



Atlântica – Instituto Universitário

Tese Mestrado em Engenharia e Gestão da Manutenção Aeronáutica

Maintenance Tracking System

Autor: Jair Simão Varela dos Reis

Orientador: Professor Doutor Rui Carvalho

Data: setembro 2025

Resumo

A presente dissertação teve como objetivo a concepção, desenvolvimento e validação de uma plataforma digital destinada à gestão integrada de manutenção aeronáutica, com especial foco no modelo de operação do Cessna 152. O trabalho foi sustentado por uma revisão bibliográfica sobre normas internacionais, como a EASA Part-M e a FAA FAR 43, aliada à análise de requisitos funcionais e não funcionais identificados junto de profissionais do setor.

O sistema proposto foi implementado segundo uma arquitetura em camadas, recorrendo a tecnologias modernas, tais como Django REST Framework no backend e Vue 3 no frontend, garantindo escalabilidade, segurança e usabilidade. Foram incluídos mecanismos de autenticação baseados em perfis de utilizador, gestão de componentes aeronáuticos, emissão de alertas em tempo real e geração automática de relatórios em conformidade com os padrões regulamentares.

A avaliação do sistema contemplou testes funcionais, automatizados e de usabilidade, que confirmaram o cumprimento dos requisitos inicialmente definidos. Os resultados demonstraram tempos de execução reduzidos, ausência de erros significativos e elevado grau de satisfação por parte dos utilizadores finais, revelando a aplicabilidade prática da solução.

Este projeto pode contribuir não apenas para a modernização dos processos de manutenção aeronáutica em pequenas frotas, mas também para a criação de bases sólidas que podem ser estendidas a contextos mais complexos, favorecendo a segurança operacional e a rastreabilidade exigidas pelas entidades reguladoras.

Palavras-Chave: Manutenção aeronáutica, CAMO, Aviação geral, Digitalização de processos, Django REST, PostgreSQL, Segurança operacional

Abstract

The objective of this dissertation was to design, develop, and validate a digital platform for integrated aeronautical maintenance management, with a special focus on the Cessna 152 operating model. The work was supported by a literature review on international standards, such as EASA Part-M and FAA FAR 43, combined with an analysis of functional and non-functional requirements identified by industry professionals.

The proposed system was implemented according to a layered architecture, using modern technologies such as Django REST Framework in the backend and Vue 3 in the frontend, ensuring scalability, security, and usability. Mechanisms based on user profiles, aeronautical component management, real-time alerts, and automatic report generation in accordance with regulatory standards were included.

The system evaluation included functional, automated, and usability tests, which confirmed compliance with the initially defined requirements. The results demonstrated reduced execution times, no significant errors, and a high degree of satisfaction among end users, revealing the practical applicability of the solution.

This project can contribute not only to the modernization of aeronautical maintenance processes in small fleets, but also to the creation of solid foundations that can be extended to more complex contexts, promoting the operational safety and traceability required by regulatory entities.

Keywords: Aeronautical maintenance, CAMO, General aviation, Process digitization, Django REST, PostgreSQL, Operational safety

Índice

1. Introdução	14
1.1. Generalidades	14
1.2. Motivação	15
1.3. Âmbito e Objetivo	15
1.4. Metodologia.....	15
1.4.1. Metodologia de desenvolvimento do <i>software</i>	17
1.5. Estrutura da Dissertação	18
1.6. Introdução ao Capítulo 2	19
2. Revisão literária.....	19
2.1. Manutenção de aeronaves.....	19
2.2. Aeronavegabilidade	20
2.2.1. Aeronavegabilidade Permanente	20
2.3. Metodologia de gestão da manutenção e regulamentação aeronáutica	21
2.4. Engenharia de <i>Software</i> aplicada à Aviação.....	22
2.5. Sistemas e métodos atuais de gestão de manutenção em ambiente controlado.....	23
3. Cessna 152.....	24
3.1. ATA 100	25
3.2. Documentação da aeronave	26
3.2.1. MSG-3 e <i>MPD – Maintenance Planing Document</i>	27
3.2.2. AMM – Manual de Manutenção de Aeronave	29
3.2.3. MMEL-Master Minimum Equipment List e MEL- Minimum Equipment List.....	30
3.2.4 Manual de isolamento de falhas	31
4. Maintenance Tracking System	32
4.1. Desenvolvimento da aplicação	33
4.1.1. Descrição do Sistema.....	34
4.1.1.1. Registo de Aeronaves e Componentes	34
4.1.1.2. Registo de Voos.....	34
4.1.1.3. Gestão de Manutenção.....	34
4.1.1.4. Emissor de Alertas.....	34

4.1.1.5.	Relatórios e Exportação.....	35
4.1.1.6.	Perfis dos Utilizadores.....	35
4.1.2.	Levantamento Requisitos	35
4.1.2.1.	Requisitos Funcionais.....	36
4.1.2.2.	Requisitos Não Funcionais	37
4.1.3.	Arquitetura do Sistema	38
4.1.3.1.	Camada de Utilizador	38
4.1.4.	Visão geral do sistema.....	39
4.1.4.1.	Caso de uso do piloto	42
4.1.4.2.	Caso de Uso do Técnico de Manutenção	44
4.1.4.3.	Caso de uso do Supervisor da Manutenção	46
4.1.4.4.	Caso de uso do Administrador do Sistema.....	47
4.1.4.5.	Tela de Login.....	50
4.1.4.6.	<i>Dashboard</i> do Piloto.....	50
4.1.4.7.	<i>Dashboard</i> do técnico de manutenção – centro operacional de manutenção	53
4.1.4.8.	<i>Dashboard</i> do Supervisor — visão-geral para decisões de aprovação	55
4.1.4.9.	Padrão geral de gestão	59
5.	Apresentação e discussão de resultados	61
5.1.	Resultados por perfil de utilizador.....	62
5.2.	Vantagens e mais-valias do Maintenance Tracking System	63
5.3.	Implicações para a gestão da manutenção aeronáutica	64
6.	Conclusão	66
	Bibliografia.....	68
	Anexo	1
1.	Arquitetura de Base de Dados	1
2.	Diagramas de Sequência.....	2
2.1.1	Registar Voo	3

2.1.2	Registrar Manutenção	4
2.1.3	Aprovar Manutenção	4
2.1.4	Rejeitar Manutenção	5
2.2	Interfaces do Sistema (<i>Frontend</i>)	6
2.3	Codificação (Backend)	6
2.4	Estrutura do <i>Backend</i> (Django REST Framework):	7
2.5	Detalhes Técnicos da Implementação	8
2.6	Codificação <i>Backend</i> – Funcionalidades Específicas.....	13
2.7	Testes.....	16

Índice de Tabelas

Tabela 1 - Funcionalidades dos utilizadores Pt.1 (Piloto)	39
Tabela 2 - Funcionalidades dos utilizadores Pt.2 (Técnico de Manutenção)	40
Tabela 3 - Funcionalidades dos utilizadores Pt.3 (Supervisor)	41
Tabela 4 - Funcionalidades dos utilizadores Pt.4 (Administrador)	42
Tabela 5 - Descrição da funcionalidade “Registar Voo”	43
Tabela 6 - Descrição da funcionalidade “Relatar Anomalia de Componente”	44
Tabela 7 - Descrição da funcionalidade "Registar Manutenção"	45
Tabela 8 - Descrição da funcionalidade "Registar Inspeção"	46
Tabela 9 - Descrição da funcionalidade "Rejeitar Manutenção"	47
Tabela 10 - Descrição da funcionalidade “Gerir Utilizadores e Perfis”	48
Tabela 11 - Descrição da funcionalidade “Consultar Logs de Auditoria”	49
Tabela 12 Verificação de Requisitos Funcionais	62
Tabela 13 - Resultado de teste de usabilidade	65

Índice de Figuras

Figura 1 - Arquitetura do Sistema	38
Figura 16 - Tela login	50
Figura 17 - <i>Dashboard</i> do piloto	50
Figura 18 - Ecrã “Voo”	51
Figura 19 - <i>Dashboard</i> do técnico de manutenção	53
Figura 20 - Ecrã “Relato Anomalia”	54
Figura 21 - Ecrã “Detalhes da Inspeção”	54
Figura 22 - <i>Dashboard</i> do supervisor	55
Figura 23 - <i>Dashboard</i> do supervisor (painel de decisão)	55
Figura 24 - Ecrã "Inspeção "	56
Figura 25 - Ecrã voo (supervisor)	57
Figura 26 - Ecrã manutenção (supervisor)	57
Figura 27 - Ecrã componentes (supervisor)	58
Figura 28 - Ecrã administrador Painel “Utilizador”	59
Figura 29 - Ecrã "Log"	60
Figura 2 - Arquitetura PostgreSQL	2
Figura 8 - Diagrama da sequência de “Login”	3
Figura 9 - Diagrama da sequência de “Registar Voo”	3
Figura 10 - Diagrama da sequência de “Registar Manutenção”	4
Figura 11-Diagrama de Sequência Aprovar Manutenção	5
Figura 12 - Diagrama da sequência de “Rejeitar Manutenção”	5
Figura 30 - Estrutura do Backend	7
Figura 31 - Modelo utilizador	8
Figura 32 - Serializador do utilizador	9
Figura 33 - <i>ViewSet</i> de utilizadores	10

Figura 34 - <i>ViewSet</i> de utilizadores cont.	10
Figura 35 - URLs utilizador.....	11
Figura 36-Serviço de alerta.....	12
Figura 37-Serviço de alerta cont.....	12
Figura 38 - Cálculo dos alertas dos componentes	13
Figura 39 - Endpoint REST “Registar Voo”	14
Figura 40 - Criação de relatórios PDF de for automática.....	15
Figura 41 - Criação de relatórios PDF de for automática cont.....	16
Figura 42 - API Registo de voo	17
Figura 43 - API Componentes em alerta	17
Figura 44 - API Autenticação de utilizadores	17

Lista de acrónimos

AMM – *Aircraft Maintenance Manual*

AMOS – *Aircraft Maintenance and Engineering System*

AMP – *Approved Maintenance Programme*

AOG – *Aircraft on Ground*

API – *Application Programming Interface*

APU – *Auxiliary Power Unit*

ARC – *Airworthiness Review Certificate*

ATA 100 – *Air Transport Association Specification 100*

CAMO – *Continuing Airworthiness Management Organisation*

CAMP – *Computerized Aircraft Maintenance Program*

CAS – *Crew Alerting System*

CAT – *Commercial Air Transport*

CM – *Condition Monitoring*

CRUD – *Create, Read, Update, Delete* (Padrão geral de gestão)

CSRF – *Cross-Site Request Forgery*

CofA – *Certificate of Airworthiness*

DAL – *Design Assurance Levels*

DRF – *Django REST Framework*

EASA – *European Union Aviation Safety Agency*

EASA Part-M – *European Union Aviation Safety Agency – Part-M* (Requisitos de aeronavegabilidade contínua)

ECAM – *Electronic Centralised Aircraft Monitoring*

EFB – *Electronic Flight Bag*

EICAS – *Engine Indicating and Crew Alerting System*

ERP – *Enterprise Resource Planning*

FAA – *Federal Aviation Administration*

FAA FAR 43 – *Federal Aviation Administration – Federal Aviation Regulations Part 43* (Manutenção, manutenção preventiva e alterações)

FADEC – *Full Authority Digital Engine Control*

FMS – *Flight Management Systems*

H/T – *Hard Time*

HTTP – *Hypertext Transfer Protocol* (Controlador HTTP)

ICAO – *International Civil Aviation Organization*

ID – *Identificador*

INOP – *Inoperative*

JSON – *JavaScript Object Notation*

JWT – *JSON Web Token (Autenticação JWT)*

LLP – *Life Limited Parts*

MCDU – *Multipurpose Control and Display Unit*

MEL – *Minimum Equipment List*

MER – *Modelo Entidade-Relação*

MMEL – *Master Minimum Equipment List*

MOC – *Minimum Operational Configuration*

MPD – *Maintenance Planning Document*

MROs – *Maintenance, Repair and Overhaul Organisations*

MSG-3 – *Maintenance Steering Group – 3rd Task Force*

MTOW – *Maximum Take-Off Weight*

MTV – *Model-Template-View*

MVC – *Model-View-Controller*

NAA – *National Aviation Authority* (Autoridade Nacional de Aviação Civil)

NCO – *Non-Commercial Operations*

NCC – *Non-Commercial Operations with Complex Motor-Powered Aircraft*

OASES – *Open Aviation Strategic Engineering System*

OEM – *Original Equipment Manufacturer*

OEMs – *Original Equipment Manufacturers*

ORM – *Object-Relational Mapper*

PWA – *Progressive Web Application*

Parte-145 – *Organizações de manutenção aprovadas*

Parte-CAMO – Parte do regulamento europeu aplicável às CAMO

Parte-CAO – Parte do regulamento europeu aplicável às CAO (*Combined Airworthiness Organisation*)

Parte-ML – Parte do regulamento europeu aplicável à manutenção de aeronaves de pequena dimensão

RNF – Requisitos Não Funcionais

SPA – *Single-Page Application*

SRM – *Structure Repair Manual*

TLS – *Transport Layer Security*

URLs – *Uniform Resource Locators*

VFR/IFR – *Visual Flight Rules / Instrument Flight Rules*

WDM – *Wiring Diagram Manual*

1. Introdução

1.1.Generalidades

Este projeto visa contribuir para o enriquecimento e um melhor desenvolvimento da manutenção preventiva na aviação geral. O projeto reside no desenvolvimento de um “*Maintenance Tracking System*” ou um software de controlo de manutenção aeronáutica. Este software é uma aplicação mobile onde o piloto e/ou proprietário da aeronave pode registar/verificar e/ou monitorizar em tempo real, as horas e ciclos da aeronave. Permite ainda verificar o intervalo entre cada inspeção do programa de manutenção, respetivas tarefas e remanescentes de utilização até à próxima inspeção, otimizando todas as tomadas de decisões relativas à utilização e toda a manutenção da mesma de forma automática e com maior controlo, rigor e precisão. Por ser um software mobile, é de rápido e fácil acesso podendo ser consultado através do Smartphone do piloto e/ou proprietário da aeronave, o que contribui para o aumento da segurança operacional. Visto que, antes de cada voo o piloto e/ou proprietário da aeronave consegue verificar o remanescente de utilização até à próxima manutenção para cada tarefa, pacote ou componente da aeronave.

A manutenção de aeronaves depende diretamente de diferentes manuais e dados operacionais do fabricante ou OEM (*Original Equipment Manufacturer*) para ser realizada com o nível aceitável de segurança operacional. Os operadores aéreos costumam ter dentro da sua estrutura organizacional, um departamento ou equipas responsáveis pelo controlo da manutenção das aeronaves, assegurando o cumprimento das tarefas de manutenção programadas e providenciar uma resposta adequada aos eventos de manutenção não programada. A equipa de desenvolvimento do programa e planeamento da manutenção, envia os designados pacotes de trabalho, com todas as tarefas necessárias a ser realizadas para uma organização de manutenção Parte-145. No caso de haver uma ocorrência ou evento não programado, também é enviado um pacote de trabalho para a organização Parte-145 para a realização da inspeção e/ou reparação, caso seja necessário. Estas tarefas de manutenção devem ser sempre realizadas com a ajuda e descrição passo-a-passo presente nos manuais da aeronave e do fabricante.

A aviação e/ou operação de aeronaves deve ser dividida em duas categorias. Operações aéreas comerciais ou CAT (*Commercial Air Transport*) e operações aéreas não comerciais ou NCO/NCC (*Non-Commercial Operations / Non-Commercial Operations with Complex Motor-Powered Aircraft*). O foco do projeto recai sobre a operação de aeronaves não comercial. Esta, é característica por ter uma base legal mais leve e menos restritiva que as operações aéreas comerciais, não existindo a necessidade do proprietário possuir ou subcontratar um centro de controlo de manutenção de aeronaves. O próprio é então responsável pelo controlo e planeamento da manutenção, podendo até, em alguns casos, ser o mesmo a executar e reparar as suas aeronaves.

1.2.Motivação

Nos Estados Unidos tem havido cada vez mais acidentes com aeronaves de operação não comercial, representando quase a totalidade dos acidentes aéreos no país. Em 2023, ocorreram 1216 acidentes aéreos, dos quais, 1150 foram com aeronaves em operação não comercial, representando quase 95 por cento dos acidentes e ocorrências [NTSB. (Maio de 2025)].

Isto despertou em mim uma vontade de desenvolver uma aplicação de rastreio de manutenção por forma a minimizar este problema. Esta aplicação permite ajudar os proprietários de aeronaves em operações aéreas não comerciais, que não têm um centro de controlo de manutenção de aeronaves, a controlar melhor e a monitorizar a aeronavegabilidade das suas aeronaves.

Com este projeto espero contribuir para diminuição do número de acidentes causados por falhas mecânicas devido ao incumprimento dos intervalos de manutenção ou substituição de componentes, estabelecidos pelos fabricantes.

1.3.Âmbito e Objetivo

Esta tese foi desenvolvida no âmbito do Mestrado Engenharia e Gestão da Manutenção Aeronáutica e tem como objetivo o desenvolvimento de uma aplicação *mobile* que faz o controlo, a programação e o planeamento da manutenção das aeronaves de forma automática, com maior rigor e precisão. Cujo publico alvo são os proprietários de aeronaves em operações aéreas não comerciais que não têm um centro de controlo de manutenção de aeronaves.

A aplicação *mobile* é direcionada para um proprietário de um Cessna C152, para efeitos de estudo e projeto mas pode no futuro ser adaptada a qualquer tipo de aeronave em regime de operação aérea não comercial. Pretende proporcionar uma fácil monitorização da manutenção um maior controlo da vida útil da aeronave e respetivos componentes e simplificar o processo de planeamento e realização das tarefas de manutenção.

1.4.Metodologia

O sistema será desenvolvido em cinco fases essenciais, que vão desde a recolha de dados técnicos dos voos à validação prática do sistema proposto. A abordagem será feita com base em métodos qualitativos e com participação ativa dos futuros utilizadores da aplicação.

As fases são as seguintes:

Fase 1 – Recolha dos dados técnicos e operacionais

Nesta fase inicial, serão recolhidas informações provenientes dos manuais de manutenção da aeronave (Cessna 152), bem como dados específicos fornecidos pelo fabricante do equipamento. Estes dados incluem os limites de utilização (em horas de voo e/ou ciclos) dos principais componentes, assim como os intervalos exigidos para inspeções e substituições. A recolha será realizada através de fontes oficiais e documentos técnicos, com especial incidência nas aeronaves ligeiras.

Fase 2 – Análise e estruturação das regras de manutenção

Com base nos dados recolhidos, haverá análise e extração de regras que serão aplicadas na lógica da aplicação, como, por exemplo: “substituir o filtro a cada 50 horas de voo”, “verificar o sistema hidráulico a cada 5 ciclos”. Estas regras constituirão a base do sistema de alertas automáticos. Também nesta fase serão definidos os requisitos funcionais da aplicação com base na realidade dos operadores da aviação geral.

Fase 3 – Desenvolvimento e implementação da aplicação

Esta fase será dedicada à programação da aplicação propriamente dita, com foco na implementação da base de dados, das *APIs* – *Application Programming Interface* e interfaces web. Será adotada uma arquitetura cliente-servidor, com separação entre o *frontend* (Vue.js) e *backend* (Django REST), utilizando o PostgreSQL para armazenamento de dados. As interfaces serão desenvolvidas de forma adaptável, privilegiando a fácil utilização por técnicos, pilotos e supervisores.

Fase 4 – Testes e validação com utilizadores reais

Após o desenvolvimento dos primeiros módulos, haverá uma sessão de testes com simulações baseadas em perfis reais. Serão testadas funcionalidades como o registo de voos, a emissão de alertas e a elaboração de relatórios de manutenção. O tempo de resposta e a usabilidade serão avaliados com base no feedback dos utilizadores, através de formulários de avaliação e de entrevistas estruturadas.

Fase 5 – Avaliação final e documentação dos resultados

Nesta última fase, será feita uma análise crítica da eficácia da aplicação, especialmente no que se refere à sua capacidade de emitir alertas em tempo útil e de fornecer apoio na gestão da manutenção. Serão documentados os pontos fortes e fracos identificados ao longo do processo, bem como sugestões de melhorias e possibilidades de futura expansão do sistema.

1.4.1. Metodologia de desenvolvimento do *software*

Tendo em consideração a natureza estruturada do sistema a ser desenvolvido e o facto de os requisitos funcionais estarem previamente definidos com um elevado grau de detalhe, opta-se pela adoção do modelo de desenvolvimento em cascata. Esta metodologia apresenta uma abordagem sequencial, permitindo o controlo sistemático de cada etapa do projeto, o que é particularmente adequado em contextos de desenvolvimento de sistemas críticos, como é o caso da aviação.

O projeto de desenvolvimento da aplicação será estruturado nas seguintes fases:

Fase 1 – Levantamento e Análise de Requisitos

Nesta etapa será realizada a recolha e análise minuciosa dos requisitos operacionais da aplicação. A base de dados será desenhada com base nos manuais técnicos dos fabricantes e nas normas de manutenção aplicáveis à aviação ligeira. Esta fase resulta na definição clara das funcionalidades a implementar, como o registo de voos, os alertas de manutenção, a gestão de componentes e os perfis de utilizadores.

Fase 2 – Projeto do Sistema

Seguidamente, será concebida a arquitetura lógica e física da aplicação. Será feita a modelação da base de dados relacional, o desenho dos fluxos de interação entre os módulos, e a especificação da API RESTful (uma interface de programação de aplicações (API) que segue os princípios arquitetónicos do estilo *REST – Representational State Transfer*. Isto permite a comunicação entre sistemas cliente-servidor de forma independente) que fará a comunicação entre o *frontend* e o *backend*. O sistema será planeado para permitir uma futura escalabilidade e fácil manutenção.

Fase 3 – Implementação

Com base no projeto técnico, proceder-se-á ao desenvolvimento das funcionalidades previstas. Esta fase envolverá a criação de módulos independentes, como: Utilizadores, Aeronaves, Voos, Componentes, Manutenções e Alertas. Serão implementadas funcionalidades como login, registo de utilizadores, registo de voos, controlo de ciclos, emissão de alertas automáticos e *dashboards* personalizados para cada perfil.

Fase 4 – Testes e Validação Funcional

Após a codificação de cada módulo, serão efetuados testes unitários e testes de integração. Os testes servirão para verificar se as regras de negócio estão corretamente implementadas, em particular no que se refere ao controlo das horas de voo dos componentes, disparo de alertas e rastreabilidade das ações dos utilizadores (através de *logs*). Serão também realizadas simulações com dados reais e tendo em conta os feedback dos utilizadores-tipo.

Fase 5 – Implantação e Documentação

Concluída a validação interna, será realizada a instalação do sistema num ambiente de demonstração. A documentação técnica e funcional será elaborada nesta fase, incluindo o modelo de dados, manuais de utilização e instruções para a manutenção do sistema.

Fase 6 – Avaliação Final

Por fim, será feita uma avaliação global do sistema desenvolvido, com destaque para a eficácia dos alertas automáticos de manutenção, a usabilidade da interface e a utilidade da solução no apoio à decisão dos operadores e técnicos da aviação geral. Serão igualmente identificadas limitações e propostas de futuras melhorias.

1.5. Estrutura da Dissertação

Como foi referido anteriormente, esta tese tem como objetivo o desenvolvimento de uma aplicação/*software* direcionado aos proprietários de aeronaves com operação aérea não comercial. Tendo a seguinte estrutura:

1º Capítulo: Introdução

É efetuada uma descrição do projeto, a apresentação dos atuais problemas que visam a operação aérea não comercial e a definição de critérios para a resolução dos mesmos. É dividida em generalidades, motivação, âmbito e objetivo e metodologia.

2º Capítulo: Revisão literária

Neste capítulo é possível encontrar o enquadramento do tema e são apresentadas algumas soluções e as respetivas comparações com aplicações/*softwares* já existentes encontradas até o momento com a aplicação/*software* que pretendo desenvolver com este projeto.

3º Capítulo: Enquadramento, documentação e ativo de estudo

O capítulo constitui uma breve descrição, características e informações obtidas da aeronave Cessna C152 e um breve enquadramento da documentação aeronáutica relevantes para o enquadramento do tema.

4º Capítulo: *Maintenance Tracking System*

Descreve toda capacidade, informação referente e todo o processo de desenvolvimento da aplicação. Constitui uma aproximação mais detalha referente às características necessárias para o desenvolvimento da mesma e considerações técnicas para o projeto.

5ºCapítulo: Apresentação e discussão de resultados.

A apresentação e discussão de resultados é descrita e desenvolvida, com todos os resultados de estudo e análise realizados e obtidos tendo em conta a capacidade de resposta da aplicação e respetiva funcionalidade tendo em conta as premissas iniciais consideradas.

6º Capítulo: Conclusão da dissertação

Neste capítulo é desenvolvido todo um resumo da dissertação e constituição, apresentação das conclusões e recomendações para trabalhos futuros.

1.6.Introdução ao Capítulo 2

O capítulo que se segue consiste numa revisão literária, enquadramento do tema, um resumo de uma organização de gestão de aeronavegabilidade contínua que visa aeronaves com operação aérea comercial e de uma organização de aeronavegabilidade combinada que visa a operação aérea não comercial, o seu funcionamento e o objetivo, o estudo de alguns sistemas utilizados para o controlo da aeronavegabilidade contínua ou permanente e planeamento das manutenções e os documentos e manuais necessários. E por fim será abordado a temática da aviação de recreio nos Estados Unidos e os problemas enfrentados.

2. Revisão literária

2.1. Manutenção de aeronaves

Podemos definir a manutenção de aeronaves como sendo um conjunto de atividades técnicas, administrativas e operacionais, cujo objetivo é garantir que a aeronave esteja em condições de segurança e de eficiência de voo. Garantindo assim que a aeronave esteja em conformidade com as especificações do fabricante e os regulamentos das autoridades aeronáuticas.

As tarefas de manutenção são executadas de diferentes formas, podendo ser utilizadas diferentes metodologias, de acordo com o tipo de aeronave e a sua operação. Essas tarefas e a forma como elas devem ser executadas são descritas de forma detalhada em documentos e legislações divulgados pelas autoridades responsáveis pela aeronave e pelo Estado de Registro. Todas as aeronaves registadas na União Europeia são obrigadas a seguir as diretivas e os requisitos definidos pela autoridade europeia, a EASA [EUR-Lex. (Maio de 2025)].

Essas tarefas de manutenção visam a preservação e/ou restauração a aeronavegabilidade contínua, corrigindo defeitos, substituindo componentes desgastados e/ou com defeitos, realizando inspeções programadas e com isso prevenir a ocorrência de falhas.

2.2.Aeronavegabilidade

“Airworthy is the status of an aircraft, engine, propeller or part when it conforms to its approved design and is in a condition for safe operation.” [ICAO. (Abril de 2025)]

Aeronavegabilidade é a aptidão da aeronave para operações de voo em todos os ambientes possíveis e circunstâncias previsíveis para os quais a ela foi concebida [Airworthiness, I.-I. F. (Maio de 2025)]. É a condição técnica de uma aeronave que permite que ela voe com segurança, de acordo com os parâmetros estabelecidos pelos fabricantes e pelas autoridades de aviação civil competentes. Sendo assim um pilar fundamental para a segurança e eficiência operacional.

Ela depende de diversos fatores, incluindo a correta manutenção da aeronave, a execução de inspeções periódicas, o cumprimento das diretrizes de aeronavegabilidade emitidas pelas autoridades e a rastreabilidade de todos os eventos técnicos relacionados ao ciclo de vida da aeronave.

2.2.1. Aeronavegabilidade Permanente

São todos os processos que asseguram que em qualquer altura da vida de uma aeronave, ela cumpra as condições técnicas exigidas para a emissão do certificado de aeronavegabilidade e que esteja em condições de operar em segurança [Airworthiness, I.-I. F. (Maio de 2025)]. Para isso, são realizadas inspeções, manutenções programadas e outras medidas preventivas e corretivas.

A aeronavegabilidade permanente é o conjunto de processos através dos quais uma aeronave, motor, hélice ou peça cumprem os requisitos de aeronavegabilidade aplicáveis e se mantêm em condições de funcionamento seguro durante a sua vida útil [ICAO. (Abril de 2025)].

O proprietário da aeronave é responsável pela conformidade dos requisitos de aeronavegabilidade permanente e deve assegurar que o voo só é realizado, se [EUR-Lex. (Maio de 2025)]:

- A aeronave for mantida em boas condições de aeronavegabilidade;
- Todos os equipamentos operacionais e de emergência da aeronave se encontrarem corretamente instalados e estiverem operacionais ou claramente identificados como não operacionais;
- O certificado de aeronavegabilidade for válido;
- A manutenção da aeronave for executada em conformidade com o programa de manutenção aprovado pela autoridade competente.

2.3. Metodologia de gestão da manutenção e regulamentação aeronáutica

Existem diferentes tipos de metodologias e métodos utilizados na manutenção aeronáutica, consoante o tipo de aeronave, a sua operação. A categoria da aeronave é um dos requisitos que define a metodologia de manutenção para a mesma, estas metodologias e métodos estão descritos de forma pormenorizada em documentos e legislações divulgadas pela autoridade responsável pela aeronave ou estado de registo [EUR-Lex. (Maio de 2025)]. Quando uma aeronave é registada na União Europeia, ela é obrigada a seguir os requisitos aplicados pela EASA.

A EASA Parte-ML (*Maintenance Light*) é a metodologia de manutenção aprovada utilizada na Europa para aviação recreativa, desde março de 2020, substituindo assim a Parte M, Subparte F do Regulamento (EU) 1321/2014, baseando-se no conceito de manutenção preventiva programada, executada dentro de um *AMP – Approved Maintenance Programme*. [SOFEMA. (de 2025)] [Lienhart, P. (Maio de 2025)].

A Parte-ML é uma norma, usado exclusivamente para a aviação privada, sem fins comerciais e/ou de formação de voo e o peso máximo à descolagem da aeronave deve igual ou inferior a 2730 kg ou 1200 kg no caso de helicópteros certificados para um máximo de 4 ocupantes.

De acordo com esta norma, o piloto proprietário declara o programa de manutenção da aeronave (AMP), sem haver a necessidade de pedir aprovação, podendo agora executar as suas próprias ações de manutenção e pedir a emissão dos certificados de aeronavegabilidade da aeronave. A Parte-ML define regulamentação técnica específica que visa simplificar a manutenção e gestão da manutenção por parte do proprietário da aeronave. Contudo não permite que o mesmo emita os certificados de aeronavegabilidade. Estes podem apenas ser emitidos por uma organização Parte-CAMO ou Parte-CAO. Pelo proprietário assumir várias funções que poderiam ser contratadas a uma organização como referida anteriormente, a Parte-ML acaba por permitir ter uma grande flexibilidade para o proprietário, porque permite não ter tantos custos com a operação da mesma. Esta Parte ML não tem um sistema de qualidade incluído no seu regulamento, tendo os requisitos de qualidade na Parte CAO ou na Parte CAMO, o que a torna como uma opção mais económica para o uso e utilização privada da aeronave.

Como referido anteriormente, o piloto/proprietário pode contratar voluntariamente uma CAMO (entidade de gestão da aeronavegabilidade permanente) ou CAO, passando assim toda a responsabilidade pela manutenção e gestão da aeronavegabilidade, assim como a emissão do seu certificado de aeronavegabilidade, da sua aeronave para esta organização. Esta organização passará a ser responsável pelo acompanhamento do estado de manutenção da aeronave, pela gestão dos pacotes de trabalho e pela emissão do Certificado de Aeronavegabilidade ou *Airworthiness Review Certificate (ARC)*. É de notar que o programa de manutenção da aeronave passará a estar sujeito a aprovação da entidade de gestão da aeronavegabilidade permanente (CAMO) ou entidade de gestão de aeronavegabilidade combinada (CAO) contratada. [SOFEMA. (de 2025)] [Lienhart, P. (Maio de 2025)]

Em síntese, podemos concluir que atualmente a Parte-ML é a metodologia de manutenção mais utilizada para aeronaves de recreio e uso privado registradas sob jurisdição da EASA, oferecendo uma maior autonomia ao proprietário da aeronave, uma menor carga administrativa, custos reduzidos e a conformidade com os padrões mínimos de segurança, dado ao proprietário a possibilidade de contratar uma organização de forma voluntária, caso o mesmo queira melhorar os padrões de segurança.

2.4. Engenharia de *Software* aplicada à Aviação

A engenharia de *software* para sistemas críticos exige práticas robustas de concepção, documentação e validação para garantir fiabilidade e segurança. De acordo com Gorton [Gorton, I. (Abril de 2025)], arquiteturas de software para sistemas complexos devem ser moduladas para permitir escalabilidade, manutenção evolutiva e resiliência a falhas.

No contexto da aviação, essas práticas são ainda mais rigorosas, uma vez que os sistemas embarcados — como FMS (*Flight Management Systems*), FADEC (*Full Authority Digital Engine Control*), e sistemas de navegação e vigilância — são classificados como sistemas críticos à segurança (*safety-critical systems*). Por isso, o desenvolvimento de software aeronáutico segue normas específicas como a DO-178C, que define níveis de criticidade e processos de verificação e validação para o ciclo de vida do software [WNDRVR. (Abril de 2025)].

A DO-178C, intitulada "*Software Considerations in Airborne Systems and Equipment Certification*", é uma norma internacional desenvolvida pela RTCA, em conjunto com a EUROCAE, que estabelece os critérios e processos para o desenvolvimento, verificação e certificação de software embarcado em sistemas aeronáuticos. A norma tem como principal objetivo garantir que o software utilizado em aeronaves seja seguro, confiável e esteja em conformidade com os requisitos de aeronavegabilidade das autoridades reguladoras, como a FAA (*Federal Aviation Administration*) e a EASA (*European Union Aviation Safety Agency*). [WNDRVR. (Abril de 2025)]

A DO-178C organiza o processo de desenvolvimento de software em diversas etapas, que incluem planejamento, requisitos, design, codificação, verificação, gestão de configuração e garantia de qualidade. Um dos seus elementos centrais é a definição dos *Design Assurance Levels (DAL)*, que variam de A (nível mais crítico) a E (menos crítico), dependendo do impacto potencial de uma falha de software na segurança da aeronave. [WNDRVR. (Abril de 2025)]

A norma também introduz a obrigatoriedade de rastreabilidade bidirecional entre os requisitos e os testes, além da documentação detalhada e da evidência de conformidade. A DO-178C é considerada uma norma base para a certificação de software aeronáutico em todo o mundo e possui suplementos que ampliam sua aplicação, como o DO-330 (ferramentas de qualificação), DO-331 (desenvolvimento baseado em modelos) e DO-332 (linguagens formais). [WNDRVR. (Abril de 2025)].

A rastreabilidade dos requisitos, testes automatizados, e métodos formais são frequentemente utilizados para garantir conformidade com os padrões regulatórios e operacionais. A integração segura entre hardware e software, a gestão de falhas, a tolerância a erros e a cibersegurança também são aspectos essenciais considerados desde as fases iniciais do projeto até a certificação e operação.

2.5.Sistemas e métodos atuais de gestão de manutenção em ambiente controlado

A manutenção eficaz de aeronaves é um pilar essencial para a segurança, a fiabilidade e a eficiência operacional no setor da aviação. Com o crescimento constante da frota aérea mundial e o aumento da complexidade das operações de manutenção, tornou-se indispensável o uso de sistemas informatizados que possibilitem uma gestão integrada e em tempo real das atividades de manutenção e da aeronavegabilidade contínua das aeronaves, havendo um acompanhamento e o registo sistemático de todas as atividades relacionadas com a manutenção da aeronave.

Nesse contexto, softwares especializados como o AMOS (*Aircraft Maintenance and Engineering System*), o OASES (*Open Aviation Strategic Engineering System*) e o CAMP (*Computerized Aircraft Maintenance Program*) desempenham um papel fundamental. Esses sistemas são amplamente utilizados por companhias aéreas, operadores privados, organizações de manutenção (MROs) e entidades de gestão de aeronavegabilidade (CAMOs), com o objetivo de garantir a conformidade regulatória, otimizar recursos e reduzir custos operacionais.

O AMOS (*Aircraft Maintenance and Engineering System*) é um dos softwares de gestão de manutenção de aeronaves mais completos e utilizados no setor da aviação, desenvolvido pela Swiss Aviation Software. É voltado tanto para operadores de frota como para organizações de manutenção (MRO), oferecendo uma solução integrada para engenharia, logística, manutenção, planeamento e documentação técnica. Tem como funcionalidade a gestão de aeronavegabilidade contínua (CAMO), o planeamento de manutenções programadas e não programadas, o controlo de inventário/logística, faz o registo completo de tarefas de manutenção, peças e tempos de voo, elabora os relatórios técnicos e suporte a auditorias, faz a Integração com ERP, EFB e outras plataformas. Fazendo uma gestão integrada de manutenção, engenharia e logística e cria módulos para controle de produção, manutenção de componentes, recursos humanos e finanças. [Software, S. A. (Maio de 2025)] [SoftwareConnect. (Maio de 2025)]

Mas por outro lado, este sistema tem algumas limitações, tais como um alto custo de aquisição e implementação; curva de aprendizagem muito acentuada para novos utilizadores, exigindo formação especializada; requer infraestrutura robusta (especialmente em grandes operadores) e tem uma interface considerada complexa por alguns utilizadores. [Capterra. (Maio de 2025)] [Software, S. A. (Maio de 2025)] [SoftwareConnect. (Maio de 2025)]

O OASES (*Open Aviation Strategic Engineering System*) é um sistema desenvolvido pela empresa britânica Commsoft, é um sistema de gestão de manutenção modular e altamente configurável. É usado principalmente por companhias de médio porte, operadores de frotas

menores, CAMOs e MROs. Tendo como funcionalidade a gestão de CAMO e MRO, o planeamento e a execução de manutenção, o controlo de inventário e componentes, o registo de tempos de voo, ciclos e inspeções, tendo uma interface de utilizador mais simples e intuitiva que o AMOS. Apresentando uma solução integrada para controle de manutenção e aeronavegabilidade e módulos para planeamento, manutenção de linha, controle de manutenção, gestão de inventário e logística internacional.

Por outro lado, o OASES apresenta uma menor profundidade de integração com sistemas de BackOffice e outros sistemas e ferramentas de análise de dados, o sistema tem alguns módulos que são vendidos separadamente, o que pode elevar o custo total, o sistema tem uma menor escalabilidade para companhias de grande porte. [OASES. (Maio de 2025)] [SoftwareConnect. (Maio de 2025)]

O CAMP (*Computerized Aircraft Maintenance Program*) é uma solução baseada na nuvem, muito popular entre frotas privadas, aviação executiva e pequenos operadores comerciais, especialmente na América do Norte. É focado na gestão da aeronavegabilidade contínua e no cumprimento das diretrizes regulatórias. Tendo como principal funcionalidade a gestão completa da aeronavegabilidade, permite o controlo e a visualização de tarefas pendentes, emite alertas automatizados para manutenções programadas e faz a gestão de inventário e registos de manutenção digitais, fazendo também o upload de documentos e integração com OEMs. [CAMP. (Maio de 2025)] [VERYON. (Maio de 2025)]

Este sistema também apresenta algumas limitações, tais como um menor capacidades de rastreamento em tempo real, dificultando o monitoramento eficaz de ativos e por isso ela não tão indicado para MROs, existe uma dependência total de conexão à internet, visto que não tem modo offline e tem um desempenho lento na geração de relatórios e extração de dados para ferramentas de análise, sua customização é limitada quando comparado com AMOS ou OASES e por ter o foco principal em aviação executiva, o que pode ser limitativo para operadores comerciais complexos. [Veryon. (Maio de 2025)]

É importante realçar que ainda existem muitas organizações que utilizam documentação técnica e registos manuais/físicos ou digitais em planilhas para o controlo de manutenções e inspeções, especialmente pequenas operações.

3. Cessna 152

O Cessna 152 é uma aeronave monomotor de asa alta, com trem de aterragem triciclo fixo, projetada para propósitos gerais e certificada na categoria utilitária. Tem de dois lugares, projetada principalmente para instrução de voo, treinamentos básicos, sendo aprovada para fazer voos VFR/IFR diurnos e noturnos quando equipada conforme os regulamentos aplicáveis. Foi introduzido em 1977 como sucessor do popular Cessna 150 e o modelo 152 foi amplamente utilizado por escolas de aviação devido à sua fiabilidade, manutenção simples e custo

operacional reduzido. Mais de 7.500 unidades foram produzidas entre 1977 e 1985, mostrando assim a sua popularidade. [Aviators, L. I. (Maio de 2025)]

O Cessna 152 apresenta uma configuração de asa alta (*high-wing*), com estrutura de alumínio totalmente metálica e trem de aterragem do tipo triciclo fixo, proporcionando uma boa visibilidade para o piloto e estabilidade durante o pouso e taxiing. A fuselagem é construída em monocoque e as asas são revestidas com painéis metálicos tipo “semi-monocoque”, com tanques de combustível integrados. [Aviators, L. I. (Maio de 2025)]

A aeronave é equipada com um motor Lycoming O-235-L2C, com 4 cilindros opostos horizontalmente, refrigerado a ar, e capaz de produzir 110 HP a 2.550 RPM. O motor aciona uma hélice metálica de passo fixo. Tendo dois tanques localizados nas asas, com capacidade total de 98 litros e a alimentação é feita por gravidade. [Aviators, L. I. (Maio de 2025)]

A velocidade de cruzeiro é de aproximadamente 107 nós a 75% de potência a 2.500 pés, uma velocidade máxima de cerca de 110 nós, uma razão de subida de 715 pés por minuto e um alcance máximo de cerca de 415 milhas náuticas (769 km) com 75% de potência e tanques cheios, tendo um teto operacional de 14.700 pés. [Aviators, L. I. (Maio de 2025)]

O Cessna 152 tem um peso máximo de decolagem (MTOW) de 1.670 libras (757 kg) e o seu peso vazio padrão é de aproximadamente 1.081 libras (490 kg). [Aviators, L. I. (Maio de 2025)]

3.1.ATA 100

O ATA 100 é um sistema de referência desenvolvido pela *Air Transport Association* em junho de 1956. Trata-se de um sistema de numeração que contém normas de referência comuns para a descrição de sistemas de aeronaves comerciais e todas as informações relacionadas com o sistema, componentes, diagramas de blocos, diagramas de cablagem e todas as informações técnicas relativas a um sistema complexo de uma aeronave. Este sistema de numeração normalizado permite uma melhor compreensão, comunicação e aprendizagem fácil para os técnicos de manutenção, engenheiros, pilotos, entre outros, sendo a base de qualquer referência feita em quase todas as operações diárias das aeronaves, manuais e documentação. [Authority, C. A. (Maio de 2025)] [SKYbrary. (Maio de 2025)]

A vantagem significativa deste sistema é o facto de os números dos capítulos de um sistema serem os mesmos em todas as aeronaves, independentemente do fabricante, o que significa que não há margem para confusão quando se fala de um capítulo com um número de referência. Por este motivo, todo o pessoal técnico sabe a que sistema se está a referir e pode facilmente compreender toda a documentação e informações adequadas relativas aos sistemas da aeronave. [Authority, C. A. (Maio de 2025)] [SKYbrary. (Maio de 2025)]

Além disso, o ATA 100 contribui diretamente para a padronização da indústria aeronáutica, otimizando a formação de novos profissionais, facilitando auditorias e inspeções de conformidade, e promovendo a interoperabilidade entre companhias aéreas, fabricantes de aeronaves, fornecedores de peças e organizações de manutenção. A estrutura lógica e

hierárquica do sistema — dividida em capítulos (por exemplo, ATA 24 para sistema elétrico, ATA 32 para trem de aterragem, ATA 61 para hélice, etc.) — permite rápida a localização da informação técnica e assegura uma gestão mais eficiente da manutenção e da aeronavegabilidade. Este sistema foi também a base para versões mais modernas como o ATA iSpec 2200 e o S1000D, que visam a digitalização e a gestão integrada da documentação técnica em ambientes digitais. [Authority, C. A. (Maio de 2025)] [SKYbrary. (Maio de 2025)]

3.2.Documentação da aeronave

Para poder operar, qualquer aeronave deve possuir um Certificado de Aeronavegabilidade, também conhecido como *CofA* (*Certificate of Airworthiness*). O *CofA* é um documento formal emitido pela Autoridade Nacional de Aviação do estado de registro (*National Aviation Authority*), que certifica que a aeronave está em conformidade com os requisitos técnicos e de segurança estabelecidos pelas normas de aeronavegabilidade vigentes, e, portanto, está apta a operar em segurança. [EUR-Lex. (Maio de 2025)] [ICAO. (Abril de 2025)]

A emissão deste certificado depende da verificação de que a aeronave cumpre com o projeto de tipo certificado (*Type Certificate*) aprovado e que está a ser mantida e operada em conformidade com os regulamentos aplicáveis. Isso inclui a conformidade com o programa de manutenção, registos técnicos atualizados, instalação de equipamentos mandatórios e a realização de inspeções periódicas.

Sem um *CofA* válido, a aeronave é considerada em estado não aeronavegável e, por consequência, está proibida de operar, exceto em casos especiais em que seja emitida uma Permissão Especial de Voo (*Permit to Fly*) para situações restritas, como testes ou ferry flights. [EUR-Lex. (Maio de 2025)]

Importa referir que nem todas as aeronaves requerem um *CofA* padrão, uma vez que isso depende do tipo e categoria da aeronave e da natureza da operação. Por exemplo, aeronaves de construção amadora ou utilizadas exclusivamente para aviação recreativa podem operar sob certificados de aeronavegabilidade específicos ou limitados, como os certificados de aeronavegabilidade restritos (*Restricted CofA*) ou em condições abrangidas pelo *Permit to Fly*, conforme previsto pelas regulamentações da EASA ou da FAA. [EUR-Lex. (Maio de 2025)] [SOFEMA. (Maio de 2025)]

A validade do *CofA* é normalmente condicionada à sua revalidação periódica, e geralmente, deve ser efetuada anualmente, sob responsabilidade de uma Organização de Gestão da Aeronavegabilidade Contínua (*CAMO – Continuing Airworthiness Management Organisation*) aprovada ou combinada (*CAO*) dependendo da operação aérea. A entidade de gestão da aeronavegabilidade permanente (*CAMO/CAO*) realiza as inspeções técnicas, análise de conformidade e documentação necessárias para garantir a aeronavegabilidade permanente da aeronave. Após dois ciclos de revalidação pela entidade de gestão da aeronavegabilidade permanente, é exigida uma nova revalidação formal pela *NAA* para garantir a conformidade de longo prazo com os requisitos regulamentares. [EUR-Lex. (Maio de 2025)]

Para que a aeronave esteja em perfeitas condições de aeronavegabilidade, devem ser cumpridos os requisitos primários presentes na *Commission Regulation (EU) 1321/2014*, e estes constituem o seguinte:

- Programa de manutenção aprovada (AMP)
- Documentação técnica da aeronave e respetivos componentes.
- Documentação das anomalias pendentes, das reparações e modificações feitas na aeronave.
- Documentação de boletins de serviço e diretivas de aeronavegabilidade.

Alguns documentos como o *MPD- Maintenance Planning Document* ou documento de planeamento de manutenção, *AMM – Aircraft Maintenance Manual* ou Manual de manutenção da aeronave, a *MMEL- Master Minimum Equipment List* ou Lista Principal de Equipamento Mínimo, o *SRM- Structure Repair Manual* ou Manual de Reparação de Estruturas e o Manual de Isolamento de Falhas são muitos importantes, visto que contêm todas as informações mais importantes e cruciais no quesito da aeronavegabilidade.

3.2.1. MSG-3 e *MPD – Maintenance Planing Document*

A metodologia usada que o fabricante da aeronave usa para o desenvolvimento do *MPD* é o *MSG-3*, ou *Maintenance Steering Group* terceira geração. Esta metodologia desenvolvida pela *Air Transport Association* que define a lógica de construção dos programas de manutenção de aeronaves. É usada uma abordagem analítica baseada em fiabilidade e risco para determinar quais tarefas de manutenção são realmente necessárias para garantir a segurança e a aeronavegabilidade contínua, analisando sistemas, estruturas e componentes de forma estruturada. A aplicação prática dessa metodologia pelos fabricantes resulta no *MPD*, ou *Maintenance Planning Document*, como referido anteriormente. É neste documento no qual estão reunidas todas as tarefas de manutenção recomendadas, já com os respetivos intervalos de execução definidos em horas de voo, ciclos ou calendário. O *MPD* funciona, portanto, como a tradução prática e real do *MSG-3* e serve de referência direta para as operadoras e organizações de gestão de aeronavegabilidade, que a partir dele elaboram o seu Programa de Manutenção da Aeronave (AMP). Este programa é ajustado às condições específicas de operação de cada companhia e deve ser submetido e aprovado pela *NAA*. A metodologia apresentada permite dividir as tarefas de manutenção em quatro grupos distintos. Estes são [Certification, F. A. (Maio de 2025)]:

- *O/C-On Condition*,
- *C/M-Condition Monitoring*
- *H/T-Hart Time*.
- *S/T-Soft Time*.

A *manutenção On Condition (OC)* é uma abordagem preventiva que permite que componentes permaneçam instalados na aeronave enquanto continuarem a funcionar dentro dos

parâmetros aceitáveis de desempenho. Diferente da manutenção baseada em tempo (*Hard Time*), o princípio do OC é que a substituição ou remoção do componente só ocorra quando evidências objetivas indicarem degradação ou risco iminente de falha. Essa decisão é fundamentada em inspeções periódicas, testes funcionais, medições ou análises visuais e técnicas, conforme detalhado no *AMM (Aircraft Maintenance Manual)*. [Certification, F. A. (Maio de 2025)]

A monitorização é realizada com base em critérios técnicos estabelecidos pelo fabricante, que especifica quais parâmetros devem ser verificados, com que frequência e através de que métodos, como análise de vibração, desgaste, desempenho funcional ou outros indicadores físicos e operacionais. O objetivo é identificar sinais de degradação antes que o componente se torne *INOP* (Inoperável) em operação normal, garantindo assim a continuidade da aeronavegabilidade e a segurança do voo, sem realizar trocas prematuras desnecessárias. [Certification, F. A. (Maio de 2025)]

A manutenção por *Condition Monitoring (CM)* é uma estratégia preventiva baseada na análise contínua do desempenho de componentes críticos da aeronave, visando identificar tendências de deterioração antes da ocorrência de falhas. Na organização *CAMO (Continuing Airworthiness Management Organisation)*, os engenheiros monitorizam dados operacionais e parâmetros técnicos de sistemas como os motores, *APU* (Unidade Auxiliar de Potência), trem de aterragem e outros sistemas vitais, cuja falha pode comprometer seriamente a segurança de voo ou a eficiência da operação. [Certification, F. A. (Maio de 2025)]

A abordagem *CM* não se baseia em substituições programadas, mas sim em análises de fiabilidade, relatórios de falhas, tendências de desempenho e indicadores técnicos, frequentemente obtidos através de sistemas integrados de monitoramento e telemetria. Esses dados permitem identificar anomalias ou padrões de degradação que justificam ações corretivas antes da falha funcional. Os componentes monitorizados possuem intervalos flexíveis de avaliação, definidos por critérios estatísticos e por recomendações do fabricante (*OEM*), proporcionando maior eficiência na gestão da manutenção, redução de custos e aumento da disponibilidade da aeronave.

A manutenção *Hard Time (H/T)* refere-se a um método preventivo em que determinados componentes ou sistemas da aeronave devem ser obrigatoriamente inspecionados, revisados ou substituídos em intervalos fixos e previamente definidos, independentemente da sua condição aparente. Esses intervalos são estabelecidos com base nas recomendações do fabricante e aprovados pelas autoridades aeronáuticas competentes. [Certification, F. A. (Maio de 2025)]

A manutenção *H/T* é geralmente aplicada a componentes críticos, cuja falha pode afetar diretamente a segurança do voo. Ela é dividida em duas categorias principais: peças com limite de vida útil (*LLP – Life Limited Parts*), que devem ser substituídas obrigatoriamente ao atingir um número específico de ciclos, horas de voo ou calendário, e componentes, que exigem verificações regulares e só são substituídos em caso de deterioração identificada. No entanto, mesmo estes últimos seguem uma frequência rígida de inspeção, definida com base em critérios de segurança e desempenho históricos. [Certification, F. A. (Maio de 2025)]

Este método, embora mais conservador, é eficaz para garantir a fiabilidade de peças de desgaste previsível, sendo amplamente utilizado na aviação comercial para componentes estruturais, sistemas hidráulicos e de controlo de voo, e certas partes do motor.

A manutenção *Soft Time* aplica-se a componentes que não afetam diretamente a segurança operacional da aeronave, sendo geralmente ligados ao conforto dos passageiros, estética da cabine ou funcionalidades secundárias. Esses itens incluem, por exemplo, assentos, iluminação de cabine, painéis internos, sistemas de entretenimento ou pequenos acessórios. As tarefas de manutenção desses componentes são definidas com base na conveniência, experiência operacional e nas prioridades do operador, podendo ser ajustadas conforme o calendário de manutenção da frota. [Certification, F. A. (Maio de 2025)]

Embora alguns desses itens possam ter recomendações de manutenção do fabricante, elas não são obrigatórias, o que dá ao operador maior flexibilidade na definição de prazos e procedimentos. Essa abordagem é útil para otimizar recursos, reduzir custos e adaptar a manutenção às preferências do cliente e à estratégia comercial, desde que não comprometa a conformidade regulamentar ou o bem-estar dos passageiros.

3.2.2. AMM – Manual de Manutenção de Aeronave

O AMM é o Manual de Manutenção da Aeronave, um documento técnico essencial e que contém instruções detalhadas para a execução correta das tarefas de manutenção aprovadas para uma aeronave específica. Embora exista uma estrutura padrão, definido pelo fabricante para o conteúdo do AMM, a sua aplicação prática pode variar entre operadores, uma vez que o *AMP (Aircraft Maintenance Program)* é adaptado de acordo com os objetivos operacionais, estratégias comerciais e perfil da operação de cada empresa. [SKYbrary. (Maio de 2025)]

O AMM inclui tarefas de manutenção programada, de lubrificação, de assistência técnica, inspeções, substituição de componentes, testes funcionais e procedimentos de *troubleshooting*, organizadas por sistema da aeronave (*ATA Chapter*). No entanto, todas as tarefas obrigatórias que afetam a segurança de voo estão incluídas nos programas de manutenção aprovados e estão diretamente associados aos requisitos mínimos definidos no *MPD (Maintenance Planning Document)*. O AMM funciona, assim, como a referência técnica principal para a execução física dessas tarefas, assegurando que o trabalho seja realizado em conformidade com os padrões estabelecidos pelo fabricante e pela autoridade aeronáutica. [SKYbrary. (Maio de 2025)]

Além disso, o AMM organiza as tarefas por zona da aeronave, sistema ou componente, permitindo uma identificação clara e eficiente das instruções aplicáveis a cada parte, contudo, nem todas as tarefas são consideradas prioritárias, ou com a mesma frequência de execução (conceito de *Soft Time*), especialmente em casos onde há uma distinção clara entre operadores que privilegiam o conforto a bordo e outros que visam redução de custos operacionais. [SKYbrary. (26 de Maio de 2025)]

Em síntese, o fabricante da Aeronave é o responsável pelo desenvolvimento do *MPD*, que contém as tarefas de manutenção recomendadas; que por sua vez, é o documento que serve de base para a criação do *AMP (EASA)*. O *AMM (Aircraft Maintenance Manual)* é o manual que fornece instruções detalhadas para a execução das tarefas de manutenção listadas no *AMP (EASA)* que contém diretivas, planeamento e toda a programação de inspeções adaptado às necessidades específicas do operador, baseado nas recomendações do *MPD*; e a execução da manutenção e o resultado da aplicação conjunta das instruções do *AMM* e das adaptações do *AMP (EASA)*.

3.2.3. MMEL-Master Minimum Equipment List e MEL- Minimum Equipment List

A *MMEL (Master Minimum Equipment List)* é um documento técnico desenvolvido pelo fabricante da aeronave (*OEM*), em colaboração com a autoridade aeronáutica do estado de projeto (como a *FAA* ou *EASA*), que estabelece a lista de equipamentos, sistemas e instrumentos que podem estar inoperativos no início de um voo sem comprometer a segurança da operação. A *MMEL* serve como um modelo específico da aeronave, e serve como referência base para a elaboração da *MEL (Minimum Equipment List)* por parte dos operadores. [Authority, C. A. (Maio de 2025)] [SKYbrary. (Maio de 2025)]

A *MMEL* especifica condições de despacho, limitações operacionais, e procedimentos compensatórios que devem ser seguidos quando um determinado item está inoperativo. Essas condições incluem, por exemplo, reduções no número de passageiros transportados, restrições meteorológicas, ou exigência de sistemas redundantes operacionais. Embora forneça uma margem de flexibilidade, a *MMEL* é conservadora e sempre prioriza a segurança do voo, a proteção dos passageiros e a integridade da aeronave.

Além disso, a *MMEL* é organizada de acordo com a estrutura ATA 100, permitindo uma navegação sistemática e padronizada das informações. A sua aplicação é fundamental para manter o equilíbrio entre a eficiência operacional e o cumprimento das normas de aeronavegabilidade, sendo periodicamente atualizada com base em dados de fiabilidade e experiência operacional.

A *MEL (Minimum Equipment List)* é um documento operacional essencial. Ela é desenvolvida pelo operador com base na *MMEL (Master Minimum Equipment List)*. A *MEL* é adaptada da *MMEL* com base na configuração específica de cada aeronave da frota do operador e deve ser formalmente aprovada pela autoridade nacional de aviação civil (*NAA*) responsável pelo certificado de aeronavegabilidade. [Authority, C. A. (Maio de 2025)] [SKYbrary. (Maio de 2025)]

A *MEL* permite alguma flexibilidade operacional, mas impõe limitações e procedimentos de compensação (como restrições de performance, revisões pré-voo adicionais ou prazos para correção da falha, como já referido anteriormente). Em caso de discrepâncias não autorizadas pela *MEL*, a aeronave deve ser imediatamente retirada de serviço (*AOG* ou *Grounded*) até que

a anomalia seja resolvida. A identificação dos itens da MEL também pode ser feita em alguns casos por meio de mensagens dos sistemas de alerta da aeronave como o *EICAS (Engine Indicating and Crew Alerting System)* ou o *CAS (Crew Alerting System)*.

A MEL também inclui detalhes sobre a quantidade mínima de unidades operacionais exigidas, limitações aplicáveis e ações de manutenção ou operacionais exigidas para cada item. Além disso, ela contribui significativamente para uma gestão segura de falhas técnicas, otimizando a disponibilidade da frota sem comprometer a segurança do voo.

A constituição da MEL (*Minimum Equipment List*) segue uma estrutura padronizada que inclui os componentes ou sistemas que podem ser deferidos, a sua descrição funcional, a quantidade instalada na aeronave, a quantidade mínima operacional (*MOC - Minimum Operational Configuration*) permitida para despacho, bem como a data limite para correção (vencimento) do item deferido. Cada item da MEL é também categorizado por letras de A a D, que representam os limites de tempo permitidos para operação com a falha, a contar a partir da data em que a avaria foi identificada:

- Categoria A: prazo definido individualmente (em dias, ciclos ou horas de voo),
- Categoria B: até 3 dias,
- Categoria C: até 10 dias,
- Categoria D: até 120 dias.

Se o item não for corrigido dentro do prazo definido, a aeronave entra em estado *AOG (Aircraft on Ground)* e está proibida de voar até que a falha seja resolvida ou prorrogada com autorização da autoridade aeronáutica nacional (*NAA*). [Authority, C. A. (Maio de 2025)] [SKYbrary. (Maio de 2025)]

Além disso, cada item MEL pode requerer ações específicas antes do voo, classificadas como:

- Procedimentos de manutenção (*M*) – a serem executados por pessoal qualificado de manutenção, conforme definido no *AMM* ou *MEL*;
- Procedimentos operacionais (*O*) – a serem cumpridos pela tripulação de voo, geralmente envolvendo ações como desligar certos sistemas, *Checklists* alternativos ou aplicação de restrições operacionais.

Estas ações asseguram que a aeronave continue a operar de forma segura e em conformidade com as regulamentações vigentes, mesmo com certos equipamentos inoperativos.

3.2.4 Manual de isolamento de falhas

O Manual de Isolamento de Falhas (*FIM*) é uma documentação técnica essencial fornecida pelo fabricante (*OEM (Original Equipment Manufacturer)*) e integra a literatura de

manutenção aprovada de uma aeronave. A sua principal função é guiar o pessoal de manutenção na identificação e resolução de falhas ou discrepâncias detetadas antes, durante ou após as operações de voo. Quando uma anomalia se torna recorrente e não pode ser resolvida com os procedimentos indicados no *FIM*, a intervenção direta do fabricante é geralmente necessária. [SKYbrary. (29 de Maio de 2025)]

A resolução de falhas, especialmente em aeronaves modernas, é fortemente baseada na leitura e interpretação dos códigos de falha e mensagens emitidos pelo Sistema de Diagnóstico de Manutenção a Bordo, descrito no *ATA* Capítulo 45. Esses dados são obtidos por meio de interfaces como o *MCDU*, *ECAM*, *EICAS* ou sistemas de manutenção centralizados, dependendo do tipo e fabricante da aeronave.

O *FIM* apresenta uma estrutura lógica de diagnóstico organizado por capítulo *ATA*, e geralmente inclui:

- Resumo e descrição da mensagem de falha,
- Descrição do sistema envolvido,
- Diagramas esquemáticos e elétricos,
- Causas prováveis da falha,
- Procedimentos passo a passo de *troubleshooting*,
- Ligações para testes específicos e recomendações baseadas em falhas conhecidas,
- Técnicas de verificação cruzada com outros manuais, como *AMM* e *WDM*.

Além disso, o *FIM* permite a aplicação de procedimentos sistemáticos para isolar e validar uma falha, reduzindo o tempo de inatividade da aeronave e evitando substituições desnecessárias de componentes. O uso adequado deste manual contribui diretamente para a eficiência da manutenção corretiva e preventiva, sendo uma ferramenta indispensável para as atividades de engenharia e manutenção em linha e em base. [SKYbrary. (Maio de 2025)]

4. Maintenance Tracking System

Como mencionado anteriormente, o *Maintenance Tracking System* será um sistema móvel desenvolvido para realizar o controlo, a programação e o planeamento da manutenção de aeronaves de forma automatizada, precisa e eficiente. O público-alvo deste sistema são os proprietários de aeronaves de recreio que, muitas vezes, não dispõem de um centro próprio de controlo de manutenção.

A aplicação visa fornecer a esses operadores uma ferramenta acessível e confiável que permita gerir as atividades de manutenção com maior rigor, apoiando-se em dados operacionais registados automaticamente ao final de cada voo. Dessa forma, promove-se uma gestão proativa

da aeronavegabilidade, reduzindo o risco de falhas operacionais e melhorando a segurança e a conformidade com os requisitos regulamentares.

Este sistema vai ter as seguintes características:

- Uma *arquitetura orientada em eventos* permitindo a detecção precoce de condições anómalas e a coordenação automática de ações. Cada registo de voo ou atualização de horas de componente dispara eventos que geram notificações em tempo real para o piloto e técnico de manutenção da aeronave, garantindo intervenções preventivas antes de falhas críticas.

Foi adotado uma estrutura que permite a comunicação entre as diferentes partes da aplicação – a parte que o utilizador vê (interface) e a parte interna que processa os dados. Para isso, utilizou-se um sistema de base de dados chamado PostgreSQL, que armazena de forma segura todas as informações sobre as aeronaves, os voos e as manutenções. Este Sistema garante que os dados são registados corretamente, que não se perdem informações e que é possível lidar com grandes quantidades de dados sem haver falhas. Sempre que o utilizador executa uma ação, como registar um voo ou consultar os componentes pendentes de manutenção, o sistema verifica primeiro se todos os dados estão corretos antes de os guardar, assegurando assim a consistência e a segurança da informação.

- Um *design centrado no utilizador* tendo uma interface intuitiva e que minimiza a probabilidade de erros humanos, contribuindo assim para o aceleração das tarefas críticas. Neste projeto, cada perfil (piloto, técnico de manutenção e o administrador do sistema) dispõe de dashboards personalizados: por exemplo, o piloto vê alertas visuais de manutenção pendente antes de criar um voo, e o técnico recebe painéis com peças que ultrapassaram o limite de horas definidas.

4.1.Desenvolvimento da aplicação

Neste capítulo apresenta-se em detalhe a aplicação idealizada para atender às necessidades de pequenos operadores de aeronaves recreativas. A partir de entrevistas semiestruturadas com um engenheiro de manutenção aeronáutica, e da conversão direta do plano OEM do Cessna 152 em regras de negócio, o sistema foi desenhado para automatizar tarefas repetitivas, fortalecer a rastreabilidade e garantir a conformidade em cada etapa do ciclo de manutenção.

4.1.1. Descrição do Sistema

O sistema foi concebido para oferecer aos pequenos operadores de aeronaves recreativas uma ferramenta única, simples e rigorosamente conforme com os regulamentos internacionais de aeronavegabilidade. A seguir, apresentam-se os elementos fundamentais:

4.1.1.1.Registo de Aeronaves e Componentes

A aplicação/sistema permite registar aeronaves (matrícula, fabricante, modelo) e os respetivos componentes (motores, trem de aterragem, filtros, etc.), incluindo parâmetros de vida útil (horas e ciclos). Cada registo define os intervalos de manutenção obrigatória extraídos do plano OEM do Cessna 152.

4.1.1.2.Registo de Voos

A Interface na qual o piloto através de um formulário intuitivo preenche a matrícula, o aeroporto de partida e chegada e o horário do voo. O sistema atualiza automaticamente o uso de cada componente associado, calculando as percentagens remanescentes e armazenando o histórico de voos para auditoria.

4.1.1.3.Gestão de Manutenção

O Painel destinado aos técnicos de manutenção, tem uma lista dinâmica de due-lists: tarefas que atingem 10 %, 50 % ou 100 % do limite programado. A partir desse painel, o técnico abre ordens de serviço, regista procedimentos e adiciona notas complementares. Cada intervenção fica vinculada ao voo e ao componente correspondente.

4.1.1.4.Emissor de Alertas

Os alertas do sistema são enviados imediatamente, assim que acontece algum evento importante — por exemplo, durante um voo ou uma manutenção. Esses alertas seguem um conjunto de regras que podem ser ajustadas através de um formato chamado JSON, que é simplesmente uma forma organizada de escrever informações que o sistema consegue entender facilmente.

Assim, quando o sistema deteta que o uso de uma peça ou componente ultrapassou o limite permitido, ele reconhece essa situação através dessas regras e gera automaticamente um alerta para avisar o utilizador.

4.1.1.5. Relatórios e Exportação

O sistema gera relatórios técnicos em formato PDF, tendo o histórico dos voos, as intervenções e os componentes substituídos. Esses documentos, em português, estão prontos para serem apresentados em auditorias ou utilizados em análises de desempenho.

4.1.1.6. Perfis dos Utilizadores

O sistema adota um modelo de controlo de acesso por funções, definindo quatro perfis:

- **Piloto:** regista voos, consulta alertas e o histórico de utilização da aeronave.
- **Técnico de manutenção:** visualiza due-lists, regista manutenções, anexa evidências e atualiza o status dos componentes.
- **Supervisor:** valida ou rejeita as manutenções, acompanha métricas de performance (tempo de resposta, número de manutenções pendentes) e exporta relatórios consolidados.
- **Administrador:** configura regras de alerta, gere utilizadores, ajusta parâmetros do plano OEM do Cessna 152 e monitoriza logs de auditoria.

4.1.2. Levantamento Requisitos

O levantamento de requisitos é a etapa inicial e mais crítica do ciclo de vida de qualquer sistema de *software*, pois é nela que se identifica, documenta e valida as necessidades dos utilizadores e das partes interessadas. Segundo Sommerville (2010), consiste em “descobrir, analisar, documentar e verificar os serviços e restrições que o sistema deve oferecer”, garantindo que o produto final corresponda exatamente ao que se espera dele.

Neste caso, o levantamento foi feito a partir de:

- Entrevistas semiestruturadas com um engenheiro de manutenção aeronáutica;
- Análise documental do fabricante como *MPD (Maintenance Planning Document)* do Cessna 152, operação e registo de voo históricos e histórico de manutenção da aeronave;
- Requisitos “*in loco*” fornecidos pelo engenheiro de manutenção aeronáutica do plano de manutenção do Cessna 152.

Nesta fase, os requisitos foram classificados em duas categorias principais:

4.1.2.1.Requisitos Funcionais

Os requisitos funcionais especificam as ações e serviços que o sistema deve oferecer aos utilizadores, descrevendo “o quê” o sistema deve fazer em cada fluxo de trabalho.

A seguir apresentam-se os requisitos funcionais identificados para o sistema.

1. RF01 – Autenticação e Controle de Acesso

Permitir login seguro através de utilizador e palavra-passe, com verificação de perfis (Piloto, Técnico, Supervisor, Administrador) de forma a restringir funcionalidades conforme a função.

2. RF02 – Registo e Gestão de Aeronaves e Componentes

Criar, editar e remover registos de aeronaves (matrícula, fabricante, modelo) e dos seus componentes (horas de vida, ciclos).

3. RF03 – Registo de Voos

Registrar novos voos informando matrícula, aeroportos de partida/chegada, horários e horas de motor; atualizar automaticamente os contadores de uso de cada componente.

4. RF04 – Relatório de Anomalias

Disponibilizar formulário para o piloto notificar defeitos em peças ou componentes e encaminhar automaticamente esses relatórios aos técnicos.

5. RF05 – Planeamento e Monitorização de Manutenção Preventiva

Exibir due-lists de tarefas pendentes com base em horas de voo, ciclos ou datas, e permitir que técnicos consultem as prioridades.

6. RF06 – Emissão de Alertas

Disparar alertas configuráveis alerta quando horas ou ciclos atinjam 10 %, 50 % ou 100 % do limite.

7. RF07 – Registo de Manutenção e Intervenção Técnica

Registrar manutenções: descrição do procedimento, atualização do estado do componente e notas técnicas.

8. RF08 – *Workflow* de Aprovação de Manutenção

Permitir que supervisores aprovem ou rejeitem intervenções registadas, registando comentários e finalizando cada ciclo de manutenção.

9. RF09 – Exportação de Relatórios Auditáveis

Gerar, quando requisitado, relatórios em PDF, em português, com histórico de voos, manutenções, peças trocadas, etc.

10. RF10 – cópia de segurança Automática de Dados

Executar cópias periódicas dos bancos de dados e arquivos anexos para garantir a recuperação em caso de falha.

4.1.2.2.Requisitos Não Funcionais

Os requisitos não funcionais (RNF) definem as qualidades e restrições do sistema, avaliando “como” o sistema deve executar essas ações em termos de desempenho, segurança, usabilidade e manutenção.

1. RNF-01 – Segurança

O sistema utiliza métodos de segurança avançados para proteger os dados e o acesso dos utilizadores. A autenticação JWT garante que apenas pessoas autorizadas conseguem entrar na aplicação, através de uma espécie de “chave digital temporária”. O CSRF (*Cross-Site Request Forgery*) é uma proteção extra que impede ataques que tentem enviar pedidos falsos em nome do utilizador.

Todas as comunicações entre o utilizador e o sistema são feitas através de TLS, um protocolo que encripta as informações trocadas — como se colocasse os dados dentro de um “cofre” durante a transmissão. As palavras-passe são guardadas de forma segura com o método PBKDF2, que transforma o texto original em códigos impossíveis de ler ou reverter. Por fim, os registos de atividades do sistema (*logs*) também são encriptados, garantindo que qualquer informação sensível fica protegida contra acessos indevidos.

2. RNF-02 – Desempenho

Resposta de dashboards em ≤ 2 s para até 100 voos e 500 componentes.

3. RNF03 – Disponibilidade e Operação Offline

Suportar modo offline-first, permitindo uso local sem conexão e sincronização automática quando a Internet retornar.

4. RNF-04 – Usabilidade

Interface responsiva em português, fluxos claros por perfil, acessível em desktop e dispositivos móveis.

5. RNF-05 – Escalabilidade

Suportar até 10 aeronaves e 10 utilizadores em simultâneo sem degradação perceptível.

4.1.3. Arquitetura do Sistema

A aplicação do sistema foi planeada sobre um modelo Cliente → Servidor → Base de Dados, mas enriquecido com uma camada intermédia de serviços de domínio. Essa quarta camada permite isolar regras críticas (alertas, cálculo de horas de componentes, geração de relatórios) sem sobrecarregar o controlador *HTTP – Hypertext Transfer Protocol* (responsável por ligar a interface do utilizador ao resto da aplicação, tratando da comunicação entre o que o utilizador pede e as respostas que o sistema devolve) nem o próprio sistema. A seguir apresenta-se, em prosa, o papel de cada bloco e o fluxo típico de trabalho.

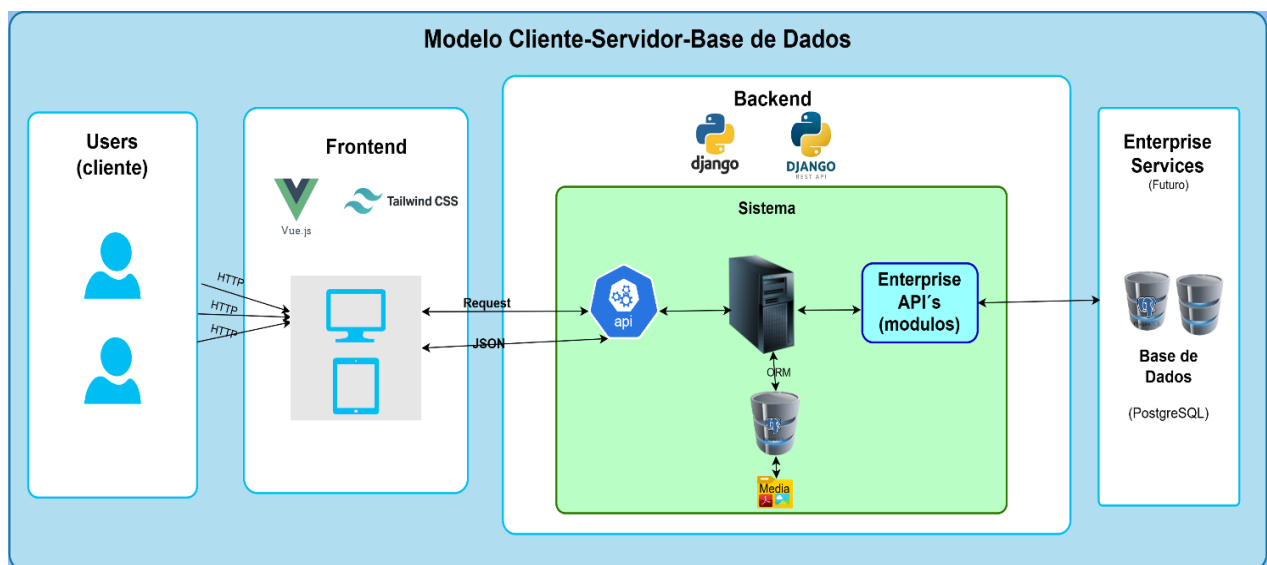


Figura 1 - Arquitetura do Sistema

4.1.3.1. Camada de Utilizador

O ponto de entrada é o navegador (ou em *smartphone*) utilizado por Pilotos, Técnicos, Supervisores e Administradores. Cada perfil recebe um *dashboard* diferente:

- O piloto vê alertas de manutenção antes de registar um voo;
- O técnico de manutenção acompanha os componentes que se encontram próximos do fim da respetiva vida útil;
- O supervisor aprova ou rejeita ordens de serviço;
- O administrador controla utilizadores e parâmetros globais.

Toda a interação acontece em tempo real, numa página única, evitando recarregamentos e diminuindo o risco de entradas duplicadas.

4.1.4. Visão geral do sistema

Tabela 1 - Funcionalidades dos utilizadores Pt.1 (Piloto)

Atores	Funcionalidades	Subfuncionalidades
Piloto	Registrar Voo	<ul style="list-style-type: none">• Introduzir dados do voo (aeronave, horários, combustível)• Calcular tempo de voo• Atualizar horas totais dos componentes
	Consultar Histórico de Voos	<ul style="list-style-type: none">• Listar por período / aeronave
	Reportar Anomalia de Componente	<ul style="list-style-type: none">• Anexar fotografia• Anexar PDF/relatório• Submeter descrição
	Consultar Componentes em Alerta	<ul style="list-style-type: none">• Ordenar por criticidade• Visualizar detalhes do componente
	Verificar Alertas do Piloto	<ul style="list-style-type: none">• Listar alertas pendentes• Marcar como lidos
	Consultar Manutenções Previstas	<ul style="list-style-type: none">• Ver plano de manutenção da aeronave• Detalhar ordem associada
	Consultar Manutenções Realizadas	<ul style="list-style-type: none">• Listar intervenções concluídas

Tabela 2 - Funcionalidades dos utilizadores Pt.2 (Técnico de Manutenção)

Atores	Funcionalidades	Subfuncionalidades
Técnico de Manutenção	Registar Manutenção	<ul style="list-style-type: none"> • Preencher checklist • Gerar relatório em PDF • Atualizar estado da manutenção
	Registar Inspeção Periódica	<ul style="list-style-type: none"> • Selecionar componentes inspecionados • Anexar observações • Criar ordem de manutenção (se necessário)
	Consultar Relatórios de Anomalia	<ul style="list-style-type: none"> • Filtrar por data / componente • Alterar estado (em análise, resolvido)
	Consultar Manutenções Agendadas	<ul style="list-style-type: none"> • Ordenar por prioridade
	Consultar Manutenções Atribuídas	<ul style="list-style-type: none"> • Atualizar progresso
	Monitorizar Componentes	<ul style="list-style-type: none"> • Mostrar todos os componentes • Listar componentes em alerta • Ordenar por horas restantes • Ver histórico de trocas
	Verificar Alertas do Técnico	<ul style="list-style-type: none"> • Listar alertas técnicos • Marcar resolução
	Consultar Estado da Frota	<ul style="list-style-type: none"> • Listar por aeronave / estado • Mostrar tabela

Tabela 3 - Funcionalidades dos utilizadores Pt.3 (Supervisor)

Atores	Funcionalidades	Subfuncionalidades
Supervisor de Manutenção	Consultar Manutenções Aguardando aprovação	<ul style="list-style-type: none"> • Ver detalhes da manutenção • Aprovar manutenção • Rejeitar manutenção
	Aprovar Manutenção	<ul style="list-style-type: none"> • Notificar técnico (aprovação) • Atualizar estado para «Concluída» Programar execução
	Rejeitar Manutenção	<ul style="list-style-type: none"> • Notificar técnico (rejeição) • Atualizar estado para «Rejeitada» • Solicitar nova inspeção
	Gerar Relatórios de Desempenho	<ul style="list-style-type: none"> • Exportar PDF
	Verificar Alertas do Supervisor	<ul style="list-style-type: none"> • Listar alertas críticos • Atribuir responsável
	Gerir Utilizadores e Perfis	<ul style="list-style-type: none"> • Criar utilizador • Editar utilizador • Desativar utilizador • Repor palavra-passe

Tabela 4 - Funcionalidades dos utilizadores Pt.4 (Administrador)

Atores	Funcionalidades	Subfuncionalidades
Administrador	Gerir Aeronaves	<ul style="list-style-type: none"> • Criar aeronave • Editar aeronave • Desativar aeronave
	Gerir Componentes	<ul style="list-style-type: none"> • Criar componente • Editar componente • Eliminar componente
	Gerir Listas de Referência	<ul style="list-style-type: none"> • Criar referência (Estado/Tipo) • Editar referência • Eliminar referência
	Consultar <i>Logs</i> de Auditoria	<ul style="list-style-type: none"> • Listar por data / módulo / utilizador •
	Verificar Alertas do Administrador	<ul style="list-style-type: none"> • Listar alertas gerais do sistema • Marcar como resolvidos

4.1.4.1.Caso de uso do piloto

O piloto desempenha um papel central no sistema de aeronaves. As funcionalidades específicas deste utilizador garantem que as operações diárias estejam corretamente documentadas, permitindo um acompanhamento rigoroso e preventivo das aeronaves.

O piloto interage com o sistema através das seguintes funcionalidades essenciais:

- **Registar Voo:** introduzir detalhes do voo (matrícula, partida e chegada), permitindo ao sistema calcular automaticamente o tempo de voo, atualizar as horas totais dos componentes e emitir notificações.
- **Reportar Anomalia:** comunicar falhas em componentes, anexando imagens e gerando alertas automáticos para os técnicos.
- **Consultar Histórico de Voos:** visualizar histórico dos voos anteriores, facilitando o acompanhamento das atividades realizadas.

- Consultar Manutenções (Previstas e Realizadas): obter informações sobre intervenções agendadas e intervenções já efetuadas na aeronave.
- Consultar Componentes em Alerta: visualizar rapidamente quais os componentes que precisam de atenção.
- Receber Notificações: ser informado automaticamente sobre eventos críticos ou relevantes relacionados à manutenção.

UC01 – Registar Voo (Piloto)

Tabela 5 - Descrição da funcionalidade “Registar Voo”

Prioridade:	Alta
Actor Primário:	Piloto
Descrição:	O piloto introduz um novo voo realizado, permitindo ao sistema atualizar o total de horas da aeronave e detetar eventuais excessos.
Gatilho:	Piloto clica em “Novo Voo” no seu painel.
Pré-condições:	1. Piloto autenticado. 2. Aeronave atribuída ativa.
Curso Normal:	1. Sistema apresenta formulário de voo. 2. Piloto preenche campos (matrícula, partida, chegada). 3. Sistema calcula tempo de voo. 4. Sistema grava registo e atualiza horas totais da aeronave. 5. Se as horas-limite forem excedidas, é criado um alerta para piloto e técnico. 6. Sistema confirma com mensagem de sucesso.
Cursos Alternativos:	2A. Campos obrigatórios em falta → sistema sinaliza a vermelho e volta ao passo 2.
Pós-condições:	<ul style="list-style-type: none"> • Voo gravado. • Horas acumuladas atualizadas. • Alertas (se aplicável) registados.
Exceções:	E1 — Ligação à BD indisponível: sistema mostra erro “Falha de gravação”; piloto pode guardar rascunho local.

UC02 – Relatar Anomalia de Componente (Piloto)

Tabela 6 - Descrição da funcionalidade “Relatar Anomalia de Componente”

Prioridade:	Média
Actor Primário:	Piloto
Descrição:	Permite ao piloto comunicar defeitos num componente, anexando evidências para análise técnica.
Gatilho:	Piloto pressiona “Relatar Anomalia” na página da aeronave.
Pré-condições:	1. Piloto autenticado. 2. Componente existente.
Curso Normal:	1. Sistema exibe formulário com lista de componentes. 2. Piloto seleciona o componente, descreve o problema e anexa fotografia. 3. Sistema grava o relato com estado “Por analisar”. 4. Sistema notifica o técnico responsável. 5. Sistema confirma operação.
Alternativos:	3A. Upload falha → sistema exibe erro e sugere repetir; regressa ao passo 2.
Pós-condições:	• Relato registado. • Técnico informado.
Exceções:	E1 — Componente não encontrado: sistema mostra mensagem “Componente inválido” e cancela operação.

4.1.4.2.Caso de Uso do Técnico de Manutenção

O técnico de manutenção é o responsável por garantir a operacionalidade segura e funcional dos componentes das aeronaves. As principais ações disponíveis para este perfil incluem:

- Registrar Manutenção: preencher dados de uma intervenção realizada, gerar automaticamente um relatório em PDF e atualizar o estado da manutenção no sistema.
- Consultar Relatórios de Anomalia: aceder a relatórios enviados pelos pilotos, filtrando por data ou componente, e alterar o estado da ocorrência (ex.: “em análise” ou “resolvido”).

- Monitorizar Componentes: acompanhar todos os componentes instalados nas aeronaves, ordenando por horas restantes até manutenção ou visualizando o histórico de trocas.
- Consultar Manutenções: ver ordens de manutenção agendadas, atribuídas ou em execução, com possibilidade de atualizar o progresso.
- Registar Inspeção: realizar inspeções periódicas, selecionar os componentes avaliados e, se necessário, criar automaticamente uma ordem de manutenção.
- Verificar Alertas: acompanhar os alertas técnicos gerados automaticamente pelo sistema, marcando os resolvidos ou pendentes.
- Atualizar Estado de Manutenção: indicar se uma ordem foi concluída, se está em progresso ou se foi reaberta após nova inspeção.

UC08 – Registar Manutenção (Técnico de Manutenção)

Tabela 7 - Descrição da funcionalidade "Registar Manutenção"

Prioridade:	Alta
Actor Primário:	Técnico de Manutenção
Descrição:	O técnico regista a intervenção executada num componente, gera o relatório PDF e atualiza o estado da manutenção.
Gatilho:	Técnico escolhe “Registar Manutenção” em manutenção atribuída.
Pré-condições:	1. Ordem com estado “Pendente”.
Curso Normal:	1. Sistema apresenta <i>checklist</i> de tarefas. 2. Técnico preenche dados, observações. 3. Sistema gera PDF do relatório. 4. Sistema muda estado para “Aguardando Aprovação”. 5. Sistema envia notificação ao piloto e supervisor. 6. Sistema grava log da ação e confirma sucesso.
Alternativos:	3A. Geração de PDF falha → sistema grava manutenção, mas alerta “Relatório não gerado”; técnico pode tentar novamente.
Pós-condições:	• Manutenção concluída e documentada. • Notificações enviadas.
Exceções:	E1 — Ordem já concluída: sistema impede edição e apresenta aviso.

UC09 – Registrar Inspeção (Técnico)

Tabela 8 - Descrição da funcionalidade "Registrar Inspeção"

Prioridade:	Média
Actor Primário:	Técnico de Manutenção
Descrição:	Regista uma inspeção de rotina, permitindo selecionar múltiplos componentes e, se necessário, criar de manutenção.
Gatilho:	Técnico seleciona “Nova Inspeção”.
Pré-condições:	Agenda de inspeções planeada pela supervisão.
Curso Normal:	<ol style="list-style-type: none">1. Sistema mostra formulário com data e lista de componentes.2. O Técnico marca componentes verificados e adiciona observações.3. Ao submeter, sistema grava inspeção.4. Se forem detetadas falhas, sistema oferece criar uma ordem de manutenção5. Sistema confirma registo.
Pós-condições:	<ul style="list-style-type: none">• Inspeção registada.• Ordens geradas conforme necessidade.

4.1.4.3.Caso de uso do Supervisor da Manutenção

O supervisor é o responsável por acompanhar, aprovar e avaliar as operações de manutenção da frota, garantindo conformidade com os padrões técnicos e operacionais. As suas principais ações incluem:

- Consultar Manutenções “Aguardando Aprovação”: visualizar as manutenções pendentes para análise, com acesso aos detalhes técnicos de cada intervenção.
- Aprovar Manutenção: confirmar que uma manutenção foi corretamente realizada, atualizar o seu estado para "Concluída" e notificar os envolvidos.
- Rejeitar Manutenção: caso encontre falhas ou inconsistências, pode rejeitar a ordem, atualizar o estado para "Rejeitada", solicitar uma nova inspeção e notificar o técnico responsável.
- Registrar Inspeção: adicionar uma nova inspeção técnica, especialmente quando uma manutenção for recusada ou precisar de verificação complementar.
- Criar Manutenção: gerar ordens de manutenção diretamente, caso identifique problemas em inspeções ou relatórios.

- Selecionar Componentes: indicar quais peças devem ser inspecionadas ou substituídas durante uma manutenção.
- Verificar Alertas: acompanhar os alertas críticos emitidos pelo sistema e atribuir responsáveis para resolução.
- Gerar Relatórios de Desempenho: exportar relatórios técnicos em formato PDF, úteis para auditorias, controle de qualidade e análise operacional.

UC22 – Rejeitar Manutenção (Supervisor)

Tabela 9 - Descrição da funcionalidade "Rejeitar Manutenção"

Campo	Conteúdo
Prioridade:	Alta
Actor Primário:	Supervisor de Manutenção
Descrição:	O supervisor recusa uma manutenção, solicita uma nova inspeção e informa o técnico.
Gatilho:	Supervisor clica em “Rejeitar” numa manutenção.
Pré-condições:	1. Manutenção com estado “Aguardando Aprovação”.
Curso Normal:	1. Supervisor muda estado para “Rejeitada”. 2. Sistema cria uma inspeção associada. 3. Sistema envia notificação ao técnico. 4. Sistema regista a ação no log.
Pós-condições:	• Manutenção rejeitada. • Inspeção agendada. • Técnico notificado.

4.1.4.4.Caso de uso do Administrador do Sistema

O administrador desempenha um papel estratégico na configuração e manutenção da infraestrutura lógica do sistema. É responsável por garantir a segurança, o controle de acessos e a organização dos recursos técnicos e operacionais. As principais funcionalidades sob sua responsabilidade incluem:

- Gerir Utilizadores e Perfis: criação, edição, desativação de contas de utilizadores e reposição de palavras-passe. Essa funcionalidade garante que cada ator tenha permissões adequadas de acordo com o seu papel (Piloto, Técnico, Supervisor).

- Gerir Componentes: administração dos componentes aeronáuticos utilizados nas manutenções e inspeções. Inclui a criação de novos itens, edição de dados técnicos e exclusão, quando necessário.
- Gerir Manutenções: permite inserir manutenções diretamente no sistema, editar ou eliminar registos antigos, em casos de ajustes administrativos ou correções.
- Gerir Aeronaves: gestão do inventário de aeronaves com funcionalidades para registar, editar informações e desativar aquelas fora de operação.
- Consultar Logs de Auditoria: garante a rastreabilidade completa de ações no sistema, listando registos por utilizador, data ou módulo.
- Verificar Notificações: acompanha alertas de sistema gerais, atuando em caso de falhas, alterações críticas ou mensagens administrativas pendentes.

Este conjunto de casos de uso posiciona o administrador como o guardião da integridade, segurança e funcionamento sistémico da aplicação.

UC25 – Gerir utilizadores e perfis (Administrador)

Tabela 10 - Descrição da funcionalidade “Gerir Utilizadores e Perfis”

Prioridade:	Alta
Actor Primário:	Administrador do Sistema
Descrição:	Permite criar, editar, desativar utilizadores e repor palavras-passes, assegurando o controlo de acesso ao sistema.
Gatilho:	Administrador abre menu “Utilizadores”.
Pré-condições:	1. Conta de administrador ativa.
Curso Normal:	1. Sistema apresenta lista de utilizadores. 2. Administrador seleciona “Novo”, “Editar” ou “Desativar”. 3. Sistema valida dados (e-mail único, perfil válido). 4. Sistema grava alterações e regista notificação de alteração. 5. Ação registada em log.
Alternativos:	2A. Reiniciar password → sistema gera token e envia e-mail de reset.
Pós-condições:	• Dados de utilizador atualizados. • Registo no log.
Exceções:	E1 — E-mail duplicado: sistema bloqueia gravação e exibe mensagem “Utilizador já existe”.

UC29 – Consultar Logs de Auditoria (Administrador)

Tabela 11 - Descrição da funcionalidade “Consultar Logs de Auditoria”

Prioridade:	Média
Actor Primário:	Administrador do Sistema
Descrição:	Permite ao administrador auditar ações, listar por data, módulo ou utilizador, garantindo a rastreabilidade.
Gatilho:	Administrador escolhe “Logs” no menu.
Pré-condições:	Logs existentes.
Curso Normal:	<ol style="list-style-type: none">1. Sistema mostra lista (data inicial/final, utilizador, módulo).2. Administrador define critérios e aplica pesquisa.3. Sistema lista registos correspondentes.4. Administrador pode exportar para PDF.
Pós-condições:	<ul style="list-style-type: none">• Informações de auditoria consultadas ou exportadas.
Exceções:	E1 — Nenhum registo encontrado: sistema exhibe mensagem “Sem resultados”.

4.1.4.5.Tela de Login

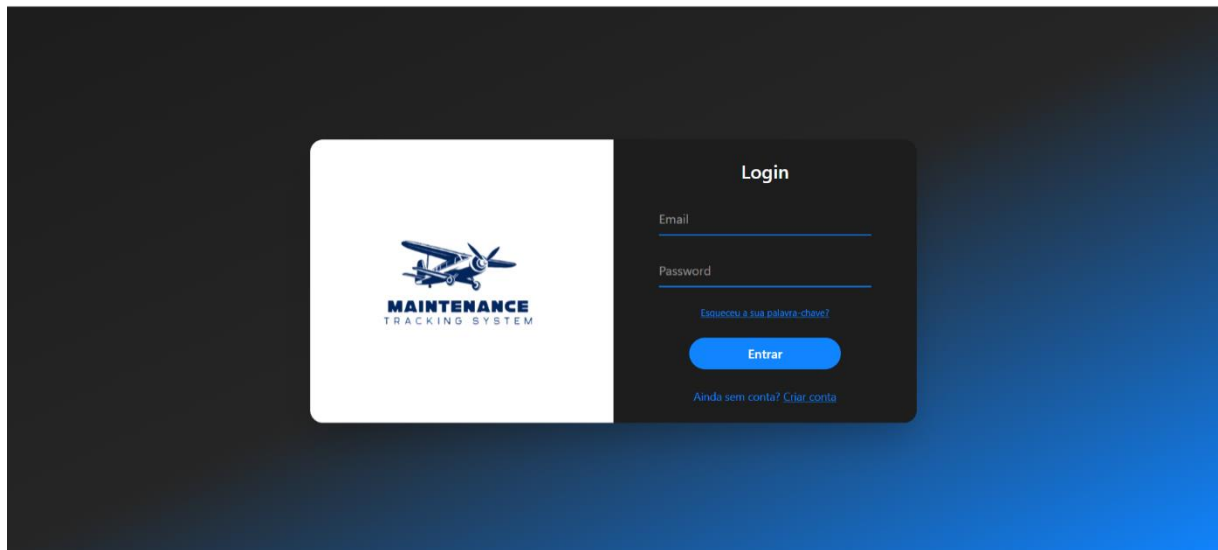


Figura 2 - Tela login

A página de login constitui a porta de entrada principal para o *Maintenance Tracking System*, oferecendo uma interface minimalista e funcional. O design simples garante rapidez no acesso, promovendo segurança e eficiência através de autenticação via email e palavra-passe. Destaca-se visualmente pela clareza das informações e pelo logo que reforça a identidade visual do sistema, proporcionando uma experiência inicial direta e intuitiva aos utilizadores.

4.1.4.6.Dashboard do Piloto

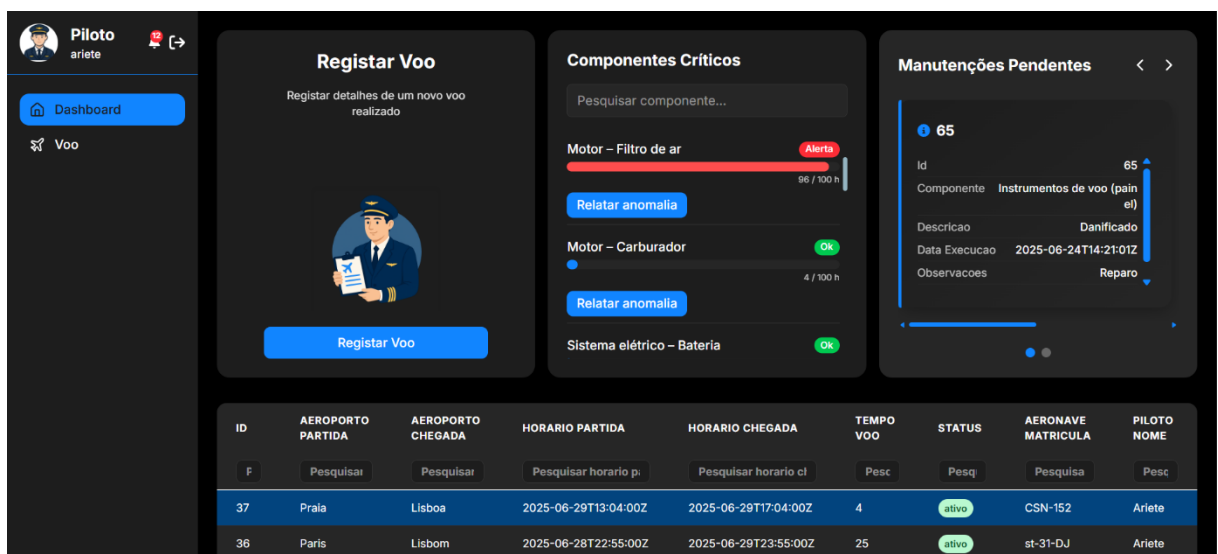


Figura 3 - Dashboard do piloto

O *dashboard* para o perfil de piloto fornece uma visão abrangente das operações essenciais para a gestão eficaz das suas atividades aéreas. A interface está estruturada em secções claramente definidas, permitindo ao piloto registar rapidamente novos voos, acompanhar o estado dos componentes críticos das aeronaves, o relato imediato de anomalias, permitindo identificar e notificar rapidamente problemas ou falhas detetadas nos componentes da aeronave, e monitorizar as manutenções pendentes de forma intuitiva. Além disso, apresenta uma área dedicada às notificações em tempo real, garantindo que o piloto esteja sempre informado sobre alertas importantes e atualizações operacionais imediatas. Complementarmente, disponibiliza uma visão geral dos voos efetuados, destacando detalhes fundamentais, contribuindo assim para uma operação eficiente, segura e transparente.

Sugestões claramente definidas permitem ao piloto registar rapidamente novos voos, acompanhar o estado dos componentes críticos das aeronaves e relatar anomalias, identificando e notificando de forma imediata problemas ou falhas detetadas nos componentes. Também é possível monitorizar as manutenções pendentes de forma intuitiva. Além disso, o sistema apresenta uma área dedicada às notificações em tempo real, garantindo que o piloto esteja sempre informado sobre alertas importantes e atualizações operacionais. Complementarmente, disponibiliza uma visão geral dos voos efetuados, destacando detalhes fundamentais, contribuindo para uma operação eficiente, segura e transparente.

Ecrã “Voo” – resumo operacional diário do piloto



Figura 4 - Ecrã “Voo”

Esta interface apresenta o painel dedicado ao piloto, com foco na gestão e monitorização das suas atividades operacionais:

- Indicadores Principais:
 - Apresenta as horas totais de voo acumuladas pelo piloto (462 horas).

- Exibe também o total de voos registrados realizados pelo piloto (3 voos).
- Manutenções Realizadas:
 - Apresenta detalhes das manutenções já concluídas nos componentes da aeronave.
 - Destaca claramente o componente (exemplo: Travões - Discos e pastilhas), data/hora da execução e o estado (concluída).
 - Observações adicionais sobre a manutenção são apresentadas, fornecendo contexto útil ao piloto.
- Manutenções Previstas:
 - Apresenta manutenções futuras agendadas, incluindo componente específico (exemplo: Motor - Velas de ignição).
 - Indica a data e hora prevista para a execução da manutenção e o estado atual (Agendada).
 - Também exibe observações adicionais, permitindo uma preparação adequada do piloto para futuras intervenções técnicas.
- Tabela de Voos:
 - Apresenta uma visão organizada das operações de voo, incluindo aeroportos de partida e chegada, horários, duração do voo, estado atual e matrícula da aeronave utilizada.
 - Facilita ao piloto o acesso rápido ao histórico das suas atividades e à situação atual dos voos registrados.

Este painel contribui para a gestão operacional eficiente, garantindo que o piloto tenha total visibilidade e controle sobre as atividades realizadas e previstas, reforçando a segurança e eficácia operacional.

4.1.4.7. *Dashboard* do técnico de manutenção – centro operacional de manutenção

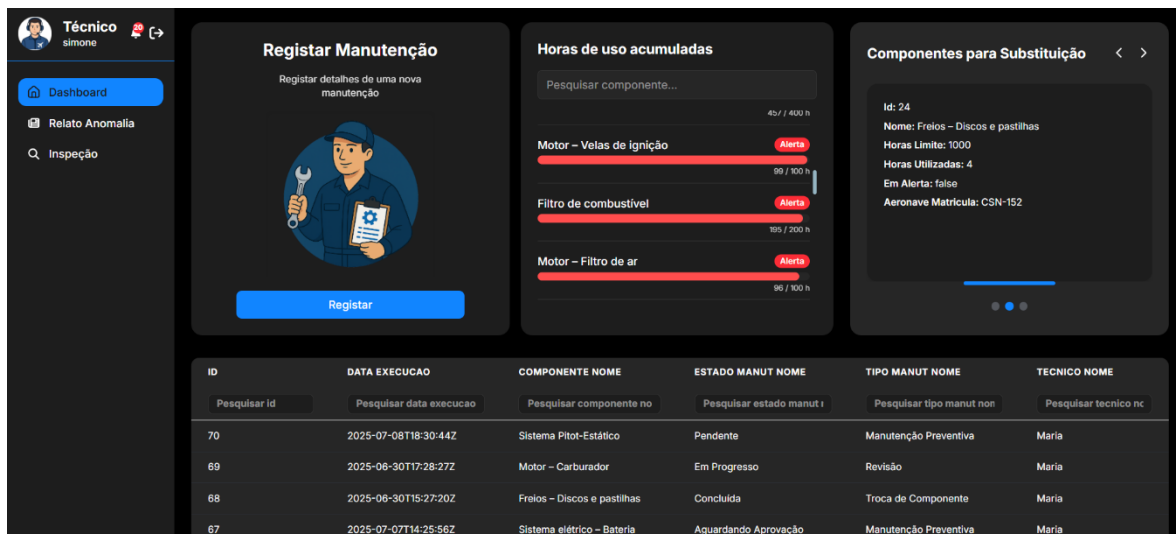


Figura 5 - *Dashboard* do técnico de manutenção

Este *dashboard* destina-se ao perfil do Técnico de Manutenção e centraliza funcionalidades para a gestão eficiente da manutenção das aeronaves. No topo da interface, o técnico pode rapidamente registrar novas manutenções através de um botão específico. Ao centro, o painel exibe uma lista detalhada das “Horas de Uso Acumuladas”, destacando claramente componentes que estão em estado de alerta, indicando que estes atingiram ou ultrapassaram o limite recomendado de utilização.

Do lado direito, é apresentado o painel de “Componentes para Substituição”, detalhando claramente o estado e as informações importantes dos componentes que precisam ser substituídos em breve, incluindo horas limite, horas já utilizadas e a matrícula da aeronave relacionada.

Na parte inferior do *dashboard* encontra-se uma tabela abrangente com todas as manutenções registadas, exibindo claramente informações relevantes como data de execução, nome do componente, estado atual da manutenção, tipo específico de manutenção e o nome do técnico responsável pela execução.

Por fim, no canto superior esquerdo, encontra-se a área de notificações, garantindo que o técnico esteja sempre atualizado sobre eventos importantes relacionados à manutenção.

Ecrã “Relato Anomalia” – painel de acompanhamento dos reports feitos pelos pilotos

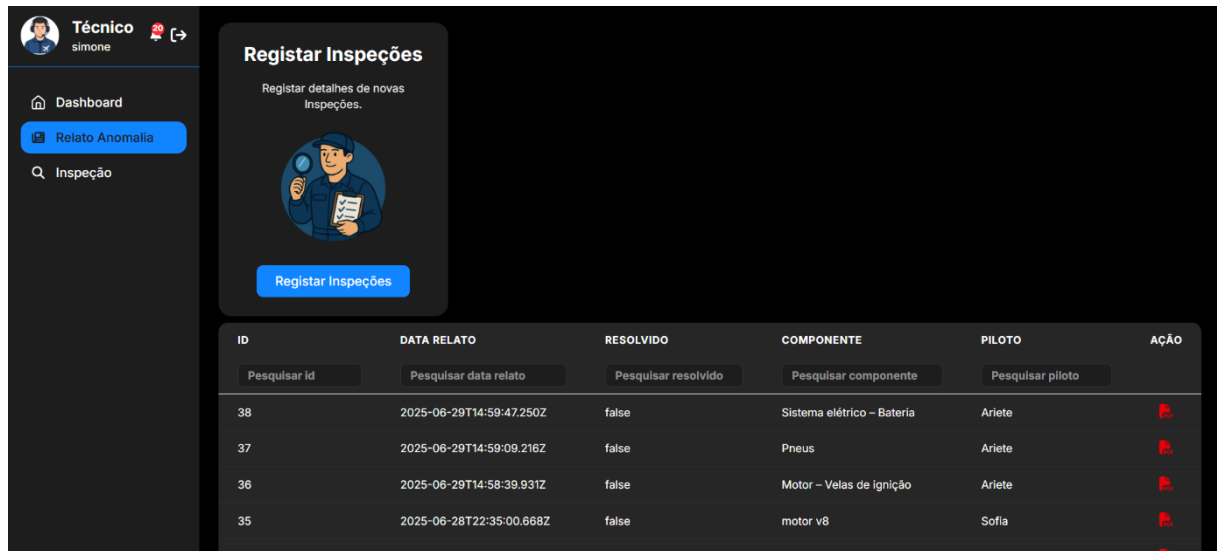


Figura 6 - Ecrã “Relato Anomalia”

Este painel de “Relato de Anomalia” permite ao perfil de Técnico monitorizar de forma rápida e centralizada todos os relatos de anomalias efetuados pelos pilotos. A interface disponibiliza uma área destacada para o registo rápido de novas inspeções relacionadas com as anomalias reportadas.

Na tabela principal, o técnico tem acesso claro e imediato aos relatos já realizados, podendo visualizar detalhes importantes como a data do relato, o estado atual (resolvido ou não), o componente afetado e o piloto responsável pelo relato. Além disso, existe uma ação direta para gerar e visualizar um relatório em PDF associado a cada anomalia, assegurando uma documentação precisa e organizada dos problemas identificados.

Janela “Detalhes da Inspeção” — visão rápida de registos já criados

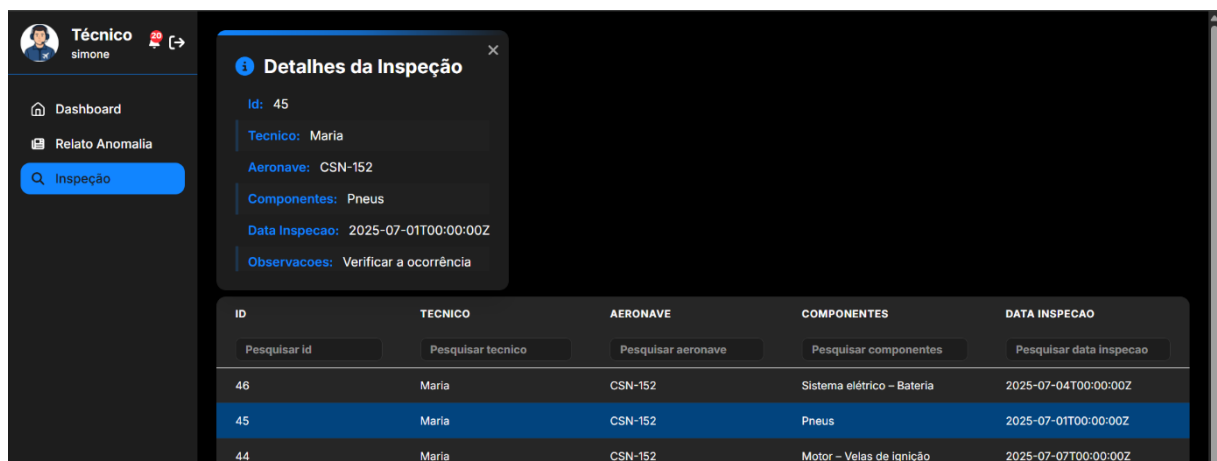


Figura 7 - Ecrã “Detalhes da Inspeção”

Esta interface apresenta o painel de Inspeção, destinado ao perfil de Técnico, permitindo-lhe acompanhar e consultar inspeções realizadas às aeronaves.

O painel apresenta de forma clara e organizada as inspeções efetuadas, com detalhes específicos como o identificador da inspeção, o técnico responsável, a aeronave inspecionada, os componentes abrangidos e a data da inspeção. Ao selecionar um registo específico, o utilizador pode visualizar uma janela detalhada com todas as informações pertinentes, incluindo observações adicionais feitas durante a inspeção. Esta funcionalidade permite uma rápida verificação e gestão eficiente das inspeções bem como das condições técnicas das aeronaves.

4.1.4.8. *Dashboard do Supervisor — visão-geral para decisões de aprovação*

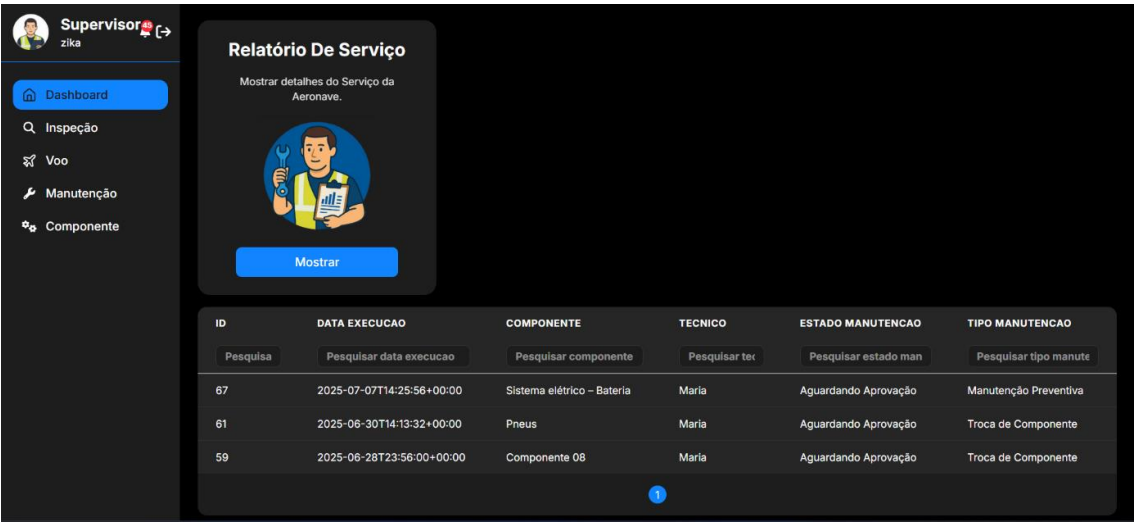


Figura 8 - *Dashboard* do supervisor

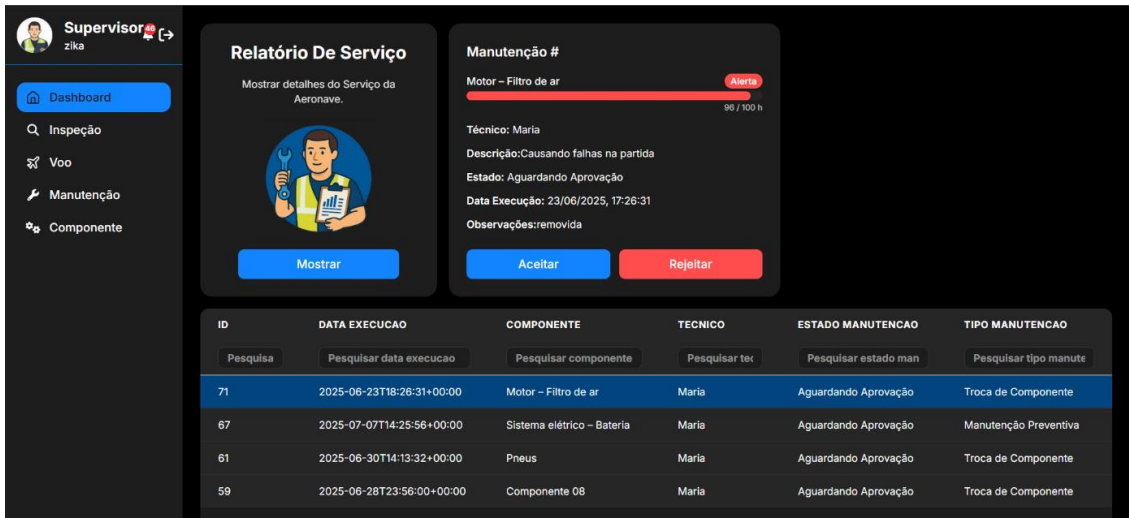


Figura 9 - *Dashboard* do supervisor (painel de decisão)

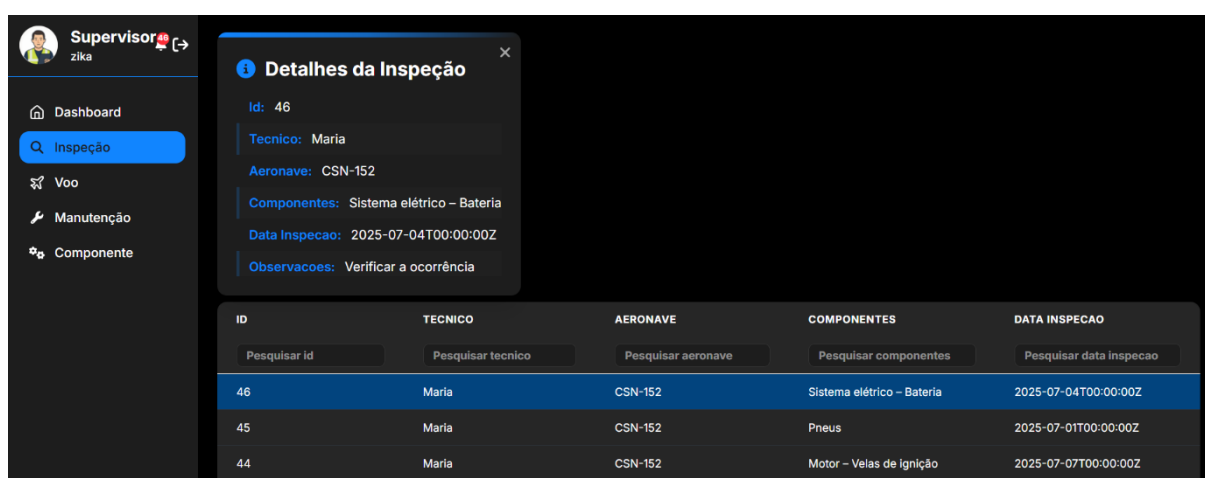
O *Dashboard* do Supervisor concentra-se no acompanhamento e gestão das manutenções realizadas nas aeronaves. A interface é composta por um painel central, onde é possível visualizar rapidamente detalhes importantes sobre os serviços de manutenção realizados pelos técnicos, incluindo a data de execução, os componentes envolvidos, o estado atual e o tipo de manutenção efetuada.

Um outro destaque é a funcionalidade para aprovação ou rejeição das manutenções que estão no aguardo de aprovação. O supervisor tem acesso direto às descrições detalhadas das manutenções realizadas, podendo tomar decisões rápidas e fundamentadas.

Através deste *dashboard*, o supervisor também pode gerar relatórios de serviço da aeronave, obtendo uma visão abrangente sobre o estado operacional dos componentes, tornando as tomadas de decisões mais eficazes, garantindo assim, a segurança operacional.

Por fim, o *dashboard* incorpora ainda um sistema de notificações que alerta o supervisor sobre ações urgentes e atualizações importantes relacionadas às atividades de manutenção e operação das aeronaves.

Ecrã Inspeção ► Detalhes da Inspeção



The screenshot displays the Supervisor Dashboard interface. On the left is a sidebar with navigation options: Dashboard, Inspeção (highlighted), Voo, Manutenção, and Componente. The main area shows a modal titled 'Detalhes da Inspeção' for inspection ID 46. The modal details include: Técnico: Maria, Aeronave: CSN-152, Componentes: Sistema elétrico - Bateria, Data Inspecao: 2025-07-04T00:00:00Z, and Observacoes: Verificar a ocorrência. Below the modal is a table with five columns: ID, TECNICO, AERONAVE, COMPONENTES, and DATA INSPECAO. The table contains three rows of inspection data.

ID	TECNICO	AERONAVE	COMPONENTES	DATA INSPECAO
46	Maria	CSN-152	Sistema elétrico - Bateria	2025-07-04T00:00:00Z
45	Maria	CSN-152	Pneus	2025-07-01T00:00:00Z
44	Maria	CSN-152	Motor - Velas de Ignição	2025-07-07T00:00:00Z

Figura 10 - Ecrã "Inspeção "

O painel de Inspeção do supervisor permite o acompanhamento detalhado das inspeções realizadas nas aeronaves. A interface apresenta uma tabela resumida das inspeções realizadas, destacando o técnico responsável, a aeronave inspecionada, os componentes verificados e a data da inspeção.

Ao selecionar uma inspeção específica, são exibidos detalhes adicionais numa janela lateral, fornecendo informações completas como as observações feitas pelo técnico e as recomendações para ações subsequentes, permitindo ao supervisor verificar o estado dos componentes e tomar decisões operacionais rapidamente.

Ecrã Voo

Histórico de voos
Mostrar detalhes de um voo realizado

Aeronaves em Alerta
Mostrar aeronaves em alerta

ID	AEROPORTO PARTIDA	AEROPORTO CHEGADA	HORARIO PARTIDA	HORARIO CHEGADA	TEMPO VOO	STATUS	AERONAVE MATRICULA	PILOTO NOME
37	Praia	Lisboa	2025-06-29T13:04:00Z	2025-06-29T17:04:00Z	4	ativo	CSN-152	Ariete
36	Paris	Lisboa	2025-06-28T22:55:00Z	2025-06-29T23:55:00Z	25	ativo	st-31-DJ	Ariete
35	praia	lisboa	2025-06-10T21:34:00Z	2025-06-28T22:34:00Z	433	ativo	ST-41-HT	Ariete
34	frança	cabo verde	2025-06-09T00:07:00Z	2025-06-10T00:07:00Z	24	ativo	ST-41-HT	Deusa

Figura 11 - Ecrã voo (supervisor)

O painel Voo do supervisor disponibiliza uma visão geral essencial das operações aéreas. Esta área exibe o histórico detalhado dos voos realizados, permitindo acompanhar os aeroportos de partida e chegada, os horários, o tempo de voo, o estado atual, matrícula da aeronave e o piloto responsável.

Além disso, disponibiliza a função "Aeronaves em Alerta", indicando rapidamente aeronaves que requerem atenção imediata devido a problemas identificados. Este recurso auxilia o supervisor a agir preventivamente, assegurando maior segurança operacional. A interface foi projetada para facilitar uma supervisão rápida e eficaz das operações aéreas diárias.

Ecrã Manutenção – visão do Supervisor

ID	DATA EXECUCAO	COMPONENTE	TECNICO	ESTADO	TIPO	AÇÃO
71	2025-06-23 18:26:31	Motor - Filtro de ar	Maria	Aguardando Aprovação	Troca de Componente	
70	2025-07-08 18:30:44	Sistema Pitot-Estático	Maria	Pendente	Manutenção Preventiva	
69	2025-06-30 17:28:27	Motor - Carburador	Maria	Em Progresso	Revisão	
68	2025-06-30 15:27:20	Freios - Discos e pastilhas	Maria	Concluída	Troca de Componente	
67	2025-07-07 14:25:56	Sistema elétrico - Bateria	Maria	Rejeitada	Manutenção Preventiva	

Figura 12 - Ecrã manutenção (supervisor)

O painel de “Manutenção” do supervisor oferece uma visão clara e detalhada de todas as intervenções de manutenção registadas no sistema. Apresenta uma tabela informativa com dados como o ID da manutenção, a data de execução, o componente afetado, o técnico responsável, o estado da manutenção (pendente, em progresso, concluída, rejeitada, etc.) e o tipo de intervenção realizada.

Além da visualização direta no sistema, este painel permite ainda exportar os dados em formato PDF, facilitando o registo, análise ou partilha das informações com outros intervenientes. A funcionalidade de filtros torna mais simples localizar manutenções específicas, permitindo ao supervisor acompanhar o estado da frota com maior agilidade e precisão.

Painel Componente – visão para o Supervisor

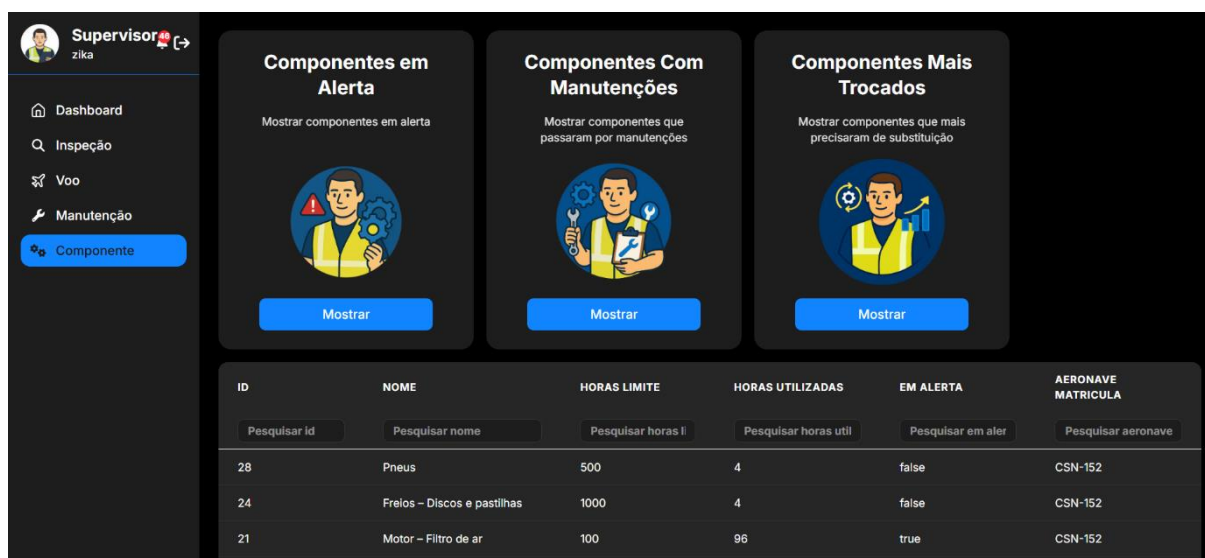
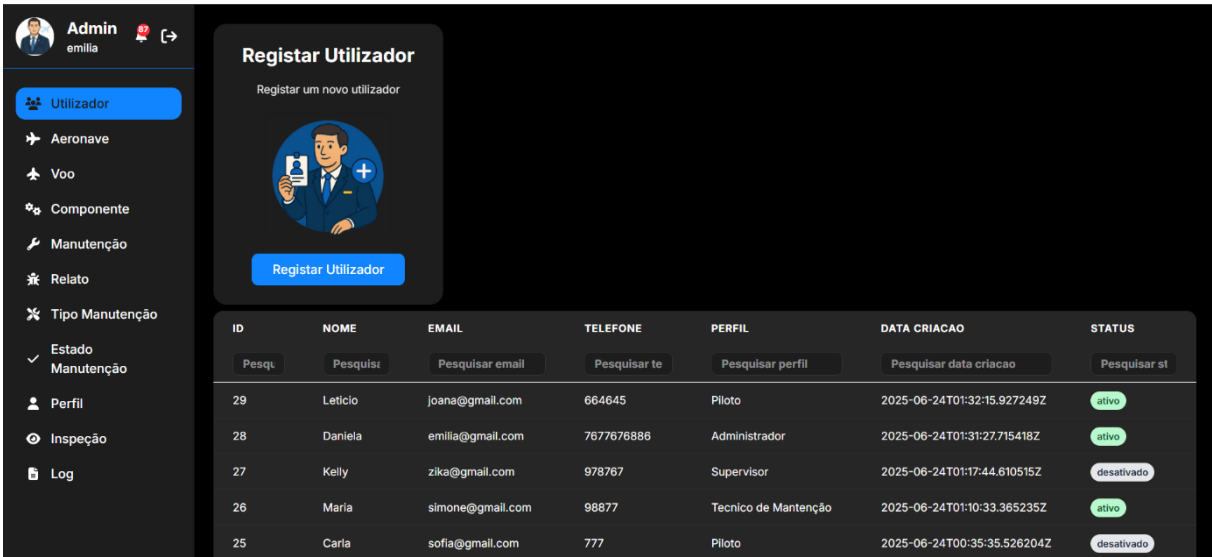


Figura 13 - Ecrã componentes (supervisor)

O painel Componente do supervisor disponibiliza uma visão centralizada sobre o estado e histórico dos principais componentes das aeronaves. É composto por três secções funcionais:

- Componentes em Alerta: identifica componentes que atingiram ou estão próximos do limite de horas de uso e que requerem atenção imediata;
- Componentes com Manutenções: lista os componentes que já passaram por manutenções, permitindo acompanhar o histórico de intervenções;
- Componentes Mais Trocados: apresenta os componentes que mais frequentemente necessitaram de substituição ao longo do tempo.

A tabela da figura que se segue, resume as informações de cada componente, incluindo nome, horas limite, horas utilizadas, estado de alerta e a matrícula da aeronave correspondente.



Registrar Utilizador
Registrar um novo utilizador

Registrar Utilizador

ID	NOME	EMAIL	TELEFONE	PERFIL	DATA CRIACAO	STATUS
29	Leticio	joana@gmail.com	664645	Piloto	2025-06-24T01:32:15.927249Z	ativo
28	Daniela	emilia@gmail.com	7677676886	Administrador	2025-06-24T01:31:27.715418Z	ativo
27	Kelly	zika@gmail.com	978767	Supervisor	2025-06-24T01:17:44.610515Z	desativado
26	Maria	simone@gmail.com	98877	Tecnico de Manutenção	2025-06-24T01:10:33.365235Z	ativo
25	Carla	sofia@gmail.com	777	Piloto	2025-06-24T00:35:35.526204Z	desativado

Figura 14 - Ecrã administrador Painel “Utilizador”

O painel “Utilizador”, acessível ao perfil de Administrador, centraliza a gestão dos utilizadores da aplicação. Na área principal, encontra-se o botão “Registrar Utilizador”, que permite adicionar novos membros à plataforma, atribuindo-lhes perfis como Piloto, Técnico de Manutenção, Supervisor ou Administrador, de acordo com as respetivas responsabilidades no sistema.

Abaixo, apresenta-se uma tabela com a listagem detalhada dos utilizadores já registados, incluindo informações como nome, e-mail, telefone, perfil, data de criação e estado da conta (ativo ou desativado). A tabela disponibiliza filtros em todas as colunas para facilitar a pesquisa de registos específicos. O painel permite editar ou eliminar utilizadores diretamente a partir da tabela, garantindo ao administrador um controlo total sobre o ciclo de vida dos acessos ao sistema.

No topo encontra-se as notificações, que alerta visualmente sobre eventos relevantes do sistema, reforçando a capacidade de monitorização e ação rápida por parte da administração.

4.1.4.9. Padrão geral de gestão

O ecrã mostra sempre uma lista paginada com dez registos por página. Cada cabeçalho possui um campo de pesquisa embutido que filtra a grelha em tempo real. No topo surge um cartão “Registrar” cujo botão abre uma janela vertical. Essa janela serve tanto para criar como para editar: campos alinhados, botões Atualizar e Eliminar no rodapé. O estado ativo/desativado é indicado por um chip; ao clicar, o chip alterna de cor

e o registo é imediatamente atualizado. Validações tradicionais — obrigatoriedade, formatos, unicidade — são aplicadas de forma consistente. Isso é aplicado em todos esses módulos aeronave, voo, componentes, manutenção, relato, tipo de manutenção, estado de manutenção, perfil e inspeção.

Histórico de Logs (Admin)

ID	AÇÃO	MÓDULO	DETALHES	DATA HORA	UTILIZADOR
581	Apagar Utilizador	Admin	Daniela apagou um utilizador na aplicação	2025-06-29T15:56:16.200778Z	Daniela
580	Apagar Utilizador	Admin	Daniela apagou um utilizador na aplicação	2025-06-29T15:56:09.341944Z	Daniela
579	Login	Login	Daniela fez login na aplicação	2025-06-29T15:55:36.818550Z	Daniela
578	Login	Login	Kelly fez login na aplicação	2025-06-29T15:50:45.165878Z	Kelly
577	Login	Login	Kelly fez login na aplicação	2025-06-29T15:45:30.528147Z	Kelly

Figura 15 - Ecrã "Log"

O painel "Log" disponibiliza ao administrador uma visão clara e organizada sobre todas as ações realizadas na aplicação. Nele, é possível acompanhar eventos como logins, eliminação de utilizadores e outras ações relevantes associadas a diferentes módulos do sistema, com detalhes como o utilizador responsável, data e hora da ocorrência.

Além da tabela com filtros por ação, módulo, detalhes, data e utilizador, o painel permite gerar relatórios em formato PDF, filtrando por tipo de registo (voo, manutenção ou serviço), o que facilita o controlo, auditoria e apresentação formal dos dados.

Este painel contribui diretamente para a transparência e rastreabilidade das atividades no sistema, reforçando a segurança e o acompanhamento administrativo da plataforma.

Nota: Para uma descrição detalhada das interfaces e funcionalidades individuais disponíveis para cada perfil de utilizador, consultar o A1- Manual de Utilização no Apêndice A.

5. Apresentação e discussão de resultados

Os resultados indiciam que o sistema proposto atende, de forma consistente, ao objetivo de apoiar a gestão proactiva da aeronavegabilidade em pequenos operadores, automatizando tarefas repetitivas, reduzindo o risco de falhas por ultrapassagem de limites de horas/ciclos e consolidando a evidência documental através de relatórios. Destacam-se:

- (i) **Atualização automática de horas de componentes a partir do registo de voo**, fazendo com os processos sejam mais precisos e ágeis, eliminando a introdução manual de dados, reduzindo drasticamente os erros humanos e assegurando que os tempos de vida dos componentes estão sempre corretos e atualizados em tempo real. Permite que a manutenção preventiva seja agendada com base em dados de uso *reais* (horas de voo), e não em estimativas, otimizando o tempo de vida útil dos componentes e evitando substituições desnecessárias.
- (ii) **Emissão de alertas configuráveis em diferentes limiares**, o que permite detetar e agir sobre potenciais problemas antes que se transformem em falhas graves (Manutenção Preditiva), reduzindo o risco de paragens não planeadas (que podem ser extremamente caras e perigosas na aviação). Os alertas críticos (por exemplo, aproximação do limite de horas de um componente) garantem que as intervenções de segurança obrigatórias são realizadas atempadamente, aumentando a aeronavegabilidade e a segurança geral.
- (iii) **Painéis personalizados por perfil, que priorizam tarefas críticas**, fazendo com que os técnicos de manutenção e os responsáveis pela gestão da manutenção possa visualizar apenas as informações e as ordens de trabalho mais relevantes e urgentes para o seu perfil, permitindo que a equipa de manutenção priorize imediatamente as tarefas críticas. Reduzindo o tempo gasto a procurar informações ou a filtrar tarefas, aumentando a produtividade da equipa no terreno.

A aplicação transforma a gestão de manutenção de um processo reativo e manual para um processo proativo, digital e baseado em dados, resultando em maior segurança, custos operacionais reduzidos e máxima disponibilidade dos ativos.

A Tabela 12 sintetiza a verificação por inspeção dos requisitos funcionais identificados (RF01–RF10). A classificação “Demonstrado” indica que a funcionalidade foi apresentada no protótipo e exercitada passo a passo; “Parcial” assinala que o comportamento principal está presente, mas requer consolidação; “A Desenvolver” corresponde a itens previstos para iterações subsequentes.

Tabela 12 Verificação de Requisitos Funcionais

Requisito Funcional	Estado	Evidência/Observações
RF01 — Autenticação e controlo de acesso	Demonstrado	Login por perfis (Piloto, Técnico, Supervisor, Administrador).
RF02 — Registo/Gestão de aeronaves e componentes	Demonstrado	CRUD com parâmetros de vida útil (horas/ciclos).
RF03 — Registo de voos e actualização de contadores	Demonstrado	Cálculo automático de tempo e atualização de horas.
RF04 — Relatório de anomalias	Demonstrado	Formulário com anexos e encaminhamento a técnicos.
RF05 — Planeamento e monitorização preventiva	Demonstrado	<i>Due-lists</i> e priorização por limiares de utilização.
RF06 — Emissão de alertas configuráveis	Demonstrado	Limiar a 10 %, 50 % e 100 % (ajustável).
RF07 — Registo de manutenção/intervenção técnica	Demonstrado	<i>Checklist</i> , notas e associação a componente/voo.
RF08 — <i>Workflow</i> de aprovação de manutenção	Demonstrado	Estados e decisões por Supervisor.
RF09 — Exportação de relatórios auditáveis (PDF)	Parcial	Histórico de voos/manutenções e peças trocadas.
RF10 — Cópia de segurança automática de dados	Parcial	Procedimento delineado; parametrização/rotina a consolidar.

5.1.Resultados por perfil de utilizador

- **Piloto**

O piloto consegue registar voos de forma expedita e visualizar alertas antes da criação de novos registos, reduzindo a probabilidade de continuar a operação com componentes em fim de vida. O relato de anomalias com anexos agiliza a comunicação com a manutenção e melhora a qualidade da informação disponibilizada.

- **Técnico de Manutenção**

O *dashboard* do técnico centraliza *due-lists*, horas acumuladas e componentes em alerta, permitindo planeamento imediato de substituições. O registo de manutenções gera evidência (PDF) e vincula a intervenção ao componente e ao voo, reforçando rastreabilidade e consistência histórica.

- **Supervisor**

O supervisor dispõe de painéis para aprovação/rejeição de intervenções, relatórios de desempenho e acompanhamento de inspeções. A decisão suportada por detalhe técnico incrementa a conformidade e a transparência do processo.

- **Administrator**

A gestão de utilizadores, parâmetros do plano OEM e *logs* de auditoria consolida a governação do sistema, permitindo ajustes operacionais sem comprometer a segurança ou a integridade dos dados.

5.2.Vantagens e mais-valias do Maintenance Tracking System

- **Rastreabilidade e Conformidade Regulamentar**

O sistema estabelece uma linha de tempo digital auditável, registando cada operação e associando-a ao responsável. Isso garante a rastreabilidade total dos eventos de manutenção e assegura a conformidade mandatória com os requisitos de CAMO e regulamentos internacionais cruciais (e.g., EASA Part-M/145, FAA FAR 43), simplificando significativamente as auditorias de aeronavegabilidade.

- **Eficiência Operacional e Redução de Custos**

A automação de processos administrativos, como o cálculo de tempo de voo/componente e a geração de relatórios, reduz o *non-productive time*. Esta otimização liberta recursos técnicos e de gestão para atividades de maior valor, resultando numa diminuição direta dos custos operacionais e do tempo de inoperatividade (*downtime*) não programado.

- **Apoio à Tomada de Decisão**

Os *dashboards* e relatórios analíticos fornecem informação *real-time* sobre o estado da frota e a posição dos limites de manutenção. Este planeamento proativo e baseado em dados permite a gestão eficaz da manutenção, prevenindo *AOG* (*Aircraft on Ground*) e otimizando a alocação de recursos (peças e mão de obra).

- **Melhoria da Segurança e Fiabilidade**

A monitorização contínua dos parâmetros operacionais (tempo/ciclos) e a emissão de alertas configuráveis promovem a manutenção preditiva. Esta capacidade de deteção precoce de tendências anómalas é crucial para reforçar a segurança operacional, mitigar a probabilidade de incidentes e manter a fiabilidade (*dispatch reliability*) da aeronave.

- **Escalabilidade e Adaptabilidade**

A arquitetura modular do sistema confere uma elevada escalabilidade, permitindo o dimensionamento eficiente para diferentes tipos de operadores (desde Aviação Geral a MROs de grande escala). A flexibilidade da infraestrutura facilita a integração bidirecional com sistemas legados (e.g., ERP, EFB, AMOS), protegendo o investimento e permitindo expansões futuras.

- **Sustentabilidade e Transformação Digital**

O sistema impulsiona a transformação digital da gestão de ativos ao centralizar dados e eliminar o suporte em papel, alinhando a organização com os princípios da Indústria 4.0. Esta digitalização contribui indiretamente para a sustentabilidade, através da redução do desperdício e da otimização dos processos logísticos.

5.3.Implicações para a gestão da manutenção aeronáutica

A implementação desta solução tecnológica *mobile* e mais acessível que faz a Gestão da Manutenção demonstra um potencial significativo na otimização dos processos da Aeronavegabilidade Permanete.

Os resultados preliminares ou as expectativas de ganhos operacionais e estratégicos convergem para uma notável melhoria na capacidade de planeamento preditivo e preventivo, dada a precisão e a tempestividade dos dados gerados.

A funcionalidade de atualização automática do tempo de operação dos componentes a partir do registo de voo assegura a integridade dos dados mestres e estabelece uma base rigorosa para o cálculo do tempo remanescente até à próxima intervenção de manutenção, um requisito crítico para a conformidade regulamentar.

Adicionalmente, o *design* de interface orientado a painéis de controlo customizados por perfil de utilizador (ex: Gestor de Manutenção, Técnico de Manutenção) não só aumenta a disciplina de registo ao digitalizar processos de *log* no ponto de trabalho, mas também otimiza a alocação de recursos e promove a tomada de decisão informada em tempo real.

Esta abordagem de priorização de tarefas críticas, aliada à visibilidade integral do estado da frota, resulta numa redução do tempo médio de inoperatividade e minimiza o trabalho administrativo repetitivo, elevando assim a segurança operacional e o índice de *dispatch reliability*.

Os testes de usabilidade foram realizados com dez utilizadores reais, divididos nos três principais perfis da aplicação: pilotos, técnicos e supervisores. Cada utilizador realizou a tarefa completa de registar um voo na aplicação seguindo um roteiro pré-definido, onde foram analisados os critérios de tempo médio de execução, o número de cliques necessários, o número de erros cometidos, a interação com a interface e satisfação final.

Tabela 13 - Resultado de teste de usabilidade

Critérios	Avaliação
1. Tempo de conclusão da tarefa	32:6 segundos
2. Número de cliques para conclusão:	
• Fluxo sem anomalia	8 – 11 cliques
• Fluxo com anomalia anexada	10 – 13 cliques
3. Número de erros cometidos	0
4. Interação com a interface gráfica	Fluida/Estável
5. Satisfação de uso	Satisfatório

Análise qualitativa dos resultados:

- A média de 32,6 segundos obtida demonstra uma excelente eficiência no processo, cumprindo facilmente o objetivo inicial estabelecido (< 45 s).
- O número reduzido de cliques e a ausência total de erros revelam uma boa conceção de interação e validação de dados.
- A interação intuitiva foi frequentemente elogiada pelos utilizadores, especialmente devido à apresentação clara das notificações e alertas imediatos após a submissão.
- A elevada satisfação dos participantes confirma que o sistema cumpre as expectativas em termos de experiência de utilizador.

6. Conclusão

O presente projeto teve como objetivo primordial conceber e implementar um sistema integrado de gestão de operações e manutenção aeronáutica, direcionado a pequenas frotas em regime de aviação não comercial. Ao longo do seu desenvolvimento, foi possível alcançar integralmente os objetivos gerais e específicos inicialmente definidos, demonstrando a viabilidade e a relevância da solução proposta.

A adoção da metodologia em cascata mostrou-se adequada, permitindo uma execução ordenada e previsível das diferentes fases do trabalho — desde o levantamento e análise de requisitos, passando pelo desenho arquitetônico, até à implementação prática e subsequente fase de testes e validação. O levantamento exaustivo de requisitos funcionais e não funcionais garantiu que o sistema desenvolvido estivesse alinhado tanto com as necessidades operacionais como com as exigências regulatórias, assegurando pertinência técnica e utilidade prática.

O desenho da arquitetura modular revelou-se fundamental para a organização e escalabilidade da solução. A aplicação foi estruturada em quatro camadas principais: frontend em Vue.js, que assegura uma interface responsiva e intuitiva; API REST desenvolvida em Django REST Framework, responsável pela lógica de negócio; serviços de domínio, que incorporam cálculos automatizados, emissão de alertas e geração dinâmica de relatórios; e persistência em PostgreSQL, que garante a integridade, segurança e rastreabilidade de todos os dados operacionais. Esta abordagem assegurou não só robustez como também flexibilidade para futuras evoluções.

Na fase de implementação prática, destacaram-se funcionalidades críticas como o registo e monitorização de voos, o cálculo automático de horas de utilização de componentes, a emissão de notificações de manutenção preventiva e a geração de relatórios técnicos em PDF. Estas ferramentas permitem uma gestão mais transparente e eficiente, reforçando a rastreabilidade das operações e o apoio à tomada de decisão por parte de supervisores e técnicos.

A validação do sistema foi assegurada através de testes rigorosos: testes funcionais de API realizados com o Insomnia garantiram a consistência das respostas, enquanto os testes exploratórios de usabilidade confirmaram que o sistema é intuitivo, ágil e adaptado às rotinas do setor aeronáutico. O feedback dos utilizadores finais foi positivo, destacando a facilidade de utilização e a utilidade prática da aplicação no contexto real de operações.

Conclui-se, assim, que a aplicação desenvolvida não apenas automatiza processos rotineiros de manutenção e gestão de voos, como também estabelece um ecossistema digital robusto e transparente, capaz de aumentar a segurança operacional, reduzir falhas decorrentes de erros humanos e consolidar a rastreabilidade exigida pelas entidades reguladoras. Para além de cumprir integralmente os objetivos estabelecidos, este trabalho constitui um avanço significativo na digitalização da manutenção aeronáutica, fornecendo

bases sólidas para futuras investigações e evoluções, como a integração de inteligência artificial para predição de falhas ou a adoção de arquiteturas orientadas a microserviços.

Em síntese, esta dissertação evidencia que a utilização de soluções digitais bem estruturadas pode representar um contributo real e transformador para a modernização e segurança das operações aeronáuticas, alinhando-se com as crescentes exigências do setor em matéria de eficiência, conformidade e inovação tecnológica.

Perspetivas para Trabalhos Futuros

Como continuidade natural deste trabalho, destaca-se a necessidade de investir na certificação oficial da aplicação junto das entidades reguladoras competentes (ex.: EASA ou FAA), assegurando que a solução cumpra todos os requisitos técnicos e normativos necessários para ser utilizada em contexto operacional real. Este passo permitirá que a aplicação seja reconhecida como uma ferramenta validada de apoio à manutenção aeronáutica, elevando a sua credibilidade e potencial de adoção no setor.

Outra evolução fundamental consiste em permitir que a aplicação descreva de forma detalhada as tarefas de manutenção previstas no plano de manutenção aprovado (AMP), apresentando ao utilizador instruções claras e alinhadas com os manuais do fabricante. Esta funcionalidade reforçará a conformidade técnica, ajudando pilotos e técnicos a executarem intervenções de forma mais precisa e segura.

Expansão para diferentes tipos de aeronaves e contextos operacionais, permitindo adaptar o sistema não apenas à aviação geral, mas também à aviação executiva e a pequenas companhias comerciais.

Em suma, os trabalhos futuros visam consolidar a maturidade da aplicação, garantir o seu reconhecimento regulamentar e ampliar as suas funcionalidades, reforçando o contributo para a modernização da manutenção aeronáutica e para a segurança operacional no setor

Bibliografia

- Airworthiness, I.-I. F. (Maio de 2025). *Continuing Airworthiness*. Obtido de <https://skybrary.aero/sites/default/files/bookshelf/2703.pdf>
- Authority, C. A. (Maio de 2025). *Master Minimum Equipment Lists (MMEL) and Minimum Equipment Lists (MEL)*. Obtido de <https://skybrary.aero/sites/default/files/bookshelf/822.pdf>
- Aviators, L. I. (Maio de 2025). *Pilot's Operating Handbook*. Obtido de <https://longislandaviators.com/wp-content/uploads/2018/08/1978-Pilots-Operating-Handbook-Cessna-152.pdf>
- CAMP. (Maio de 2025). Obtido de <https://www.campsystems.com/>
- Capterra. (Maio de 2025). *AMOS*. Obtido de https://www.capterra.ae/software/87390/amos?utm_source=chatgpt.com
- Certification, F. A. (Maio de 2025). *Soft Time, Hard Time, and OC CM Components*. Obtido de <http://www.faa-aircraft-certification.com/soft-time-hard-time-and-oc-cm-components.html>
- EUR-Lex. (Maio de 2025). *Commission Regulation (EU) No 1321/2014*. Obtido de <https://eur-lex.europa.eu/eli/reg/2014/1321/oj/eng>
- Gorton, I. (Abril de 2025). *Essential Software Architecture*. Obtido de https://www.researchgate.net/profile/Ian-Gorton/publication/220690558_Essential_Software_Architecture_2_ed/links/02e7e5154531cb0eca000000/Essential-Software-Architecture-2-ed.pdf
- ICAO. (Abril de 2025). *Airworthiness*. Obtido de <https://www.icao.int/safety/OPS/OPS-Normal/Pages/Airworthiness.aspx>
- ICAO. (Abril de 2025). *Annexes 1 to 18*. Obtido de <https://skybrary.aero/sites/default/files/bookshelf/3370.pdf>
- Lienhart, P. (Maio de 2025). Obtido de <https://www.peter-ftp.co.uk/aviation/patrick-other/euroga-pl-easa-ml.pdf>
- NTSB. (Maio de 2025). *US Civil Aviation Accident Dashboard: 2008-2023*. Obtido de <https://www.nts.gov/safety/StatisticalReviews/Pages/CivilAviationDashboard.aspx>
- OASES. (Maio de 2025). *MRO SOFTWARE FOR AVIATION MAINTENANCE MANAGEMENT | DIGITAL, COMPLIANT, DATA-DRIVEN*. Obtido de <https://oases.aero/>
- SKYbrary. (Maio de 2025). *Aircraft Maintenance*. Obtido de <https://skybrary.aero/articles/aircraft-maintenance>

- SKYbrary. (Maio de 2025). *Aircraft Maintenance Manual (AMM)*. Obtido de <https://skybrary.aero/articles/aircraft-maintenance-manual-amm>
- SKYbrary. (Maio de 2025). *ATA Classification*. Obtido de <https://skybrary.aero/articles/ata-classification>
- SKYbrary. (Maio de 2025). *Fault Investigation Manual (FIM)*. Obtido de <https://skybrary.aero/articles/fault-investigation-manual-fim>
- SKYbrary. (Maio de 2025). *Minimum Equipment List (MEL)*. Obtido de <https://skybrary.aero/articles/minimum-equipment-list-mel>
- SOFEMA. (Maio de 2025). Obtido de <https://www.sofemaonline.com/about/blog/entry/airworthiness-considerations-related-to-easa-part-ml>
- SOFEMA. (Maio de 2025). *Considerations Related to EASA Certificates of Airworthiness (C of A) and Restricted Certificate of Airworthiness (RC of A)*. Obtido de <https://sofemaonline.com/about/blog/entry/considerations-related-to-easa-certificates-of-airworthiness-c-of-a-and-restricted-certificate-of-airworthiness-rc-of-a>
- Software, S. A. (Maio de 2025). *AMOS*. Obtido de <https://www.swiss-as.com/amos-mro>
- SoftwareConnect. (Maio de 2025). *Awiss-AS AMOS*. Obtido de <https://softwareconnect.com/reviews/swissas-amos/>
- SoftwareConnect. (Maio de 2025). *OASES MRO Software*. Obtido de <https://softwareconnect.com/reviews/oases-aviation-mro-software/>
- Veryon. (Maio de 2025). *Improving Aircraft Management with Real-Time Data: flyExclusive's Transition to Veryon*. Obtido de https://veryon.com/blog/improving-aircraft-management-with-real-time-data-flyexclusives-transition-to-veryon?utm_source=chatgpt.com
- VERYON. (Maio de 2025). *Improving Aircraft Management with Real-Time Data: flyExclusive's Transition to Veryon*. Obtido de <https://veryon.com/blog/improving-aircraft-management-with-real-time-data-flyexclusives-transition-to-veryon>
- WNRDVR. (Abril de 2025). *What is DO-178C?* Obtido de <https://www.windriver.com/solutions/learning/do-178c>

Anexo

1. Arquitetura de Base de Dados

A gestão de manutenção aeronáutica impõe requisitos de integridade referencial, rastreabilidade histórica e consistência transacional que se alinham mais naturalmente a um sistema relacional do que a soluções NoSQL orientadas a documentos. Por esse motivo o projeto adota PostgreSQL como repositório principal de dados.

A base de dados do sistema foi concebida para assegurar integridade, rastreabilidade e desempenho — três exigências centrais num domínio regulamentado pela EASA/FAA.

O esquema é relacional e normalizado até à 3.^a Forma Normal. Sete entidades formam o núcleo: Utilizador, Aeronave, Componente, Voo, Manutenção, Notificação e Log Auditoria. As relações entre elas são impostas por chaves estrangeiras indexadas, o que permite consultas como «quais os componentes desta aeronave que passaram de 90 % do intervalo?» serem executadas em poucos milissegundos.

Toda alteração estrutural é introduzida por uma *Django migration* controlada por versões em Git. Isto cria um trilho de auditoria do próprio esquema, possibilitando reconstruir o estado exato da base num dia específico — requisito habitual em investigações de incidentes.

O PostgreSQL carrega extensões que reforçam segurança e exatidão temporal: *pgcrypto* cifra campos sensíveis (e-mail, n.º de licença) e *timezone* garante cálculos corretos entre o Tempo Universal Coordenado e hora local. Já se planeia particionar a tabela de *logs*, evitando crescimento ilimitado.

Os ficheiros volumosos — fotos de anomalias e relatórios PDF — não entram no sistema. São guardados em `MEDIA_ROOT` no sistema de ficheiros e servidos por Nginx; a base grava apenas o caminho relativo, reduzindo o peso dos backups e melhorando performance.

Há uma estratégia de replicação assíncrona em estudo: um nó secundário, só de leitura, alimentará relatórios CAMO e futuros painéis de *Business Intelligence* sem sobrecarregar o primário.

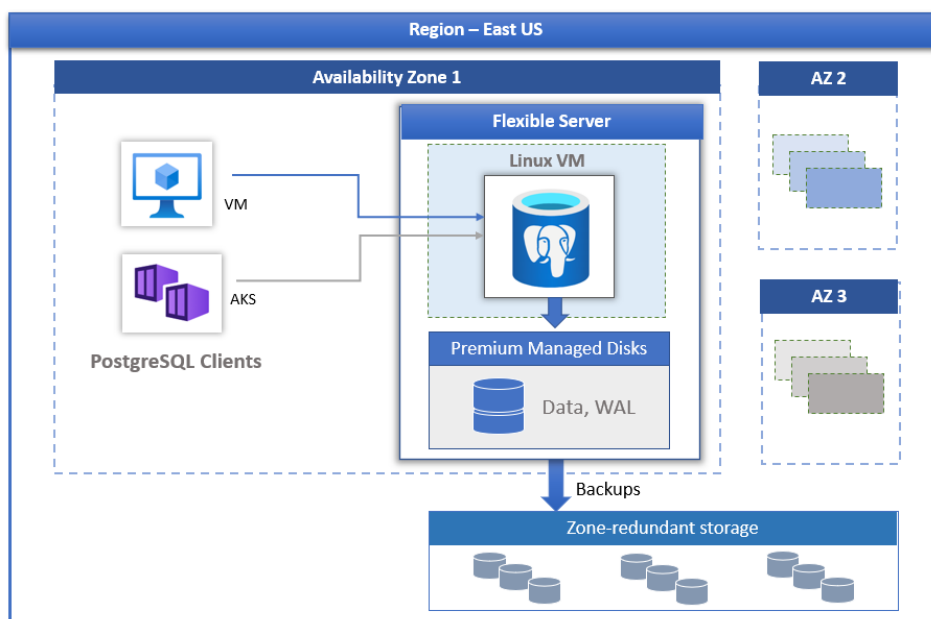


Figura 16 - Arquitetura PostgreSQL

2. Diagramas de Sequência

Os diagramas de sequência representam a interação entre os utilizadores e o sistema ao longo do tempo, mostrando a troca de mensagens entre os componentes em ordem cronológica. Eles permitem compreender como determinadas funcionalidades são executadas passo a passo, desde a ação inicial do utilizador até à resposta do sistema. (Sommerville, I. 2011) Neste projeto, estes diagramas ajudam a visualizar claramente o comportamento do sistema em cenários chave, como o registo de voo, a execução de uma manutenção ou a autenticação. A sua utilização facilita a validação da lógica e o alinhamento com os requisitos definidos. A seguir, serão apresentados os principais diagramas de sequência desenvolvidos.

2.1 Login de Utilizador

Este diagrama descreve o processo de autenticação. O utilizador insere as credenciais, que são enviadas ao *backend* (Auth API). A API procura os dados do utilizador na base de dados e devolve a informação necessária. A palavra-passe é verificada e, conforme o resultado, o sistema toma um dos seguintes caminhos: se as credenciais forem inválidas, é devolvida uma mensagem de erro; se forem válidas, o utilizador é redirecionado para o painel (*dashboard*). O diagrama contempla ainda uma estrutura condicional (*alt*) que define estas duas possibilidades.

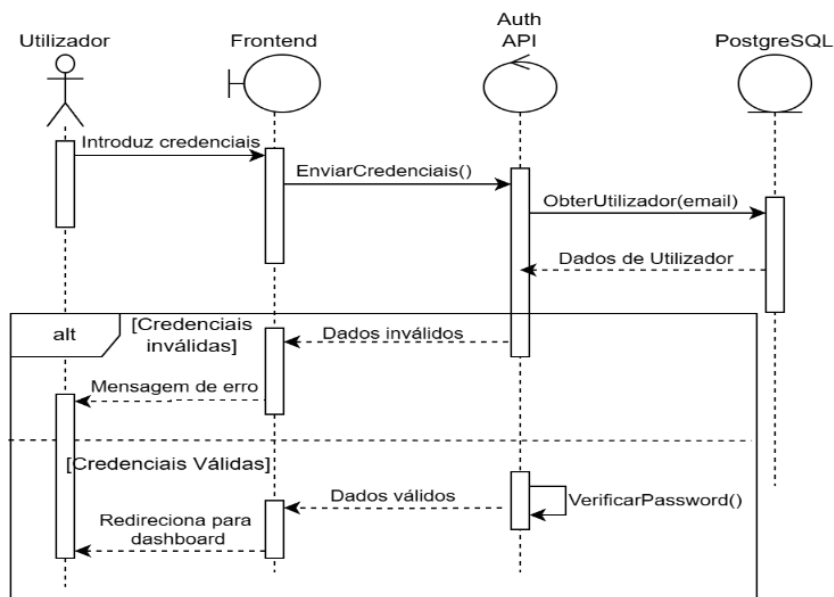


Figura 17 - Diagrama da sequência de “Login”

2.1.1 Registar Voo

Este diagrama representa o fluxo de ações que ocorre quando um piloto regista um novo voo. A sequência inicia com o envio dos dados do voo através da interface, que são então processados pela API responsável. A API cria o registo do voo e procede à atualização das horas acumuladas dos componentes da aeronave. Após calcular o novo total de horas, é realizada uma verificação de possíveis excessos. Caso algum componente ultrapasse o limite estabelecido, é gerado automaticamente um alerta que é enviado tanto ao piloto como ao técnico de manutenção.

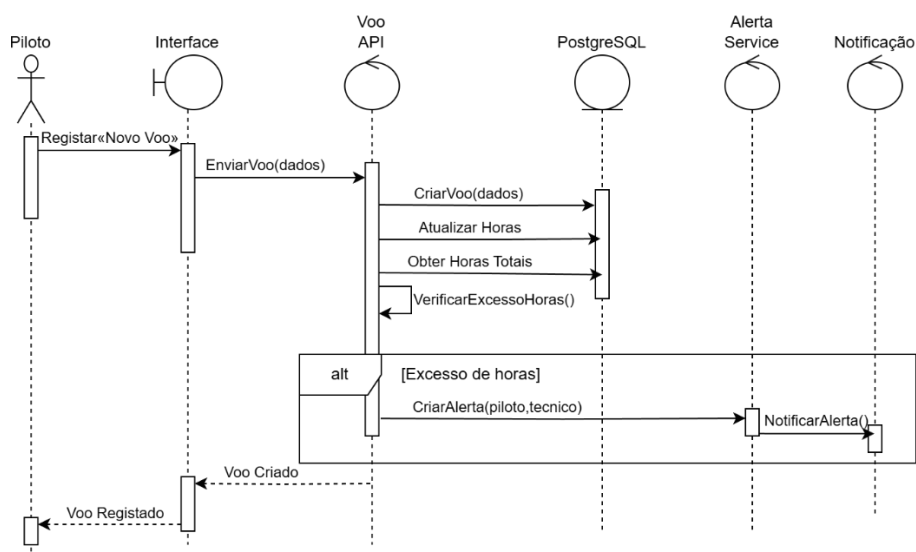


Figura 18 - Diagrama da sequência de “Registar Voo”

2.1.2 Registrar Manutenção

Neste cenário, um técnico de manutenção preenche o formulário com os dados da intervenção. A interface envia estas informações à API de manutenção, que registra o serviço e aciona o serviço de geração de relatórios (PDF). O relatório gerado é armazenado e o seu link é retornado ao sistema. Em seguida, o sistema envia notificações ao piloto e ao supervisor informando que a manutenção foi realizada e encontra-se pendente de aprovação. O fluxo termina com a confirmação ao técnico de que a manutenção foi devidamente registrada.

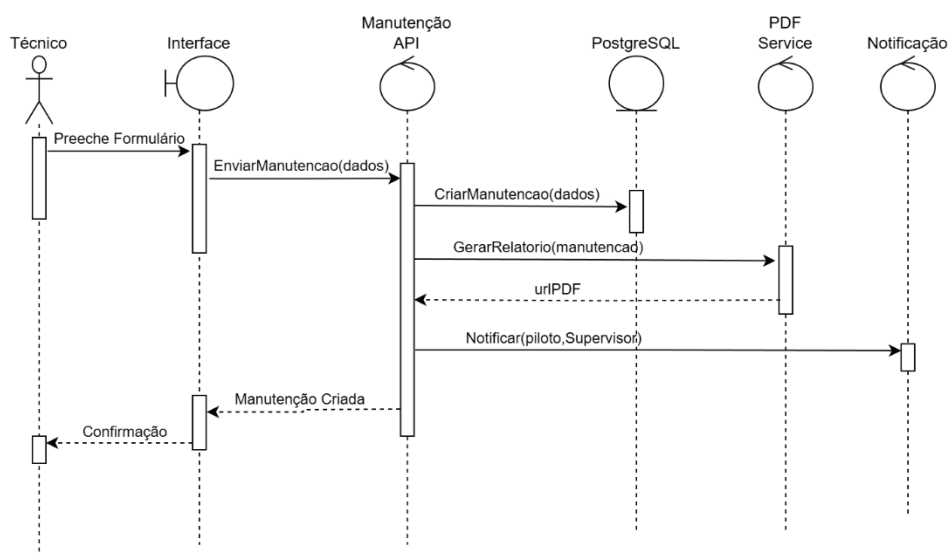


Figura 19 - Diagrama da sequência de “Registrar Manutenção”

2.1.3 Aprovar Manutenção

Este fluxo é iniciado pelo supervisor ao clicar em "Aprovar". A interface comunica essa ação à API de manutenção, que atualiza o estado da ordem para “Concluída”. Posteriormente, é feito o agendamento da execução e a informação é gravada na base de dados. Após o sucesso da operação, o sistema envia uma notificação ao técnico responsável informando que a manutenção foi aprovada. A sequência termina com a confirmação visual ao supervisor.

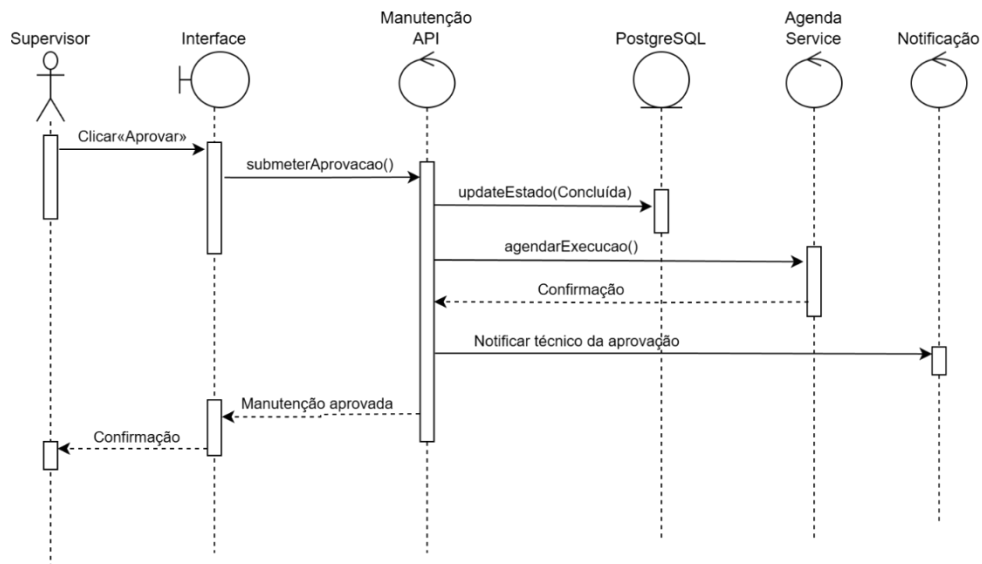


Figura 20-Diagrama de Sequência Aprovar Manutenção

2.1.4 Rejeitar Manutenção

Neste caso, o supervisor opta por rejeitar uma ordem de manutenção. A API atualiza o estado da manutenção para “Rejeitada” e, em seguida, cria uma inspeção associada. Esta nova inspeção é gravada na base de dados e o sistema invoca a API de inspeção para registrar os dados. Por fim, o técnico é notificado da rejeição e da necessidade de realizar uma nova inspeção. O supervisor recebe uma confirmação visual de que a operação foi executada com sucesso.

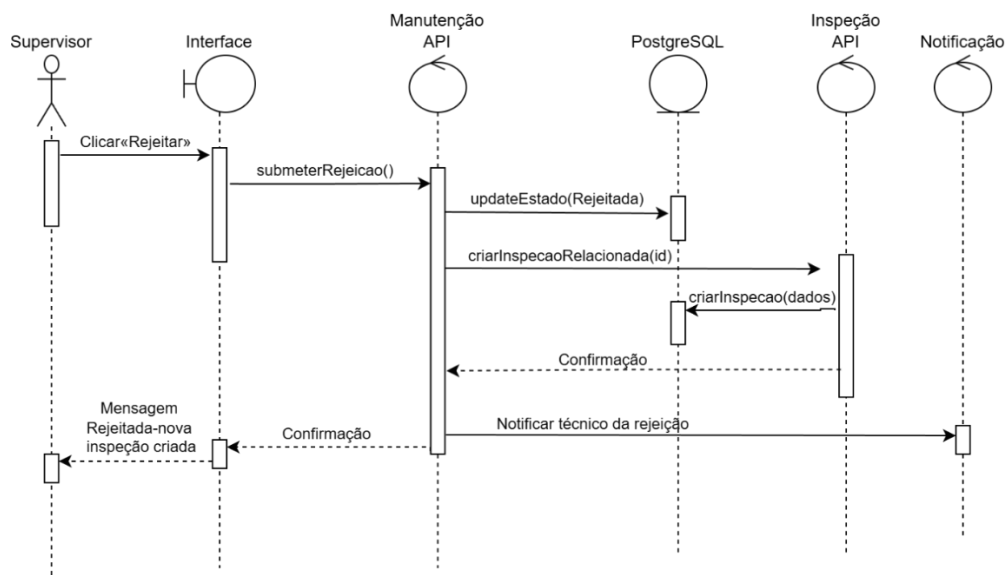


Figura 21 - Diagrama da sequência de “Rejeitar Manutenção”

2.2 Interfaces do Sistema (*Frontend*)

A interface do sistema representa o ponto de interação entre o utilizador e a aplicação. Desenvolvida com Vue.js e Tailwind CSS, garante uma navegação intuitiva, responsiva e eficiente, comunicando com o backend via API. O mapeamento das interfaces organiza as telas por perfil de utilizador (Piloto, Técnico, Supervisor, Administrador), permitindo visualizar o fluxo de navegação e o acesso às funcionalidades. As telas foram desenhadas com foco na usabilidade, clareza e cumprimento dos requisitos definidos. As imagens seguintes ilustram essas estruturas visuais, facilitando o entendimento do percurso de cada utilizador no sistema.

2.3 Codificação (*Backend*)

A codificação *backend* é a camada fundamental do projeto, responsável pela implementação de toda a lógica que permite ao sistema gerir dados, responder a requisições e executar as operações necessárias ao correto funcionamento das funcionalidades. É nesta camada que são definidos os métodos, regras e procedimentos que o sistema utiliza para processar e disponibilizar informações aos diferentes tipos de utilizadores (pilotos, técnicos, supervisores e administradores).

Neste projeto, o backend foi desenvolvido com Django e Django REST Framework (DRF), duas tecnologias robustas e altamente confiáveis para a criação e gestão de APIs RESTful. A escolha destas ferramentas foi motivada pela sua eficiência na estruturação lógica do sistema, facilidade de manutenção e escalabilidade futura.

A codificação segue um padrão arquitetural inspirado em MVC (*Model-View-Controller*), que no contexto Django está representado como MTV (*Model-Template-View*):

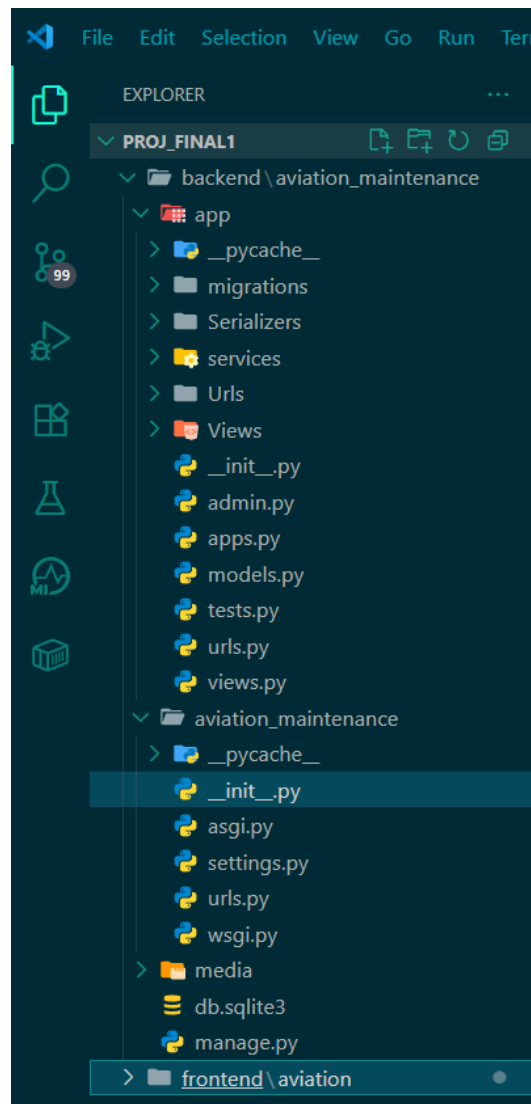


Figura 22 - Estrutura do Backend

2.4 Estrutura do *Backend* (Django REST Framework):

O *backend* do projeto encontra-se estruturado em diferentes componentes organizados da seguinte forma:

- *Models*: representam as entidades do domínio do sistema, traduzindo diretamente a estrutura das tabelas na base de dados PostgreSQL. Incluem entidades como Utilizador, Aeronave, Voo, Componente, Manutenção, Anomalia, entre outras. Os modelos definem atributos, relações e métodos específicos que garantem a integridade e consistência dos dados armazenados.
- *Serializers*: Camada intermediária essencial que converte objetos Python (modelos) em formatos adequados para transmissão via JSON e vice-versa. Cada

modelo tem um serializador próprio que permite validar e manipular dados antes de serem persistidos ou enviados ao cliente.

- *Views* (ViewSets e APIViews): Representam o controlador no contexto MVC, responsáveis pela lógica de tratamento das requisições HTTP feitas ao servidor. As *views* interagem diretamente com os *serializers* para receber e validar dados das requisições, bem como retornar as respostas adequadas ao *frontend*.
- URLs (*Endpoints*): Definem as rotas específicas que o *frontend* pode aceder para obter ou enviar dados ao *backend*, especificando claramente quais *views* devem responder a cada requisição.
- *Services*: Camada adicional onde é implementada toda a lógica específica do negócio, separando a lógica complexa das *views*. Esta organização facilita a manutenção, reutilização e testes automáticos das funcionalidades do sistema.

2.5 Detalhes Técnicos da Implementação

- **Modelos (*Models*):**

Os modelos representam as entidades fundamentais do domínio do sistema, cada uma traduzindo uma tabela na base de dados PostgreSQL. Por exemplo, o modelo de utilizadores é responsável por guardar as credenciais e informações relacionadas com o perfil (Piloto, Técnico, Supervisor ou Administrador), permitindo realizar autenticação segura, bem como gerir permissões baseadas em roles.

Exemplo simplificado do modelo Utilizador:

```
backend > aviation_maintenance > app > models.py
14
15 from django.db import models
16 from django.contrib.auth.hashers import make_password, check_password
17 from datetime import timedelta
18 from django.utils.timezone import now
19 from django.contrib.auth.models import User
20
21 class Utilizador(models.Model):
22     id = models.AutoField(primary_key=True)
23     nome = models.CharField(max_length=150)
24     user = models.OneToOneField(User, on_delete=models.CASCADE, null=True,
25                                blank=True)
26     telefone = models.CharField(max_length=200, null=True, blank=True)
27     perfil = models.ForeignKey(Perfil, on_delete=models.PROTECT)
28     data_criacao = models.DateTimeField(auto_now_add=True)
29     email = models.EmailField(unique=True, null=True, blank=True)
30     status = models.TextField(null=False, default="ativo")
31     last_login = models.DateTimeField(null=True, blank=True)
32
33     def __str__(self):
34         return self.user.get_full_name() or self.user.username
35
36
37
```

Figura 23 - Modelo utilizador

- Serializadores (*Serializers*)

Os serializadores são usados para traduzir objetos complexos em JSON, assegurando que os dados estejam devidamente validados antes de serem enviados ao cliente ou armazenados na base de dados.

Exemplo simplificado do serializador do modelo Utilizador:

```
backend > aviation_maintenance > app > Serializers > serializer.py
40 class UtilizadorSerializer(serializers.ModelSerializer):
41     username = serializers.CharField(write_only=True, required=False, allow_blank=True)
42     email = serializers.EmailField(write_only=True)
43     password = serializers.CharField(write_only=True, min_length=8, required=False, allow_blank=True)
44     first_name = serializers.CharField(write_only=True, required=False, allow_blank=True)
45     last_name = serializers.CharField(write_only=True, required=False, allow_blank=True)
46
47     perfil_id = serializers.PrimaryKeyRelatedField(
48         queryset=Perfil.objects.all(), source="perfil", write_only=True
49     )
50     perfil = PerfilSerializer(read_only=True)
51     user = UserSerializer(read_only=True)
52
53     class Meta:
54         model = Utilizador
55         fields = [
56             "id", "username", "email", "password",
57             "first_name", "last_name", "nome",
58             "telefone", "perfil_id", "perfil",
59             "status", "data_criacao", "user"
60         ]
61         read_only_fields = ["data_criacao", "user"]
62
63     def create(self, validated_data):
64         username = validated_data.pop("username", "").strip()
65         email = validated_data.pop("email")
66         password = validated_data.pop("password")
67         first_name = validated_data.pop("first_name", "")
68         last_name = validated_data.pop("last_name", "")
69
70         if not username:
71             username = email.split("@")[0]
72
73         if User.objects.filter(username=username).exists():
74             raise serializers.ValidationError({"username": "Já existe."})
75         if User.objects.filter(email=email).exists():
76             raise serializers.ValidationError({"email": "Já existe."})
```

Figura 24 - Serializador do utilizador

- **Views (Controladores)**

As *views* representam os pontos de acesso (*controllers*) do *backend*. Elas recebem as requisições HTTP feitas pelo *frontend*, processam as informações, interagem com serializadores e modelos, e enviam as respostas em JSON ao *frontend*.

Exemplo simplificado da *ViewSet* de utilizadores:

```
backend > aviation_maintenance > app > Views > utilizadorView.py
18 # Utilizador Views
19 @api_view(['GET'])
20 def utilizador_list(request):
21     if request.method == 'GET':
22         utilizadores = Utilizador.objects.all()
23         serializer = UtilizadorSerializer(utilizadores, many=True)
24         data = serializer.data
25         # Remove 'user' field from each object
26         for item in data:
27             item['email'] = item['user']['email'] if 'user' in item and 'email' in item['user'] else
28             item.pop('user', None)
29         return JsonResponse(data, safe=False)
30
31 @api_view(['GET'])
32 def utilizador_get(request, id):
33     try:
34         utilizador = Utilizador.objects.get(id=id)
35     except Utilizador.DoesNotExist:
36         utilizador={}
37         return JsonResponse({"message": "Utilizador not found."}, status=status.HTTP_208_ALREADY_GONE)
38
39     if request.method == 'GET':
40         serializer = UtilizadorSerializer(utilizador)
41         data=serializer.data
42         # Remove 'user' field from the object
43         data['email'] = data['user']['email'] if 'user' in data and 'email' in data['user'] else
44         data.pop('user', None)
45         # Add email field from nested user
46         return JsonResponse(data, safe=False)
47
48 @api_view(['POST'])
49 @permission_classes([AllowAny]) # registo público
50 def utilizador_create(request):
```

Figura 25 - *ViewSet* de utilizadores

```
backend > aviation_maintenance > app > Views > utilizadorView.py
52 @api_view(['POST'])
53 @permission_classes([AllowAny]) # registo público
54 def utilizador_create(request):
55     serializer = UtilizadorSerializer(data=request.data)
56     if serializer.is_valid():
57         utilizador = serializer.save()
58         return Response(UtilizadorSerializer(utilizador).data,
59                         status=status.HTTP_201_CREATED)
60     return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
61
62 @api_view(['PUT'])
63 def utilizador_update(request, id):
64     try:
65         utilizador = Utilizador.objects.get(id=id)
66     except Utilizador.DoesNotExist:
67         return Response(status=status.HTTP_404_NOT_FOUND)
68
69     serializer = UtilizadorSerializer(utilizador, data=request.data)
70     if serializer.is_valid():
71         serializer.save()
72         return Response(serializer.data, status=status.HTTP_200_OK)
73     return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
74
75 @api_view(['DELETE'])
76 def utilizador_delete(request, id):
77     try:
78         utilizador = Utilizador.objects.get(id=id)
79     except Utilizador.DoesNotExist:
80         return JsonResponse({"message": "Utilizador not found."}, status=status.HTTP_208_ALREADY_GONE)
81
82     if request.method == 'DELETE':
83         utilizador.status = "desativado" # Update status instead of deleting
84         utilizador.save()
85         return JsonResponse({"message": "Utilizador desativado successfully"}, status=status.HTTP_200_OK)
```

Figura 26 - *ViewSet* de utilizadores cont.

- **URLs (Rotas)**

As URLs são utilizadas para definir os caminhos de acesso aos diferentes *endpoints*, mapeando as requisições específicas para as *views* correspondentes.

Exemplo simplificado de URLs:

```
backend > aviation_maintenance > app > Urls > utilizadorUrl.py
1  from django.urls import path
2  from ..Views import utilizadorView, password_reset
3  from django.shortcuts import render
4  from app.Views.utilizadorView import utilizador_login
5
6
7  urlpatterns = [
8      # Utilizador URLs
9      path('utilizador/', utilizadorView.utilizador_list),
10     path('password-reset-request/', password_reset.password_reset_request, name='password-reset-request'),
11     path('password-reset-confirm/', password_reset.password_reset_confirm, name='password-reset-confirm'),
12     path('utilizador/piloto/', utilizadorView.utilizador_list_piloto),
13     path('utilizador/tecnico/', utilizadorView.utilizador_list_tecnico),
14     path('utilizador/<int:id>/', utilizadorView.utilizador_get),
15     path('utilizador/registar/', utilizadorView.utilizador_create),
16     path('utilizador/editar/<int:id>/', utilizadorView.utilizador_update),
17     path('utilizador/eliminar/<int:id>/', utilizadorView.utilizador_delete),
18     path('utilizador/login/', utilizador_login, name='utilizador-login'),
19     path('utilizador/recuperar-password/', password_reset.password_reset_request, name='password-reset-request'),
20     path('utilizador/recuperar-password/confirm/', password_reset.password_reset_confirm, name='password-reset-confirm'),
21
22
23 ]
```

Figura 27 - URLs utilizador

- **Serviços (Services)**

A camada de serviços permite abstrair a lógica mais complexa e detalhada do negócio, simplificando as *views* e garantindo que a manutenção e evolução futura sejam mais fáceis.

Exemplo simplificado de serviço de alerta:

```
backend > aviation_maintenance > app > services > alertas.py
4 from django.db.models import F, Q
5 from app.models import Utilizador, Aeronave, Voo, Componente, Manutencao, RelatoAnomalia, TipoManutencao, EstadoManutencao, Perfil,
6
7
8 LIMIAR_ALERTA = 0.90      # 90 % das horas-Limite
9
10 def verificar_alertas_para_piloto(piloto_id: int) -> int:
11     """
12     aeronaves_ids = (
13         Voo.objects
14         .filter(piloto_id=piloto_id)
15         .values_list('aeronave_id', flat=True)
16         .distinct()
17     )
18
19     # componentes acima de 90 % do limite
20     componentes = (
21         Componente.objects
22         .filter(aeronave_id__in=aeronaves_ids)
23         .annotate(pct=F('horas_utilizadas') / F('horas_limite'))
24         .filter(pct__gte=LIMIAR_ALERTA)
25     )
26
27     criados = 0
28     for c in componentes:
29         titulo = (
30             f"Componente '{c.nome}' da aeronave {c.aeronave.matricula} "
31             f"está com {c.horas_utilizadas}/{c.horas_limite} h"
32         )
33
34         # evita duplicar a mesma notificação
35         notif, created = Notificacao.objects.get_or_create(
36             destinatario_id=piloto_id,
37             entidade_id=c.id,
38             entidade='componente',
39             defaults={
```

Figura 28-Serviço de alerta

```
25     )
26
27     criados = 0
28     for c in componentes:
29         titulo = (
30             f"Componente '{c.nome}' da aeronave {c.aeronave.matricula} "
31             f"está com {c.horas_utilizadas}/{c.horas_limite} h"
32         )
33
34         # evita duplicar a mesma notificação
35         notif, created = Notificacao.objects.get_or_create(
36             destinatario_id=piloto_id,
37             entidade_id=c.id,
38             entidade='componente',
39             defaults={
40                 'titulo': titulo,
41                 'mensagem': titulo,
42                 'lida': False,
43                 'tipo': 'alerta',
44                 'criada_em': timezone.now(),
45             },
46         )
47         criados += int(created)
48
49     return criados
50
```

Figura 29-Serviço de alerta cont.

2.6 Codificação *Backend* – Funcionalidades Específicas

Nesta secção são apresentados os detalhes técnicos adicionais relacionados com as funcionalidades essenciais do sistema, exemplificando com clareza os principais aspetos que contribuíram diretamente para o sucesso da aplicação desenvolvida.

- **Cálculo de Alerta de Componente**

Para garantir segurança e conformidade, implementou-se um serviço específico (*verificar_alerta_componentes*) responsável pela gestão dos alertas de componentes das aeronaves. Esta função verifica continuamente se os componentes atingem um limiar crítico (por exemplo, menos de 10 horas para manutenção), gerando automaticamente alertas de manutenção preventiva:

```
backend > aviation_maintenance > app > services > alertas.py
57 def verificar_alerta_tecnico(tecnico_id: int) -> int:
58     """
59     criados = 0
60     agora = timezone.now()
61
62     # -----
63     # 1) COMPONENTES em limite
64     # -----
65     componentes_limite = (
66         Componente.objects
67         .annotate(pct=F('horas_utilizadas') / F('horas_limite'))
68         .filter(
69             Q(horas_limite__gt=0),
70             pct__gte=LIMIAR_ALERTA
71         )
72     )
73
74     for c in componentes_limite:
75         titulo = (
76             f"[Alerta] • Componente '{c.nome}' da aeronave "
77             f"{c.aeronave.matricula} atingiu "
78             f"{c.horas_utilizadas}/{c.horas_limite} h"
79         )
80
81         created = Notificacao.objects.get_or_create(
82             destinatario_id=tecnico_id,
83             entidade='componente',
84             entidade_id=c.id,
85             defaults={
86                 "titulo": titulo,
87                 "mensagem": titulo,
88                 "tipo": "alerta",
89                 "lida": False,
90                 "criada_em": agora,
91             },
92         )
93     criados += int(created)
```

Figura 30 - Cálculo dos alertas dos componentes

- **Endpoint REST “Registrar Voo”**

A funcionalidade para registrar novos voos é central para a operação do sistema. Através de uma API RESTful, o piloto fornece dados fundamentais (matrícula, horários de partida e chegada, etc.), permitindo o cálculo automático de tempo de voo e atualização das horas totais de componentes:

```
backend > aviation_maintenance > app > Views > vooView.py
73
74 @api_view(['POST'])
75 def voo_create(request):
76     if request.method != 'POST':
77         return Response(status=405)
78
79     serializer = VooSerializer(data=request.data)
80     if not serializer.is_valid():
81         return Response(serializer.errors, 400)
82
83     # 1) grava voo
84     voo = serializer.save() # ← objecto Voo
85
86     # 2) actualiza horas dos componentes da aeronave
87     tempo = voo.tempo_voo or 0
88     if tempo:
89         (Componente.objects
90          .filter(aeronave=voo.aeronave)
91          .update(horas_utilizadas=F('horas_utilizadas') + tempo))
92
93     # opcional: recalcular flag em_alerta
94     componentes = Componente.objects.filter(aeronave=voo.aeronave)
95     for c in componentes:
96         if c.horas_limite:
97             c.em_alerta = (c.horas_utilizadas / c.horas_limite) >= 0.90
98             c.save(update_fields=['em_alerta'])
99
100     return Response(VooSerializer(voo).data, 201)
101
102 @api_view(['PUT'])
```

Figura 31 - Endpoint REST “Registrar Voo”

- **Geração Automática de Relatórios PDF**

A geração de PDFs foi implementada usando a biblioteca *ReportLab*, permitindo compilar relatórios completos sobre as aeronaves diretamente pelo *backend*. Um exemplo simplificado de geração é apresentado abaixo:

```
backend > aviation_maintenance > app > services > pdf_generator.py
772 def gerar_relatorio_servico_aeronave(aeronave_id: int,
807     pendente_id = (
809         .filter(nome__icontains="pend")
810         .values_list('id', flat=True)
811         .first()
812     )
813
814     m_realizadas = (
815         Manutencao.objects
816         .filter(componente__aeronave=aero,
817               data_execucao__date__gte=inicio_range)
818         .select_related('componente', 'tecnico', 'tipo_manutencao')
819         .order_by('-data_execucao')
820     )
821
822     m_pendentes = (
823         Manutencao.objects
824         .filter(componente__aeronave=aero)
825         .filter(Q(data_execucao__isnull=True) | Q(estado_manutencao_id=pendente_id))
826         .select_related('componente', 'tipo_manutencao')
827         .order_by('id')
828     )
829
830     inspecoes = (
831         Inspecao.objects
832         .filter(aeronave=aero)
833         .select_related('tecnico')
834         .order_by('-data_inspecao')[:5]
835     )
836
837     # duas queries: primeiro conta críticos, depois corta lista
838     anomalias_qs = (
839         RelatoAnomalia.objects
840         .filter(componente__aeronave=aero)
841         .select_related('componente', 'piloto')
842         .order_by('-data_relato')
843     )
```

Figura 32 - Criação de relatórios PDF de for automática

```

backend > aviation_maintenance > app > services > pdf_generator.py
772 def gerar_relatorio_servico_aeronave(aeronave_id: int,
807     pendente_id = (
809         .filter(nome__icontains="pend")
810         .values_list('id', flat=True)
811         .first()
812     )
813
814     m_realizadas = (
815         Manutencao.objects
816         .filter(componente__aeronave=aero,
817             data_execucao__date__gte=inicio_range)
818         .select_related('componente', 'tecnico', 'tipo_manutencao')
819         .order_by('-data_execucao')
820     )
821
822     m_pendentes = (
823         Manutencao.objects
824         .filter(componente__aeronave=aero)
825         .filter(Q(data_execucao__isnull=True) | Q(estado_manutencao_id=pendente_id))
826         .select_related('componente', 'tipo_manutencao')
827         .order_by('id')
828     )
829
830     inspecoes = (
831         Inspecao.objects
832         .filter(aeronave=aero)
833         .select_related('tecnico')
834         .order_by('-data_inspecao')[:5]
835     )
836
837     # duas queries: primeiro conta criticos, depois corta lista
838     anomalias_qs = (
839         RelatoAnomalia.objects
840         .filter(componente__aeronave=aero)
841         .select_related('componente', 'piloto')
842         .order_by('-data_relato')
843     )

```

Figura 33 - Criação de relatórios PDF de for automática cont.

2.7 Testes

Os testes representam uma etapa essencial no desenvolvimento de qualquer sistema, garantindo a qualidade, robustez e eficácia das funcionalidades implementadas. O objetivo principal é assegurar que o sistema esteja de acordo com as expectativas iniciais e os requisitos estabelecidos, evitando falhas, bugs e erros operacionais antes da aplicação entrar em produção.

Nesta fase, foram realizados dois tipos principais de testes:

- **Testes Automatizados da API**

Os testes automatizados da API foram realizados utilizando o Insomnia, uma ferramenta robusta que permitiu validar com eficiência as funcionalidades essenciais do sistema, nomeadamente:

- Registo de voo;

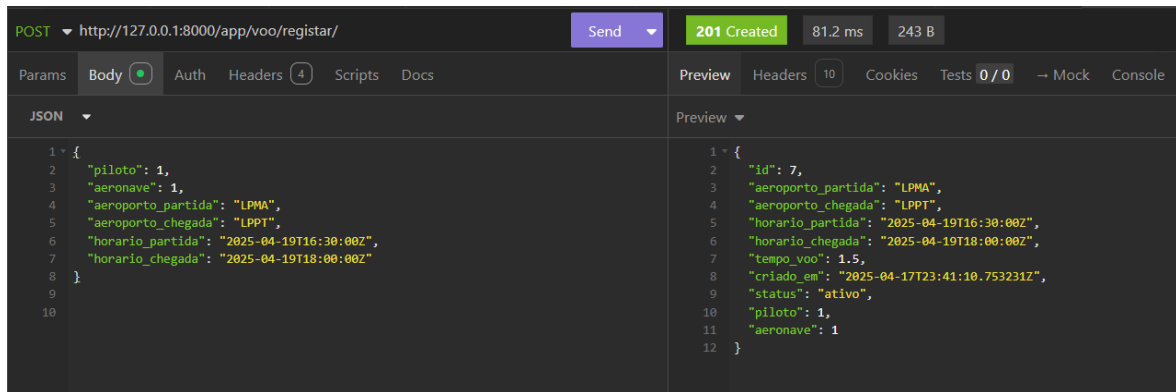


Figura 34 - API Registo de voo

- Componentes em alerta;

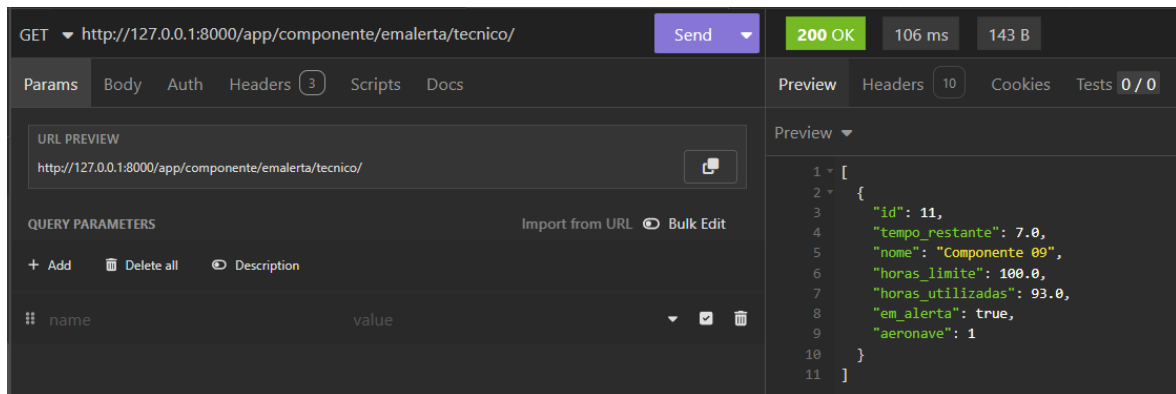


Figura 35 - API Componentes em alerta

- Autenticação de utilizadores:

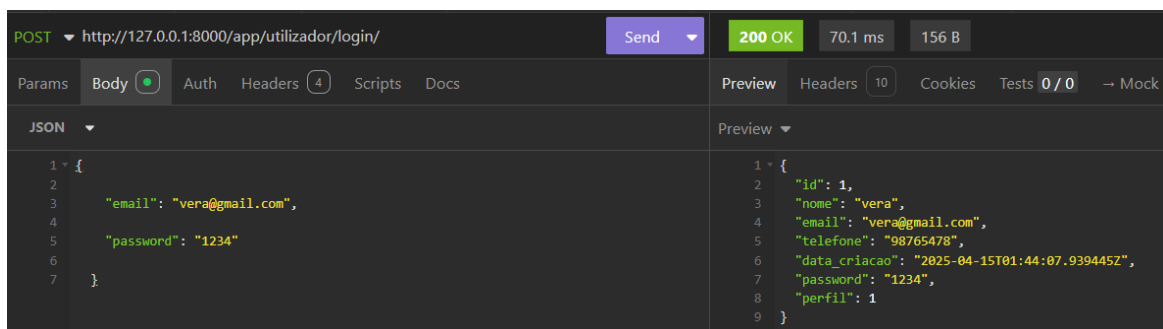


Figura 36 - API Autenticação de utilizadores

Cada *endpoint* foi testado através da execução de requisições HTTP específicas (GET, POST), sendo validados aspetos como o tempo de resposta, formato dos dados (JSON Schema), consistência dos resultados e gestão correta dos estados de erro.