



Universidade do Minho  
Mestrado em Cibersegurança

Aplicações e Serviços de Computação  
em Nuvem Fase-1  
Grupo nº33

Carlos Daniel Silva Fernandes  
(A102499)

Afonso Miguel da Silva Ribeiro  
(PG60232)

Pedro Augusto Ennes Camargo  
(A102504)

Renato Pereira Garcia  
(A101987)

2 de novembro de 2025

# Conteúdo

<b>1</b>	<b>Contextualização</b>	<b>3</b>
1.1	Visão Geral da Aplicação . . . . .	3
1.2	Principais Tecnologias . . . . .	3
1.3	Arquitetura . . . . .	3
1.4	Componentes . . . . .	3
1.5	Infraestrutura . . . . .	4
<b>2</b>	<b>Arquitetura e principais componentes</b>	<b>5</b>
2.1	Web Server . . . . .	5
2.2	Base de dados . . . . .	5
<b>3</b>	<b>Conclusão</b>	<b>6</b>

# Capítulo 1

## Contextualização

### 1.1 Visão Geral da Aplicação

Numa primeira fase deste projeto, é feito o estudo da aplicação de tracking de voos airtrail. Neste relatório estão presentes detalhes sobre a arquitetura da aplicação, as suas funcionalidades e APIs, bem como uma reflexão e discussão sobre potenciais pontos de falha da aplicação, gargalos de desempenho e possíveis dificuldades na instalação.

### 1.2 Principais Tecnologias

- **Base de Dados:** *PostgreSQL*
- **Web Server:** Desenvolvido com *SvelteKit* e *TypeScript*
- **Backend:** Integrado no próprio frontend através do *SvelteKit*
- **Acesso à Base de Dados:** *Kysely* (SQL Builder)
- **Schema e Migrações:** Geridas com *Prisma*

### 1.3 Arquitetura

A arquitetura adotada é do tipo **cliente-servidor**, em que o cliente interage com o servidor web que, por sua vez, comunica com a base de dados.

### 1.4 Componentes

- **Web Server:** Container `airtrail`
- **Base de Dados:** Container `airtrail_db`

## 1.5 Infraestrutura

A infraestrutura é composta por dois containers Docker que comunicam entre si através de uma *bridge network* criada automaticamente pelo *docker compose*:

NETWORK ID	NAME	DRIVER	SCOPE
b16b80e06117	airtrail_default	bridge	local

A comunicação entre a base de dados e o servidor web é facilitada através de um **ORM** (*Object-Relational Mapping*), utilizando as bibliotecas *Prisma* e *Kysely* para abstrair e simplificar o acesso aos dados.

## **Capítulo 2**

# **Arquitetura e principais componentes**

A aplicação tem uma arquitetura cliente-servidor, com o servidor ligado a uma base de dados PostgreSQL. Tanto o servidor como a base de dados correm em containers Docker, e comunicam através de uma bridge network criada automaticamente pelo docker compose. Entre a base de dados e o web server é utilizado um ORM (Object-Relational Mapping) para facilitar a comunicação entre os dois componentes

### **2.1 Web Server**

### **2.2 Base de dados**

## Capítulo 3

# Conclusão

Durante o desenvolvimento desta fase, o principal desafio passou por compreender e implementar corretamente as curvas de **Bezier** e de **Catmull-Rom**, bem como ajustar as transformações de forma a garantir um movimento natural dos objetos. Apesar destas adversidades, consideramos que concluímos o trabalho com a qualidade exigida pela equipa docente.