. No Conceito de uma Plataforma RedService é idealizada uma plataforma onde o cliente possa continuar a utilizar as formas de contacto anteriores, mas também implementar novos meios, mais rápidos e/ou mais simples de fazer um agendamento. Com a crescente utilização da Internet e especialmente a utilização de *smartphones* (Marktest, 2020), torna-se uma das formas mais eficientes de efetuar

um agendamento numa oficina RedService. Neste seguimento, o desenvolvimento de uma aplicação móvel grátis disponível para todos os utilizadores que possuem um *smartphone* IOS ou Android, sendo estes sistemas operativos os mais utilizados (StatCounter, 2022), vai permitir cativar o maior número de clientes para a rede RedService.

Com a reformulação de todo o conceito da RedService, o site existente vai ser reformulado, tornando-o mais apelativo para o cliente, mostrando as novas funcionalidades que poderão encontrar na plataforma. O sistema de agendamento online será melhorado, tornando-o mais perceptível para a oficina bem como para o cliente.

Após o agendamento por parte do cliente, cada oficina irá gerir a sua agenda, efetuando os seus orçamentos em plataformas próprias ou efetuar a subscrição em outros serviços, deixando esta parte deixada a cargo de cada oficina. Com o desenvolvimento deste conceito, foi planeado um *backoffice*, denominado de *WMS* (*Workshop Management System*), onde é possível cada oficina visualizar a sua agenda e onde vai ser possível orçamentar os pedidos efetuados pelos clientes, através do seu ERP (*Enterprise Resource Planning*) ou de uma plataforma externa o *Portal Leirilís* que estarão integrados no *WMS*. A Plataforma externa, o portal Leirilís, tem objetivo o acesso à área de orçamentos, que cruza informação obtida através do *HaynesPro* e as peças que existentes no Portal de vendas da Leirilís, permitindo logo à oficina saber de forma rápida e eficaz o valor da reparação do veículo do cliente, e ainda exportar estes dados para o *WMS* para ser possível enviar esta informação ao cliente de modo que este possa decidir se pretende efetuar o serviço ou se pretende efetuar ainda alguns ajustes com a oficina.

Após a decisão por parte do cliente, caso esta seja revogada, ainda existirá a possibilidade da oficina alterar o seu orçamento e enviá-lo novamente. Caso esta seja deferida, a marcação ficará no estado de "Aceite", sendo o próximo passo do cliente quando efetuar a deslocação à oficina no dia acordado.

No dia da marcação, a oficina irá receber o veículo nas suas instalações, através de uma receção ativa, que consiste na avaliação do estado do veículo, como, por exemplo, o estado da pintura, amolgadelas na chapa, quantidade de combustível, entre outros pontos que o mecânico considerar relevantes e reportar na receção através do uso de fotografias que serão posteriormente validadas pelo cliente a fim de finalizar a receção do carro na oficina.

Com a receção do veículo concluída, a partir deste momento os mecânicos poderão começar a realizar o trabalho pedido pelo cliente. Através da aplicação os mecânicos

irão poder registrar os tempos de reparação de cada serviço e ainda sugerir novos serviços caso exista necessidade, permitindo ao mecânico tirar uma fotografia para reportar a anomalia ao gerente da oficina e de seguida efetuar um orçamento para enviar ao cliente para ele poder aceitar ou revogar. No final de todos os serviços estarem concluídos, o mecânico irá dar o agendamento como concluído e enviar uma notificação ao cliente em como o seu veículo já se encontra disponível para levantamento. Neste momento, no *WMS*, será possível finalizar todo o processo através da criação de uma fatura recorrendo a um ERP integrado na aplicação RedService, com base nos serviços que foram efetuados no veículo. No final de todo o processo entre o cliente e a oficina o agendamento será dado como finalizado.

Para efeitos de consultas e marcações o cliente terá acesso a uma aplicação Web, onde poderá efetuar o login e obter todos os dados sobre as suas marcações assim como efetuar novas marcações dos seus veículos. Esta aplicação *Web* será uma versão simplificada da versão mobile. Na versão mobile deverá estar disponível para IOS e Android e irá permitir ao cliente efetuar as marcações, consultar o histórico de marcações com mais detalhe, uma interação mais direta com a oficina e acesso exclusivos nas promoções na rede RedService.

#### 2.4.3 *Conclusão*

Atualmente, a análise das oficinas pertencentes à rede RedService evidenciam a ausência de sistemas informáticos e uma grande diversidade nos métodos de trabalho. O desenvolvimento da plataforma visa combater essas lacunas, proporcionando às oficinas uma melhor preparação para enfrentar a concorrência e permitindo o aumento do volume de negócios.

O objetivo desta plataforma é aproximar o cliente da oficina, garantindo que a comunicação entre ambas as partes seja clara, concisa e simples. A padronização dos processos de trabalho e a introdução de ferramentas tecnológicas irão contribuir para a melhoria dos serviços, permitindo a competitividade das oficinas no mercado.

Através dessa solução integrada, as oficinas poderão otimizar suas operações, garantir maior transparência no relacionamento com os clientes e, conseqüentemente, melhorar a qualidade do atendimento e a fidelização dos mesmos.

## ARQUITETURA

Com base no capítulo anterior, vai ser proposto o desenvolvimento de uma arquitetura que tem como objetivo responder a todas as necessidades que foram idealizadas para o desenvolvimento da Plataforma RedService. De seguida, irá ser contextualizada toda a arquitetura da Plataforma RedService, explicando de forma sucinta as funcionalidades que cada servidor integrado na Plataforma RedService vai desempenhar.

### 3.1 ARQUITETURA GERAL DA SOLUÇÃO

Nesta secção vai ser apresentada a arquitetura da solução, com as várias integrações que vão ser necessárias para que as metas estipuladas, possam ser cumpridas. Na figura 3, podemos identificar os vários servidores que estão integrados com a Plataforma RedService, e que efetuam a comunicação entre si, através de *API's*.

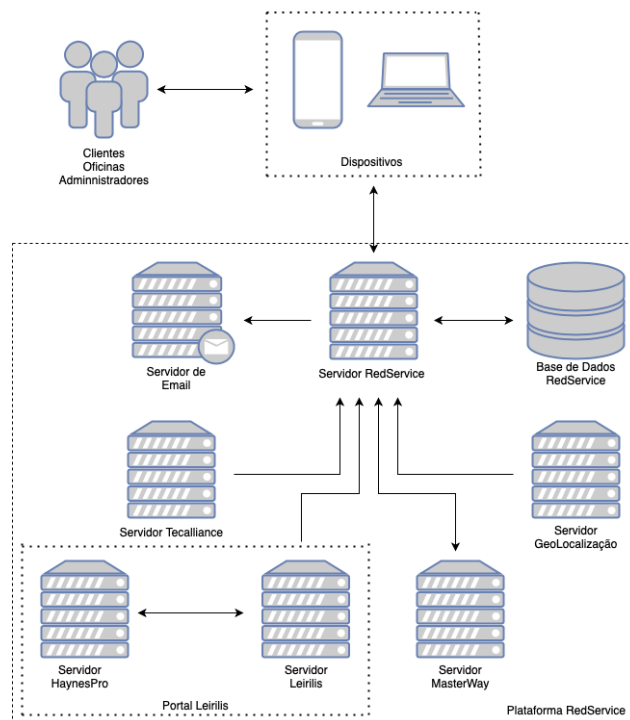


Figura 3: Arquitetura geral da solução

Esta arquitetura visa responder a todas as necessidades identificadas no capítulo anterior, como referido anteriormente e nesta sequência, vão ser contextualizadas todas as funcionalidades das interligações propostas.

### 3.1.1 *Servidores*

Como se pode observar na figura 3, a arquitetura geral da Plataforma RedService integra vários servidores. Já que um servidor é um *software* ou dispositivo que fornece um serviço para outro *software*, ou seja, um servidor recebe e responde a solicitações de outros *softwares*/servidores (Oracle, 2022a). Nesta ótica, a Plataforma RedService vai utilizar vários servidores que serão descritos nas secções abaixo.

#### 3.1.1.1 *Servidor de Email*

Para o desenvolvimento da plataforma RedService, existe a necessidade de utilizar um servidor de email para a comunicação com os clientes, possibilitando o envio de novidades e campanhas que estejam a decorrer na RedService, bem como facilitar a comunicação nos agendamentos online, nomeadamente o envio de orçamentos, propostas de serviços e envio de documentação de faturação. Para responder a esta necessidade vai ser utilizado o servidor *Plesk* (Plesk, 2023), que atualmente já é utilizado para fornecer serviço de email a todas as oficinas RedService.

#### 3.1.1.2 *Servidor TecAllience - TecDoc*

De forma a facilitar a identificação dos veículos por parte dos clientes e das oficinas, vai ser integrado o servidor da *TecAllience* para utilizar o serviço de *TecDoc*. Este serviço vai permitir enviar a matrícula do veículo para o servidor deles e obter toda a identificação completa do veículo. Com o veículo bem identificado, vai ser possível reduzir significativamente a margem de erro na identificação do veículo, permitindo assim uma menor margem de erro na identificação de peças ou avarias que o veículo possa ter.

#### 3.1.1.3 *Servidor HaynesPro*

Através do Portal DpAutomotive - Leirilil, vai ser possível utilizar os recursos da servidor HaynesPro <sup>1</sup>. Este servidor fornece através da identificação do veículo os

---

<sup>1</sup> HaynesPro (<https://www.haynespro.com>)

tempos estimados para efetuar os serviços num veículo, a identificação das referências originais, problemas crónicos, documentação sobre uma determinada peça, sendo este detalhe levado a cada marca e modelo.

#### 3.1.1.4 *Servidor Leirilis*

Como forma de agilizar o processo de orçamentação na Plataforma RedService, vai ser integrado o servidor Leirilis, que vai possibilitar a utilização do Serviço HaynesPro com a orçamentação do Portal DpAutomotive - Leirilis. Este serviço da Plataforma DpAutomotive, vai permitir que a oficina no agendamento do *WMS*, abra diretamente a orçamentação do Portal DpAutomotive com o veículo já identificado, só existindo a necessidade da oficina selecionar o tipo de serviço que deseja efetuar no veículo do cliente e de seguida o portal irá cruzar as referências originais do *HaynesPro* com o stock existente no Portal DpAutomotive, dando a opção de escolha para a oficina, efetuando um orçamento que posteriormente poderá ser enviada para o cliente através do servidor de email.

#### 3.1.1.5 *Servidor MasterWay*

Como referido anteriormente, para uniformizar todo o processo dentro da plataforma RedService, vai ser integrado um **ERP** (*Enterprise Resource Planning*), o MasterWay (Group, 2022). Um **ERP** tem como grandes funcionalidades gerir as atividades comerciais da empresa, como as finanças, compras e gestão dos produtos (Orracle, 2022). Para integração com a Plataforma RedService, vai ser integrada a parte financeira, para permitir através da plataforma gerar orçamentos, folhas de obras e faturas.

#### 3.1.1.6 *Servidor de GeoLocalização*

Para agilizar o processo de atualização de uma oficina, foi integrada uma *API* de geolocalização. Esta irá permitir ao utilizador obter a morada completa através do sistema de coordenadas (Latitude e Longitude).

### 3.1.2 *Dispositivos*

Para que toda a Plataforma RedService, seja utilizada pelo maior número de utilizadores, foi desenvolvida uma arquitetura que seja possível ser acedida de

diversos dispositivos. Esta arquitetura vai permitir que os utilizadores possam utilizar a Plataforma *WEB* e o *WMS* em todos os dispositivos que permitam a utilizam de um *browser*, como por exemplo o Safari, Google Chrome, Microsoft Edge e disponibilidade da aplicação móvel em todos os dispositivos com o sistema operativo Android ou IOS.

### 3.1.3 *Conclusão*

A arquitetura proposta para a Plataforma RedService foi planeada para atender a todas as necessidades levantadas durante o processo de desenvolvimento.

Esta abordagem garante não apenas a automação e otimização de processos internos, como também facilita a interação com os clientes, permitindo à RedService oferecer serviços mais precisos e ágeis, como a orçamentação, identificação de veículos e o envio de documentação. A compatibilidade com diversos dispositivos reforça o carácter acessível e abrangente da Plataforma.

## TECNOLOGIAS

---

Este capítulo tem como objetivo fazer a descrição das tecnologias usadas para a realização do projeto, bem como a comparação com outras tecnologias emergentes no mercado atual. Na secção 4.1 serão abordadas várias tecnologias para o desenvolvimento de *API* (*Application Programming Interfaces*), na secção 4.2 serão abordadas várias tecnologias para o desenvolvimento de aplicações *Web* e na secção 4.3 serão abordadas várias tecnologias para o desenvolvimento de aplicações móveis híbridas.

### 4.1 APPLICATION PROGRAMMING INTERFACE

Uma *API*, acrónimo de *Application Programming Interface*, é uma forma fácil e eficaz de qualquer tipo de aplicação ter acesso remoto de dados, permitindo que os dispositivos troquem informação entre si e/ou controlar outros dispositivos através da Internet. Com o aumento do numero de dispositivos a serem usados diariamente e o aumento do fluxo de dados a serem enviados/recebidos nas *API*'s, estas necessitam de ser bem estruturadas para que seja possível obter uma boa *performance*, ou seja, o tempo desde que o pedido é feito pelo dispositivo até obter uma resposta da *API*. As *API*'s também incentivam a reutilização de código, permitindo implementações mais rápidas, reduzindo, por consequente, o tempo e custo de desenvolvimento (Karlsson, 2021).

De seguida vão ser apresentadas algumas *framework*'s de *API*'s que foram analisadas antes de começar o desenvolvimento da aplicação RedService.

#### 4.1.1 *Express.js*

*Express.js* é uma *framework open-source* que foi lançada em 2010, com o objetivo agilizar e facilitar o desenvolvimento de uma *API* em Node.js. Esta *framework* é executada pelo mecanismo da Google, o JavaScript v8, que se descreve como *Fast, unopinionated, minimalist Web framework for Node.js* (Rápido, sem opinião

, minimalista, framework *Web* para Node.js)(Foundation, 2022). Uma das razões para a popularidade desta framework, está relacionada com o uso da linguagem JavaScript, pois esta permite, que com a mesma linguagem seja desenvolvido o *front-end* e o *banck-end* de uma aplicação, segundo os artigos (Greiff e Johansson, 2019) e (Lund, 2017).

#### 4.1.2 *Spring*

Spring é uma *framework* Java para desenvolver aplicações de alto nível. Esta *framework* oferece uma ampla gama de recursos para o desenvolvimento de vários tipos de aplicações, como aplicações *Web*, aplicações móveis e micros-serviços. Uma das principais vantagens do Spring é sua modularidade, o que permite que os desenvolvedores escolham apenas os componentes de que precisam para sua aplicação. Ela também oferece uma série de recursos úteis, como injeção de dependência, um modelo de segurança e uma framework de acesso a dados.

#### 4.1.3 *WebAPI*

WebAPI é um *framework* para construir serviços HTTP em .NET. É utilizada para desenvolver aplicações RESTful, que usam o protocolo HTTP para obter e enviar dados. A WebAPI é baseada em ASP.NET e fornece uma série de recursos para facilitar o desenvolvimento de *API's*, incluindo uma arquitetura flexível, suportando diversos formatos de dados e a integração com *Entity Framework*, uma framework de acesso a dados para .NET.

#### 4.1.4 *ASP.Net Core*

ASP.net core é uma *framework open-source* que foi lançada em 2016, tornando-se esta nova versão, uma reestruturação da antiga versão ASP.net. ASP.Net Core usa uma abordagem de *multithread* para lidar com muitos pedidos em simultâneo, permitindo garantir níveis de *performance* superiores. Esta *framework* usa a linguagem *c-sharp*, tendo como necessidade a compilação do código. Esta compilação é executado pelo compilador Roslyn (Microsoft, 2022a) convertendo o código em *c-sharp* em assembly com a extensão .dll ou .exe e sendo esta compilação executada pela maquina virtual .Net Core CoreCLR.

#### 4.1.5 Conclusão

Como referido anteriormente, uma das grandes vantagens relativamente ao Express.js é a possibilidade de desenvolver a parte de *front-end* e o *back-end* de uma aplicação ao invés do C-sharp. Segundo um artigo de comparação entre as duas *framework* (Karlsson, 2021), é possível obter melhor performance em regra geral com o *Express.js*, quando o volume de dados é superior. No entanto é de salientar, que a *framework ASP .net core* irá funcionar melhor em aplicações que necessitem do uso intensivo do CPU ou na obtenção de pequenas quantidades de dados da base de dados.

Para a escolha da tecnologia a utilizar para o desenvolvimento do projeto, foi selecionada a *framework ASP .net core*, uma vez que esta tecnologia já é utilizada na empresa e nenhuma das vantagens relativamente à *framework Express.js* foi suficiente para efetuar uma mudança na *API* da empresa. Tornando-se também uma vantagem para a restante equipa de desenvolvimento.

## 4.2 SISTEMA DE BASE DE DADOS

Como parte essencial da estrutura da Plataforma RedService, foi necessário analisar e escolher um motor de base de dados adequado para manipular os dados de forma eficiente e segura. Uma base de dados é uma coleção de dados organizada de maneira estruturada, que pode ser armazenada eletronicamente (Oracle, 2022b). A escolha do motor da base de dados impacta diretamente o desempenho, a escalabilidade e a robustez da aplicação, especialmente no contexto de uma plataforma que gere informações sensíveis e de grande volume.

A seguir, serão analisados os diferentes motores de base de dados disponíveis, avaliando suas capacidades de manipulação de dados através dos métodos CRUD (*Create, Read, Update, Delete*).

### 4.2.1 SQL - Structured Query Language

As bases de dados relacionais baseadas em SQL (*Structured Query Language*) são amplamente utilizadas devido à sua estrutura tabular, que facilita a organização e a consulta de dados. Bases de dados como *MySQL*, *PostgreSQL* e *Microsoft SQL Server* são exemplos de sistemas que utilizam SQL.

Uma das principais vantagens das bases de dados SQL é o suporte de transações ACID (*Atomicidade, Consistência, Isolamento, Durabilidade*), o que garante que todas as operações na base de dados sejam realizadas de maneira confiável e previsível. Além disso, a linguagem SQL é padronizada, o que facilita a transição entre diferentes sistemas de gestão de bases de dados (SGBD).

#### 4.2.2 *Firebird*

O *Firebird* é um sistema de gestão de base de dados relacional que também utiliza SQL como linguagem principal. É conhecido por ser leve, eficiente e por ter um código-fonte aberto (*open-source*), o que o torna uma opção popular para pequenas e médias aplicações (Firebird, 2024). O *Firebird* oferece suporte a transações ACID e é altamente configurável, permitindo às equipas de desenvolvimento a adaptação do sistema às necessidades específicas da aplicação.

Uma das vantagens do *Firebird* é o baixo consumo de recursos, o que o torna ideal para aplicações que precisam ser executadas em ambientes com limitações de hardware. Além disso, ele oferece recursos como os *triggers* e *stored procedures*, que podem ser utilizados para otimizar a lógica de negócios no lado do servidor.

#### 4.2.3 *MongoDB*

O MongoDB é uma base de dados *NoSQL*, orientada a documentos, que se destaca pela sua flexibilidade e capacidade de escalar horizontalmente. Diferente das bases de dados relacionais, o MongoDB armazena dados em documentos no formato *BSON* (versão binária do *JSON*), o que permite armazenar estruturas de dados complexas e não estruturadas de forma mais eficiente (mongodb, 2024).

A flexibilidade do MongoDB em lidar com estruturas dinâmicas torna-o uma escolha ideal para aplicações que exigem mudanças frequentes na estrutura dos dados ou que seja necessário armazenar grandes volumes dos mesmos. Além disso, o MongoDB oferece suporte a consultas *ad hoc*, indexação poderosa e agregações, o que permite a manipulação e a análise de dados de forma eficiente.

#### 4.2.4 Conclusão

A escolha do motor de base de dados para a Plataforma RedService foi baseada em critérios como desempenho, escalabilidade, integridade dos dados e facilidade de manutenção. Cada um dos sistemas analisados apresenta características específicas que os tornam adequados para diferentes cenários.

No caso da Plataforma RedService, dado que os tipos de dados não serão alterados com muita frequência, o MongoDB foi excluído como opção, pois sua flexibilidade em lidar com estruturas dinâmicas não era uma necessidade para o projeto. Entre as alternativas restantes, o *Firebird* e o *Microsoft SQL Server*, optou-se pelo *Microsoft SQL Server*. Essa escolha foi motivada principalmente pelo fato de que o *Microsoft SQL Server* já ser amplamente utilizado em outras aplicações desenvolvidas pela empresa requerente do projeto, o que facilita a integração e a manutenção dentro do ecossistema tecnológico existente.

### 4.3 APLICAÇÃO WEB

Uma aplicação *Web* é uma aplicação que é desenvolvida para ser executada através de um *browser*. Hoje em dia uma aplicação *Web* tem como "requisitos" mínimos ser *responsive*, isto quer dizer que as páginas *Web* tem de se redimensionar consoantes os tipos de ecrãs. Este tipo de aplicação tem como vantagem, não ser necessária a instalação, só existindo a necessidade de um *browser*, como por exemplo o Google Chrome, Microsoft Edge ou Safari (AMA - Agência para a Modernização Administrativa, 2022).

Nesta secção vão ser analisadas várias *frameworks* para o desenvolvimento de aplicações *Web* e no final,

#### 4.3.1 ASP.NET

ASP.NET é uma *framework open-source* desenvolvida pela Microsoft para o desenvolvimento de aplicações *Web* e serviços em .NET (Microsoft, 2023), baseia-se na utilização de C# (*c-sharp*) e F# (*F-sharp*) e utiliza um modelo MVC (*Model, View, Controller*). A *framework* é multi-plataforma, ou seja, as suas aplicações podem correr em MacOS, Windows, Linux e Docker.

#### 4.3.2 *Vue.js*

Vue é uma *framework JavaScript* para o desenvolvimento de interfaces para os utilizadores (páginas *Web*), e baseia-se na utilização de HTML, CSS e *JavaScript* e fornece um modelo de programação baseada em componentes para garantir o desenvolvimento de interfaces com eficiência, tornando-se esta uma estrutura progressiva (Vue, 2022).

#### 4.3.3 *React.js*

React.js é uma tecnologia *JavaScript* para o desenvolvimento de aplicações *Web*. Esta tecnologia baseia-se na utilização de HTML, CSS e *JavaScript* para desenvolver componentes encapsulados que gerem o seu próprio estado, permitindo desenvolver interfaces mais complexas através de vários componentes simples. O React pode ser renderizado no servidor usando Node e no desenvolvimento de aplicações móveis através do React Native (Meta Platforms, 2022).

#### 4.3.4 *Conclusão*

As tecnologias descritas anteriormente têm bastantes semelhanças, ambas funcionam com componentes que se gerem automaticamente, permitindo garantir um nível de performance superior, pois os componentes só renderizam quando existem alterações no seu estado. Outra particularidade é que ambas as tecnologias basearem-se na utilização de HTML, CSS e *JavaScript*.

A escolha feita para o desenvolvimento deste projeto foi React.js, uma vez que ambas as tecnologias são muito semelhantes e uma vez que irá ser desenvolvida uma aplicação móvel e o React tem a componentes do desenvolvimentos de aplicações híbridas, optou-se por manter todo o ecossistema da aplicação RedService em React.

### 4.4 APLICAÇÃO MÓVEL

Com o crescente aumento da utilização de dispositivos móveis pela população em geral (Marktest, 2020), procedeu-se à análise, de quais os sistemas operativos mais utilizados, dos existentes no mercado. Segundo a statcounter (statcounter, 2022),

cerca de 99% dos dispositivos móveis utilizam os sistemas operativos Android e IOS. De forma a satisfazer esta necessidade, pretende-se desenvolver uma aplicação que possa ser utilizada num dispositivo Android e IOS para ser possível chegar ao maior número de utilizadores. Para isso analisou-se a opção de fazer uma aplicação híbrida, que num único desenvolvimento pode ser compilado para várias plataformas, ou nativa que o código desenvolvido dá unicamente para a plataforma que foi desenvolvida. Face ao tempo de desenvolvimento, entre uma aplicação híbrida e nativa, optou-se por desenvolver uma aplicação híbrida porque tem a grande vantagem do ciclo de desenvolvimento ser substancialmente mais curto e todas as alterações a serem efetuadas ao longo do tempo ficarão logo disponíveis para as várias plataformas.

De seguida, vão ser apresentadas várias tecnologias de aplicações híbridas.

#### 4.4.1 *Apache Cordova*

Apache Cordova é uma *framework open-source* de desenvolvimento de aplicações híbridas em Android, IOS e *Web*. Esta *framework* utiliza as tecnologias padrão de desenvolvimento *Web*, como o HTML, CSS e *JavaScript*, permite a partilha de código entre as várias plataforma, existe a possibilidade de utilizar *plugins* para a utilização de componente nativos das plataformas e ainda a possibilidade de utilizar *core plugins* que permitem a utilização de recursos do dispositivo, como bateria, câmara, contatos (Software, 2023).

#### 4.4.2 *Xamarin*

Xamarin é uma *framework open-source* de desenvolvimento de aplicações para IOS, Android e Windows em .NET. Esta *framework* permite o desenvolvimento de aplicações híbridas, partilhando 90% do código entre as várias plataformas, mantendo o desempenho, aparência e os comportamentos nativos em cada plataforma (Microsoft, 2022b). Esta *framework* ainda tem a particularidade de lidar automaticamente com tarefas como alocação de memória, gestão do "lixo" em memória e interoperabilidade com plataformas subjacentes.

#### 4.4.3 *React Native*

React Native é uma *framework* de desenvolvimento de aplicações híbridas desenvolvida pela empresa Meta.inc, que permite a criação de uma aplicação nativa de IOS e Android, ou seja *cross-platform*. Nesta *framework* grande parte do código pode ser partilhada entre as várias plataformas, mesmo que seja necessário algum código específico para uma das plataformas. Além desta particularidade, esta *framework* é *open-source* e permite que a comunidade de programação contribua para o seu desenvolvimento (Hansson e Vidhall, 2016).

#### 4.4.4 *Flutter*

Flutter é uma *framework open-source* de desenvolvimento de aplicações híbridas desenvolvida pela Google. O Flutter é desenvolvido através da linguagem de programação Dart (orientada a objetos) que pode ser compilada para código ARM nativo de 32 e 64 bits para Android e IOS e opcionalmente compilado para JavaScript para a versão *Web* (Boukhary e Colmenares, 2019). O componente mais importante desta *framework* são os *Widgets*. Estes controlam a visualização dos ecrãs e as ações do utilizador, tornando-se crucial que estes tenham um desempenho rápido, incluindo a renderização e animação. No caso do React Native a *framework* utiliza os *widgets* nativos, enquanto no Flutter estes são desenvolvidos pela equipa Flutter da Google, permitindo que estes sejam mais personalizáveis e extensíveis, no entanto torna a aplicação mais pesada (Wu, 2018).

#### 4.4.5 *Conclusão*

As *frameworks* descritas previamente preenchem os objetivos propostos, o desenvolvimento de uma aplicação híbrida de Android e IOS, mas quando são analisadas mais ao pormenor, conseguimos identificar algumas vantagens e desvantagens em ambas. No caso do Flutter foi possível constatar a sua melhor performance relativamente ao React Native, embora o desenvolvimento em Flutter iria requerer a aprendizagem da linguagem Dart, com isto tornando-se assim também mais rápido o início do desenvolvimento em React Native. Também é possível constatar a existência de uma grande comunidade que poderá ajudar na resolução de alguns problemas durante o desenvolvimento da aplicação. Como referido anteriormente um dos pontos para a

escolha da *framework* React Native para o desenvolvimento da aplicação híbrida foi a possibilidade de manter todo o ecossistema da aplicação RedService em React.



## METODOLOGIA

---

### 5.1 METODOLOGIAS DE TRABALHO

No desenvolvimento do projeto, foram adotadas metodologias ágeis. Embora a equipa seja pequena, estas metodologias trazem diversas vantagens, como a possibilidade das entregas incrementais, que permitem obter *feedback* contínuo do cliente. Isto contribui para uma melhoria contínua do projeto e garante que as funcionalidades sejam analisadas e ajustadas de acordo com as expectativas das oficinas.

As metodologias ágeis adotadas no desenvolvimento da aplicação foram Kanban e Scrum. Em seguida é explicada a forma como estas duas metodologias definiram os padrões de desenvolvimento e como foram utilizadas.

#### 5.1.1 *Kanban*

O *Kanban* é uma metodologia ágil focada na melhoria contínua do fluxo de trabalho (Rehkopf, 2024). Esta metodologia utiliza um quadro visual para acompanhar o progresso das tarefas, divididas em categorias "To do", "In Progress", "To Validate" e "Complete". Pode ser observado na figura 4 uma parte do *kanban* utilizado no desenvolvimento da Plataforma RedService. O principal objetivo do Kanban é garantir que o fluxo de trabalho seja contínuo e que a equipa de desenvolvimento esteja sempre ciente das tarefas em progresso e das próximas a serem realizadas.

No projeto, o uso do quadro *Kanban* facilitou o controlo das tarefas a serem desenvolvidas, além de permitir a gestão eficaz de novas tarefas que surgiram à medida que o desenvolvimento avançava. Isso possibilitou a visualização clara do estado atual do trabalho, garantindo que todos os intervenientes no projeto tivessem uma visão global do progresso e das mudanças nas prioridades, conforme as novas necessidades iam sendo identificadas.

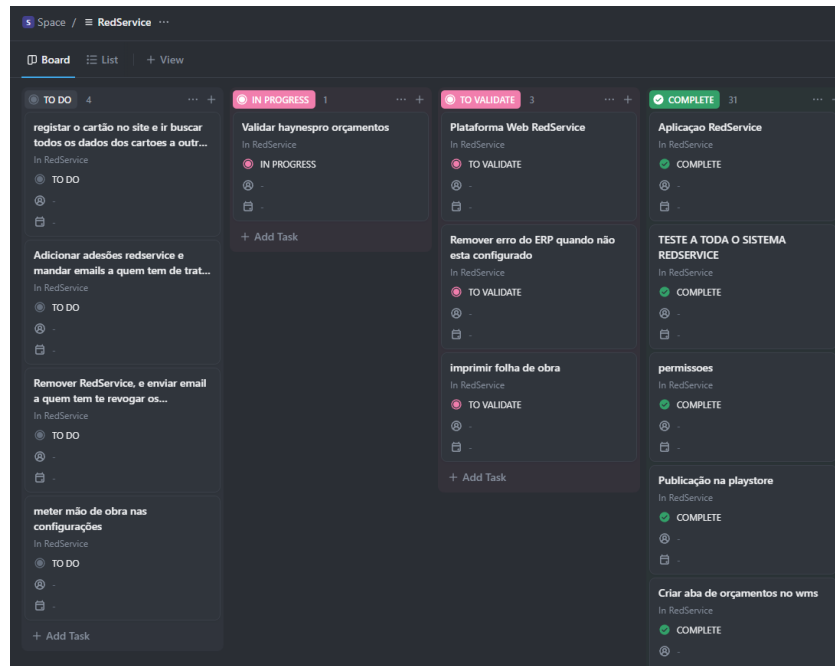


Figura 4: Excerto de quadro *Kanban* utilizado no desenvolvimento da Plataforma RedService

### 5.1.2 *Scrum*

*Scrum* é outra metodologia ágil amplamente utilizada para gerir projetos complexos. Baseia-se em ciclos curtos de trabalho, conhecidos como *sprints*, que normalmente duram entre duas a quatro semanas (Meet Ken Schwaber, 2024). No final de cada *sprint*, há uma entrega incremental do *software* que deve ser potencialmente utilizável.

No desenvolvimento do projeto, o *Scrum* proporcionou uma estrutura clara para a execução dos *sprints*, com reuniões a cada quinze dias, realizadas com o CEO da Leirilil onde era monitorizado o progresso e identificados obstáculos. Estas *sprints* foram realizadas durante todo o processo de desenvolvimento, aproximadamente 25 no total. As *reviews* e retrospectivas realizadas no final de cada *sprint*, geralmente a cada duas semanas, ajudaram a refletir sobre o trabalho concluído, sendo ajustado o planeamento do próximo ciclo, como se pode observar na figura 5, o gráfico de Gantt onde eram definidas as tarefas a realizar. Além disso, as entregas incrementais permitiram uma comunicação constante com o CEO da Leirilil e com algumas oficinas RedService que faziam alguns testes na plataforma, garantindo que as funcionalidades estavam de acordo com as expectativas.





## IMPLEMENTAÇÃO

---

Nesta secção vai ser descrita a implementação técnica e detalhada da Plataforma RedService. Esta apresentação vai começar pela descrição geral da solução, onde vai ser explicado o fluxo de funcionamento de toda a Plataforma. De seguida, vai ser feita a análise de requisitos funcionais e não-funcionais da Plataforma.

### 6.1 ANÁLISE DE REQUISITOS

Com base no estudo prévio realizado em torno de todas as oficinas concorrentes à Plataforma RedService e olhando diretamente para a implementação do modelo de negócio e aplicações já existentes na Plataforma RedService, foi possível estipular um conjunto de requisitos funcionais para a Plataforma RedService.

#### 6.1.1 *Requisitos funcionais*

Nesta secção vão ser descritos os requisitos funcionais para cada uma das componentes propostas para desenvolvimento.

#### 6.1.2 *Aplicações móveis - Permissões de cliente*

- O cliente deve marcar agendamentos online com ou sem login.
- O cliente pode ter informações sobre o cartão RedService.
- O cliente tem de conseguir visualizar todas as oficinas existentes e as que aceitam agendamento.
- O cliente tem de estar informado sobre as campanhas em vigor.
- O cliente tem de ter a possibilidade de criar uma conta para poder associar todos os dados (veículos, agendamento, marcações, dados pessoais) a sua conta.

- O cliente tem de conseguir adicionar, editar e eliminar os seus veículos na sua conta.
- O cliente tem a possibilidade de criar, editar e eliminar a sua conta e dados da Plataforma RedService.
- O cliente deve conseguir visualizar o histórico das manutenções efetuadas no seu veículo.
- O cliente deve conseguir visualizar os agendamentos efetuados, e os seus detalhes (veículo, estado do agendamento, oficina, e serviços pedidos).
- O cliente deve poder adicionar revisões que foram efetuados noutras oficinas ao seu histórico de manutenções de um veículo específico, podendo estes ser eliminados posteriormente.
- O cliente tem a possibilidade de enviar os dados e histórico do seu veículo para outro utilizador com conta na aplicação RedService, através de *QRCode* ou email, passando o outro utilizador a ter o veículo na sua conta e saindo os dados do veículo do primeiro utilizador.
- O cliente deve ter a possibilidade de importar um veículo para a sua conta.
- O cliente deve ter a possibilidade de escolher e/ou editar a sua oficina principal.
- O cliente deve ter a possibilidade de visualizar as informação Legal sobre a RedService.
- O cliente deve poder visualizar as notificações que são enviadas.

### 6.1.3 *Aplicações móveis - Permissões de oficina*

- Os Roles da oficina, devem ter acesso a uma área na aplicação dos agendamento aceites pela oficina, os que se encontram já na oficina em serviço e os que já se encontram concluídos, mas que o cliente ainda não fez o levantamento do mesmo.
- Os Roles da oficina, devem poder visualizar os dados do agendamento.
- Os Roles da oficina, devem poder selecionar um agendamento e efetuar uma receção ativa (detalhar o estado do carro quando chega a oficina).
- Os Roles da oficina, quando o veículo já se encontra na oficina, caso necessário, devem poder sugerir ao cliente reparações que se encontrem necessárias a fazer no veículo, exigindo uma validação por parte do Gestor da oficina antes de

ser enviado ao cliente um orçamento, que deverá ser aprovado ou reprovado posteriormente pelo cliente.

- Os Roles da oficina, quando tem um agendamento selecionado, devem poder visualizar todos os serviços que devem ser efetuados, podendo selecionar o serviço e iniciar um temporizador, efetuar uma pausa no serviço e dar o serviço como concluído.

#### 6.1.4 *Workshop Management System*

- O Administrador tem acesso a gráficos de dados de toda a aplicação.
- O Administrador tem a possibilidade de criar, editar e eliminar oficinas.
- O Administrador tem a possibilidade de criar editar e eliminar utilizadores com vários tipos de Roles.
- O Administrador tem a possibilidade de visualizar todos os agendamentos dentro da Plataforma RedService.
- O Administrador tem acesso a edição do *Front-end* da Plataforma RedService (site do cliente).
- O Administrador deve gerir os *banners* da Plataforma.
- O Administrador pode criar, editar e eliminar serviços para toda a Plataforma.
- O Administrador pode adicionar e remover documentação para as oficinas, sendo esta documentação geral ou específica para cada oficina.
- O Administrador gere os pedidos para aderir as oficinas RedService, podendo dar como tratado cada um dos pedidos.
- O Administrador deve ter uma pesquisa avançada de todos os agendamentos efetuados na Plataforma RedService, bem como todos os dados desse agendamento.
- O Gestor da oficina, vai poder criar, editar e eliminar os utilizadores da sua oficina (Gestores, Oficina, mecânicos).
- O Gestor da oficina, vai poder editar todos os dados sobre a sua oficina na Plataforma RedService.

- O Gestor da oficina, vai ter acesso a uma área de definições, onde poderá adicionar e editar os acessos ao Portal Leirilis, configurar o acesso ao email RedService e configurar o sistema de faturação MasterWay<sup>1</sup>.
- O Gestor da oficina, vai poder aceder ao ao seu email RedService.
- O Gestor da oficina e o mecânico, vão ter acesso aos gráficos correspondentes à sua oficina.
- O Gestor da oficina e o mecânico, vão ter acesso unicamente aos agendamentos da sua oficina, com a possibilidade de visualizar os detalhes de cada um dos agendamentos.
- O Gestor da oficina e o mecânico, vão poder consultar a documentação disponibilizada pela Leirilis.
- O Gestor da oficina e o mecânico, vão ter acesso a uma área de pesquisa, onde poderão efetuar uma pesquisa sobre todos os agendamento efetuados na sua oficina.
- O Gestor da oficina e o mecânico, nos agendamentos, tem a possibilidade de criar um agendamento pelo cliente.
- O Gestor da oficina e o mecânico, nos agendamentos, tem a possibilidade de editar um agendamento que se encontre num estado de "Pendente", podendo alterar a data, hora, serviços que o cliente agendou e orçar os serviços.
- O Gestor da oficina e o mecânico, nos agendamentos, pode alterar o estado dos agendamentos.
- O Gestor da oficina e o mecânico, nos agendamentos, quando o agendamento se encontra concluído, pode mandar gerar a fatura no MasterWay através dos serviços do agendamento.
- O Gestor da oficina e o mecânico, nos agendamentos, pode cancelar um agendamento quando este se encontra pendente de aceitação por parte do cliente.

#### 6.1.5 Site RedService

- O Utilizador sem autenticação tem a possibilidade de visualizar os serviços prestados pelas oficinas RedService, aconselhamentos sobre os veículos, campanhas em vigor nas rede de oficinas, acordos, visualizar num mapa as oficinas

---

<sup>1</sup> MasterWay (<https://masterway.net/>)

que existem e as que estão a receber agendamentos online e a informação sobre o seu horário de funcionamento.

- O Utilizador sem autenticação, tem a possibilidade de efetuar um registo na Plataforma RedService.
- O Utilizador sem autenticação, pode efetuar o login no site da RedService à área de cliente.
- O Utilizador autenticado, na área de cliente, pode visualizar todos os veículos associados a sua conta, adicionar novos veículos, remover o veículo da sua conta e ver todos os serviços que foram efetuados no seu veículo.
- O Utilizador autenticado, na área de cliente, pode visualizar todos os agendamentos associados a sua conta e os detalhes de cada agendamento.
- O Utilizador autenticado na área de cliente, pode visualizar os dados da sua conta, permitindo ainda a edição dos mesmos.
- O Utilizador autenticado na área de cliente, com o Role de Gestor de Oficina, Mecânico ou Oficina, tem ainda um menu de redirecionamento para o WMS.

#### 6.1.6 *Requisitos não-funcionais*

- A Plataforma tem de ser *responsive*.
- A Plataforma deverá comunicar com uma base de dados SQL
- A Plataforma deverá correr em todos os *Browsers*.
- A Plataforma deve respeitar o RGPD (Regulamento Geral de Proteção de Dados).
- A Plataforma tem de ser *user friendly*.
- A Plataforma deverá ter vários privilégios de acesso à informação (*Roles*).
- A aplicação móvel deve correr na Versão 13.0 ou superior em IOS e versão 10 ou superior em Android.

## 6.2 FERRAMENTAS UTILIZADAS

Para o desenvolvimento do projeto foi necessário escolher um ambiente de desenvolvimento que pudesse responder a todas as linguagens que me propus a desenvolver,

uma vez que existem várias linguagens a serem desenvolvidas. Neste contexto, para o desenvolvimento da plataforma utilizei o IDE (*Integrated development environment*) Microsoft Visual Studio Professional<sup>2</sup>, uma vez que este garantia-me a utilização do C# para o desenvolvimento da API, bem como as aplicações móveis, com a utilização do *JavaScript* e *TypeScript*. Para a compilação das aplicações móveis, foi necessário recorrer ao IDE Android Studio<sup>3</sup>, para compilar a aplicação nativa em Android e respetivamente ao XCode<sup>4</sup> para compilar a aplicação nativa para IOS. Para a implementação da Base de dados em SQL, foi utilizado o *software Microsoft SQL Server*<sup>5</sup>.

De forma a garantir o registo de todas as alterações efetuadas no projeto, foi utilizada o sistema de controlo de versões Git (Git, 2023). O serviço utilizado foi o GitHub, onde foram criados 3 repositórios, um repositório para cada uma das aplicações desenvolvidas. Ambos os repositórios são privados e não acessíveis por terceiros, uma vez que toda a Plataforma RedService é propriedade da Leirilil. Na figura 6 é possível observar uma amostra da utilização do GitHub e os *commits* efetuados em uma das aplicações.

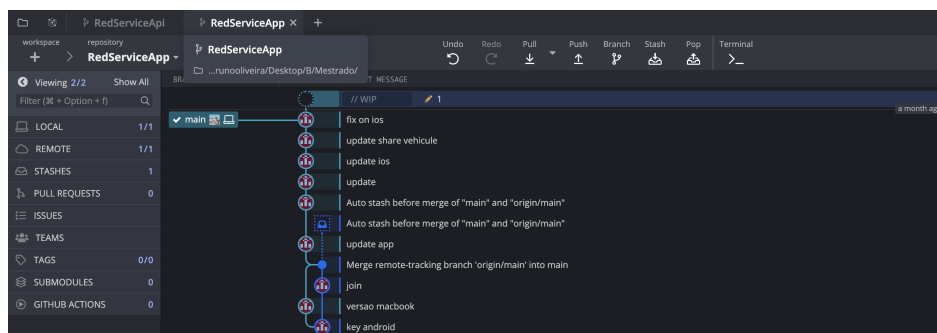


Figura 6: GitHub

Para organizar o trabalho e as tarefas propostas para desenvolvimento ao longo do projeto, utilizei a ferramenta *ClickUp*<sup>6</sup>, que já utilizava para organização de tarefas durante o horário laboral de trabalho. Esta ferramenta permite planejar as tarefas visualizar todas as tarefas e as várias fases em que se encontra cada uma delas, através dos quadros "TO DO", "IN PROGRESS", "TO VALIDATE" e "COMPLETE", como se pode observar na figura 6.

<sup>2</sup> Microsoft Visual Studio Professional (<https://visualstudio.microsoft.com/vs/professional/>)

<sup>3</sup> Android Studio (<https://developer.android.com/studio>)

<sup>4</sup> Xcode (<https://developer.apple.com/xcode/>)

<sup>5</sup> Microsoft SQL Server (<https://www.microsoft.com/pt-br/sql-server>)

<sup>6</sup> ClickUp (<https://clickup.com/>)

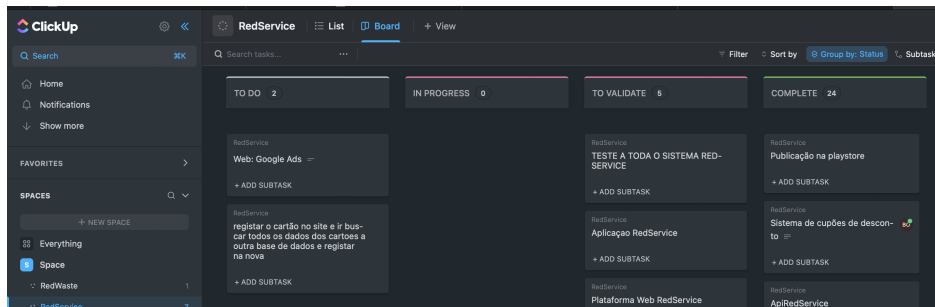


Figura 7: Quadro Kanban

No desenvolvimento dos protótipos para a Plataforma, foi utilizada a plataforma online *Moqups*<sup>7</sup>, que permitiu fazer o design de todas as aplicações desenvolvidas.

Para o desenvolvimento da Plataforma RedService foram estas as ferramentas utilizadas.

### 6.3 PROTÓTIPOS

Na primeira fase do desenvolvimento da Plataforma RedService, passou pela conceção dos ecrã para as várias aplicações a serem desenvolvidas, de forma a acomodar todos os requisitos e funcionalidades definidas na análise de requisitos. No desenvolvimento dos vários protótipos, foi necessário garantir a melhor experiência do utilizador, nesse contexto, teve-se em consideração a necessidade de criar uma lógica na organização das aplicações e a simplicidade na sua utilização.

A aplicação *web* foi a primeira a ser desenvolvida, denominada de *WMS - Workshop Management System*, destina-se à parte administrativa da plataforma RedService. Aqui vão estar concentradas muitas funcionalidades para gerir a Plataforma, existindo a necessidade de criar vários menus, dessa forma desenvolveu-se um protótipo com um menu lateral, tornando-se simples a navegação entre as várias paginas com as suas diversas funcionalidades, como se pode observar na figura 8.

Na figura 8 é apresentada a página *WMS*, que é considerada a página inicial da Plataforma *WMS*. Aqui são apresentados os vários gráficos sobre a plataforma, que podem variar consoante o tipo de utilizador que efetua o login. Nesta figura, com as permissões de administrador da plataforma, é apresentado, como exemplificado na figura, o gráfico de utilizadores novos registados na Plataforma RedService, o total de agendamento efetuados em cada oficina, os agendamentos efetuados em

<sup>7</sup> Moqups (<https://moqups.com/>)

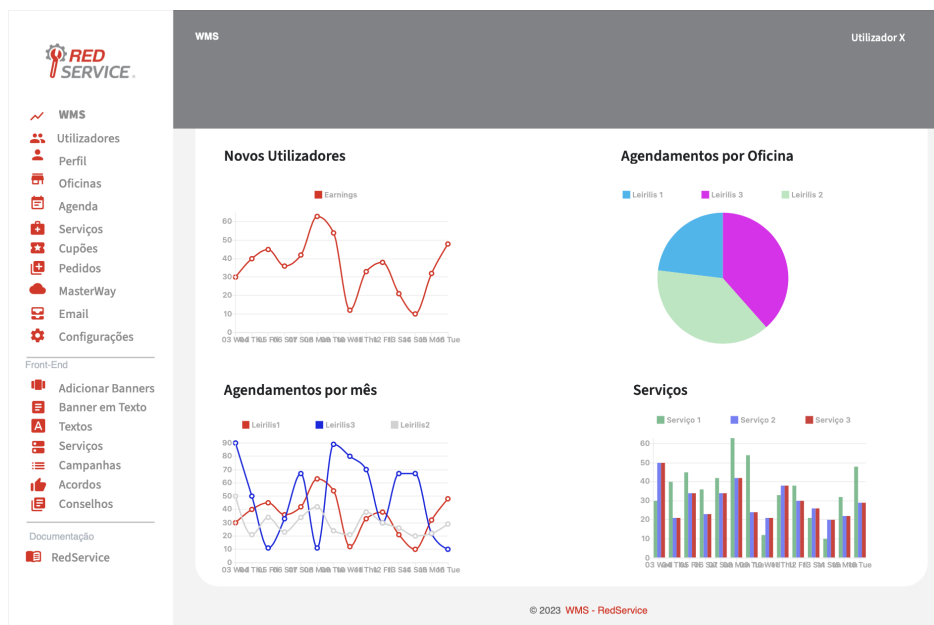


Figura 8: WMS - Página principal

cada uma das oficinas no mês corrente e por fim um gráfico com o volume de tipos de serviços, efetuado em todas as oficinas.

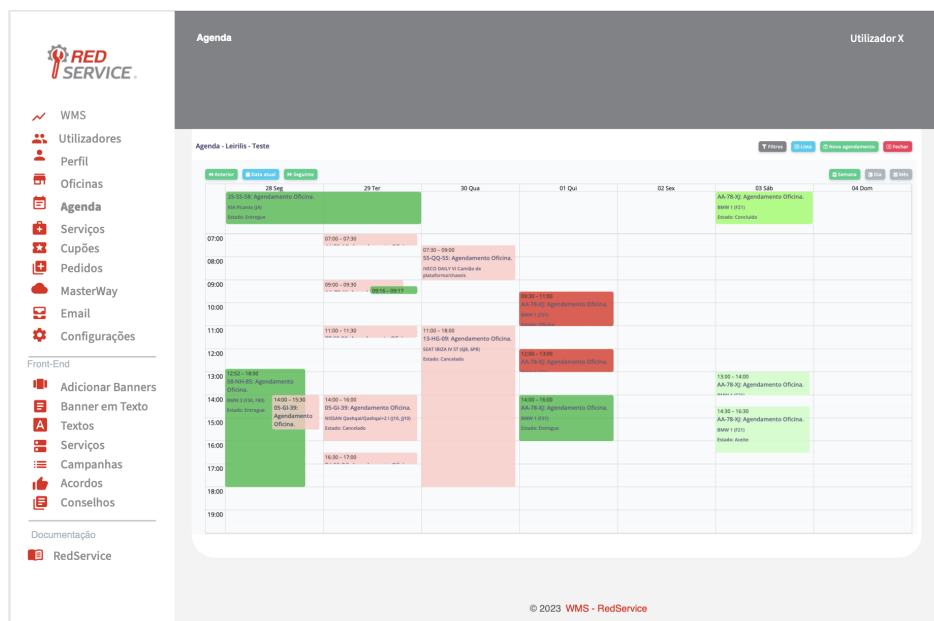


Figura 9: WMS - Agendamentos

Na figura 9 é apresentada a página "Agenda". Nesta página vai ser possível aos utilizadores autenticados e com permissões adequadas, visualizar todos os seus agendamentos, ou seja, cada oficina vai poder visualizar, editar, criar e/ou eliminar os seus agendamentos. O utilizador dentro de cada agendamento, ainda vai ter

várias funcionalidades, tais como, a edição de dados (datas do agendamento, dados do veículo, colaborador, observações, serviços) e vai poder criar editar ou eliminar os serviços pedidos pelo cliente, ajustando da melhor forma o serviço pedido às necessidades do veículo. Dentro de cada serviço pedido pelo cliente vai ser possível adicionar artigos (peças necessárias para a manutenção do veículo, mão de obra, entre outros valores definidos pela oficina) com o respetivo preço, com a finalidade de gerar um orçamento.

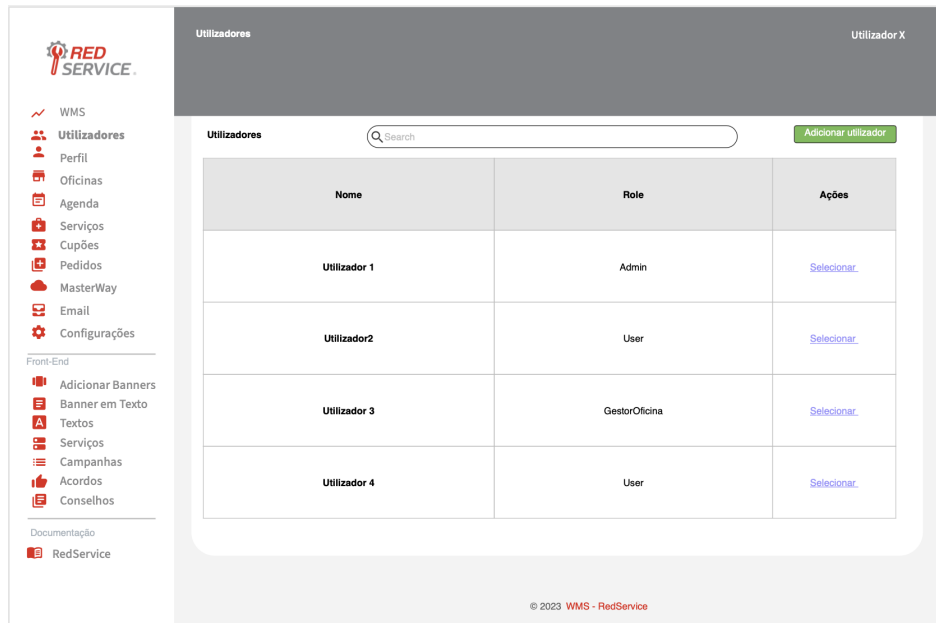


Figura 10: WMS - Utilizadores

Na figura 10 é apresentada a página "Utilizadores". Nesta página vai ser possível ao administrador visualizar todas as contas criadas na plataforma, e no caso dos gestores de oficina, vai ser possível visualizar todas as contas dos seu funcionários (gestores e mecânicos), ambos vão ter a possibilidade de criar, editar e eliminar essas contas.

Perfil Utilizador X

Conta

Informações do utilizador

Role: Admin Oficina: Leiria

Nome: Bruno Apellido: Oliveira Data de nascimento: 15/12/1995

Email: Bruno.oliveira@leirite.com Telemovel: 912312567 Contribuinte: 123456789

Informações de contacto

Morada: Rua Paulo VI

Cidade: Leiria País: Portugal Código Postal: 2400-147

Veículos associados

Marca	Modelo	Ações
Brw	Serie 1	Selecionar

© 2023 WMS - RedService

Figura 11: WMS - Formulário

Na figura 11 é apresentada a página "Perfil". Nesta página vai ser possível ao administrador visualizar, editar, eliminar, fazer a reposição da *password*, adicionar e eliminar veículos associados à conta de cada utilizador. Esta página é reutilizável, ou seja, consoante o tipo de "Role" do utilizador autenticado, vão ser disponibilizadas ou removidas funcionalidades ao utilizador.

Como podemos visualizar nas figuras 9, 10 e 11, existe uma barra lateral do lado esquerdo, que corresponde ao menu do WMS. Algumas páginas deste menu já foram explicadas anteriormente, não havendo a necessidade de descrevê-las novamente. As páginas restantes seguem uma estrutura semelhante em termos de *design* e funcionalidades. As únicas diferenças dizem respeito ao contexto e aos dados apresentados.

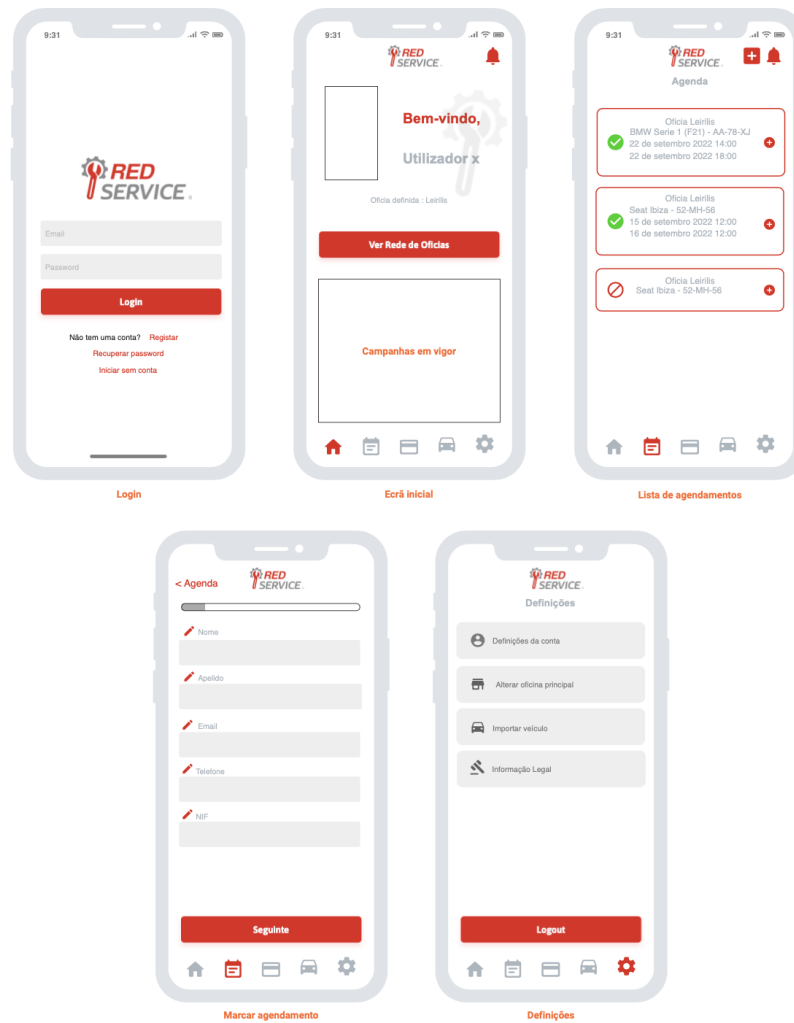


Figura 12: *Mockups* das aplicações móveis

Na figura 12, pode-se visualizar o protótipo das aplicações móveis, tornando-se cada uma das imagens, *Mockups*, referentes aos principais ecrãs da aplicação.

O primeiro *Mockup* é o *design* do login da aplicação, onde também vai ser possível aceder ao ecrã de *login*, recuperação da *password* e a opção de se seguir para o ecrã principal da aplicação sem o recurso ao *login*, sendo esta parte limitada nas funcionalidades disponíveis.

No segundo *Mockup*, correspondente ao ecrã principal da aplicação, vai ser possível visualizar na parte superior, informação sobre as oficinas e no caso do *login*, informação da conta; a meio do ecrã existe um botão que vai redirecionar o utilizador para uma página onde poderá visualizar todas as oficinas num mapa; na parte inferior do ecrã vão aparecer *banners* com publicidade sobre a Rede de Oficinas.

No terceiro *Mockup*, o ecrã da lista de agendamento, que só está disponível para os utilizadores autenticados, vai ser possível ao utilizador ver uma lista com o estado dos agendamentos efetuados, bem como ao pressionar o agendamento e visualizar todos os detalhes sobre esse agendamento.

No quarto *Mockup*, o ecrã para efetuar um agendamento, pode-se visualizar alguns dos *inputs* que o utilizador vai ter de preencher para efetuar o agendamento.

No quinto *Mockup*, o ecrã com o menu de definições, pode-se visualizar os vários sub-menus, que variam se o utilizador estiver autenticado ou não.

### 6.4 DESENVOLVIMENTO

Com o início do desenvolvimento da plataforma proposta, é necessário garantir que todos os objetivos delineados anteriormente são cumpridos, deste modo, iniciou-se o desenvolvimento da base de dados em SQL com base nos critérios definidos anteriormente, de seguida foi iniciado o desenvolvimento da *API* em .Net paralelamente com o desenvolvimento da Aplicação Web - *WMS*, permitindo assim garantir que a manipulação dos dados da aplicação seja feita através da interface WEB recorrendo aos métodos CRUD (*Create, Read, Update, Delete*) desenvolvidos na aplicação. De seguida, foi dado início ao desenvolvimento da aplicação móvel.

#### 6.4.1 Base de dados

Com base nos objetivos propostos para o projeto, foi necessário desenvolver uma base de dados que responda a todas as necessidades da Plataforma RedService.

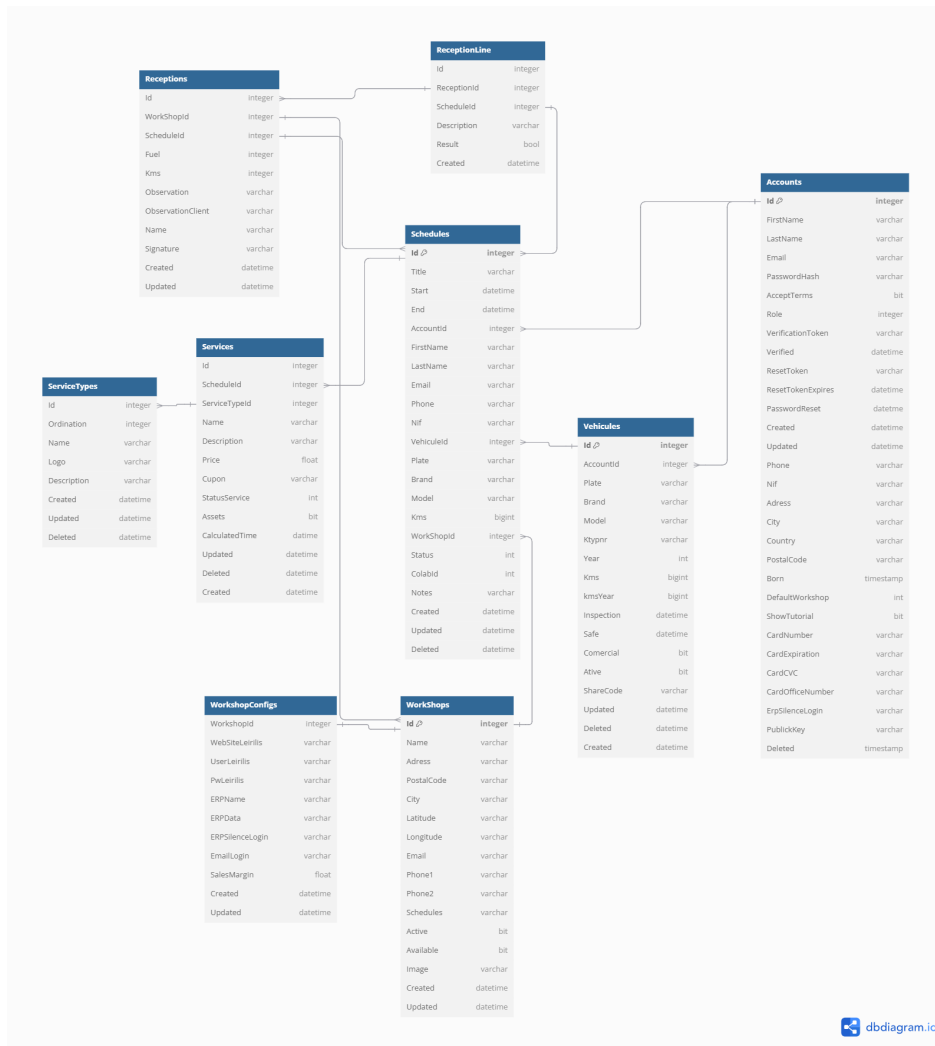


Figura 13: Modelo físico da Base de dados RedService

Na figura 13, é possível visualizar toda a arquitetura geral da base de dados RedService. Nesta arquitetura é importante destacar algumas das tabelas, tornando-as os pontos principais para garantir o funcionamento da Plataforma RedService, desta forma de seguida vão ser explicadas as suas características.

Tabelas:

- **"Accounts"** - Nesta tabela, vão ser guardados os dados de todos os utilizadores da aplicação, tendo como uma das características o campo *email* que deverá ser único dentro da Plataforma WMS, o campo *Role*, que permite dar privilégios de acesso a utilizadores da Plataforma, o campo *DefaultWorkshop* que dá a possibilidade de definir uma oficina principal associada à conta. Os restantes campos, não menos importantes, terão um papel de auxílio durante a utilização da Plataforma RedService.

- **"Vehicules"** - Nesta tabela, vão ser guardados os dados dos veículos, que cada utilizador tem associado à sua conta. Desta forma, esta tabela tem uma chave estrangeira *AccountId*, que permite associar um veículo a uma conta específica, garantindo também assim que cada conta possa ter vários veículos associados (1 : N). Nesta tabela também existem outros campos como *Plate* - matrícula, *Brand* - marca, *Model* - modelo que servem para o utilizador fazer a identificação do seu veículo, o campo *Ktypnr* que vai ser preenchido pela aplicação através da *API* externa *TecDoc*<sup>8</sup> e o campo *shareCode* que vai permitir ao utilizador enviar os dados do seu veículo para outro utilizar enquanto o código de partilha estiver válido.
- **"WorkShops"** - Nesta tabela, vão ser guardados os dados sobre as oficinas RedService. Desta forma, aqui vamos poder visualizar no campo *Name* - nome da oficina, nos campos *Adress*, *PostalCode* e *City* a morada completa da oficina, nos campos *Latitude* e *Longitude* as coordenadas sobre a localização específica da oficina, nos campos *Email*, *Phone1* e *Phone2* os meios de comunicação da oficina, o campo *Ative* que irá informar se a oficina se encontra ativa na rede RedService, o campo *Available* que informa se a oficina se encontra no momento a aceitar marcações da parte do cliente e o campo *Image* que vai conter a localização da imagem da oficina.
- **"WorkShopsConfigs"** - Nesta tabela, vão ser guardadas algumas configurações adicionais referente a cada oficina. Desta forma, esta tabela tem o campo *WorkShopId* que tem como chave primária o *Id* da oficina(tabela "*WorkShops*"), os campos *UserLeirilis* e *PwLeirilis* que vão guardar as credenciais de acesso ao Portal Leirilis, os campos *ERPNome*, *ERPData* e *ERPSilenceLogin* que vão armazenar os dados de acesso à *API* do ERP<sup>9</sup> e o *EmailLogin* que vai guardar o acesso ao email RedService.
- **"Schedules"** - Nesta tabela, vão ser guardados todos os dados sobre os agendamentos efetuados na Plataforma RedService. É composta pela chave estrangeira *WorkShopId* referente à chave primária da tabela *WorkShop*, permitindo que uma oficina tenha vários agendamentos (1 : N). Na tabelas dos agendamentos vão existir vários campos informativos sobre o os dados do veículo como os campos *Plate*, *Brand*, *Model*, *Kms* e *VehiculeId* que será preenchido caso o veículo já se encontre registado pelo utilizador na Plataforma RedService, os campos *Start* e *End* que delimita o horário da marcação, os campos *FirstName*, *LastName*, *Email*, *Phone*, *Nif* e *AccountId* que também

---

<sup>8</sup> TecDoc - Identificação de veículos (<https://www.tecalliance.net>)

<sup>9</sup> ERP - Sistema integrado de gestão empresarial (*Enterprise resource planning*)

será preenchido caso o email do utilizador já esteja registado na Plataforma RedService, o campo *Status* que informa o estado do agendamento, o campo *Notes* que vai permitir à oficina escrever notas internas.

- **"Services"** - Nesta tabela, são guardados os dados relativos aos agendamentos efetuados pelo cliente/oficina. Cada serviço guardado vai estar associado através de uma chave estrangeira *ScheduleId* ao agendamento e cada serviço pode ou não ter um tipo associado à tabela *ServiceTypes*, definido pelos administradores da Plataforma. Esta tabela ainda contempla vários campos como o nome, descrição, preço, tempo de serviço e estado.
- **"ServicesTimer"** - Esta tabela tem como funcionalidade contabilizar os tempos de serviço efetuados pelo mecânico da oficina. Cada *ServicesTimer* está associado a um serviço através da chave estrangeira *ServiceId*, cada tempo contabilizado está associado a um colaborador através da chave estrangeira *AccountId*. Na tabela está também guardado o início e o fim de serviço, sendo posteriormente efetuado o cálculo de tempo e atualizado o tempo total na tabela *Services*.
- **"Receptions"** - A tabela *Receptions* tem como objetivo guardar os dados do veículo quando este dá entrada na oficina. Esta tabela está associada com uma chave estrangeira à tabela de *Schedules* e guarda valores como os kms, nível de combustível, nome do cliente que entregou o veículo na oficina e fotografias do estado do veículo.
- **"ReceptionLine"** - A tabela *ReceptionLine* está associada à tabela *Reception* através de uma chave estrangeira e tem como finalidade guardar os serviços extra, adquiridos pelo cliente na entrega do veículo.

#### 6.4.2 *Application Programming Interface*

Para fazer a interligação entre a base de dados e o utilizador, foi necessário iniciar o desenvolvimento de uma *API*. Como referido anteriormente, foi selecionada a *framework ASP .net core*. Neste tópico vão ser explicadas as medidas de segurança que tiveram de ser implementadas na *API* de forma a garantir a segurança dos dados, bem como o acesso aos mesmos com a utilização de permissões por utilizador.

### 6.4.3 Segurança

Na implementação da segurança da *API*, foi desenvolvido um método de autenticação que utiliza os fundamentos da biblioteca da *.NET Microsoft.AspNetCore.Authentication*. Para a implementação funcionar, quando o utilizador da Plataforma cria uma nova conta, este deve preencher algum requisitos, um email e uma password para poder efetuar login na plataforma, posteriormente.

Listagem 1: Código de registo de utilizador

---

```
1 public void Register(RegisterRequest model, string origin)
2 {
3     // validate
4     if (_context.Accounts.Count() > 0)
5     {
6         if (_context.Accounts.Any(x => x.Email.Equals(model.Email)))
7             { // send already registered error in email to prevent account enumeration
8                 sendAlreadyRegisteredEmail(model.Email, origin);
9                 return;
10            }
11    }
12
13    // map model to new account object
14    var account = _mapper.Map<Account>(model);
15
16    // first registered account is an admin
17    var isFirstAccount = _context.Accounts.Count() == 0;
18    account.Role = isFirstAccount ? Role.Admin : Role.User;
19    account.Created = DateTime.Now;
20    account.VerificationToken = randomTokenString();
21
22    // hash password
23    account.PasswordHash = BC.HashPassword(model.Password);
24
25    // save account
26    _context.Accounts.Add(account);
27    _context.SaveChanges();
28
29    // send email
30    var response = sendVerificationEmail(account, origin);
31    if (response == false)
32    {
33        throw new Exception("Erro a enviar email");
34    }
35 }
```

---

Como podemos observar no enxerto de código da listagem 1, é feita logo uma verificação se o email já se encontra registado na plataforma, caso exista, será

despoletado um email para o utilizador a informar, e um *token* para poder recuperar a password da sua conta. No caso de não existir, vai ser feito um novo registo deste utilizador na plataforma, sendo-lhe atribuído o *Role user*, ou seja, de utilizador, que terá um acesso "mínimo a aplicação", como um cliente final da aplicação. Aquando da criação também é gerado um *token* aleatório para fazer a verificação do email, só depois deste passo, a conta fica funcional.

No passo seguinte, o utilizador pode efetuar o login da Plataforma RedService.

#### Listagem 2: Código de Autenticação

---

```

1 public AuthenticateResponse Authenticate(AuthenticateRequest model, string
  ↪ ipAddress)
2 {
3     try
4     {
5         var account = _context.Accounts.SingleOrDefault(x => x.Email ==
  ↪ model.Email);
6         if (account == null)
7             throw new ApplicationException("Conta não existe.");
8         if (!account.IsVerified || !BC.Verify(model.Password,
  ↪ account.PasswordHash) || account.Deleted != null)
9             throw new ApplicationException("Email ou password incorretas.");
10
11         // authentication successful so generate jwt and refresh tokens
12         var jwtToken = generateJwtToken(account);
13         var refreshToken = generateRefreshToken(ipAddress);
14         account.RefreshTokens.Add(refreshToken);
15
16         // remove old refresh tokens from account
17         removeOldRefreshTokens(account);
18
19         // save changes to db
20         _context.Update(account);
21         _context.SaveChanges();
22         var response = _mapper.Map<AuthenticateResponse>(account);
23         response.JwtToken = jwtToken;
24         response.RefreshToken = refreshToken.Token;
25         response.NameWorkshop = _context.Workshops.Where(x => x.Id.ToString() ==
  ↪ response.DefaultWorkshop).FirstOrDefault()?.Name;
26         return response;
27     }
28     catch (Exception ex)
29     {
30         throw new ApplicationException(ex.Message);
31     }
32 }
33 }

```

---

No excerto de código apresentado na listagem 2, podemos observar o método de autenticação da Plataforma RedService, quando o utilizador e a *password* estão corretas, é gerado um novo *token* e revogados todos os *tokens* antigos, caso existam, associados a conta em que está a ser feito o *login* e devolvida toda a informação necessária, como o *bearer token* para futuras chamadas à *API* como método de autenticação nos métodos protegidos.

#### 6.4.4 Roles/Perfis

Como referido anteriormente, para o desenvolvimento da plataforma existe a necessidade de criar vários tipos de utilizadores com vários tipos de privilégios. Como forma de superar esta necessidade, foram definidos vários tipos de roles/perfis onde vão ser discriminados os principais pontos de divergência entre eles.

Tabela 2: Plataforma *Web*

Funcionalidades/Roles	Administrador	Utilizador	Oficina	Gestor Oficina	Mecânico
Acesso a Plataforma Web	X		X	X	X
Acesso a <i>Dashboard</i>	X		X	X	X
Editar dados dos utilizadores	X				
Histórico de agendamentos	X		X	X	X
Criar/editar/eliminar agendamentos	X		X	X	X
Criar orçamentos	X			X	
Criar/editar/eliminar serviços para a plataforma	X				
Acesso a email	X		X	X	X
Editar dados das oficinas	X			X	
Pedidos de contacto	X				
Visualizar documentação da plataforma	X		X	X	
Gerir documentação da plataforma	X				

Tabela 3: Aplicação *mobile*

Funcionalidades/Roles	Administrador	Utilizador	Oficina	Gestor Oficina	Mecânico
Acesso à aplicação Mobile	X	X	X	X	X
Histórico de agendamentos		X			
Ver rede de oficinas	X	X	X	X	X
Criar agendamento		X			
Histórico de agendamentos	X	X	X	x	X
Aprovar/recusar orçamentos		X			
Agendamento na oficina	X		X	X	X
Receção ativa	X		X	X	X
Contabilizar tempos de serviços				X	X
Adicionar novo serviço em veículo			X	X	X
Visualizar notificações	X	X	X	X	X
Editar dados da conta	X	X	X	X	X
Definir oficina principal		X			
Registar/visualizar cartão RedService		X			
Criar/editar/eliminar veículos		X			
Consultar serviços efetuado nos veículos		X			
Importar veículo		X			

Nas tabelas 2 e 3, apresentadas anteriormente é possível observar as principais funcionalidades da aplicação *Web* e aplicação móvel com as diferenças entre os roles.

#### 6.4.5 Controladores

No desenvolvimento da *API*, de modo a estruturar as funções necessárias para toda a plataforma, para criados *controllers* com os variados *endpoints* da *API*. Na tabela 4, são apresentados os vários controladores desenvolvidos e explicadas de forma sucinta as suas funcionalidades.

Tabela 4: Controladores

Controlador	Descrição
AccountsController	Gere todos os pedidos associados a conta do utilizador, métodos CRUD
BannersController	Métodos CRUD para controlos dos <i>Banners</i> da aplicação mobile e <i>Front-end</i>
CardController	Associa os cartões RedService a uma conta de utilizador
CheckListController	Tem como objetivo efetuar a gestão dos serviços em <i>checkList</i>
DocumentationController	Adicionar, editar e eliminar documentação e possibilidade de associar a oficinas específicas
DocumentController	Criar, editar e eliminar documentos internos da aplicação
DpController	Controlar para efetuar os acessos às <i>API's</i> da Leirilis
ERPController	Controlar com os métodos de acesso ao ERP integrado na Plataforma
FrontEndController	Controlador para carregar os dados no site de <i>Front-End</i> da RedService
GraphController	Controlar para gerar os gráficos
JoinUsController	Controlador que recebe os pedidos de informação sobre a plataforma
KmsHistoryController	Guarda e gere os dados dos Kms fornecidos pelos utilizadores/oficinas
NewsletterController	Controlador que recebe informação dada pelos utilizadores para futura gestão de <i>marketing</i>
NotificationsController	Gere as notificações da Plataforma (Email e notificações <i>push</i> )
OtherServicesController	Criar, editar e eliminar serviços adicionais associados a um agendamento
ProductsController	Cria, editar e elimina produtos/serviços
ReceptionController	Gere a receção dos veículos
SchedulesController	Cria, editar e elimina todos os agendamentos associados a plataforma
ServicesController	Cria, editar e elimina serviços associados a um agendamento
ServicesTimerController	Gere todo o sistema de contabilização de tempos associado a um serviço
ServiceTypesController	Gere os serviços transversais a toda a plataforma
TecDocController	Controlador de acesso a <i>API</i> externa "TecDoc"
VehiculesController	Criar, editar e eliminar veículos da plataforma
WorkshopConfigs	Criar, editar e eliminar as configurações associadas às oficinas
WorkshopsController	Criar, editar e eliminar oficinas da plataforma

#### 6.4.6 Aplicação Web

No desenvolvimento da aplicação *Web*, foram criadas várias funcionalidades que serão explicadas e exemplificadas nesta secção.

Na implementação dos *Dashboards*, foram definidas diversas métricas para apresentar tanto ao administrador da oficina quanto aos administradores da plataforma.





## IMPLEMENTAÇÃO

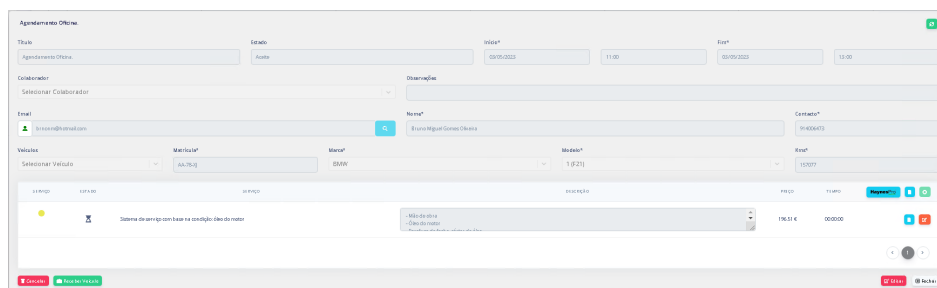


Figura 18: Agendamento: Detalhes

Na janela de detalhes da tabela dos serviços solicitados pelo cliente, é possível elaborar com precisão os artigos necessários para complementar o serviço. O administrador dos agendamentos pode definir esses valores manualmente, carregar referências através do ERP MasterWay (mediante uma integração realizada), ou pelo portal Leirilil. A integração desenvolvida com a ferramenta de orçamentação do portal Leirilil permite, utilizando a matrícula do carro e o tipo de serviço a ser realizado, cruzar a informação fornecida pelo *HaynesPro* com o catálogo de peças da Leirilil. Dessa forma, a plataforma retorna automaticamente o valor da peça com a margem de lucro definida.

Nesta janela, é possível visualizar diversos documentos, como orçamentos, folhas de obra e o documento final para emissão da fatura. Esses detalhes podem ser observados na Figura 19.

ID	QTY	Descrição	Preço	Valor	Total	Status	
8001	3,00	Mão de obra	10,00	30,00	30,00	✓	
13 M30201	1,00	Óleo de motor	27,26	27,26	41,21	✓	
512 20100511	20 10 0001	Parafuso de Pêdo, lâmpar do disco	3,46	23,00	4,252	✓	
402 2101040	2101040	Parafuso, parafuso de drenagem do disco	0,04	23,00	1,029	✓	
922487205	9 022 487 205	Filtro de disco	29,54	23,00	24,933	✓	
402 01461500	01461500	Junta, lâmpar do freno do disco	0,05	23,00	0,022	✓	
130742353	1 307 423 503	Filtro, ar do habitáculo	0,00	23,00	71,831	✓	
		Total Itm	31,73	Total Equipos	166,40	Total	191,51

Figura 19: Agendamento: Documento

Para melhorar a consulta dos agendamentos já realizados pela oficina, foi desenvolvido um histórico que permite ao utilizador pesquisar por nome, email, matrícula, data, marca ou modelo. Este recurso tem como finalidade fornecer uma visão completa de todos os serviços realizados pelo cliente em seus veículos. Os detalhes desta funcionalidade podem ser observados na Figura 20.

ID	Nome	Descrição	Data	Status
701	E-commerce	E-commerce	2023-10-10	Ativo
697	E-commerce	E-commerce	2023-10-10	Ativo
696	E-commerce	E-commerce	2023-10-10	Ativo
695	E-commerce	E-commerce	2023-10-10	Ativo
622	E-commerce	E-commerce	2023-10-10	Ativo
150	E-commerce	E-commerce	2023-10-10	Ativo
90	E-commerce	E-commerce	2023-10-10	Ativo
55	E-commerce	E-commerce	2023-10-10	Ativo
347	E-commerce	E-commerce	2023-10-10	Ativo
346	E-commerce	E-commerce	2023-10-10	Ativo
328	E-commerce	E-commerce	2023-10-10	Ativo
327	E-commerce	E-commerce	2023-10-10	Ativo
300	E-commerce	E-commerce	2023-10-10	Ativo
300	E-commerce	E-commerce	2023-10-10	Ativo
442	E-commerce	E-commerce	2023-10-10	Ativo

Figura 20: Histórico

Estas funcionalidades foram desenvolvidas com o objetivo de otimizar a gestão da oficina e melhorar a experiência do utilizador, proporcionando uma interface intuitiva e eficiente para a gestão de agendamentos. Tanto os utilizadores da oficina como os administradores podem monitorizar, controlar e analisar os serviços prestados de maneira mais eficaz.

#### 6.4.7 Aplicação Móvel

No desenvolvimento da aplicação móvel híbrida para Android e iOS, foram implementadas diversas funcionalidades que visam proporcionar uma experiência intuitiva e eficiente para o utilizador final. Nesta secção, são destacadas as principais funcionalidades do aplicativo, acompanhadas de exemplos explicativos para facilitar a compreensão.

Além disso, para auxiliar na compreensão das funcionalidades descritas, a Tabela 5 apresenta uma descrição detalhada dos diferentes estados possíveis na aplicação.

Tabela 5: Estados da aplicação

<b>Icon</b>	<b>Descrição</b>
✔	Quando o agendamento está aceite.
⊘	Quando o agendamento é cancelado.
✕	Quando a montagem/serviço é cancelado.
👤	Quando o agendamento está concluído.
🚗	Quando a montagem/serviço está completa.
🏠	Quando o estado está aceite.
👨‍🔧	Quando o veículo se encontra na oficina.
✂	Quando a montagem/serviço está aceite pelo cliente.

Quando a aplicação é aberta, é apresentado ao utilizador uma página de login, caso as suas credenciais estejam já guardados no dispositivo móvel, será pedido no caso do sistema operativo IOS o *FaceId* ou impressão digital, e no caso do Android a impressão digital para efetuar o login na aplicação.

Nesta página inicial, o utilizador ainda tem a possibilidade de efetuar a recuperação da *password*, efetuar o registo na plataforma e iniciar sessão sem conta.

Esta página inicial pode observada na Figura 21.

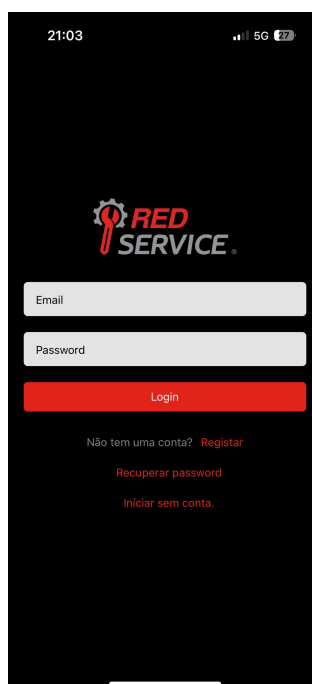


Figura 21: Login

Quando o utilizador faz o login, é apresentada a página principal da aplicação. Nesta página, o utilizador pode visualizar diversos dados sobre a sua conta, um botão que mostra um mapa com toda a rede oficial RedService e *banners* com as campanhas em vigor. No mapa, o utilizador tem a possibilidade de efetuar um agendamento na oficina selecionada, desde que esta esteja a aceitar agendamentos. As oficinas que aceitam agendamentos são destacadas no mapa com o logótipo da marca a cores, enquanto as que não aceitam são representadas com o logótipo a preto e branco.

Estas funcionalidades podem ser observadas nas Figuras 22 e 23.

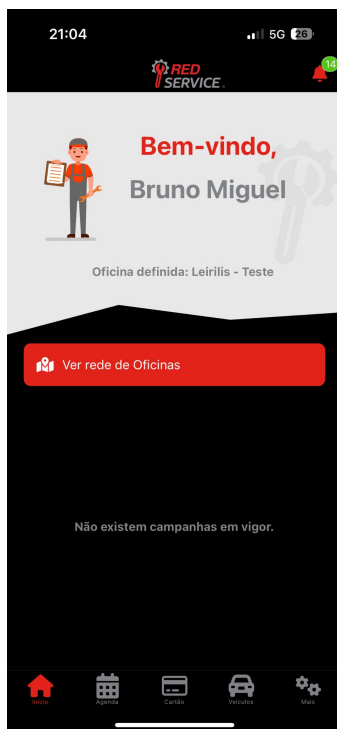


Figura 22: Página principal

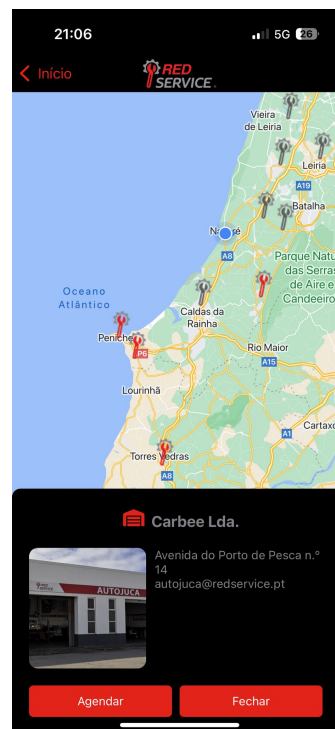


Figura 23: Rede RedService

Na segunda página da aplicação, o utilizador pode visualizar a lista de agendamentos, conforme apresentado na Figura 24. Nesta lista, é possível identificar o estado de cada agendamento através da primeira imagem da linha. Os estados podem variar entre "Pendente" (aguardando aceitação da oficina), "Aceite" (aceitação pela oficina dos horários e serviços propostos pelo cliente), "Oficina" (quando o veículo já se encontra nas instalações da oficina) e "Entregue" (quando o veículo já foi devolvido ao cliente).

Na Figura 25, é possível obter mais detalhes sobre o agendamento. Aqui, o utilizador pode observar com precisão os serviços que estão sendo ou já foram efetuados no veículo, bem como o respetivo valor associado. Nesta página, é possível

## IMPLEMENTAÇÃO

acompanhar o estado do veículo enquanto este se encontra na oficina. Assim que um dos serviços é dado como concluído, os detalhes do agendamento são atualizados e o utilizador recebe uma notificação, como podemos observar na Figura 26.



Figura 24: Listagem



Figura 25: Detalhes

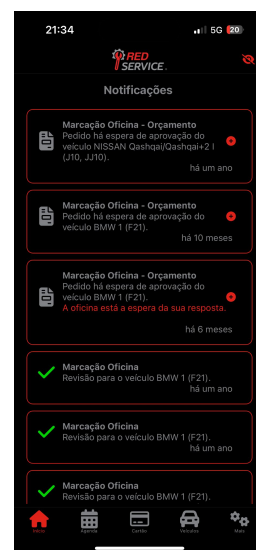


Figura 26: Notificações

Na terceira página da aplicação, zona do cartão RedService (figura 27), é possível efetuar o registo do cartão ou associar o cartão físico que já tem, na conta da plataforma.



Figura 27: Cartão RedService

Na quarta página da aplicação, é possível visualizar todos os veículos associados à sua conta, com podemos observar na figura 28, com alguns detalhes sobre o mesmo.

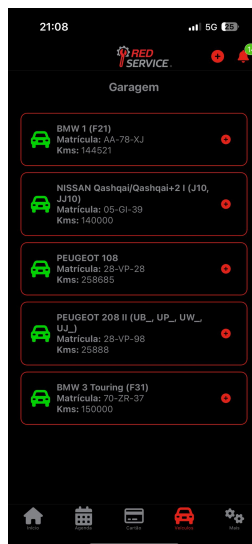


Figura 28: Listagem

Ao selecionar um veículo, o utilizador pode visualizar e editar diversas informações sobre o mesmo. Na primeira aba (figura 29), são exibidas informações básicas como marca, modelo, ano e quilometragem.

Na segunda aba, (figura 30), podem ser adicionadas informações detalhadas, como a data exata de registo do veículo, a média de quilómetros percorridos pelo utilizador e a data do seguro. Estas informações são utilizadas para gerar alertas sobre revisões, inspeções obrigatórias e a necessidade de manutenção do veículo.

Na terceira aba, (figura 31), o utilizador tem acesso a um histórico de manutenções efetuadas nas oficinas RedService que são adicionadas automaticamente pela plataforma.

Para tornar a aplicação mais versátil, foi criada uma opção que permite ao utilizador introduzir manualmente manutenções externas à plataforma, utilizando o botão "+" no canto superior direito. Além disso, o utilizador pode partilhar os dados do seu veículo com outros utilizadores através do botão de *share*, como ilustrado nas figuras 28 , 30 e 31.

## IMPLEMENTAÇÃO

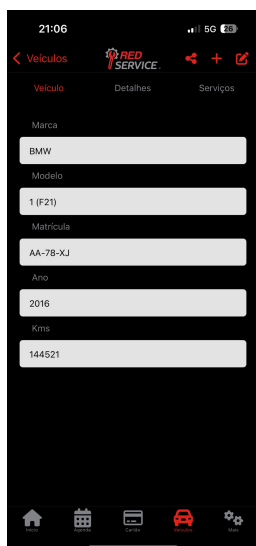


Figura 29: Informação

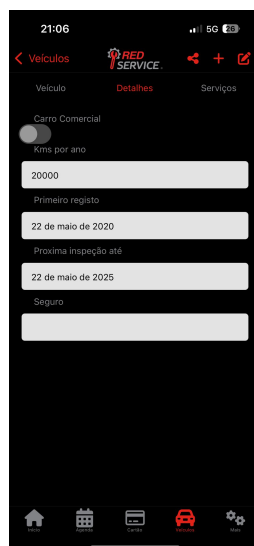


Figura 30: Detalhes

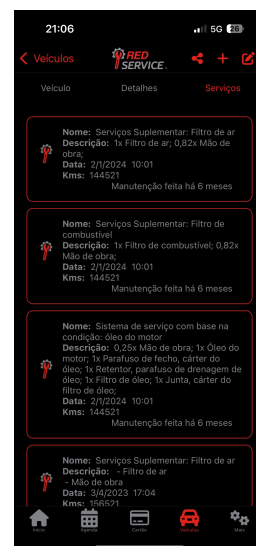


Figura 31: Serviços

Na última página da aplicação, as definições (figura 32), é possível editar as informações da conta, alterar a oficina principal, importar veículos para a conta do utilizador e consultar toda a informação legal da Plataforma RedService.

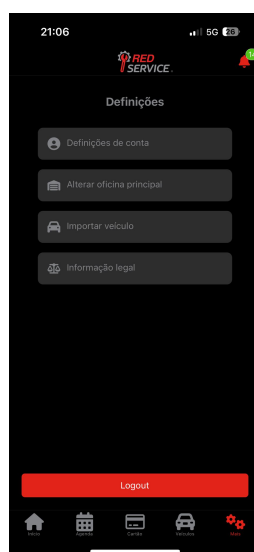


Figura 32: Definições

Na ótica da oficina, o utilizador irá verificar diferentes funcionalidades. Na figura 33 é possível observar a menor quantidade de menus disponíveis.

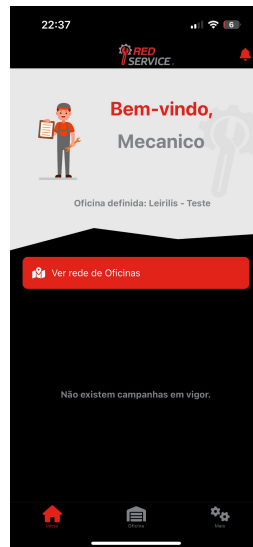


Figura 33: Página inicial da oficina

No segundo menu, o mecânico irá encontrar todos os agendamentos para a sua oficina e quando um agendamento se encontra afeto a si, este aparecerá na aplicação a cor branca, como se pode observar na figura 34.



Figura 34: Agendamentos oficina

Quando um agendamento é selecionado, o mecânico tem acesso a todos os serviços que deve realizar, conforme ilustrado na Figura 35. Nesta janela de detalhes, o mecânico deve selecionar o(s) serviço(s) a ser(em) executado(s) e iniciar o contador de tempo. Este recurso permite à oficina contabilizar o tempo efetivo que o mecânico levou para completar o serviço proposto.

O contador de tempo também oferece funcionalidades adicionais, como a visualização da listagem dos tempos registrados e a possibilidade de efetuar pausas durante a execução do serviço, como é possível observar nas figuras 36 e 37.



Figura 35: Detalhes

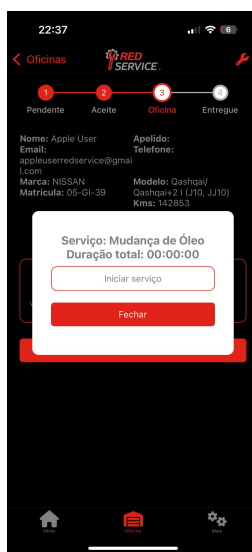


Figura 36: Iniciar tempos



Figura 37: Lista de tempos

Ainda na página de detalhes, o mecânico tem a possibilidade de propor um novo serviço adicional, utilizando a opção representada pela "Chave" no canto superior direito. Esse novo serviço proposto deve ser aprovado pelo cliente antes de ser realizado. A Figura 38 ilustra o formulário que o mecânico precisa preencher para sugerir o serviço adicional.

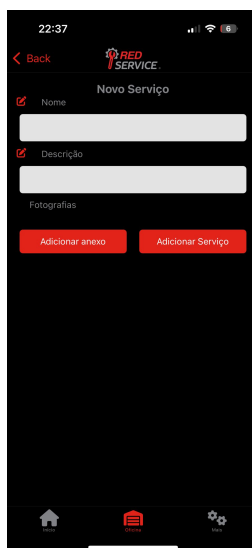


Figura 38: Agendamentos oficina

O desenvolvimento da aplicação móvel híbrida para Android e IOS foi desenvolvida para fornecer uma experiência funcional tanto para os clientes como para as oficinas da rede RedService. Através da implementação de uma interface intuitiva e várias funcionalidades, a aplicação não só facilita a gestão de agendamentos e serviços, mas também melhora a comunicação entre mecânicos e clientes.

A aplicação destaca-se pela integração de funcionalidades que respondem diretamente às necessidades dos utilizadores, promovendo uma experiência de utilização fluída e prática. Assim, tanto a rede de oficinas quanto os clientes beneficiam de um sistema que melhora significativamente a organização e a transparência no processo de manutenção de veículos.

## 6.5 TESTES

Durante o desenvolvimento da Plataforma RedService, foi adotada uma abordagem de testes contínuos em todas as funcionalidades. Desde a fase inicial, o código foi sendo avaliado, procurando explorar vulnerabilidades e garantir que eventuais falhas fossem corrigidas.

Na fase final de desenvolvimento, foi realizada uma implantação da plataforma em algumas oficinas da rede RedService. Esta etapa foi fundamental para validar a solução num ambiente real. As oficinas começaram a utilizar a plataforma nas suas operações diárias e, ao longo desse processo, reportaram eventuais bugs e sugeriram melhorias. O *feedback* das oficinas foi crucial, uma vez que permitiu adaptar o sistema às suas necessidades específicas. A implementação dessas melhorias baseadas nas sugestões dos próprios utilizadores tornou a plataforma mais alinhada com os fluxos de trabalho reais, o que contribuiu significativamente para a usabilidade e a eficiência do sistema.

Adicionalmente, a aplicação móvel, desenvolvida para ser utilizada principalmente pelos clientes finais, passou por uma série de testes feitos pelos colaboradores da empresa Leirilis. Esses colaboradores, atuando como utilizadores finais, relataram diretamente os problemas encontrados durante o uso da aplicação, possibilitando que as falhas fossem resolvidas de forma rápida e eficiente. Essa participação ativa dos colaboradores no processo de testes foi essencial para garantir que a experiência do utilizador final fosse satisfatória e que a aplicação fosse otimizada para atender às expectativas dos clientes.

## IMPLEMENTAÇÃO

Desta forma, a combinação de testes contínuos e a validação em ambientes de trabalho e *feedback* dos utilizadores finais permitiu que a Plataforma RedService fosse aprimorada ao longo do desenvolvimento, resultando em uma solução adaptada às necessidades tanto das oficinas quanto dos seus clientes.

## CONCLUSÃO

---

O desenvolvimento da Plataforma RedService resultou numa solução inovadora e eficaz para centralizar a gestão de oficinas automóveis, atendendo às necessidades específicas da empresa Leirilis. A Plataforma permitiu a uniformização dos processos nas oficinas da rede, proporcionando um controlo mais eficaz das operações diárias e uma melhoria significativa na experiência dos clientes. A mesma integra funcionalidades cruciais, como o agendamento online, orçamento e monitorização dos serviços em tempo real.

Do ponto de vista técnico, a escolha de tecnologias como ASP.NET Core, React.js e Microsoft SQL Server revelaram-se essenciais para garantir a robustez e a escalabilidade. A utilização de uma arquitetura modular permitiu a fácil integração com aplicações externas, como um ERP, os serviços de identificação de veículos por matrícula do TecDoc e orçamentação da HaynesPro. Além disto, a aplicação móvel híbrida e a Plataforma *web* tornam-na acessível em diferentes dispositivos, aumentando a usabilidade por parte dos utilizadores.

A metodologia ágil adotada, a combinação de Scrum e Kanban, facilitou o desenvolvimento incremental, o que possibilitou a rápida adaptação às mudanças solicitadas pela empresa e uma entrega contínua de valor. A utilização de quadros Kanban ajudou a manter o fluxo de trabalho organizado e garantiu uma visão clara das tarefas a serem desenvolvidas, enquanto as *sprints* do Scrum permitiram que as funcionalidades fossem validadas e ajustadas com os objetivos definidos.

Os principais resultados da implementação da RedService são a otimização dos recursos das oficinas, a uniformização dos processos operacionais, a redução de custos operacionais e melhorar a comunicação com os clientes. Estas melhorias devem resultar num aumento da produtividade e na capacidade das oficinas atenderem um volume maior de clientes com mais eficiência e qualidade.

Relativamente ao trabalho futuro, existem algumas otimizações que se podem identificar. A plataforma pode evoluir com a integração de tecnologias de inteligência artificial e *machine learning* para automatizar ainda mais os processos, como a previsão de manutenção preditiva baseada em dados históricos de serviços e o comportamento dos veículos. Outro ponto a ser explorado é o desenvolvimento de

## CONCLUSÃO

funcionalidades focadas na fidelização dos clientes, como programas de pontos e descontos personalizados a cada cliente.

Por fim, a RedService não só atende aos requisitos atuais da Leirilis como também estabelece uma base sólida para futuras inovações. Ao integrar tecnologia de ponta com processos de negócios definidos, coloca a rede RedService numa posição competitiva no setor automóvel, preparada para crescer e adaptar-se às mudanças e evoluções no mercado.

## BIBLIOGRAFIA

---

- AMA - Agência para a Modernização Administrativa, I. P. (2022). *Tipos de aplicações*. URL: <https://usabilidade.gov.pt/tipos-de-aplicacoes>.
- automóvel, A Oficina - Especialista no seu (2022). *A Oficina*. URL: <https://www.aoficina.net>.
- Boukhary, Shady e Eduardo Colmenares (2019). «A Clean Approach to Flutter Development through the Flutter Clean Architecture Package». Em: *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1115–1120. DOI: [10.1109/CSCI49370.2019.00211](https://doi.org/10.1109/CSCI49370.2019.00211).
- Consumer Guidance Institute Portugal, Lda (2020). *As maiores redes de oficinas independentes em Portugal*. URL: <https://www.cgip.pt/consumer-infos/as-maiores-redes-de-oficinas-independentes-em-portugal/>.
- (2022). *As melhores oficinas 2022*. URL: <https://www.cgip.pt/tests/as-melhores-oficinas-2022/>.
- Eco-Oficinas (2022). *Eco-Oficinas*. URL: <http://www.eco-oficina.pt>.
- Firebird (2024). *Firenird 2019*. URL: <https://www.firebirdsql.org/>.
- Foundation, OpenJs (2022). «Express.js». Em: URL: <https://expressjs.com/>.
- Git (2023). *Git*. URL: <https://git-scm.com>.
- Greiff, Magnus e André Johansson (2019). *Symfony vs Express: A Server-Side Framework Comparison*.
- Group, Sendys (2022). *MasterWay*. URL: <https://www.masterway.net>.
- Hansson, Niclas e Tomas Vidhall (2016). *Effects on performance and usability for cross-platform application development using React Native*.
- Karlsson, Oliver (2021). *A Performance comparison Between ASP.NET Core and Express.js for creating Web APIs*.
- Leirilis (2022). *Leirilis - Valores*. URL: <http://www.leirilis.com>.
- Lund, Niklas (2017). *Node.js vs PHP : En jämförelse av serverside-tekniker*.
- Marktest (2020). *Internet 2020: cresce a liderança dos smartphones*. URL: <https://www.marktest.com/wap/a/n/id~269e.aspx>.
- Meet Ken Schwaber, Meet Jeff Sutherland e (2024). *Scrum Guide*. URL: <https://scrumguides.org/scrum-guide.html>.
- Meta Platforms, Inc. (2022). *React.js*. URL: <https://reactjs.org>.
- Mforce (2022). *Rede de oficinas Mforce*. URL: <https://www.oficinasmforce.pt>.

- Microsoft (2022a). *Roslyn*. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/roslyn-sdk/>.
- (2022b). *What is Xamarin?* URL: <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>.
- (2023). *What is ASP.NET?* URL: <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>.
- mongodb (2024). *MongoDB*. URL: <https://docs.mongodb.com/manual>.
- Norauto (2022). *Norauto*. URL: <https://www.norauto.pt>.
- Oracle (2022a). *What Is a Database?* URL: <https://www.techtarget.com/whatis/definition/server>.
- (2022b). *What is a Server?* URL: <https://www.oracle.com/database/what-is-database/>.
- Orracle (2022). *What is ERP?* URL: <https://www.oracle.com/pt/erp/what-is-erp/>.
- Plesk (2023). *Plesk Documentation and Help Portal*. URL: <https://docs.plesk.com/en-US/obsidian/administrator-guide/mail.71434/>.
- RedService (2022). *RedService - Quem Somos*. URL: <https://www.redservice.pt/A-RedService>.
- Rehkopf, Max (2024). *Kanban - Board*. URL: <https://www.atlassian.com/agile/kanban/boards>.
- Service, Bosh Car (2022). *Bosch Car Service -Encontre a oficina mais próxima*. URL: <https://www.boschcarservice.com/pt/pt/localizador-de-oficinas>.
- Software, The Apache (2023). *Overview*. URL: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>.
- StatCounter (2022). *Mobile Operating System Market Share Portugal*. URL: <https://gs.statcounter.com/os-market-share/mobile/portugal>.
- statcounter (2022). *Mobile Operating System Market Share Worldwide*. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- Vue (2022). *Vue*. URL: <https://vuejs.org/guide/introduction.html>.
- Wu, Wenhao (2018). «React Native vs Flutter, Cross-platforms mobile application frameworks». Em.

## DECLARAÇÃO

---

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título “*Plataforma Redservice*”, é original e foi realizado por Bruno Miguel Gomes Oliveira (2202591) sob orientação de professora Dulce Cristina dos Santos Iria Gonçalves ([dulce.goncalves@ipleiria.pt](mailto:dulce.goncalves@ipleiria.pt)).

*Leiria, setembro de 2024*

---

Bruno Miguel Gomes Oliveira