INSTITUTO TECNOLÓGICO DE AERONÁUTICA

Estudos para Aviação do Hoje e do Amanhã

TED n. 11525720240005-003882/2024

PRODUTO 2

AIRDATA – Artificial Intelligence for Aviation Data Analysis

airdata

Histórico de versões:

Versão	*Data*	*Responsável*	*Descrição da Alteração*
1.0	4 de mar. 2026	Prof. Dr. Marcelo Xavier Guterres	Versão inicial do produto II

Coordenação Geral
Prof. Dr. Cláudio Jorge Pinto Alves
claudioj@ita.br

Gerente da Etapa
Prof. Dr. Marcelo Xavier Guterres
guterres@ita.br

Equipe ITA

Prof. Dr. Flávio Mendes Neto

Prof. Dr. Dimas Betioli Ribeiro

Prof. Dr. João Basílio Tarelho Szenczuk

Msc. Guilherme Trindade Tolentino Bernardo

Msc. Jean Phelipe de Oliveira Lima

Vitor Lucas Kohls Correa

Felipe Silva Ramos Lelis

Produto II

Meta 1 | Etapa 6: AirData

Sistema de Integração de Dados

Sumário Executivo

O presente Produto faz parte da relação de entregas da Etapa 6 da Meta 1 - do TED - Termo de Execução Descentralizada n. 1525720240005-003882/2024, firmado entre a Secretaria de Aviação Civil, cujo número de Processo é 50020.008564/2024-14. Tal TED foi decorrente de estruturação entre a SAC – Secretaria de Aviação Civil e o ITA – Instituto Tecnológico de Aeronáutica, com foco em *"Estudos para Aviação de Hoje e do Amanhã"*. O ITA respondeu à demanda da SAC e o TED citado foi estruturado em 02 (duas) Metas com 16 (dezesseis) Etapas. O citado TED foi firmado no dia 20/12/2024.

Dados Referenciais:

- TED n. 1525720240005-003882/2024
- Processo n. 50020.008564/2024-14
- Etapa 6 da Meta 1
- Produto II

Para tanto, o presente relatório documenta o segundo Produto do **AirData**.

Conteúdo

Lista de Siglas

ACID *Atomicity, Consistency, Isolation, Durability*

ADS-B *Automatic Dependent Surveillance-Broadcast*

AIXM *Aeronautical Information Exchange Model*

ANAC Agência Nacional de Aviação Civil

API *Application Programming Interface*

ATM *Air Traffic Management*

CAPE *Convective Available Potential Energy*

CAT *Clear Air Turbulence*

CIN *Convective Inhibition*

CORS *Cross-Origin Resource Sharing*

CPU *Central Processing Unit*

CSS *Cascading Style Sheets*

CSV *Comma-Separated Values*

DAG *Directed Acyclic Graph*

DECEA Departamento de Controle do Espaço Aéreo

ERA5 *ECMWF Reanalysis v5*

ETL *Extract, Transform, Load*

FIXM *Flight Information Exchange Model*

GNN *Graph Neural Networks*

GPU *Graphics Processing Unit*

HNSW *Hierarchical Navigable Small World*

HTML *HyperText Markup Language*

HTTP *HyperText Transfer Protocol*

HTTPS *HyperText Transfer Protocol Secure*

IA *Inteligência Artificial*

ICA Instruções do Comando da Aeronáutica

ICAO *International Civil Aviation Organization*

INMET Instituto Nacional de Meteorologia

IP *Internet Protocol*

ITA Instituto Tecnológico de Aeronáutica

JSON *JavaScript Object Notation*

KPI *Key Performance Indicator*

LLM *Large Language Model*

METAR *Meteorological Aerodrome Report*

NCEP *National Centers for Environmental Prediction*

NOTAM *Notice to Airmen*

OWL *Web Ontology Language*

PDF *Portable Document Format*

RAG *Retrieval-Augmented Generation*

RBAC Regulamentos Brasileiros da Aviação Civil

RBHA Regulamentos Brasileiros de Homologação Aeronáutica

RDF *Resource Description Framework*

REST *Representational State Transfer*

SGBD *Sistema de Gerenciamento de Banco de Dados Relacional*

SHACL *Shapes Constraint Language*

SISCEAB *Sistema de Controle do Espaço Aéreo Brasileiro*

SQL *Structured Query Language*

SSE *Server-Sent Events*

SSH *Secure Shell*

SSR *Secondary Surveillance Radar*

TLS *Transport Layer Security*

TTL *Time To Live*

URI *Uniform Resource Identifier*

URL *Uniform Resource Locator*

UUID *Universally Unique Identifier*

VRA Voo Regular Ativo

WIDOCO *WIzard for DOCumenting Ontologies*

WXXM *Weather Information Exchange Model*

XML *Extensible Markup Language*

Lista de Figuras

Lista de Tabelas

1 Infraestrutura *Extract, Transform, Load*

1.1 Introdução

Este capítulo apresenta o status da implementação de infraestrutura de *Extract, Transform, Load* (ETL) de dados no escopo do projeto **AirData**. Essa implementação integra o Produto II do projeto.

1.1.1 Produto II: Sistema de Integração de Dados

O Produto II é um sistema integrado com infraestrutura de armazenamento, componentes de ETL, interfaces para OpenSky, mecanismos de padronização e validação. Inclui documentação técnica completa.

O objetivo técnico da Infraestrutura ETL é construir uma plataforma que permita a orquestração e monitoramento das tarefas de ETL, além de prover o acesso e validação de dados. Como orquestrador de tarefas foi utilizado o **Apache Airflow**. O **PostgreSQL** foi utilizado como banco de dados relacional para armazenamento de dados.

O ecossistema do Produto II integra múltiplos componentes especializados:

Tabela 1.1: Componentes do Produto II e suas finalidades

Componente	**Finalidade**	**Ferramentas**
Ontologia	Representação semântica do domínio	Protégé, OWL, *Resource Description Framework* (RDF), SPARQL
Pipeline ETL + Validação	Orquestração e checagem de consistência entre bases	GitHub Actions, Docker, Apache Airflow
Interface OpenSky	Conectores de dados de trajetória	*Application Programming Interface* (API) *Representational State Transfer* (REST), Python, Pandas, FastAPI
Chatbot AirData	Consulta semântica com RAG	LangChain, HuggingFace
Data Warehouse	Camada DW com versionamento	Docker, Postgres
Documentação Técnica	Descritivos de API, ontologias, pipelines	Overleaf, LaTeX

1.2 Arquitetura - Lessonia

A máquina virtual, criada no Lessonia, destinada ao **AirData**, está configurada no *Internet Protocol* (IP) `161.24.29.22`. Trata-se de um IP válido, disponível para internet, porém apenas as portas de *HyperText Transfer Protocol* (HTTP) e *HyperText Transfer Protocol Secure* (HTTPS) estão expostas. Por isso, para acessar os serviços sem necessidade de mapeamento de porta via *Secure Shell* (SSH) (`ssh -L`) foi instalado o **NGINX**, um proxy reverso, para lidar com as requisições HTTP (porta 80).

1.2.1 Serviços Implantados

A infraestrutura computacional do projeto foi organizada de forma modular, contemplando serviços responsáveis pela orquestração de pipelines de dados, armazenamento persistente e visualização administrativa. A seguir são descritos os principais componentes implantados no ambiente.

1. **Apache Airflow**

   O *Apache Airflow* foi adotado como o principal orquestrador de pipelines de dados do sistema. Sua função é coordenar a execução das tarefas de ETL (Extract, Transform and Load), permitindo a definição de fluxos de processamento por meio de *Directed Acyclic Graphs* (DAGs).

Nesse contexto, o Airflow centraliza a execução dos scripts responsáveis pela coleta, transformação e carga dos dados utilizados no projeto **AirData**. Além disso, a ferramenta oferece recursos importantes para ambientes de engenharia de dados, tais como:

- agendamento automático de tarefas;
- monitoramento do estado das execuções;
- registro detalhado de logs;
- reexecução de tarefas em caso de falhas;
- visualização gráfica das dependências entre tarefas.

A interface web administrativa do Airflow foi publicada através de um proxy reverso configurado no servidor *NGINX*, permitindo acesso via navegador. O serviço pode ser acessado através do endereço:

<http://161.24.29.22/airflow/>

2. **PostgreSQL**

   O *PostgreSQL* foi utilizado como sistema de gerenciamento de banco de dados relacional (SGBD) da infraestrutura. Este banco de dados exerce duas funções principais no sistema:

   - armazenamento dos metadados operacionais do Apache Airflow;
   - armazenamento dos dados processados pelo sistema **AirData**.

   Para melhor organização das informações e separação lógica dos dados, foram utilizados dois *schemas* distintos:

   - `public`: responsável por armazenar as tabelas internas utilizadas pelo Airflow para controle de execução, histórico de tarefas, logs e agendamentos;
   - `airdata`: destinado ao armazenamento das informações processadas pelo pipeline de dados do projeto, incluindo dados coletados, transformados e preparados para análise.

   Por questões de segurança da infraestrutura, o banco de dados PostgreSQL não possui sua porta padrão (5432) exposta diretamente à internet, restringindo o acesso apenas aos serviços internos do servidor.

3. **pgweb**

   O *pgweb* foi implantado como uma interface web de administração do banco de dados PostgreSQL. Trata-se de uma ferramenta leve que permite realizar operações de inspeção e análise sobre os dados armazenados.

   Entre as funcionalidades disponibilizadas pela ferramenta destacam-se:

   - visualização de tabelas e estruturas de dados;
   - execução manual de consultas SQL;
   - inspeção de registros armazenados;
   - validação rápida de dados ingeridos pelos pipelines.

   A utilização do pgweb foi particularmente importante para facilitar o processo de validação e auditoria dos dados gerados pelo pipeline ETL, principalmente pelo fato de o banco PostgreSQL não estar diretamente exposto à internet.

   Assim como o Airflow, o serviço também foi publicado por meio do proxy reverso NGINX, podendo ser acessado através do endereço:

   <http://161.24.29.22/pgweb/>

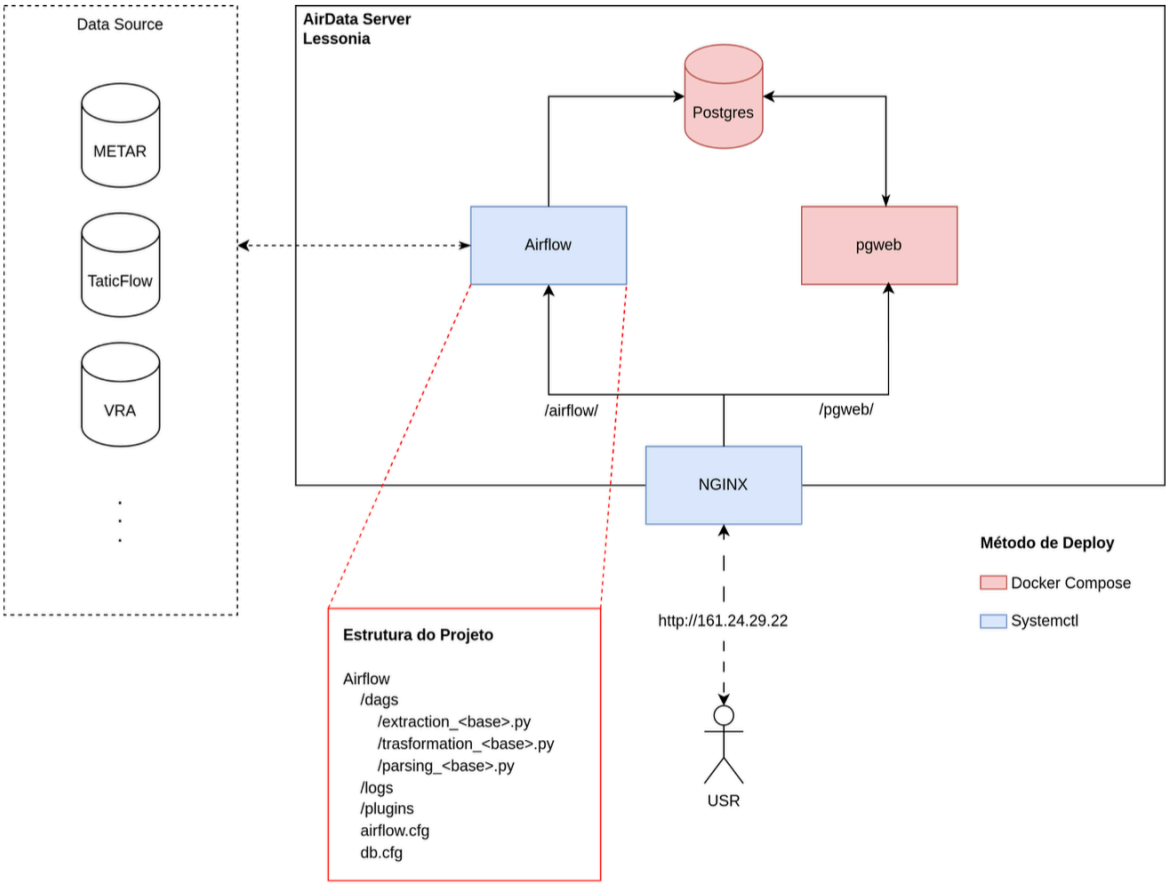

Figura 1.1: Arquitetura computacional do Produto II

1.3 Versionamento

O controle de versionamento do pipeline ETL foi estruturado com o objetivo de garantir organização no desenvolvimento colaborativo, rastreabilidade das modificações e estabilidade do ambiente de produção. Para isso, adotou-se um fluxo de trabalho baseado em *branches*, permitindo separar o desenvolvimento de novas funcionalidades do código utilizado em produção.

A estratégia de versionamento foi implementada utilizando o sistema de controle de versões *Git*, com hospedagem do repositório na plataforma GitHub. Foram definidas duas branches principais para gerenciamento do ciclo de desenvolvimento: **dev** e **main**.

1.3.1 Fluxo de Desenvolvimento

O fluxo de desenvolvimento adotado segue uma abordagem simplificada inspirada no modelo *Git Flow*, onde as atualizações passam por uma etapa de desenvolvimento e validação antes de serem disponibilizadas em produção.

1. **Branch dev**

   A branch `dev` é utilizada como ambiente principal de desenvolvimento. Nessa branch são realizados:

   - desenvolvimento de novas DAGs do Apache Airflow;
   - ajustes em pipelines ETL;
   - correções de bugs;
   - testes de integração com os serviços de banco de dados e infraestrutura.

   Todos os commits relacionados a experimentação, testes e implementação de novas funcionalidades são realizados inicialmente nessa branch.

2. **Homologação**

   Após a conclusão das modificações na branch `dev`, o código passa por um processo de validação técnica (homologação). Essa etapa tem como objetivo verificar:

   - o funcionamento correto das DAGs;
   - a integridade dos dados processados;

- a estabilidade do pipeline de execução.

O processo de homologação é supervisionado pelo pesquisador Jean Lima, responsável por validar se as alterações estão adequadas para serem promovidas ao ambiente de produção.

3. **Deploy**

   Uma vez aprovadas na etapa de homologação, as alterações são integradas à branch `main`, que representa a versão estável do sistema em produção.

   O processo de deploy consiste nas seguintes etapas:

   (a) realização do *merge* da branch `dev` na branch `main`;

   (b) atualização do repositório local no servidor de produção através do comando *git pull*;

   (c) recarregamento dos serviços do Apache Airflow para reconhecer as novas DAGs ou alterações realizadas.

   Esse procedimento é executado diretamente no servidor operacional do projeto, denominado *Lessonia*, garantindo que as atualizações aprovadas sejam refletidas na infraestrutura ativa do sistema.

O repositório remoto do projeto encontra-se disponível em:

<https://github.com/airdatagit-ops>

A Figura 1.2 apresenta uma representação visual do fluxo de desenvolvimento e deploy adotado no projeto.

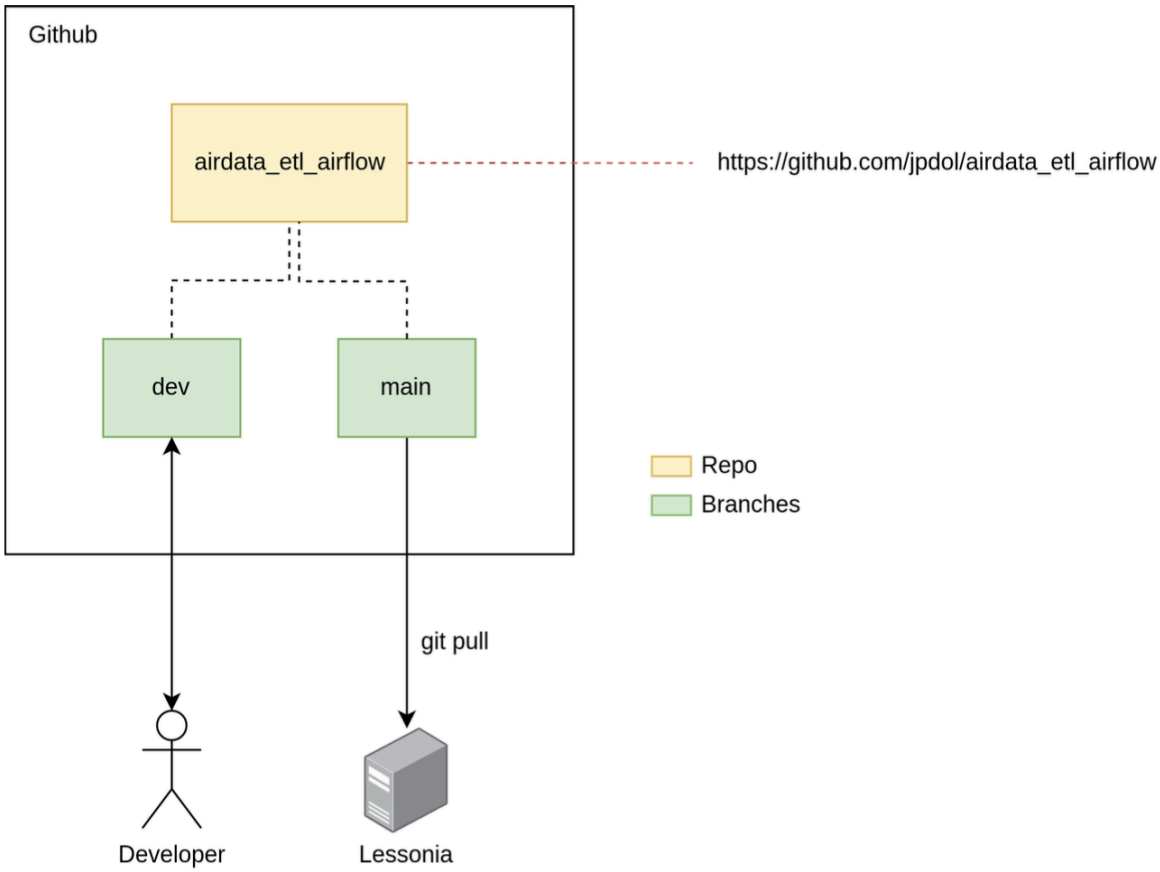

Figura 1.2: Processo de desenvolvimento e deploy na máquina do Lessonia.

1.4 PostgreSQL

O PostgreSQL é um *Sistema de Gerenciamento de Banco de Dados Relacional* (SGBD) open-source, robusto e amplamente consolidado na indústria, reconhecido por sua aderência a padrões, extensibilidade e alta confiabilidade.

Ele oferece suporte nativo a transações *Atomicity, Consistency, Isolation, Durability* (ACID), forte integridade referencial, capacidades avançadas de indexação e otimização, além de extensões poderosas, como PostGIS, TimescaleDB e funções procedurais em diversas linguagens. No contexto do **AirData**, o PostgreSQL foi escolhido por combinar desempenho, estabilidade operacional e flexibilidade de modelagem, essenciais para integrar dados heterogêneos do *Sistema de Controle do Espaço Aéreo Brasileiro* (SISCEAB). Sua compatibilidade com ferramentas modernas de ETL (como Apache Airflow), facilidade de escalabilidade mediante particionamento e replicação, e o ecossistema maduro de administração tornam-no a escolha ideal para sustentar um data warehouse relacional estruturado, garantindo consistência, auditabilidade e evolução futura da plataforma.

1.4.1 Implementação via Docker Compose

O serviço de banco de dados PostgreSQL foi implantado no servidor do projeto, denominado *Lessonia*, utilizando a ferramenta *Docker Compose*. Essa abordagem permite definir e gerenciar serviços containerizados de forma declarativa, facilitando a reprodução do ambiente em diferentes infraestruturas e simplificando o processo de manutenção.

A configuração do serviço foi definida por meio de um arquivo `docker-compose.yml`, no qual são especificados os parâmetros necessários para a execução do container. Entre os principais elementos configurados destacam-se:

- utilização da imagem oficial do PostgreSQL, garantindo estabilidade e compatibilidade com versões amplamente utilizadas na comunidade;
- definição das variáveis de ambiente responsáveis pela criação automática do usuário administrador, senha de acesso e banco de dados inicial;
- inicialização automática do *schema* `airdata`, destinado ao armazenamento dos dados gerados pelos pipelines do processo de ETL;
- exposição da porta padrão do PostgreSQL (5432) apenas para comunicação interna entre os serviços da infraestrutura;
- implementação de mecanismos de *healthcheck*, responsáveis por verificar periodicamente a disponibilidade do banco de dados;
- configuração de política de reinicialização automática do container (*restart policy*), garantindo maior resiliência em caso de falhas do serviço;
- conexão do container a uma rede Docker dedicada ao projeto, permitindo comunicação segura e isolada entre os serviços implantados.

Essa abordagem baseada em containers contribui para maior portabilidade do ambiente, possibilitando que toda a infraestrutura necessária ao funcionamento do pipeline de dados seja replicada de forma consistente em diferentes máquinas ou ambientes de execução.

1.5 Apache Airflow

O *Apache Airflow* é uma plataforma open-source amplamente utilizada para a orquestração de workflows e pipelines de dados. Sua principal função é permitir a definição,

execução e monitoramento de processos complexos de processamento de dados de forma programática, escalável e reproduzível.

A ferramenta foi originalmente desenvolvida pela empresa Airbnb e posteriormente incorporada à Apache Software Foundation, tornando-se um dos principais padrões da indústria para gerenciamento de pipelines de dados em ambientes de engenharia de dados e plataformas de análise.

1.5.1 Características Principais

Projetado para lidar com fluxos de dados complexos e integração entre múltiplas fontes heterogêneas, o Apache Airflow oferece um conjunto de funcionalidades que o tornam particularmente adequado para aplicações de engenharia de dados e automação de pipelines ETL. Entre suas principais características destacam-se:

- definição de workflows através de *Directed Acyclic Graphs* (DAGs), permitindo representar dependências entre tarefas de forma explícita;
- agendamento automático de execuções periódicas, possibilitando a execução de pipelines em intervalos definidos ou em horários específicos;
- monitoramento detalhado do estado de execução das tarefas, incluindo histórico de execuções e logs completos;
- capacidade de reprocessamento de tarefas em caso de falhas, aumentando a robustez dos pipelines;
- integração nativa com diversos sistemas de armazenamento, bancos de dados e serviços externos;
- escalabilidade horizontal através da utilização de múltiplos *workers*, permitindo a execução paralela de tarefas.

Essas características tornam o Apache Airflow particularmente adequado para ambientes de processamento de dados científicos e aplicações que demandam alto nível de rastreabilidade, controle operacional e confiabilidade na execução de pipelines de dados.

- Controle preciso sobre dependências entre tarefas
- Paralelização de execução

- Versionamento e reexecução
- Rastreabilidade completa de cada etapa do fluxo de dados

No contexto do projeto **AirData**, o Airflow foi adotado por sua capacidade de estruturar uma arquitetura de ingestão e processamento modular, transparente e extensível. Essa abordagem facilita a incorporação contínua de novas bases do SISCEAB, ao mesmo tempo em que possibilita o monitoramento operacional centralizado e a padronização dos processos de carga e transformação de dados. Adicionalmente, a integração nativa com o PostgreSQL e com ambientes distribuídos, aliada ao amplo ecossistema de operadores, hooks e sensores disponíveis, viabiliza uma automação robusta, auditável e preparada para expansão, atendendo aos requisitos de um sistema nacional de dados aeronáuticos.

1.5.2 Conceito de DAG

O funcionamento do Airflow baseia-se na execução de DAGs, que representam abstrações de pipelines de dados, nas quais:

- Cada nó do grafo corresponde a uma **task**
- As arestas definem as dependências de execução entre essas tarefas

Os DAG são definidos por meio de scripts escritos na linguagem de programação **Python**, permitindo elevada flexibilidade na modelagem dos fluxos. Esses scripts devem ser armazenados em um diretório específico do ambiente Airflow, denominado `/dags`, localizado na raiz do projeto no servidor Lessonia.

1.5.3 Exemplo: DAG VRA

A Figura 1.3 apresenta um exemplo de DAG desenvolvido para a coleta de dados da base VRA da Agência Nacional de Aviação Civil (ANAC).

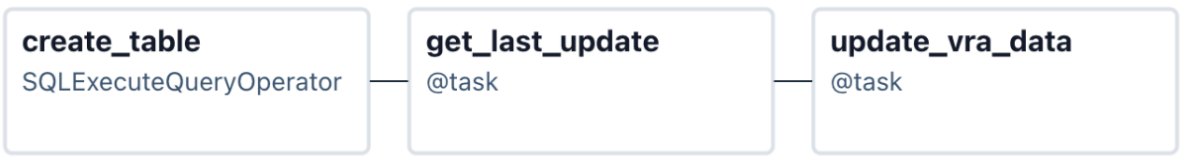

Figura 1.3: DAG do processo de coleta de dados do VRA.

1.5.4 Detalhamento da DAG VRA

A DAG `vra_extraction` foi projetada para automatizar o processo de extração incremental, tratamento e armazenamento dos dados da base Voo Regular Ativo (VRA) disponibilizada pela ANAC. Sua definição segue o paradigma declarativo do Apache Airflow, no qual o pipeline é modelado como um grafo acíclico direcionado, permitindo controle explícito de dependências, rastreabilidade e reexecução controlada.

A DAG é registrada por meio do decorador `@dag`, com identificador `vra_extraction`, agendamento diário às 06:00 (cron `0 6 * * *`) e restrição de execução concorrente (`max_active_runs=1`). Essa configuração garante que apenas uma instância do fluxo seja executada por vez, evitando sobreposição de cargas e possíveis duplicidades de dados durante o processo de ingestão.

1.5.4.1 Task: create_table

O fluxo inicia-se com a task `create_table`, implementada por meio do operador `SQLExecuteQueryOperator`. Essa etapa é responsável por assegurar a existência da tabela `airdata.vra` no banco de dados PostgreSQL, utilizando o comando `CREATE TABLE IF NOT EXISTS`. Ao incorporar essa verificação diretamente na DAG, o pipeline torna-se mais robusto e autossuficiente, reduzindo dependências de inicializações manuais e facilitando a implantação em novos ambientes.

1.5.4.2 Task: get_last_update

Em seguida, a task `get_last_update`, definida com o decorador `@task`, realiza a consulta à base de dados para identificar a última data de referência (`dt_referencia`) já armazenada. Essa etapa é fundamental para viabilizar a extração incremental dos dados. Caso a tabela ainda não possua registros, o código define explicitamente uma data inicial padrão, permitindo a primeira carga histórica controlada. O valor retornado por essa task é propagado automaticamente pelo Airflow para as etapas subsequentes, por meio do mecanismo de **XComs**.

1.5.4.3 Task: update_vra_data

A task `update_vra_data` constitui o núcleo do processo de extração e carga. A partir da última data registrada, o código define dinamicamente o intervalo de datas a ser consultado, iniciando no dia subsequente à última atualização e avançando até a data corrente. Para cada dia do intervalo, é realizada uma requisição HTTP à API pública

da ANAC, utilizando o endpoint oficial do VRA.

Os dados retornados são tratados em formato tabular por meio da biblioteca **pandas**. O código executa explicitamente o parsing das colunas temporais, convertendo strings em objetos `datetime` e `date`, assegurando consistência tipológica antes da inserção no banco de dados. Funções auxiliares dedicadas ao tratamento de datas e horários são utilizadas para centralizar essa lógica e reduzir a probabilidade de erros de conversão.

1.5.4.4 Carregamento de Dados

Após o pré-processamento, os dados são inseridos na tabela `airdata.vra` utilizando o método `to_sql` do pandas, com estratégia de inserção em lote (`chunksize=5000`) e operação em modo `append`. Essa abordagem equilibra desempenho e segurança, permitindo a ingestão eficiente de grandes volumes de registros sem sobrescrever informações previamente armazenadas.

1.5.4.5 Ordenação de Execução

A ordem de execução das tasks é explicitamente definida no final da DAG por meio do operador >>, estabelecendo a seguinte sequência lógica:

1. **Criação da tabela**

   A primeira etapa consiste em garantir a existência da tabela de destino no banco de dados PostgreSQL. Caso a tabela ainda não esteja presente no schema `airdata`, ela é criada com a estrutura necessária para armazenar os dados processados pelo pipeline.

2. **Identificação da última atualização**

   Na segunda etapa, o sistema consulta o banco de dados para identificar o timestamp ou marcador correspondente ao último registro inserido na tabela. Essa informação é utilizada como referência para determinar a partir de qual ponto novos dados devem ser coletados ou processados.

3. **Atualização incremental dos dados**

   Por fim, o pipeline executa a atualização incremental, inserindo apenas os registros mais recentes que ainda não estão presentes no banco de dados. Essa estratégia reduz significativamente o volume de dados processados a cada exe-

cução da DAG, aumentando a eficiência do pipeline e evitando duplicidade de registros.

Essa estrutura garante que cada etapa só seja executada após a conclusão bem-sucedida da etapa anterior, reforçando a consistência e a confiabilidade do pipeline.

Por fim, a chamada da função `vra_extraction()` efetiva o registro da DAG no ambiente Airflow, tornando o fluxo disponível para agendamento, monitoramento e auditoria por meio da interface web da plataforma.

1.6 NGINX - Reverse Proxy e Servidor Web

O NGINX foi adotado no projeto **AirData** como servidor web e reverse proxy, atuando como ponto único de entrada para os serviços internos da plataforma. Essa abordagem permite expor aplicações distintas, como o Apache Airflow e o pgweb, de forma organizada, segura e transparente, sem a necessidade de acesso direto às portas internas onde esses serviços estão efetivamente em execução.

No contexto geral do **AirData**, o uso do NGINX como reverse proxy proporciona benefícios importantes, como centralização do acesso aos serviços, isolamento das portas internas, maior controle sobre redirecionamentos e cabeçalhos HTTP, além de preparar a infraestrutura para futuras extensões, como autenticação, controle de acesso, registro de logs unificados e habilitação de HTTPS. Essa camada de abstração contribui diretamente para a robustez, segurança e escalabilidade do ecossistema de dados aeronáuticos. Os aplicativos `owl.airdata.ita.br`, `chatbot.airdata.ita.br` e `data.airdata.ita.br` também são acessíveis através do redirecionamento de portas do NGINX.

1.6.1 Configuração

A configuração do servidor NGINX foi estruturada a partir de um bloco `server` responsável por receber e encaminhar as requisições HTTP destinadas aos serviços da infraestrutura. Esse bloco está configurado para escutar na porta 80 utilizando o parâmetro `default_server`, o que indica que ele atuará como o servidor padrão para todas as requisições recebidas que não correspondam explicitamente a outro *virtual host* definido na mesma instância do NGINX.

Adicionalmente, foi definido o parâmetro `server_name _`, que indica que o servidor

responderá de forma genérica a qualquer nome de host recebido na requisição. Essa configuração é particularmente adequada para ambientes internos, laboratoriais ou de pesquisa, nos quais o acesso aos serviços ocorre diretamente por meio do endereço IP do servidor ou por um único domínio institucional.

Nesse contexto, o NGINX atua como um *proxy reverso*, centralizando o acesso externo aos serviços internos da infraestrutura, como o Apache Airflow e o pgweb. Dessa forma, todas as requisições provenientes de clientes web são inicialmente recebidas pelo NGINX, que então as encaminha para os respectivos serviços executados em portas internas do servidor.

1.6.1.1 Redirecionamentos

Para garantir consistência na navegação e evitar problemas relacionados ao tratamento de caminhos relativos nas aplicações web, foram definidos redirecionamentos explícitos para determinados contextos de acesso.

Especificamente, quando um usuário acessa os caminhos `/airflow` ou `/pgweb` sem a presença da barra final, o servidor NGINX retorna um redirecionamento HTTP temporário (código 302) direcionando a requisição para os caminhos `/airflow/` e `/pgweb/`, respectivamente.

Essa abordagem é importante porque diversas aplicações web assumem que a raiz do contexto de execução inclui a barra final. Na ausência desse elemento, podem ocorrer problemas na resolução de *Uniform Resource Locator* (URL) relativas, carregamento de recursos estáticos ou funcionamento de rotas internas da aplicação.

Assim, a configuração dos redirecionamentos garante maior compatibilidade entre o servidor proxy e as aplicações hospedadas, contribuindo para uma experiência de navegação mais estável e previsível.

1.6.1.2 Proxy para Airflow

O bloco `location /airflow/` configura o NGINX como reverse proxy para a interface web do Apache Airflow, que está em execução localmente na porta 8080. A diretiva `proxy_pass http://127.0.0.1:8080;` encaminha as requisições recebidas pelo NGINX para o serviço interno correspondente, abstraindo do usuário final detalhes da topologia interna do servidor.

As diretivas adicionais de proxy asseguram a correta propagação de informações de

contexto da requisição original:

- `proxy_set_header Host $host`: Preserva o domínio acessado pelo cliente
- `X-Real-IP` e `X-Forwarded-For`: Mantêm o endereço IP de origem, fundamental para auditoria, logging e controle de acesso
- `X-Forwarded-Proto`: Informa ao serviço de destino o protocolo utilizado na requisição original, permitindo comportamento adequado em cenários futuros de HTTPS
- `proxy_redirect off`: Impede a reescrita automática de URL retornadas pelo serviço proxied, garantindo consistência nos endereços apresentados ao usuário

1.6.1.3 Proxy para pgweb

De forma análoga, o bloco `location /pgweb/` estabelece o encaminhamento das requisições para o serviço pgWeb, responsável pela interface web de administração do banco de dados PostgreSQL, executado localmente na porta 8081. A simetria entre as configurações de proxy do Airflow e do pgweb reforça a padronização da arquitetura e simplifica a manutenção e expansão do ambiente.

2 AirData Ontology

2.1 Visão Geral

O **AirData Ontology** é um subsistema especializado em representação semântica do domínio aeronáutico, constituindo um dos três pilares do **AirData** junto ao **Data Check** e **RAG System**. A ontologia opera como camada de conhecimento que estrutura conceitos, relacionamentos e regras de negócio do domínio *Air Traffic Management* (ATM) de forma compreensível tanto para humanos quanto para máquinas.

2.1.1 Objetivos Estratégicos

O desenvolvimento da ontologia aeronáutica visa:

- **Representação Semântica**: Modelagem formal de conceitos aeronáuticos (aeronaves, voos, aeródromos, rotas) e seus relacionamentos
- **Interoperabilidade**: Garantir compatibilidade com padrões de conhecimento aberto (RDF, OWL, SPARQL)
- **Raciocínio Automático**: Habilitar inferências lógicas sobre dados aeronáuticos através de reasoners
- **Validação Semântica**: Verificar conformidade de dados com regras de negócio do domínio
- **Integração com IA**: Fornecer base de conhecimento estruturado para LLMs e sistemas RAG

2.2 Workflow de Validação e Qualidade Ontológica

Foi desenvolvido um workflow automatizado para garantir que a ontologia OWL seja validada, analisada e documentada de forma consistente e reproduzível. O processo segue três fases principais:

1. **Validação**: Verificação de integridade e qualidade (sintaxe, lógica, SPARQL, legibilidade)
2. **Documentação**: Geração de documentação navegável e visualizações gráficas
3. **Publicação**: Disponibilização via web para acesso e revisão colaborativa

A Figura 2.1 ilustra os componentes desse sistema.

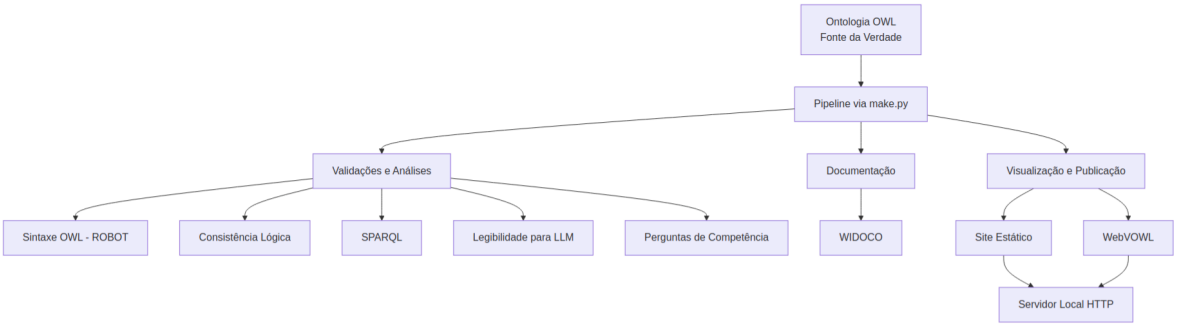

Figura 2.1: Workflow do sistema de ontologia do **AirData**

O pipeline utiliza os arquivos OWL como única fonte de verdade, evitando divergências entre modelo e documentação.

2.2.1 Tipos de Validação Implementados

O workflow executa seis categorias distintas de validação, cada uma avaliando aspectos complementares da qualidade ontológica:

2.2.1.1 Validação Sintática

Esta etapa verifica se a ontologia está gramaticalmente correta, análogo a um verificador ortográfico. Um dicionário com erros de digitação seria inútil para consultas. Aqui, utiliza-se uma ferramenta especializada (ROBOT) que garante que todos os elementos ontológicos (classes, propriedades, definições) obedecem às regras sintáticas do padrão OWL.

Importância: Erros de sintaxe ou estrutura podem invalidar resultados posteriores, gerar documentação incorreta ou quebrar visualizações. O ROBOT automatiza essa

checagem, reduzindo falhas humanas e oferecendo um filtro rápido e confiável para manter a base ontológica saudável.

Saída: Relatório com indicador binário (verde se válido, vermelho se detectados erros).

2.2.1.2 Análise Semântica

Enquanto a validação sintática questiona "está escrito corretamente?", a análise semântica questiona "faz sentido logicamente?". Por exemplo: uma classe chamada `Avião` não deveria estar definida como subclasse de `Comida`.

Esta etapa detecta inconsistências lógicas e estruturais que mesmo um arquivo gramaticalmente correto poderia apresentar, incluindo:

- Classes órfãs (sem conexão hierárquica)
- Relações circulares inválidas
- Domínios e ranges incompatíveis
- Violações de restrições de cardinalidade

Ferramentas: A análise semântica é suportada principalmente pela biblioteca `rdflib` (v7.5.0), padrão Python para processamento de grafos RDF/OWL que permite carregar, parsear e navegar estruturas ontológicas em formato *Extensible Markup Language* (XML). Complementando essa camada, `beautifulsoup4` (v4.14.3) processa metadados em HTML/XML, enquanto `lxml` oferece parsing XML otimizado. Para análises avançadas de conectividade e estrutura do grafo, utiliza-se `networkx`, que detecta componentes desconexos, mede profundidade de hierarquias e calcula métricas de centralidade.

2.2.1.3 Axiomas Lógicos OWL

Esta é a base da ontologia. Aqui verificamos se foram definidas regras formais que permitem ao sistema raciocinar automaticamente. Por exemplo: "todo Voo tem exatamente uma aeronave". Se essas regras não são definidas, o sistema não consegue validar dados ou realizar inferências inteligentes.

Objetivo. Verificar a riqueza e correção dos axiomas lógicos OWL, garantindo que a ontologia não seja apenas uma "hierarquia de classes", mas um modelo semântico rico capaz de suportar:

- Inferências automáticas (*reasoners*);
- Validação de instâncias;
- Interoperabilidade semântica (alinhamentos ontológicos);
- Consultas SPARQL complexas.

Tipos de Axiomas Verificados

Restrições de Cardinalidade Definem quantas relações uma entidade pode possuir.

Exemplo. Um voo tem exatamente uma aeronave (`owl:cardinality 1`).

Justificativa. Previnem dados inconsistentes e permitem validação automática.

Classes Disjuntas (`owl:disjointWith`) Declaram que duas classes não podem compartilhar indivíduos.

Exemplo. `Aeronave` é disjunta de `Aeroporto`.

Justificativa. Evitam classificações incorretas e permitem que *reasoners* detectem inconsistências automaticamente.

Características de Propriedades

- **Funcional**: máximo um valor (ex.: `temMatricula`);
- **Inversa funcional**: identifica unicamente (ex.: `temICAO`);
- **Simétrica**: $A \to B$ implica $B \to A$ (ex.: `conectadoCom`);
- **Transitiva**: $A \to B$ e $B \to C$ implica $A \to C$ (ex.: `parteDe`);
- **Reflexiva / Irreflexiva**.

Justificativa. Definem o comportamento esperado das relações, garantindo inferências corretas.

Restrições Complexas

- `owl:allValuesFrom` (∀): todos os valores devem satisfazer a restrição;
- `owl:someValuesFrom` (∃): existe pelo menos um valor que satisfaz a restrição;
- `owl:hasValue`: possui um valor específico.

Exemplo. Todo voo comercial deve possuir pelo menos um piloto certificado.

Justificativa. Permitem modelar regras de domínio mais sofisticadas.

Propriedades Inversas (`owl:inverseOf`) `temOrigem` é inversa de `origemDe`, permitindo navegação bidirecional no grafo.

Justificativa. Enriquecem consultas SPARQL e o raciocínio automático.

Score de Riqueza Lógica

Métrica de 0–100 que avalia a cobertura de axiomas lógicos na ontologia:

- Cobertura de cardinalidades: 30%;
- Cobertura de disjunções: 25%;
- Características de propriedades: 20%;
- Restrições complexas: 15%;
- Propriedades inversas: 10%.

Sem Axiomas Lógicos

- Ontologia reduzida a uma estrutura hierárquica semelhante a banco de dados;
- *Reasoners* não realizam inferências;
- Não há validação automática de dados;
- Dados inconsistentes não são detectados.

Com Axiomas Lógicos

- Detecção automática de inconsistências;
- Suporte a inferências formais;
- Ontologia torna-se autodocumentada;
- Maior interoperabilidade semântica.

2.2.1.4 Usabilidade para LLMs

LLMs (como ChatGPT) precisam entender a ontologia para utilizá-la em consultas semânticas e sistemas RAG. A formatação utilizada é usada para verificar se a ontologia está bem documentada e legível para máquinas:

- Possui `rdfs:label` em português para todas as classes
- Contém `rdfs:comment` com descrições claras
- Nomes de *Uniform Resource Identifier* (URI) são semanticamente significativos
- Anotações seguem padrões de metadados (Dublin Core, SKOS)

Se um manual técnico estiver confuso, ninguém (nem humanos nem máquinas) consegue usar efetivamente.

2.2.1.5 Validação SPARQL

SPARQL é uma "linguagem de consulta" para ontologias (similar a SQL para bancos de dados relacionais). Aqui é testado se as perguntas principais que alguém faria à ontologia funcionam corretamente.

Exemplo de query testada:

```
PREFIX : <http://airdata.org/ontology#>
SELECT ?voo ?aeronave
WHERE {
  ?voo a :Voo .
  ?voo :operadoCom ?aeronave .
}
```

Esta etapa verifica se a ontologia permite executar essas consultas sem erros e se os resultados fazem sentido dentro do domínio.

2.2.1.6 Perguntas de Competência

Perguntas de competência são questões estratégicas do domínio que a ontologia DEVE ser capaz de responder. Elas validam se a ontologia foi bem modelada para seus casos de uso reais.

Princípio: "A ontologia é boa se consegue responder as perguntas que o domínio faz."

*Exemplos de perguntas de competência para **AirData**:*

- Qual aeronave opera um voo específico?
- Que operações ocorrem em um aeródromo?
- Quais são as rotas aéreas entre duas cidades?
- Qual é a altitude de cruzeiro de uma classe de aeronave?
- Quantos voos partiram de SBGR nas últimas 24 horas?
- Quais waypoints compõem a rota SBSP-SBBR?

Se a ontologia consegue responder todas essas perguntas através de consultas SPARQL ou inferências, significa que foi bem modelada para seu propósito.

2.3 Documentação e Visualização

O macroprocesso de Visualização e Publicação existe para tornar a ontologia compreensível e acessível fora do formato OWL bruto, o que é imprescindível para estudo, validação colaborativa e manutenção contínua.

2.3.1 Ferramentas de Documentação

2.3.1.1 WIzard for DOCumenting Ontologies (WIDOCO)

Utiliza-se o WIDOCO no workflow porque ele transforma a ontologia OWL (fonte da verdade) em documentação HTML padronizada e navegável, facilitando entendimento, validação e comunicação com diferentes públicos.

Benefícios da integração WIDOCO:

- Documentação sempre atualizada e consistente com versões da ontologia
- Reduz trabalho manual de documentação
- Evita divergências entre modelo OWL e documentação publicada
- Melhora transparência e acelera revisões internas/externas
- Fortalece reutilização da ontologia ao oferecer visão clara de classes, propriedades e axiomas

A Figura 2.2 apresenta a página inicial da documentação da ontologia do **AirData** disponível no site `owl.airdata.ita.br`

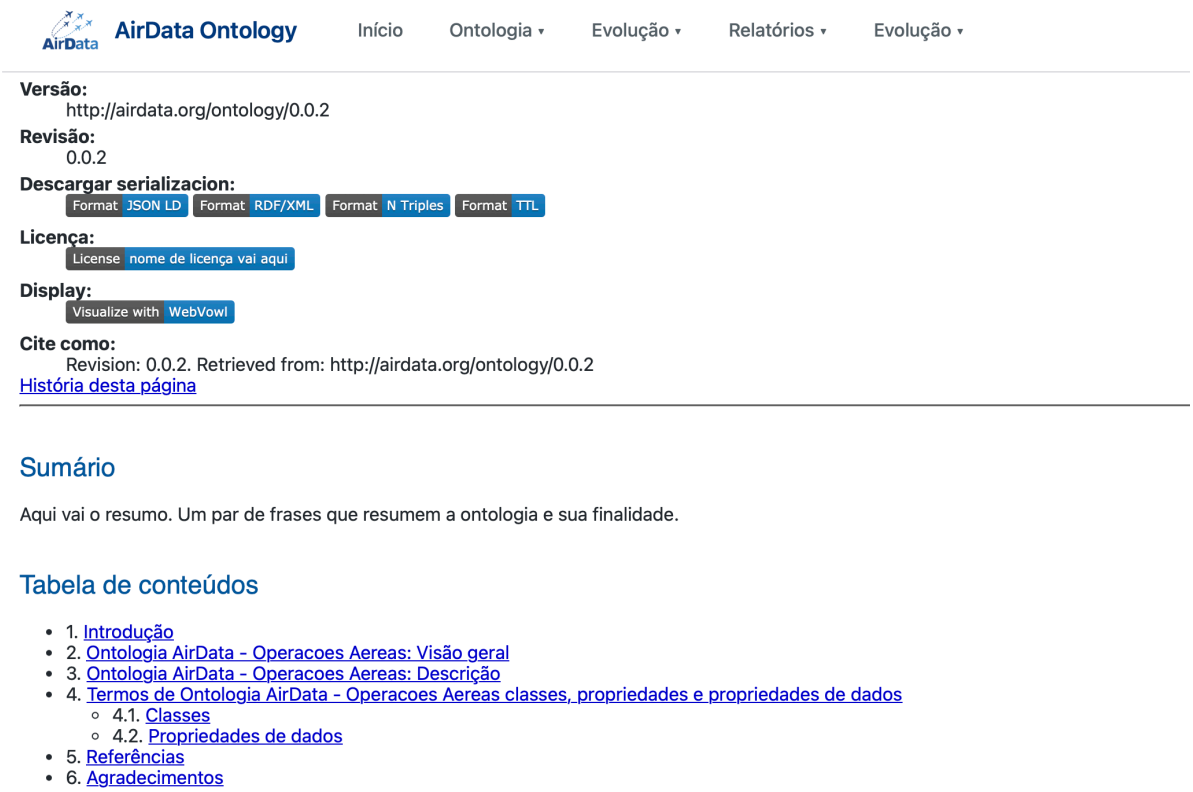

Figura 2.2: Documentação da ontologia com WIDOCO

A documentação HTML gerada pelo WIDOCO inclui:

- Metadados da ontologia (autores, versão, licença)
- Diagrama de visão geral
- Índice navegável de classes e propriedades
- Definições formais e descrições textuais
- Exemplos de uso
- Referências cruzadas entre elementos

2.3.1.2 WebVOWL - Visual Notation for OWL Ontologies

WebVOWL é utilizado para visualização gráfica interativa das classes e relações da ontologia. O grafo resultante facilita detectar lacunas e inconsistências estruturais que não são óbvias em formato textual. A Figura 2.3 exibe o grafo da primeira versão da ontologia.

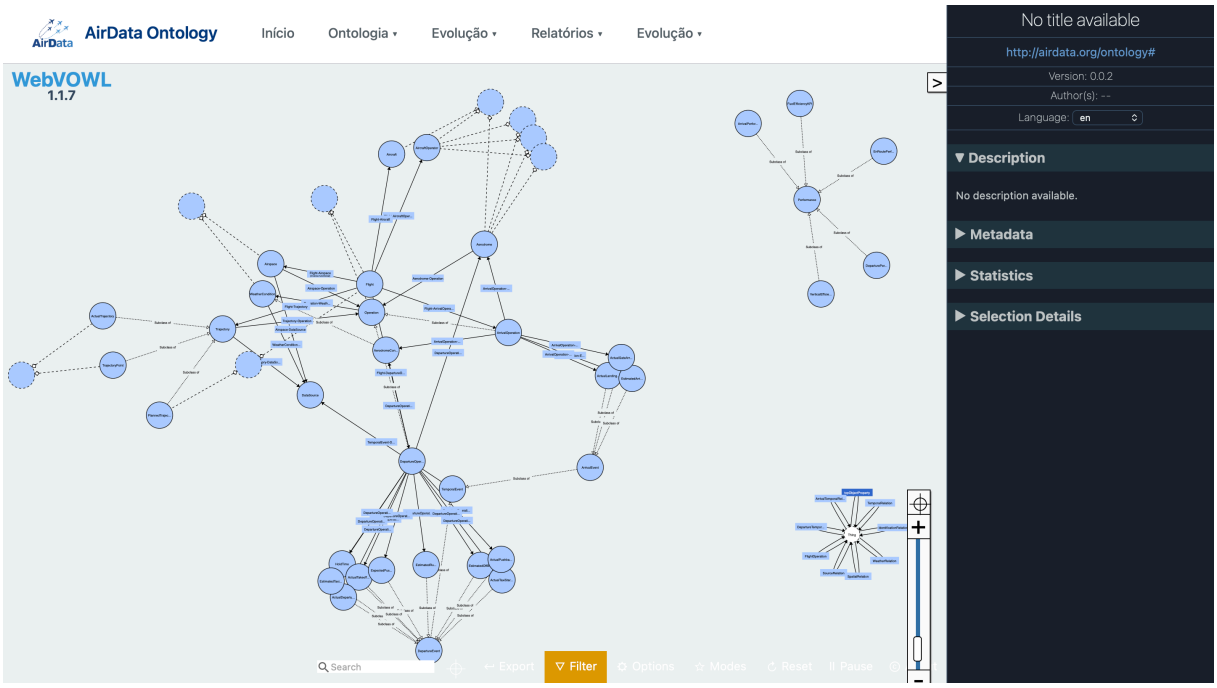

Figura 2.3: Grafo da primeira versão da ontologia **AirData**

Funcionalidades do grafo interativo:

- Visualização com caixas (conceitos) e linhas (relações)

- Navegação interativa: clicar, arrastar, aproximar e afastar
- Filtros por tipo de entidade (classes, propriedades de objeto, propriedades de dados)
- Identificação visual de hierarquias (subclasses, super classes)
- Detecção de padrões estruturais (hubs, componentes desconexos)

É útil para "enxergar" a estrutura geral e qual conceito conecta a qual, quem depende de quem, permitindo compreensão holística da arquitetura ontológica.

2.3.2 Publicação Web

Toda a documentação e visualizações são publicadas em um site estático servido via HTTP para garantir acesso estável e evitar problemas de *Cross-Origin Resource Sharing* (CORS). Esta abordagem:

- Reduz custo de entendimento da ontologia
- Acelera processos de revisão por stakeholders
- Facilita onboarding de novos desenvolvedores
- Mantém a ontologia alinhada com necessidades do projeto ao longo do tempo

2.4 Relatórios de Qualidade Ontológica

O sistema gera múltiplos relatórios automatizados que oferecem visões complementares sobre a saúde da ontologia.

2.4.1 Relatório de Validação OWL (ROBOT)

O ROBOT executa uma bateria de "checks" padronizados (report queries) e gera um relatório HTML tabulado. A tabela apresenta, para cada regra:

- **Nível**: Erro (Error) ou Aviso (Warning)
- **Tipo de Problema**: Categoria da violação

- **Subject**: Recurso afetado (URI da classe/propriedade)
- **Property**: Propriedade relacionada ao problema
- **Value**: Valor encontrado (ou ausência dele)

Propósito: Avaliar rapidamente a qualidade da OWL em termos de:

- Completude de anotações (ex: ausência de `rdfs:label`)
- Consistência de metadados
- Boas práticas de modelagem
- Aderência a padrões de documentação

Aponta lacunas objetivas que impactam legibilidade, interoperabilidade e manutenção da ontologia.

2.4.2 Guia de Interpretação

Documento que explica como ler os relatórios de validação. Funciona como manual de instruções, ajudando usuários a entender o significado de cada número, gráfico e indicador, além de orientar ações corretivas quando indicadores aparecem em vermelho.

2.4.3 Relatório de Estatísticas

Apresenta números-chave sobre a ontologia, funcionando como "boletim de saúde":

- Número total de classes
- Número de propriedades de objeto (`owl:ObjectProperty`)
- Número de propriedades de dados (`owl:DatatypeProperty`)
- Profundidade máxima da hierarquia de classes
- Número de axiomas lógicos
- Grau médio de conectividade do grafo

- Taxa de documentação (porcentagem de classes com `rdfs:label` e `rdfs:comment`)

Oferece visão rápida do tamanho, complexidade e maturidade da ontologia.

2.4.4 *Relatório Delta de Qualidade*

Relatório comparativo que analisa duas versões da ontologia lado a lado, apresentando:

- Evolução quantitativa: "na versão anterior tínhamos 20 classes, agora temos 25"
- Mudanças qualitativas: "antes havia 3 axiomas lógicos, agora tem 8"
- Métricas de progressão: score de riqueza lógica, cobertura de documentação
- Classes/propriedades adicionadas, removidas ou modificadas

Propósito: Entender se a qualidade melhorou, piorou ou se manteve estável entre versões, auxiliando decisões de versionamento e validação de melhorias implementadas.

2.5 Integração com Sistemas de IA

2.5.1 *Chatbot **AirData** com RAG*

O Chatbot **AirData** utiliza a ontologia como base de conhecimento estruturado para consultas semânticas através da arquitetura RAG. A ontologia fornece:

- **Vocabulário Controlado**: Termos padronizados para interpretação de consultas em linguagem natural
- **Contexto Semântico**: Relacionamentos entre conceitos para enriquecer respostas
- **Validação de Inferências**: Regras lógicas contra as quais LLMs podem verificar geração de texto
- **Mapeamento para Dados**: Links entre conceitos ontológicos e tabelas do Data Warehouse

2.5.2 Stack Tecnológico de IA

- **LangChain**: Framework para construção de aplicações LLM com acesso a bases de conhecimento
- **HuggingFace Transformers**: Modelos de linguagem pré-treinados para processamento de consultas

2.6 Mapeamento de Conceitos ICAO

A ontologia **AirData** está sendo desenvolvida com aderência aos padrões e terminologias da ICAO, garantindo:

- Alinhamento com *Aeronautical Information Exchange Model* (AIXM)
- Compatibilidade com *Flight Information Exchange Model* (FIXM)
- Conformidade com *Weather Information Exchange Model* (WXXM)
- Interoperabilidade com sistemas ATM globais

O mapeamento ICAO permite que a ontologia **AirData** seja utilizada em contextos internacionais e integrada com sistemas de outras autoridades aeronáuticas.

2.7 Casos de Uso da Ontologia

2.7.1 Validação Semântica de Dados

A ontologia é utilizada para verificar se dados aeronáuticos coletados obedecem às regras de negócio do domínio:

- Um voo não pode ter origem e destino idênticos
- Aeronaves devem ter matrícula única
- Waypoints devem ter coordenadas geográficas válidas
- Altitude de cruzeiro deve estar dentro de níveis de voo permitidos

2.7.2 Enriquecimento de Consultas

Consultas SQL simples podem ser enriquecidas semanticamente:

```
-- Consulta SQL simples
SELECT * FROM voos WHERE origem = 'SBGR'

-- Consulta semântica enriquecida via ontologia
-- "Quais voos partiram de aeródromos da região metropolitana de São Paulo?"
-- A ontologia sabe que SBGR, SBKP, SBSP e SBMT fazem parte desta região
```

2.7.3 Detecção de Anomalias

Reasoners podem identificar automaticamente dados inconsistentes:

- Voo registrado em aeronave que estava em manutenção
- Altitude reportada incompatível com categoria da aeronave
- Tempo de voo fisicamente impossível para distância percorrida

2.8 Próximos Passos

A ontologia **AirData** encontra-se em desenvolvimento ativo, com as seguintes etapas planejadas:

1. **Expansão de Classes**: Modelagem completa de todos os conceitos das 18 bases de dados
2. **Definição de Axiomas**: Aumento do score de riqueza lógica para > 80%
3. **Mapeamento ICAO**: Alinhamento formal com padrões AIXM/FIXM/WXXM
4. **Integração com Data Quality**: Utilizar ontologia para geração automática de checks de validação
5. **Versionamento Semântico**: Continuar com o controle de versões com OWL Versioning
6. **APIs SPARQL**: Expor endpoints públicos para consultas semânticas

A ontologia **AirData** representa um ativo estratégico que potencializa capacidades analíticas, garante qualidade semântica dos dados e habilita integrações inteligentes com sistemas de IA e outros repositórios de conhecimento aeronáutico.

3 AirData RAG System

3.1 Visão Geral

O **AirData** RAG System é uma plataforma de inteligência artificial especializada em regulamentações da aviação civil brasileira. Utilizando tecnologia RAG, o sistema combina busca semântica avançada com modelos de LLM para fornecer respostas precisas, contextualizadas e verificáveis sobre normas da ANAC, do Departamento de Controle do Espaço Aéreo (DECEA) e outros documentos normativos aeronáuticos.

Desenvolvido pelo Instituto Tecnológico de Aeronáutica (ITA) como parte do Projeto **AirData**, a plataforma oferece uma interface conversacional intuitiva onde partes interessadas podem obter informações regulatórias de forma rápida e confiável, com rastreabilidade completa das fontes e sistema integrado de avaliação de qualidade.

3.1.1 Objetivos do Sistema

O sistema foi concebido em torno de seis objetivos centrais. O primeiro é a **democratização do conhecimento regulatório**: facilitar o acesso a normas técnicas da aviação civil brasileira por profissionais de diferentes áreas, eliminando a necessidade de navegar manualmente por múltiplos portais e documentos. O segundo objetivo é garantir **respostas fundamentadas**, assegurando que toda informação gerada esteja ancorada em documentos oficiais verificáveis, com citação explícita das fontes. Isso leva ao terceiro ponto, a **rastreabilidade**: cada resposta traz pontuações de relevância (*scores*) e o trecho exato extraído do documento original, permitindo auditoria imediata.

A **eficiência operacional** é o quarto objetivo, visando reduzir significativamente o tempo gasto em buscas e na interpretação de regulamentações. Em quinto lugar está a **melhoria contínua**: um sistema de avaliação multidimensional permite monitorar e refinar a qualidade das respostas ao longo do tempo. Por fim, o suporte à **vigência temporal** possibilita consultar normas vigentes em datas específicas, o que é especialmente útil para análises históricas e investigações de conformidade retroativa.

3.2 Arquitetura do Sistema

O sistema é composto por dois serviços principais que operam em conjunto: uma **API RAG** (porta 8083), responsável por toda a lógica de busca semântica, geração de respostas e gerenciamento de sessões; e uma **Interface Web** (porta 8001), que provê a experiência de usuário com chat em tempo real. A API e a interface web são executadas diretamente no servidor como serviços **systemd**, gerenciados via `systemctl`, e expostas ao exterior por meio de um proxy reverso **Nginx**. Apenas o banco de dados vetorial **Qdrant** opera de forma conteinerizada via Docker, garantindo isolamento e portabilidade para o armazenamento e busca de *embeddings*. O servidor de modelos **Ollama** também é executado nativamente no servidor.

3.2.1 Componentes Principais

Tabela 3.1: Componentes da arquitetura **AirData** RAG

Componente	**Descrição**	**Tecnologia**
Interface Web	Chat em tempo real com *streaming*, avaliação por estrelas e histórico persistente de conversas	FastAPI, Jinja2, JavaScript *Server-Sent Events* (SSE)
API RAG	Servidor que processa consultas, gerencia sessões e coordena a geração de respostas	Python, FastAPI (porta 8083)
Banco Vetorial	Armazenamento e busca de *embeddings* com índice *Hierarchical Navigable Small World* (HNSW) e filtros por metadados temporais	Qdrant (porta 6333)
Modelo de Embeddings	Geração de representações vetoriais de 1024 dimensões para textos jurídicos em PT-BR	Legal-BERTimbau-sts-large-ma-v3
Servidor LLM	Modelos de linguagem para síntese e explicação de normas, executados localmente	Ollama (porta 11434)
Base de Conhecimento	Documentos normativos indexados: ICAs do DECEA e normas da ANAC (Regulamentos Brasileiros de Homologação Aeronáutica (RBHA), Regulamentos Brasileiros da Aviação Civil (RBAC)) coletadas via LexML	Instruções do Comando da Aeronáutica (ICA)s DECEA, RRBHA, RBAC, LexML

3.2.2 Stack Tecnológico

3.2.2.1 Backend

O backend é construído em **Python 3.11** com o framework **FastAPI**, escolhido por sua natureza assíncrona (*async/await*), alto desempenho e geração automática de documentação OpenAPI. Em produção, o servidor é executado via **Gunicorn** com *workers* Uvicorn, permitindo paralelismo real entre requisições. A validação e serialização de dados é feita com **Pydantic v2**, e o sistema de *logging* utiliza **Loguru**, com rotação e retenção configuráveis via variáveis de ambiente.

3.2.2.2 Frontend

A interface web é renderizada no servidor com **Jinja2 Templates**, evitando a complexidade de frameworks JavaScript pesados como React ou Vue.js. O *streaming* de respostas em tempo real — exibidas token por token conforme são geradas pelo LLM — é implementado via SSE, o que exige uma camada de JavaScript leve para consumo do fluxo de dados no navegador.

3.2.2.3 Banco de Dados Vetorial

O **Qdrant** é o banco de dados vetorial responsável por armazenar e recuperar os *embeddings* dos documentos normativos. A busca utiliza **similaridade de cossenos** como métrica de distância e o índice HNSW, que oferece complexidade de busca $O(\log N)$ e permite recuperação em dezenas de milissegundos mesmo com coleções de grande escala. Além dos vetores, o Qdrant armazena metadados de cada documento (datas de vigência, status, categoria), sobre os quais são criados índices de *payload* para filtragem eficiente durante as consultas.

3.2.2.4 Modelos de IA

Para geração de *embeddings*, o sistema utiliza o **Legal-BERTimbau-sts-large-ma-v3** (`rufimelo/Legal-BERTimbau-sts-large-ma-v3`), um modelo especializado em textos jurídicos em português brasileiro, baseado na arquitetura BERTimbau com *fine-tuning* para similaridade semântica via *Sentence Transformers*. Esse modelo produz vetores de 1024 dimensões, oferecendo representações ricas do conteúdo normativo.

Para a geração de respostas, o sistema suporta múltiplos modelos LLM executados localmente via **Ollama**: o **Llama 3.2 3B** é o padrão do sistema, equilibrando qualidade e velocidade; o **Llama 3.1 8B** e o **Llama 3.1 70B** estão disponíveis para cenários que exigem maior capacidade de raciocínio; o **Phi-3 3.8B** (Microsoft) e o **Gemma 2B**

(Google) completam o conjunto como alternativas leves. A execução local de todos os modelos elimina dependência de APIs externas e garante a privacidade dos dados consultados.

3.2.2.5 Segurança

O acesso à API é protegido por **autenticação via API Key**, transmitida no cabeçalho HTTP `X-API-Key` em todas as requisições a *endpoints* protegidos. Um mecanismo de **rate limiting** (biblioteca `slowapi`) impõe limite de 100 requisições por minuto por endereço IP, configurável via `RATE_LIMIT`. As origens CORS permitidas são definidas via variável de ambiente `CORS_ORIGINS`, e todas as entradas são validadas com Pydantic v2 antes de qualquer processamento. No ambiente de produção, o **Nginx** atua como proxy reverso, podendo ser configurado com terminação *Transport Layer Security* (TLS) para conexões seguras.

3.3 Fluxo de Dados e Processamento

O sistema opera em duas fases principais: **ingestão de documentos** (executada *offline*, uma única vez ou periodicamente conforme publicação de novas normas) e **consulta interativa** (em tempo real, por requisição do usuário).

3.3.1 Pipeline de Ingestão de Documentos

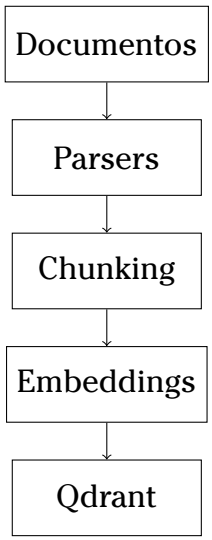

Figura 3.1: Pipeline de processamento de documentos para indexação vetorial

3.3.1.1 Coleta de Documentos

O sistema conta com três mecanismos de coleta, cada um especializado em uma fonte de dados.

LexML Scraper (`parsers/lexml_scraper.py`) é a ferramenta responsável por coletar normas do portal LexML (<https://www.lexml.gov.br>), que agrega regulamentações federais brasileiras. O *scraper* realiza extração automatizada de regulamentações aeronáuticas — RBHA, RBAC, Leis e Decretos — filtrando por palavras-chave relevantes. Cada documento coletado preserva seus metadados (data de publicação, número da norma, órgão emissor) e tem sua estrutura hierárquica de artigos, incisos e parágrafos mantida. Um sistema de deduplicação incremental via `document_tracker.py` garante que documentos já processados não sejam reingeridos. Os arquivos resultantes são salvos em formato *JavaScript Object Notation* (JSON) no diretório `data/lexml/`.

DECEA Scraper (`parsers/decea_scraper.py`) lida especificamente com o portal do DECEA, cujas páginas utilizam carregamento dinâmico via JavaScript. Por esse motivo, a ferramenta é implementada com **Selenium**, que controla um navegador real para navegar e fazer o download das ICAs. O *scraper* suporta tanto a coleta completa do catálogo quanto a coleta seletiva por sigla (ex.: `--slugs ICA-100-12,ICA-63-47`), salvando os arquivos *Portable Document Format* (PDF) originais em `data/originals/ica/`.

PDF Parser (`parsers/pdf_parser.py`) processa documentos PDF locais — tanto ICAs baixadas pelo DECEA Scraper quanto outros regulamentos em formato PDF. A extração de texto é feita primariamente via **pdfplumber** e **PyMuPDF (fitz)**, com **PyPDF2** como *fallback*. Para documentos escaneados ou com texto não selecionável, o módulo **PaddleOCR** pode ser ativado via `ENABLE_OCR=true`, reconhecendo o texto visualmente. Em todos os casos, o parser aplica correção de *encoding* e normalização de caracteres especiais do português.

3.3.1.2 Chunking Inteligente

O processo de *chunking* divide documentos longos em segmentos semanticamente coerentes, pois modelos de *embedding* possuem limite de contexto e a busca vetorial é mais precisa sobre trechos focados do que sobre documentos inteiros. O sistema implementa duas estratégias especializadas.

O **ArticleChunker** é utilizado para normas estruturadas por artigos (RBHA, RBAC e documentos LexML em geral). Ele detecta fronteiras de artigos pelo padrão "Art. Nº" e divide o conteúdo respeitando a hierarquia natural dos textos jurídicos: artigos,

parágrafos e incisos. Cada *chunk* gerado é limitado a **512 tokens** e carrega metadados como número do artigo, nível hierárquico e referências cruzadas.

O **ICAChunker** é especializado nas ICAs do DECEA, que seguem uma estrutura técnica distinta. Ele segmenta os documentos por artigos e, quando necessário, realiza subdivisões progressivas em parágrafos (§), incisos (I, II...), alíneas (a), b)...) e seções ou capítulos. O limite de 512 tokens por *chunk* também se aplica aqui, e o cabeçalho do documento é sempre preservado no primeiro *chunk* para manter o contexto do regulamento.

Em ambas as estratégias, *chunks* adjacentes compartilham uma sobreposição (*overlap*) de **50 tokens**. Esse recurso preserva o contexto entre segmentos contíguos, evita quebras no meio de conceitos importantes e melhora o *recall* durante a busca vetorial.

3.3.1.3 Geração de Embeddings

Cada *chunk* é transformado em um vetor de **1024 dimensões** pelo modelo **Legal-BERTimbau-sts-large-ma-v3**. O comprimento máximo de entrada é de 512 tokens por sequência. O processamento ocorre em lotes de 32 *chunks* (configurável via `EMBEDDING_BATCH_SIZE`), com detecção automática de CUDA para aceleração por *Graphics Processing Unit* (GPU) e *fallback* transparente para *Central Processing Unit* (CPU).

3.3.1.4 Indexação no Qdrant

Vetores são armazenados na coleção `aviation_regulations` do Qdrant com a seguinte estrutura:

```
{
  "id": "uuid",
  "vector": [0.234, -0.567, ...],  // 1024 dimensões
  "payload": {
    "text": "Art. 5º - Todo voo deve...",
    "regulation_id": "rbha-147-art-5",
    "version": "2023-05-12",
    "effective_date": "2023-05-12T00:00:00Z",
    "expiry_date": null,
    "status": "active",
    "metadata": { "category": "operações", ... }
```

```
  }
}
```

O índice HNSW é configurado com **M = 16** (conexões por nó), **ef_construct = 100** (parâmetro de construção do grafo) e **ef_search = 64** (parâmetro de busca em tempo de consulta). Sobre os campos do *payload*, são criados índices específicos para `effective_date` e `expiry_date` (tipo DATETIME), `status` e `regulation_id` (tipo KEYWORD) e
`metadata.category` (tipo KEYWORD), permitindo filtros eficientes sem varredura completa da coleção.

3.3.2 Pipeline de Consulta Interativa

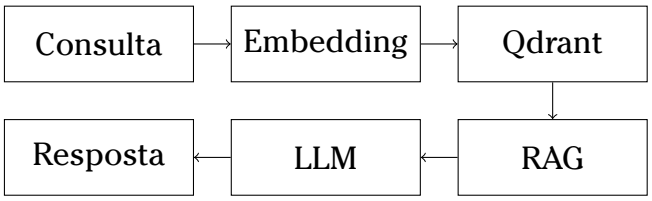

Figura 3.2: Fluxo de processamento de consulta do usuário até geração de resposta

3.3.2.1 Busca Semântica com Filtros Temporais

Ao receber uma pergunta, o sistema a converte imediatamente em um vetor de 1024 dimensões usando o mesmo modelo Legal-BERTimbau empregado na ingestão, garantindo compatibilidade no espaço vetorial. Em seguida, realiza uma busca por similaridade de cossenos no Qdrant, retornando por padrão os **5 *chunks*** mais relevantes com *score* acima de **0.3** (ambos configuráveis via `SEARCH_TOP_K` e `SEARCH_SCORE_THRESHOLD`). Quando o usuário especifica uma data de referência (`rag_date`), a busca inclui filtros temporais automáticos sobre os campos `effective_date` e `expiry_date`, garantindo que apenas normas vigentes naquela data sejam consideradas. Os *chunks* recuperados são então ordenados por *score* e combinados em um *prompt* estruturado.

Exemplo de consulta com filtro temporal:

```
Query: "requisitos para licença de piloto comercial"
Filtros temporais: {
  "effective_date": {"lte": "2023-01-01"},
  "expiry_date":   {"gte": "2023-01-01"}  // null = ainda vigente
```

```
}
Score Threshold: 0.3
Top-K: 5
```

3.3.2.2 Geração de Resposta

Os *chunks* recuperados são combinados com a consulta original em um *prompt* estruturado enviado ao LLM via Ollama:

```
Você é um assistente especializado em regulamentações
aeronáuticas brasileiras. Use APENAS as informações das
fontes abaixo para responder.

FONTES:
[1] Art. 147.45 - RBHA 147 (2023-05-12, score: 0.89)
"Todo candidato a licença de piloto comercial deve..."

[2] Seção 4.2 - ICA 100-12 (2022-11-03, score: 0.84)
"Os requisitos mínimos de horas de voo são..."

PERGUNTA: {user_query}

RESPOSTA:
```

O LLM (padrão: Llama 3.2 3B) processa este *prompt* e gera a resposta em *streaming*, com temperatura 0.3 para favorecer respostas factuais e determinísticas.

3.4 Recursos e Funcionalidades

3.4.1 Chat Conversacional com Gerenciamento de Sessões

A interface de chat exibe as respostas em *streaming* token por token via SSE, de modo que o usuário começa a ler a resposta imediatamente, sem aguardar sua geração completa. Cada conversa é identificada por um *Universally Unique Identifier* (UUID) único e mantida em memória com *Time To Live* (TTL) configurável de 60 minutos; um processo de limpeza automático descarta sessões expiradas a cada 5 minutos. O histórico de mensagens é preservado dentro da sessão (até 50 trocas), permitindo

que o usuário faça perguntas de acompanhamento sem repetir o contexto. A infraestrutura suporta até 10.000 sessões simultâneas e fornece indicadores visuais de processamento (*typing indicators* e tempo de resposta).

3.4.2 Sistema de Avaliação de Qualidade

Cada resposta pode ser avaliada pelo usuário em 5 dimensões independentes, em escala de 1 a 5 estrelas:

1. **Precisão Técnica**: a resposta está correta segundo as normas?
2. **Completude**: todos os aspectos da pergunta foram abordados?
3. **Clareza**: a resposta é compreensível e bem estruturada?
4. **Qualidade das Citações**: as fontes são relevantes e bem referenciadas?
5. **Relevância**: a resposta atende ao que foi perguntado?

As avaliações são registradas e alimentam métricas agregadas de qualidade por categoria de pergunta, identificação de respostas problemáticas para refinamento, comparação de performance entre diferentes modelos LLM e a construção futura de conjuntos de dados para *fine-tuning* supervisionado.

3.4.3 Fontes Oficiais Verificáveis

Toda resposta é acompanhada de referências completas às fontes utilizadas: nome do regulamento, data de publicação, artigo ou seção específico, *score* de relevância (valor entre 0.0 e 1.0) e o trecho exato extraído do documento original, sem paráfrase. Esse mecanismo permite que o usuário verifique de forma independente qualquer informação fornecida pelo sistema.

Exemplo de citação:

> *Fonte: RBHA 147 – Art. 5º (publicado em 2023-05-12) [Relevância: 0.89]*
>
> "Todo voo comercial deve ser operado por piloto com licença válida emitida pela ANAC, conforme estabelecido neste regulamento."

3.4.4 Busca Vetorial Direta (sem LLM)

O sistema oferece também um modo de busca semântica pura, acessível via *endpoint* `POST /api/vector-search`, que retorna os *chunks* mais similares à consulta ranqueados por *score*, sem acionar o LLM para geração de resposta. Essa funcionalidade é útil para auditar a qualidade e cobertura do índice vetorial, verificar quais documentos o sistema encontra para uma dada consulta e realizar depurações, com latência significativamente menor por não envolver inferência do modelo de linguagem.

3.4.5 Histórico de Conversas

Além do histórico em memória gerenciado por sessão, as conversas são registradas em arquivos JSON no diretório `web/chat_history/` do servidor, cada uma identificada por um UUID único. Esses arquivos contêm a transcrição completa das trocas de mensagens com *timestamps*, servindo como registro permanente das interações.

3.5 Implantação e Infraestrutura

3.5.1 Modelo de Implantação

Na configuração atual de produção, o sistema adota um modelo de implantação híbrido. A **API RAG** e a **Interface Web** são executadas diretamente no servidor como serviços **systemd**, gerenciados pelo `systemctl`. Essa abordagem permite controle direto do ciclo de vida dos processos (inicialização automática no *boot*, reinicialização em caso de falha, gerenciamento de logs via `journalctl`) e simplifica o acesso aos recursos locais do servidor, como o modelo de *embeddings* em cache e o sistema de arquivos para persistência de histórico.

O **Nginx** atua como proxy reverso, recebendo as requisições externas e encaminhando-as para os serviços internos nas portas apropriadas (8083 para a API RAG, 8001 para a interface web). O Nginx também é responsável pela terminação TLS, compressão de respostas e controle de acesso por IP, quando configurado.

Apenas o **banco de dados vetorial Qdrant** opera de forma conteinerizada via **Docker**, aproveitando a imagem oficial `qdrant/qdrant:latest`. Essa escolha garante isolamento do armazenamento vetorial, facilidade de atualização do Qdrant (basta trocar a tag da imagem), portabilidade dos dados via volumes Docker e separação clara entre a camada de aplicação e a camada de dados vetoriais. O container expõe as

portas 6333 (HTTP/REST) e 6334 (gRPC) e utiliza um volume persistente para manter os dados entre reinicializações.

O servidor de modelos **Ollama** também é executado nativamente no servidor, sem conteinerização, para acesso direto à GPU e máxima performance na inferência dos modelos LLM.

A Tabela 3.2 resume a forma de implantação de cada componente do sistema.

Tabela 3.2: Componentes e seus modos de implantação em produção

Componente	Modo de Implantação	Porta	Descrição
API RAG	Serviço systemd (Gunicorn + Uvicorn)	8083	API principal do sistema RAG
Interface Web	Serviço systemd (Uvicorn)	8001	Chatbot com interface web
Nginx	Serviço nativo do sistema	80 / 443	Proxy reverso com TLS
Qdrant	Container Docker	6333 / 6334	Banco de dados vetorial (único componente conteinerizado)
Ollama	Serviço nativo do servidor	11434	Servidor de modelos LLM locais

3.5.2 Serviços systemd

Cada serviço da aplicação é registrado como uma *unit* systemd, permitindo gerenciamento padronizado via `systemctl`. Os principais comandos de operação são:

```
# Gerenciamento da API RAG
sudo systemctl start ragapi
sudo systemctl stop ragapi
sudo systemctl restart ragapi
sudo systemctl status ragapi

# Gerenciamento da Interface Web
sudo systemctl start ragweb
sudo systemctl stop ragweb
sudo systemctl restart ragweb
sudo systemctl status ragweb

# Verificar logs em tempo real
sudo journalctl -u ragapi -f
sudo journalctl -u ragweb -f
```

Os arquivos de *unit* ficam em `/etc/systemd/system/` e definem o usuário de exe-

cução, o diretório de trabalho, o comando de inicialização (Gunicorn para a API, Uvicorn para a interface web), variáveis de ambiente e a política de reinicialização automática (`Restart=on-failure`).

3.5.3 Container Docker do Qdrant

O Qdrant é o único componente executado via Docker no ambiente de produção atual. O container pode ser iniciado diretamente:

```
docker run -d --name qdrant \
  -p 6333:6333 -p 6334:6334 \
  -v qdrant_data:/qdrant/storage \
  --restart unless-stopped \
  qdrant/qdrant:latest
```

O volume `qdrant_data` persiste as coleções vetoriais entre reinicializações do container. A flag `--restart unless-stopped` garante que o container reinicie automaticamente após uma falha ou *reboot* do servidor.

3.5.4 Principais Variáveis de Configuração

Todo o comportamento do sistema é controlado por variáveis de ambiente definidas no arquivo `.env`. Os parâmetros mais relevantes estão listados na Tabela 3.3.

Tabela 3.3: Principais variáveis de configuração do sistema

Variável	**Padrão**	**Descrição**
`OLLAMA_MODEL`	`llama3.2:3b`	Modelo LLM padrão
`EMBEDDING_MODEL`	`rufimelo/Legal-BERTimbau-sts-large-ma-v3`	Modelo de *embeddings*
`SEARCH_TOP_K`	5	Número de *chunks* recuperados por consulta
`SEARCH_SCORE_THRESHOLD`	0.3	Limiar mínimo de similaridade
`CHUNK_MAX_TOKENS`	512	Tamanho máximo de cada *chunk*
`CHUNK_OVERLAP`	50	Sobreposição entre *chunks* adjacentes (em tokens)
`LLM_TEMPERATURE`	0.3	Temperatura do LLM (0 = mais determinístico)
`LLM_MAX_TOKENS`	500	Limite de tokens na resposta gerada pelo LLM
`RATE_LIMIT`	100	Requisições permitidas por minuto por IP
`ENABLE_OCR`	`false`	Ativar OCR para PDFs escaneados
`EMBEDDING_BATCH_SIZE`	32	Número de *chunks* por *batch* de *embedding*
`API_WORKERS`	4	Número de *workers* Gunicorn em produção

3.6 Integração com **AirData** Platform

O sistema RAG integra-se com os demais componentes do **AirData** de forma complementar. O módulo de **Data Quality** pode utilizar a ontologia do RAG para validar a consistência semântica das respostas geradas, enquanto o **Data Warehouse** pode ser enriquecido com contexto regulatório extraído nas consultas. Para os usuários, o chatbot está acessível diretamente via menu principal da plataforma **AirData**, e seus *endpoints* REST, documentados automaticamente via OpenAPI/Swagger, permitem integração com sistemas externos de forma padronizada.

3.7 Performance e Escalabilidade

3.7.1 Métricas de Performance

Tabela 3.4: Benchmarks de performance do sistema RAG

Operação	**Latência Média**	**Throughput**
Busca vetorial (top-5)	8–15ms	~1000 req/s
Geração de *embedding* (1 *chunk*)	25–40ms	~500 *chunks*/s
Resposta LLM – Llama 3.1 70B	2–5s	~20 tokens/s
Resposta LLM – Llama 3.1 8B	0.8–1.5s	40–60 tokens/s
Resposta LLM – Llama 3.2 3B (padrão)	0.5–1s	60–80 tokens/s
Resposta LLM – Phi-3 3.8B	0.5–1s	60–80 tokens/s
Resposta LLM – Gemma 2B	0.3–0.8s	80–120 tokens/s
Pipeline completo (busca + LLM)	3–6s	~15 req/s

Nota: Valores medidos com aceleração por GPU. A performance em CPU é significativamente menor, especialmente para os modelos de maior porte.

3.7.2 Escalabilidade

A arquitetura foi projetada para crescer conforme a demanda. O **Qdrant**, já conteinerizado, suporta *sharding* automático para coleções de grande volume e pode ser facilmente migrado para um cluster dedicado. A **FastAPI** com *async/await* processa múltiplas requisições de forma concorrente sem bloqueio, e o **Gunicorn** permite escalar o número de *workers* de forma simples (`API_WORKERS=4` por padrão). Os modelos LLM via **Ollama** podem ser replicados em múltiplas instâncias com distribuição de carga. Futuramente, a conteinerização completa de todos os serviços via Docker

Compose poderá ser adotada para simplificar o *deploy* e permitir orquestração via Kubernetes.

4 AirData Data Check

4.1 Visão Geral

O **AirData Data Check** é o subsistema principal de coleta, processamento e validação de dados aeronáuticos. Este capítulo cobre sua implementação, componentes e operação completa.

O sistema monitora **18 bases de dados** distintas, cada uma com sua frequência e horário específico de execução. Este ecossistema de dados foi projetado para prover uma visão completa **Gate-to-Gate** da operação aérea.

4.1.1 Objetivo e Escopo

Este capítulo tem como objetivo descrever a ferramenta **AirData Data Check**, sua arquitetura, principais funcionalidades e abordagem metodológica para avaliação da qualidade de dados operacionais e meteorológicos. Além da descrição técnica da solução, o capítulo apresenta uma análise qualitativa do conjunto de dados utilizado, com base em dimensões consolidadas de qualidade de dados. O escopo abrange dois eixos complementares. O primeiro é estrutural, detalhando como a ferramenta foi projetada, quais mecanismos de validação implementa e como organiza regras, métricas e verificações. O segundo é analítico, aplicando esses critérios ao conjunto de dados em estudo, a fim de avaliar sua adequação para análises estatísticas, uso operacional e aplicações de *Inteligência Artificial* (IA). O propósito é transformar dados operacionais brutos em um ativo confiável, com regras explícitas, mensuráveis e passíveis de monitoramento contínuo.

Por fim, o capítulo discute o impacto dessas dimensões na utilização analítica dos dados, examinando sua adequação para geração de indicadores operacionais, análises estatísticas e suporte a modelos de IA.

O ecossistema de dados do **AirData** é dividido em quatro categorias que representam diferentes aspectos da operação aeronáutica:

1. **Operacional e Regulatório**: Dados que regem a intenção e o registro oficial da

operação

2. **Meteorologia**: Condições ambientais em diferentes escalas e altitudes
3. **Vigilância e Infraestrutura**: A execução física e o mapa estrutural do espaço aéreo
4. **Planejamento e Fluxo**: Gerenciamento de slots e fluxo de tráfego aéreo

A Figura 4.1 apresenta uma referência visual para o ecossistema de dados.

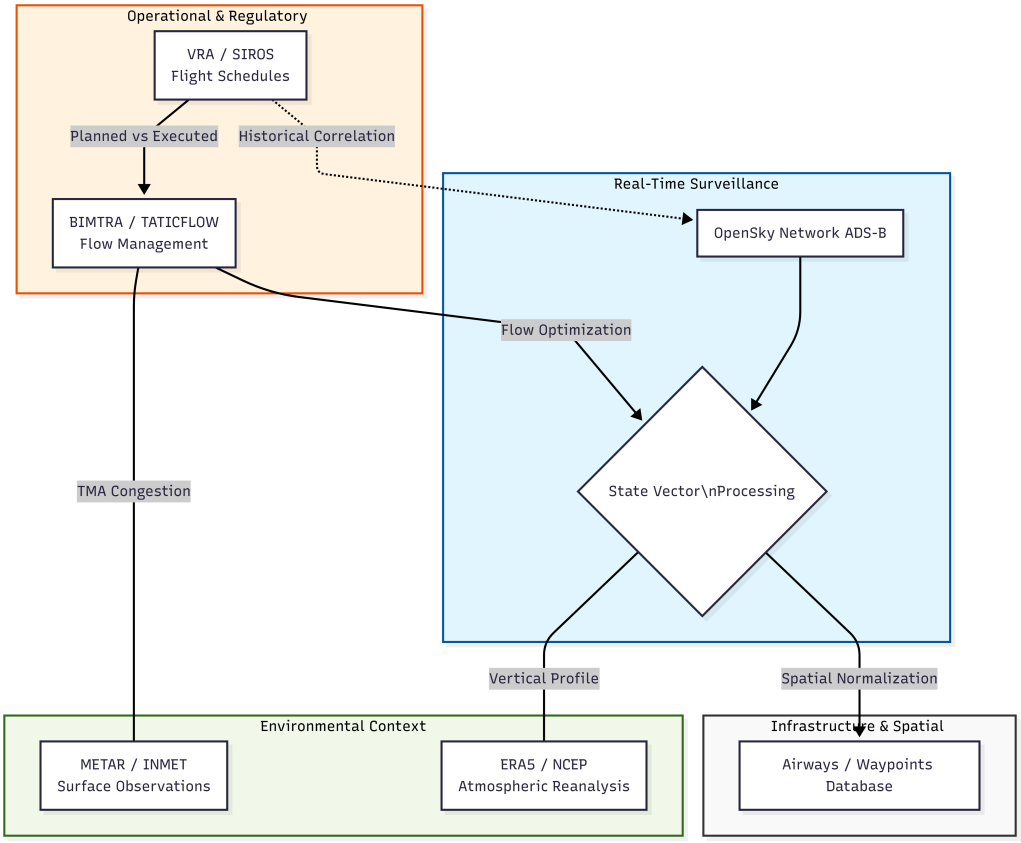

Figura 4.1: Ecossistema geral de dados do **AirData**.

4.1.1.1 Fora de Escopo

Não fazem parte do escopo atual a implementação técnica detalhada de regras de validação (código executável) e a documentação exaustiva de todos os campos do banco de dados. Também não serão abordados modelos específicos de Machine Learning, métricas de performance de modelos, análises de fontes externas não descritas ou a avaliação de infraestrutura e performance de sistemas.

4.1.2 Matriz de Conectividade Técnica

A potência analítica deste ecossistema reside na interconexão das diferentes camadas de dados. A tabela abaixo ilustra as principais relações entre datasets:

Tabela 4.1: Conectividade entre Datasets

Dataset A	**Dataset B**	**Chave de Ligação**	**Aplicação Analítica**
OpenSky	Waypoints	Geo-Position	Map-Matching e detecção de desvios de rota
VRA / SIROS	OpenSky	Callsign / Time	Validação de execução real vs planejado
VRA / BIMTRA	METAR	ICAO / Time	Análise de impacto meteorológico no solo
OpenSky	ERA5 / NCEP	Lat/Lon/Alt/Time	Turbulência e Wind Shear em altitude

4.1.3 Valor para Inteligência Artificial

Este ecossistema foi projetado para sustentar modelos de alta fidelidade:

1. **Engenharia de Features**: A combinação de vigilância com meteorologia permite prever o consumo de combustível e os tempos de voo;
2. **Detecção de Anomalias**: O cruzamento entre o regulatório (VRA) e o real *Automatic Dependent Surveillance-Broadcast* (ADS-B) permite identificar desvios operacionais ou inconsistências;
3. ***Graph Neural Networks* (GNN)**: A estrutura de Waypoints permite modelar o espaço aéreo como um grafo dinâmico para previsão de congestionamento

4.1.4 Governança e Qualidade

Cada base de dados é submetida a um framework estruturado de qualidade da informação, fundamentado em sete pilares essenciais de governança de dados, garantindo confiabilidade analítica, rastreabilidade e integridade operacional:

- **Completude**: Garante cobertura integral dos campos críticos necessários para análise operacional. Ex.: um voo no BIMTRA sem horário de pouso inviabiliza o cálculo de tempo de voo ou do atraso.

- **Consistência**: Assegura coerência lógica e temporal entre sistemas distintos. Ex.: um voo registrado às 14h no BIMTRA e às 16h no SIROS indica desalinhamento sistêmico.
- **Acurácia**: Verifica o realismo e a plausibilidade operacional dos dados. Ex.: um voo Rio - São Paulo com duração registrada de 4 minutos caracteriza erro material.
- **Validade**: Confirma aderência estrita a padrões formais (ICAO, IATA, formatos de data). Ex.: código ICAO "ZZZZ" ou data "2026-13-40" violam regras estruturais.
- **Pontualidade**: Mede a disponibilidade do dado dentro da janela útil para decisão. Ex.: dados da REDEMET recebidos 3 horas após o voo possuem valor histórico, mas não operacional.
- **Unicidade**: Garante inexistência de duplicidades ou registros redundantes. Ex.: dois registros idênticos do mesmo voo no mesmo dia distorcem métricas estatísticas.
- **Rastreabilidade**: Permite auditoria completa de alterações, origem e versionamento. Ex.: identificação da data e horário de upload de um dado e seus respectivos responsáveis.

Esses sete pilares estruturam o modelo de governança do **AirData**, assegurando que qualquer aplicação de IA ou Analytics opere sobre bases confiáveis, auditáveis e semanticamente coerentes com o domínio ATM.

4.2 Arquitetura do Sistema

O **AirData Data Check** é estruturado em uma arquitetura de três camadas, composta pelos níveis de ingestão, armazenamento e apresentação. Essa organização garante separação de responsabilidades, escalabilidade e facilidade de manutenção. As camadas que compõem o sistema são:

- **Camada de Ingestão**: Apache Airflow (orquestração)
- **Camada de Armazenamento**: PostgreSQL (banco de dados central)
- **Camada de Apresentação**: FastAPI + HTML5/CSS3/JavaScript

4.2.1 Apache Airflow

A camada de ingestão é implementada com o Apache Airflow, responsável por orquestrar os pipelines de ETL. O serviço está instalado em um servidor dedicado (161.24.29.22) e atualmente monitora 18 DAG, sendo uma para cada base de dados integrada ao sistema. Cada DAG possui agendamento configurado conforme as características da respectiva fonte de dados, podendo operar em horários fixos ou em frequências específicas. Os logs de execução são armazenados em banco de dados e podem ser acessados por meio da interface do próprio Airflow, permitindo rastreabilidade e auditoria das execuções.

4.2.2 PostgreSQL

A camada de armazenamento utiliza PostgreSQL 13+ como banco de dados central. O serviço também está hospedado no servidor 161.24.29.22, com acesso realizado por meio de túnel SSH para garantir comunicação segura. A estrutura do banco está organizada em dois schemas principais: o schema `airdata`, que armazena os dados coletados, e o schema `airflow`, que contém os metadados relacionados à orquestração. O sistema possui 18 tabelas principais, além de tabelas auxiliares de metadados, totalizando atualmente mais de 87 milhões de registros armazenados.

4.2.3 FastAPI

A camada de backend é desenvolvida com FastAPI, um framework web assíncrono moderno, executado na porta 9010. Essa API é responsável por intermediar o acesso entre o frontend e o banco de dados, oferecendo endpoints para consulta de DAGs, listagem de bases monitoradas, execução de consultas SQL ad hoc e acesso ao catálogo de dados. A autenticação e o acesso ao banco remoto são realizados por meio de túnel SSH, reforçando a segurança da comunicação entre os componentes.

4.2.4 Frontend - HTML5/CSS3/JavaScript

A camada de frontend é construída com HTML5, CSS3 e JavaScript puro (Vanilla JavaScript), sem dependências externas estruturais. A interface oferece suporte a modo claro e escuro com persistência de preferência, abas interativas para navegação entre bases de dados, editor SQL com destaque de sintaxe, visualização geográfica com Leaflet.js e gráficos de desempenho com Chart.js. Além disso, o sistema

possui mecanismos de paginação e filtragem de dados, garantindo melhor experiência na exploração de grandes volumes de informação. A interface é totalmente responsiva, adaptando-se a dispositivos móveis, tablets e desktops.

O fluxo de dados do sistema inicia-se com a extração realizada pelas DAGs do Airflow, que coletam dados de fontes síncronas. Em seguida, ocorre a etapa de transformação, na qual os dados são processados e padronizados. Após essa fase, os registros são carregados no PostgreSQL, incluindo a coluna `dt_insercao` para controle temporal. Na sequência, um motor de qualidade executa validações sobre os dados inseridos, assegurando consistência e integridade. Por fim, os dados validados são disponibilizados por meio da API e podem ser explorados pelos usuários na interface web, através de dashboards e consultas interativas.

4.2.5 Conectividade e Rede

No que se refere à infraestrutura de rede, o servidor de banco de dados está acessível via SSH pelo endereço 161.24.29.22 na porta 2222. Toda a comunicação é realizada por meio de túnel SSH criptografado, garantindo confidencialidade no tráfego de dados. As credenciais são armazenadas em variáveis de ambiente, evitando exposição sensível no código-fonte, e o firewall do servidor é configurado para permitir apenas os acessos estritamente necessários à operação do sistema.

4.3 BIMTRA

4.3.1 Visão Geral dos Dados

O conjunto de dados é composto por registros de movimentos operacionais de voo, abrangendo informações desde o planejamento até a execução e o registro final do evento no sistema. Esses dados são gerados e consolidados a partir de sistemas operacionais de controle e gestão do tráfego aéreo. As informações refletem eventos reais do domínio ATM e podem ser usadas para acompanhamento operacional, análise de desempenho e auditoria de processos, além de suportar tomada de decisão e aplicações de IA. Na Figura 4.2 está o fluxo geral do dado.

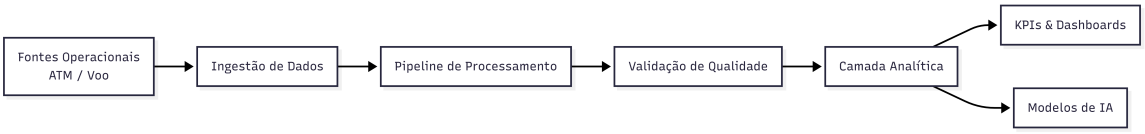

Figura 4.2: Fluxo geral do dado.

4.3.1.1 Natureza e Granularidade

Cada registro representa um movimento individual de voo, associado a uma operação de partida ou chegada. O conjunto de dados contém campos categóricos, temporais e textuais. A temporalidade é marcada por múltiplos timestamps por registro, capturando fases do ciclo do voo. A atualização ocorre de forma incremental, refletindo a evolução dos eventos desde o planejamento e estimativa até a execução e registro final.

4.3.1.2 Estrutura Geral do Conjunto de Dados

O dataset pode ser organizado em quatro grupos: identificação e contexto do voo, localização e espaço aéreo, temporalidade e eventos, e características da aeronave e procedimentos. Essa organização facilita a avaliação de qualidade e a integração semântica futura.

4.3.1.3 Fluxo de Dados no Pipeline

O ciclo de vida do dado segue o fluxo representado na Figura 4.2, do registro operacional até a ingestão e disponibilização no pipeline.

4.3.2 Dicionário de Dados

Esta seção apresenta um resumo dos principais campos, com foco em atributos críticos para análise, validação de qualidade e aplicações de IA. O objetivo não é documentar todos os campos, mas estabelecer base comum sobre os elementos estruturantes do dataset.

4.3.2.1 Identificação e Contexto do Voo

O campo `codmovimentovalidado` identifica unicamente o movimento validado no sistema. O campo `numvoo` representa o callsign do voo e é fundamental para agregações

e correlações. O campo `deparr` indica o tipo de operação (partida ou chegada), essencial para interpretar timestamps e regras operacionais.

4.3.2.2 Aeródromos e Espaço Operacional

Os campos `adpartida` e `addestino` representam os códigos ICAO de partida e destino. Os campos `altn` e `altn2` descrevem aeródromos alternativos. O campo `orgaoats` representa o órgão ATS responsável pelo controle do movimento.

4.3.2.3 Aeronave e Características Técnicas

O campo `matricula` identifica a aeronave, permitindo histórico e recorrência. O campo `tipoaeronave` descreve o tipo de equipamento. O campo `esteiraturb` indica a categoria de esteira de turbulência. O campo `transponder` representa o código *Secondary Surveillance Radar* (SSR).

4.3.2.4 Temporalidade e Eventos do Movimento

Os campos `dhmovprev`, `dhmovestm` e `dhmovreal` representam, respectivamente, o horário planejado, a última estimativa e o horário real do movimento. Os campos `dhinseridobd` e `dt_insercao_airflow` registram a ingestão no banco e no pipeline, essenciais para rastrear latência.

4.3.2.5 Procedimentos e Infraestrutura

Os campos `saida` e `proceddescida` indicam procedimentos de saída e descida. O campo `pista` identifica a pista utilizada. Os campos `taxiway` e `tqaligado` trazem informações de taxiamento.

4.3.2.6 Campos Complementares

Os campos `rmk` e `rmkapp` armazenam observações textuais, enquanto `outrosdados` agrega dados complementares, exigindo tratamento específico.

4.3.3 Dimensões de Qualidade Avaliadas

A avaliação foi estruturada a partir de dimensões clássicas de qualidade de dados, adaptadas ao contexto de tráfego aéreo. As dimensões incluem completude, validade, consistência temporal e conformidade semântica. Na Figura 4.3 está o mapa das dimensões.

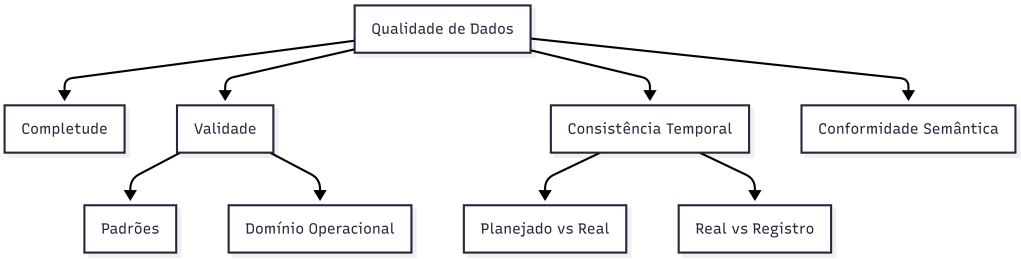

Figura 4.3: Mapa de dimensões de qualidade.

4.3.3.1 Completude

A completude avalia a presença de valores nos campos esperados, considerando obrigatoriedade técnica e contexto operacional. Campos críticos incluem `numvoo`, `adpartida`, `addestino`, `dhmovprev`, `dhinseridobd` e `orgaoats`. Campos operacionais, como `dhmovreal`, `dhmovestm`, `saida`, `proceddescida`, `pista` e `taxiway`, devem ser avaliados de forma condicional. Campos informativos como `rmk` e `outrosdados` são opcionais.

A avaliação distingue completude absoluta e completude condicional. Por exemplo, `dhmovreal` é obrigatório apenas para voos executados, e procedimentos de descida são esperados apenas em chegadas. Essa abordagem reduz falsos positivos e alinha a avaliação à realidade operacional. A métrica de completude é a razão entre registros preenchidos e registros esperados.

4.3.3.2 Validade

A validade avalia se os valores respeitam padrões esperados, domínios operacionais e significado semântico. A validade sintática verifica formato (ex.: ICAO, transponder). A validade por domínio operacional garante que valores pertençam ao conjunto permitido. A validade semântica básica avalia coerência lógica imediata (ex.: origem e destino distintos, procedimentos compatíveis com operação).

Dados inválidos afetam agregações, análises categorizadas, modelos de ML e integração com grafos de conhecimento, aumentando cardinalidade e ruído. Recomenda-se

monitoramento contínuo integrado ao pipeline.

4.3.3.3 Conformidade Semântica

A conformidade semântica verifica se valores e relações preservam o significado do domínio ATM. Entidades conceituais incluem voo, aeronave, aeródromo, órgão ATS e procedimentos. Problemas semânticos incluem duplicidade de entidades, atributos tratados como texto livre e relações implícitas não preservadas. Essa dimensão é essencial para integração de múltiplas fontes, inferência lógica e grafos de conhecimento.

Regras semânticas em SHACL são restrições formais aplicadas a dados modelados em RDF que verificam se entidades e suas relações obedecem à estrutura, cardinalidade e coerência lógica definidas por um modelo conceitual.

Na Figura 4.4 está um exemplo conceitual relacionado à validação semântica com SHACL.

Figura 4.4: Conceito de validação semântica com SHACL.

4.3.4 Regras e Métricas de Qualidade

4.3.4.1 Regras Básicas

As regras básicas garantem estrutura mínima consistente, com presença, tipagem, padronização e unicidade. Elas funcionam como filtro inicial, reduzindo falsos positivos em validações mais complexas e garantindo integridade estrutural para validações semânticas.

Em implementação SHACL, essas regras seriam restrições de presença, tipo e cardinalidade, com validações simples de consistência estrutural.

4.3.4.2 Regras Temporais de Consistência

Recomenda-se expressar as regras temporais como SHACL aplicadas a entidades de voo. A implementação conceitual inclui três camadas: tipagem e presença, regras

cross-field de comparação temporal e regras condicionais por estado (DEP vs ARR, voo concluído, etc.). O fluxo recomendado inclui mapear registros para entidades, aplicar Shapes como Quality Gate e gerar relatório de validação com contagens e exemplos de violações.

4.3.4.3 Regras Cross-field

As regras cross-field avaliam coerência lógica entre múltiplos campos do mesmo registro. Exemplos incluem coerência entre tipo de operação e campos temporais, relação entre aeródromos e procedimentos, coerência entre aeronave e características técnicas, relação entre planejamento e execução e compatibilidade entre infraestrutura utilizada e tipo de movimento. Essas regras são fundamentais para detectar incoerências semânticas que não aparecem em validações campo a campo.

Violações devem ser classificadas por severidade para priorização operacional. Na Figura 4.5 está uma referência visual relacionada ao monitoramento dessas violações.

Figura 4.5: Métrica e monitoramento de violações de qualidade.

4.3.5 Potencial Analítico e Aplicações em IA

A seção avalia a adequação do conjunto de dados para análises estatísticas, operacionais e aplicações de IA. São apresentadas análises dos principais campos da base e suas possibilidades de integração com modelos de IA nas próximas etapas do projeto. Em outras palavras, descreve-se como os modelos de IA poderão ser alimentados a partir desses campos, uma vez que estejam representados e instanciados em grafos correspondentes à ontologia do domínio. A Figura 4.6 fornece uma referência visual relacionada à aplicação de IA.

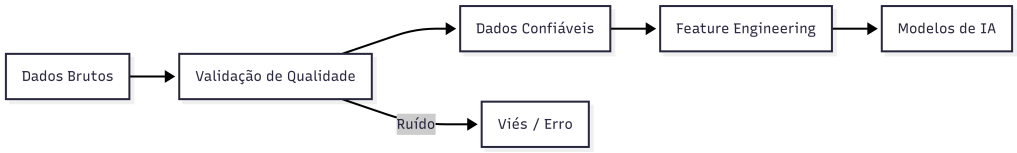

Figura 4.6: Contexto de IA e analytics.

4.3.5.1 Adequação para Estatística

O dataset apresenta alto potencial devido à riqueza temporal e granularidade por movimento. Pontos fortes incluem múltiplos timestamps e campos categóricos padronizados. Riscos envolvem inconsistências temporais e campos textuais livres. É adequado para estatística desde que filtros de qualidade sejam aplicados.

É viável gerar KPIs de pontualidade, atraso médio e volume de movimentos. Esses indicadores dependem de timestamps consistentes, aeroportos válidos e completude mínima.

4.3.5.2 Adequação para Machine Learning

Aplicações possíveis incluem predição de atraso, detecção de anomalias, clusterização de padrões e análise de carga. Riscos incluem rótulos ruidosos, explosão de cardinalidade e data leakage.

4.3.5.3 Modelos Temporais e Séries de Tempo

A presença de eventos ao longo do tempo permite análise de tendência, mudança de regime e previsão agregada. É necessário garantir ordenação temporal consistente, ausência de timestamps futuros e definição clara de granularidade.

4.3.5.4 Integração com Ontologias e Grafos

O dataset é adequado para grafos de conhecimento e inferência semântica, desde que validade e semântica estejam garantidas.

4.4 ERA5

O dataset contém variáveis atmosféricas físicas contínuas, incluindo vento, temperatura, pressão, umidade, precipitação, instabilidade convectiva e parâmetros dinâmicos. Seu uso é relevante para modelagem preditiva de atraso aeronáutico, análise de turbulência e vento em altitude, estudo de instabilidade atmosférica, integração com dados de voo e com a base METAR, além de modelagem espaço-temporal com IA. O foco da avaliação é garantir coerência física, integridade estrutural e estabilidade temporal.

4.4.1 Visão Geral dos Dados

Cada registro representa uma observação ou estimativa atmosférica para um instante específico (`time`), uma posição geográfica (`lat`, `lon`), uma altitude (`geoaltitude`) e múltiplas variáveis físicas associadas. Trata-se de uma reanálise meteorológica (modelada e assimilada), composta por dados numéricos contínuos com dimensão espacial (latitude e longitude), dimensão vertical (altitude ou geopotencial) e dimensão temporal (timestamp).

A estrutura conceitual pode ser agrupada em: (1) dimensão espaço-temporal, (2) dinâmica do vento, (3) temperatura e umidade, (4) pressão atmosférica, (5) precipitação, (6) nuvens, (7) instabilidade atmosférica e (8) parâmetros dinâmicos avançados.

4.4.2 Dicionário de Dados

4.4.2.1 Dimensão Espaço-Temporal

Os campos `time`, `lat`, `lon` e `geoaltitude` definem o contexto espaço-temporal da observação. O campo `geopotential` representa o geopotencial associado à altitude.

4.4.2.2 Vento

A dinâmica do vento é representada por `u_component_of_wind` e `v_component_of_wind` (componentes zonal e meridional), além das componentes a 10 metros `10m_u_component_of_wind` e `10m_v_component_of_wind`. A rajada é registrada em `instantaneous_10m_wind_gust`, e a velocidade vertical em `vertical_velocity`.

4.4.2.3 Temperatura e Umidade

A temperatura atmosférica é registrada em `temperature`. A umidade é representada por `specific_humidity` e o ponto de orvalho a 2 metros por `2m_dewpoint_temperature`.

4.4.2.4 Pressão

Os campos `surface_pressure` e `mean_sea_level_pressure` descrevem pressão na superfície e ao nível do mar, respectivamente.

4.4.2.5 Nuvens

A cobertura de nuvens é representada por `total_cloud_cover`, `fraction_of_cloud_cover`, `low_cloud_cover`, `medium_cloud_cover` e `high_cloud_cover`. Alturas relevantes são
`cloud_base_height` e `boundary_layer_height`. O conteúdo de água em nuvem é descrito
`specific_cloud_ice_water_content` e `specific_cloud_liquid_water_content`.

4.4.2.6 Precipitação

A precipitação total é registrada em `total_precipitation`, com decomposição em `convective_precipitation` e `large_scale_precipitation`.

4.4.2.7 Instabilidade Atmosférica

A instabilidade é caracterizada por `convective_available_potential_energy` (*Convective Available Potential Energy* (CAPE)), `convective_inhibition` (*Convective Inhibition* (CIN)), `k_index` e `total_totals_index`.

4.4.2.8 Parâmetros Dinâmicos Avançados

Os campos `potential_vorticity` e `zero_degree_level` representam, respectivamente, a vorticidade potencial e a altitude do nível de $0\,^{\circ}$C. O campo `dt_insercao` registra a data de ingestão no banco.

4.4.3 Dimensões de Qualidade Avaliadas

O dataset ERA5 exige avaliação sob dimensões estruturais, físicas e espaço-temporais, pois representa campos atmosféricos contínuos. As dimensões avaliadas são completude, validade física, consistência espaço-temporal e conformidade dinâmica.

4.4.3.1 Completude

Campos críticos incluem `time`, `lat`, `lon`, `temperature`, `u_component_of_wind`, `v_component_of_wind` e `surface_pressure`. A ausência compromete a modelagem atmosférica.

4.4.3.2 Validade Física

Os valores devem respeitar limites plausíveis: temperatura dentro de faixa atmosférica ($-100°C$ a $+60°C$), pressão positiva, umidade específica ≥ 0, CAPE ≥ 0, e cobertura de nuvens entre 0 e 1. O ponto de orvalho deve ser menor ou igual à temperatura.

4.4.3.3 Consistência Espaço-Temporal

O `time` deve ser monotônico crescente, sem saltos abruptos incompatíveis com a resolução esperada. As coordenadas `lat` e `lon` devem estar nos intervalos $[-90, 90]$ e $[-180, 180]$.

4.4.3.4 Conformidade Dinâmica

As componentes u/v devem gerar magnitude coerente, a `vertical_velocity` deve ser fisicamente plausível, CAPE elevado deve estar associado a parâmetros convectivos consistentes, e o `zero_degree_level` deve ser coerente com o perfil térmico.

4.4.4 Regras e Métricas de Qualidade

As regras asseguram integridade estrutural, coerência física e estabilidade espaço-temporal. Devido à natureza contínua e tridimensional do dataset, as validações incluem restrições físicas e dinâmicas.

4.4.4.1 Regras Básicas

Devem estar preenchidos `time`, `lat`, `lon`, `temperature`, `u_component_of_wind`, `v_component_of_wind` e `surface_pressure`. O campo `time` deve ser timestamp válido; `lat` e `lon` devem respeitar limites geográficos; `geoaltitude` deve ser ≥ 0; e as variáveis físicas devem ser numéricas. Não deve haver duplicidade para a combinação `time` + `lat` + `lon` + `geoaltitude`.

4.4.4.2 Regras Físicas

Temperatura deve permanecer em faixa plausível e o ponto de orvalho deve ser menor ou igual à temperatura. `surface_pressure` deve ser positiva e `mean_sea_level_pressure` deve permanecer em faixa plausível (ex.: 850–1100 hPa). A pressão deve diminuir com aumento de altitude. Umidade específica deve ser ≥ 0. A magnitude do

vento $\sqrt{u^2 + v^2}$ deve ser coerente e a rajada a 10 m deve ser maior ou igual ao vento médio quando aplicável. Cobertura total de nuvens deve estar em $[0, 1]$, e as frações de nuvem baixa, média e alta devem ser coerentes com a cobertura total. Alturas de base de nuvem e camada limite devem ser ≥ 0. CAPE deve ser ≥ 0 e CIN tipicamente ≤ 0.

4.4.4.3 Regras Espaço-Temporais e Dinâmicas

O tempo deve ser monotônico crescente e sem lacunas abruptas incompatíveis com a resolução. Variações espaciais muito abruptas entre células adjacentes devem ser analisadas. Temperatura e pressão devem decrescer com altitude em condições normais, e o `zero_degree_level` deve ser coerente com o perfil térmico. Eventos de alta CAPE devem coincidir com instabilidade convectiva plausível, e a `potential_vorticity` deve respeitar limites físicos. Para precipitação, `total_precipitation` deve ser ≥ 0, e `convective_precipitation` e `large_scale_precipitation` devem ser $\leq$ total.

4.4.4.4 Indicadores e Métricas

O ERA5 representa campos atmosféricos contínuos no espaço e no tempo. A qualidade deve ser monitorada sob perspectivas estruturais, físicas e espaço-temporais. A degradação pode ocorrer por erro de interpolação, artefatos numéricos, inconsistência vertical, falhas de ingestão ou recortes espaciais incorretos.

4.4.4.5 Indicadores de Completude

O indicador de completude estrutural mede a proporção de registros válidos em relação ao total, considerando o preenchimento dos campos críticos de espaço, tempo e variáveis meteorológicas essenciais. A completude estrutural pode ser medida por:

$$completude\_estrutural = \frac{registros\_validos}{total\_registros}$$

considerando preenchimento de `time`, `lat`, `lon`, `geoaltitude`, `temperature`, `u_component_of_wind`, `v_component_of_wind` e `surface_pressure`.

4.4.4.6 Indicadores de Validade Física

Avaliar percentuais de valores fora de faixa física para `temperature`, `surface_pressure`, `specific_humidity`, `total_cloud_cover` e `convective_available_potential_energy`. Monitorar violações verticais onde pressão ou temperatura não variam corretamente com altitude e incoerências no `zero_degree_level`.

4.4.5 Potencial Analítico e Aplicações em IA

Diferentemente de dados observacionais pontuais, o ERA5 representa uma reconstrução global baseada em assimilação de múltiplas fontes, permitindo análise tridimensional de vento, temperatura, instabilidade e precipitação. Sua integração com dados operacionais amplia o potencial preditivo de modelos de IA.

4.4.5.1 Adequação para Análise Estatística

O dataset é adequado para análise de padrões regionais de vento em altitude, instabilidade convectiva, sazonalidade, gradiente térmico vertical e regimes de precipitação, com agregações por região, altitude e período. Features potenciais incluem magnitude do vento, `vertical_velocity`, CAPE, `total_precipitation`, `total_cloud_cover`, `zero_degree_level` e `boundary_layer_height`.

4.4.5.2 Integração com Dados Operacionais

O ERA5 pode ser combinado com METAR, *Notice to Airmen* (NOTAM), dados de voo, dados de aeroportos e dados de tráfego aéreo. Essa integração permite modelos híbridos que combinam precisão observacional, cobertura espacial do ERA5 e contexto operacional.

4.4.5.3 Exemplos de Aplicações em IA

Os dados do ERA5 permitem estimativas de turbulência em rota (*Clear Air Turbulence* (CAT), detecção de condições de gelo, conteúdo de água líquida e nível de zero grau, previsão de wind shear a partir de gradientes de vento e rajadas, previsão de tempestades com CAPE, CIN, `vertical_velocity` e umidade, e otimização de rotas.

4.5 INMET

O dataset contém observações meteorológicas de superfície, incluindo temperatura, umidade, vento, precipitação e pressão atmosférica, registradas por estações distribuídas pelo território nacional. Esses dados são relevantes para análise climática regional, validação cruzada com METAR, modelagem de impacto meteorológico em aeroportos, treinamento de modelos de Inteligência Artificial e estudos de correlação clima e desempenho operacional.

4.5.1 Visão Geral dos Dados

Cada registro representa uma observação meteorológica realizada por uma estação específica em um instante temporal. Trata-se de um dataset observacional terrestre, composto majoritariamente por variáveis numéricas contínuas, com dimensão espacial definida pelas coordenadas da estação e dimensão temporal definida pela coluna `data`. A cobertura é nacional, com distribuição de estações por todas as regiões e unidades federativas.

A estrutura conceitual pode ser entendida como um conjunto de campos de identificação da estação, dimensão temporal e variáveis meteorológicas de temperatura, umidade, vento, precipitação e pressão atmosférica.

4.5.2 Dicionário de Dados

A identificação da estação é representada por `estacao`, `codigo`, `regiao` e `uf`, permitindo vincular cada observação à sua origem geográfica e administrativa. A localização é dada por `latitude`, `longitude` e `altitude`, que definem a posição e a elevação da estação.

A dimensão temporal é representada pela coluna `data`, que registra o timestamp da observação. As variáveis meteorológicas incluem `temperatura`, `temp_max`, `temp_min`, `temperatura_orvalho`, `temperatura_orvalho_max` e `temperatura_orvalho_min`. Umidade é representada por `umidade`, `umidade_max` e `umidade_min`. Vento é descrito por `dir_vento`, `vel_vento` e `rajada_vento`. Precipitação é registrada em `chuva`. Pressão atmosférica é representada por `pressao`, `pressao_max` e `pressao_min`. Radiação é registrada em `radiacao`. O campo `dt_insercao` registra a data de ingestão no banco.

4.5.3 Dimensões de Qualidade Avaliadas

O dataset INMET exige validação estrutural e física por representar medições reais de sensores terrestres. As dimensões avaliadas incluem completude, validade física, consistência temporal e consistência espacial. A completude garante presença de campos críticos, a validade física verifica limites plausíveis dos fenômenos, a consistência temporal avalia monotonicidade e intervalos regulares de observação, e a consistência espacial assegura plausibilidade geográfica e coerência regional entre estações.

4.5.3.1 Completude

Campos críticos incluem `estacao`, `codigo`, `data`, `temperatura`, `umidade`, `vel_vento` e `pressao`. A ausência desses campos compromete análises climáticas, validações cruzadas e modelagens operacionais.

4.5.3.2 Validade Física

Os valores devem respeitar limites plausíveis: temperatura em $[-100, +60]$ $°C$, umidade em $[0, 100]$, velocidade do vento ≥ 0, pressão > 0, chuva ≥ 0 e radiação ≥ 0 quando aplicável. Além disso, `temp_max` deve ser maior ou igual a `temp_min`, e `temperatura` deve permanecer entre esses limites.

4.5.3.3 Consistência Temporal

A coluna `data` deve ser monotonicamente crescente por estação, sem repetições de timestamp para a mesma estação. Intervalos entre registros devem ser regulares (por exemplo, 1h, 3h ou 24h), e saltos abruptos ou lacunas excessivas devem ser sinalizados.

4.5.3.4 Consistência Espacial

As coordenadas devem respeitar latitude em $[-90, 90]$, longitude em $[-180, 180]$ e altitude ≥ 0. Estações devem estar em limites geográficos plausíveis e, quando próximas, devem apresentar padrões climáticos compatíveis dentro de variações físicas razoáveis.

4.5.4 Regras e Métricas de Qualidade

As regras de qualidade asseguram integridade estrutural e validade física dos dados do INMET. Campos essenciais devem estar preenchidos (`estacao`, `codigo`, `data`, `temperatura`, `umidade`, `vel_vento`, `pressao`). A tipagem deve ser consistente, com `data` como timestamp válido, coordenadas dentro dos limites geográficos e variáveis físicas numéricas.

Não deve existir duplicidade para a combinação `estacao` + `data`. Para temperatura, `temp_max` deve ser maior ou igual a `temp_min`, e `temperatura` deve estar entre elas quando disponíveis. Para umidade, os valores devem estar em $[0, 100]$ e `temperatura_orvalho` deve ser menor ou igual à `temperatura`. Para vento, `vel_vento` deve ser ≥ 0, `rajada_vento` deve ser maior ou igual à velocidade, e `dir_vento` deve estar em $[0, 360]$. Para precipitação, `chuva` deve ser ≥ 0 e, quando registrada, deve ser coerente com o acumulado do período. Para pressão, `pressao` deve ser positiva e `pressao_max` deve ser maior ou igual a `pressao_min`. A consistência temporal deve garantir monotonicidade e intervalos regulares, e a consistência espacial deve garantir coordenadas válidas e coerência regional.

4.5.5 Potencial Analítico e Aplicações em IA

O dataset INMET é valioso para análise climática e modelagem de impacto meteorológico, especialmente quando combinado com dados aeronáuticos. Sua estrutura de observações pontuais permite validação cruzada com METAR, análise de microclimas regionais, estudos de correlação clima x desempenho de voo e treinamento de modelos de IA para previsão de eventos locais.

Para análise estatística, o dataset é adequado para padrões de temperatura e umidade, regimes de precipitação, identificação de extremos climáticos, análise de vento e correlação com aeroportos próximos. Para modelos preditivos, pode ser usado para prever condições meteorológicas locais, estimar o impacto operacional em aeroportos, treinar modelos de previsão de atrasos, identificar padrões de risco meteorológico e validar modelos baseados em dados observacionais.

A integração com METAR, NOTAM, dados de voo, dados de aeroportos e dados de tráfego aéreo permite modelos híbridos que combinam precisão observacional com cobertura espacial, preveem atrasos e ampliam a robustez analítica em ambientes operacionais.

4.6 NCEP Pressão

Esses dados são relevantes para análise sinótica, estudo de sistemas de alta e baixa pressão, modelagem de dinâmica atmosférica, integração com ERA5 e análise de impacto meteorológico em operações aéreas. Cada registro representa um ponto da grade atmosférica em determinado instante temporal.

4.6.1 Visão Geral dos Dados

Trata-se de um dataset modelado em grade contínua, com dimensão espacial definida por `latitude` e `longitude`, dimensão vertical representada por `pressure_level` e dimensão temporal em `timestamp`.

A estrutura conceitual pode ser agrupada em: (1) dimensão espaço-temporal, (2) pressão atmosférica, (3) variáveis derivadas e (4) estrutura vertical por níveis de pressão.

4.6.2 Dicionário de Dados

4.6.2.1 Dimensão Espaço-Temporal

Os campos `timestamp`, `latitude` e `longitude` definem o contexto espaço-temporal da observação. O campo `pressure_level` representa o nível de pressão ao qual as variáveis se referem.

4.6.2.2 Pressão e Geopotencial

A variável `hgt` representa altura geopotencial e está diretamente relacionada à pressão e temperatura. A variável `air` representa temperatura do ar no nível de pressão.

4.6.2.3 Vento e Dinâmica

Os componentes do vento são `uwnd` e `vwnd`. A variável `omega` representa a velocidade vertical em coordenadas de pressão (movimento vertical do ar).

4.6.2.4 Umidade

A umidade relativa é representada por `rhum`.

4.6.2.5 Controle de Ingestão

O campo `dt_insercao` registra o timestamp de ingestão no banco.

4.6.3 Dimensões de Qualidade Avaliadas

O dataset NCEP exige validação estrutural, física e espaço-temporal, pois representa campos atmosféricos contínuos.

As dimensões avaliadas são completude, validade física, consistência espaço-temporal e coerência sinótica.

4.6.3.1 Completude

Campos críticos incluem `timestamp`, `latitude`, `longitude`, `pressure_level`, `air` e `uwnd`/`vwnd`. A ausência compromete análises sinóticas e modelagem.

4.6.3.2 Validade Física

Valores devem respeitar limites físicos plausíveis: pressão em faixa típica, latitude em $[-90, 90]$ e longitude em $[-180, 180]$. Temperatura deve estar em faixa plausível para o nível de pressão e altitude correspondente. Umidade relativa deve estar em $[0, 100]$.

4.6.3.3 Consistência Espaço-Temporal

As variáveis devem variar no espaço e no tempo. Variações bruscas devem ser investigadas, e a coerência com dados vizinhos (como ERA5) deve ser monitorada.

4.6.3.4 Coerência Sinótica

Sistemas de alta e baixa pressão devem ser fisicamente plausíveis, e os gradientes de pressão devem ser consistentes com os campos de vento disponíveis.

4.6.4 Regras e Métricas de Qualidade

4.6.4.1 Regras Básicas

Devem estar preenchidos `timestamp`, `latitude`, `longitude`, `pressure_level` e `air`. O campo `timestamp` deve ser timestamp válido, `latitude` e `longitude` devem respeitar limites geográficos, `pressure_level` deve estar em faixa plausível e variáveis físicas devem ser numéricas. Não deve existir duplicidade para a combinação `timestamp` + `latitude` + `longitude` + `pressure_level`.

4.6.4.2 Regras Físicas

A pressão deve permanecer em faixa típica. A altura geopotencial `hgt` deve ser consistente com o nível de pressão. A temperatura `air` deve ser compatível com altitude e pressão. A umidade `rhum` deve permanecer em $[0, 100]$.

4.6.4.3 Regras de Consistência Espaço-Temporal

A variação de pressão entre pontos adjacentes deve permanecer dentro de limites típicos e as variações temporais devem ser plausíveis para a resolução dos dados. Diferenças sistemáticas em relação ao ERA5 devem ser monitoradas.

4.6.4.4 Regras de Coerência Sinótica

Centros de alta e baixa pressão devem ser coerentes com padrões atmosféricos conhecidos, e gradientes devem respeitar a dinâmica atmosférica.

4.6.4.5 Métricas de Qualidade

As métricas avaliam completude, validade física, consistência espaço-temporal e coerência sinótica. Monitorar percentuais de valores válidos para `pressure_level`, `air`, `hgt` e `rhum`.

4.6.4.6 Consistência Espaço-Temporal

Avaliar percentuais de variação suave, coerência com vizinhos e plausibilidade de sistemas sinóticos.

4.6.5 Potencial Analítico e Aplicações em IA

O dataset NCEP é valioso para análise sinótica e modelagem de dinâmica atmosférica, especialmente quando combinado com ERA5. Sua estrutura de grade contínua permite análise de padrões de pressão, identificação de sistemas de alta e baixa pressão, previsão de condições meteorológicas e integração com dados de voo. Também pode treinar modelos de IA para previsão de demanda, detecção de congestionamentos, otimização de rotas e previsão de atrasos.

4.6.5.1 Adequação para Análise Estatística

O dataset é adequado para análise de padrões de pressão, estudo de sistemas de alta e baixa pressão, correlação com dados de aeroportos próximos e análise de rotas reais.

4.6.5.2 Adequação para Modelos Preditivos

O NCEP pode ser usado para prever demanda de tráfego, detectar congestionamentos, prever atrasos de voo, identificar padrões de risco e treinar modelos de IA para previsão de demanda, detecção de conflito, otimização de rotas e previsão de atrasos.

4.6.5.3 Integração com Bases de Dados Meteorológicas

A combinação com ERA5, NOTAM, dados de voo, dados de aeroportos e dados de tráfego aéreo permite criar modelos híbridos que combinam precisão observacional com cobertura espacial, preveem atrasos e otimizam rotas, melhorando a segurança do tráfego aéreo.

Recomenda-se padronizar as unidades de medida, implementar validação automática pós-ingestão (near-real-time), adicionar metadados de cobertura e monitorar a degradação sistêmica. Para IA, recomenda-se integrar com ERA5, validar com dados locais, monitorar variações sazonais e verificar sistemas de alta e baixa pressão.

4.7 NCEP Superfície

O dataset contém campos meteorológicos modelados ao nível da superfície, organizados em grade espacial e dimensão temporal contínua.

Esses dados são relevantes para análise sinótica de superfície, estudo de sistemas de alta e baixa pressão, modelagem de vento próximo ao solo, integração com METAR e INMET, além de estudos de impacto meteorológico nas operações aéreas.

4.7.1 Visão Geral dos Dados

Cada registro representa um ponto da grade atmosférica ao nível da superfície em determinado instante temporal. Trata-se de um dataset modelado em grade de 2,5°, com dimensão espacial definida por `latitude` e `longitude`, dimensão temporal em `timestamp` e variáveis incluindo pressão, temperatura e vento.

A estrutura conceitual pode ser agrupada em: (1) dimensão espaço-temporal, (2) temperatura de superfície, (3) pressão ao nível do mar, (4) vento de superfície, (5) precipitação e (6) variáveis derivadas.

4.7.2 Dicionário de Dados

4.7.2.1 Dimensão Espaço-Temporal

Os campos `timestamp`, `latitude` e `longitude` definem o contexto espaço-temporal do ponto de grade.

4.7.2.2 Temperatura

O campo `air_2m` representa a temperatura do ar a 2 metros de altura, medida padrão para observações de superfície.

4.7.2.3 Pressão

O campo `pres_sfc` representa a pressão ao nível da superfície. O campo `mslp` (mean sea level pressure) representa a pressão reduzida ao nível médio do mar, útil para análise sinótica.

4.7.2.4 Vento

Os componentes `uwnd_10m` e `vwnd_10m` representam as componentes zonal e meridional do vento a 10 metros de altura, medição padrão para vento de superfície.

4.7.2.5 Controle de Ingestão

O campo `dt_insercao` registra o timestamp de ingestão dos dados no banco.

4.7.3 Dimensões de Qualidade Avaliadas

O dataset NCEP Superfície exige validação estrutural, física e espaço-temporal, pois representa campos atmosféricos modelados próximos ao solo.

As dimensões avaliadas são completude, validade física, consistência espaço-temporal e coerência sinótica.

4.7.3.1 Completude

Campos críticos incluem `timestamp`, `latitude`, `longitude`, `air_2m`, `mslp`, `uwnd_10m` e `vwnd_10m`. A ausência compromete análises sinóticas e modelagem de superfície.

4.7.3.2 Validade Física

Valores devem respeitar limites físicos: temperatura em $[-100, 50]$ °C, pressão em $[800, 1100]$ hPa, velocidade de vento em $[0, 100]$ m/s, latitude em $[-90, 90]$ e longitude em $[-180, 180]$.

4.7.3.3 Consistência Espaço-Temporal

Os campos devem variar suavemente no espaço e no tempo. Variações bruscas devem ser investigadas e a coerência com dados vizinhos (METAR, Instituto Nacional de Meteorologia (INMET), ERA5) deve ser monitorada.

4.7.3.4 Coerência Sinótica

Sistemas de alta e baixa pressão devem ser fisicamente plausíveis e os gradientes de pressão devem ser consistentes com os campos de vento.

4.7.4 Regras e Métricas de Qualidade

4.7.4.1 Regras Básicas

Devem estar preenchidos `timestamp`, `latitude`, `longitude`, `air_2m`, `mslp`, `uwnd_10m` e `vwnd_10m`. O campo `timestamp` deve ser timestamp válido, `latitude` e `longitude` devem respeitar limites geográficos, temperatura deve estar em faixa plausível e componentes de vento devem ser numéricas. Não deve existir duplicidade para a combinação `timestamp` + `latitude` + `longitude`.

4.7.4.2 Regras Físicas

Temperatura deve estar em $[-100, 50]\,°C$, com variação diária menor que $30\,°C$ e variação sazonal menor que $50\,°C$. Pressão deve estar em $[800, 1100]$ hPa, com variação diária menor que 50 hPa e sazonal menor que 100 hPa. Magnitude do vento deve estar em $[0, 100]$ m/s, com variação diária menor que 50 m/s e sazonal menor que 80 m/s.

4.7.4.3 Regras de Consistência Espaço-Temporal

Variação de temperatura entre pontos adjacentes deve ser $\leq 10\,°C$, variação de pressão ≤ 20 hPa e variação de vento ≤ 20 m/s. Diferenças para METAR e INMET devem ser $\leq 5\,°C$, diferenças para ERA5 $\leq 2\,°C$. Centros de alta e baixa pressão devem ser coerentes com padrões conhecidos.

4.7.4.4 Regras de Conformidade

O dataset deve cobrir a região de interesse com consistência geográfica e resolução temporal adequada para análise planejada.

4.7.4.5 Métricas de Qualidade

As métricas avaliam completude, validade física, consistência espaço-temporal e coerência sinótica.

4.7.4.6 Validade Física

Monitorar percentuais de valores válidos para temperatura, pressão, vento e componentes do vento.

4.7.4.7 Consistência Espaço-Temporal

Avaliar percentuais de variação suave, coerência com vizinhos e plausibilidade de sistemas sinóticos.

4.7.5 Potencial Analítico e Aplicações em IA

O dataset NCEP Superfície é valioso para análise sinótica e modelagem de dinâmica atmosférica, especialmente quando combinado com METAR, INMET e ERA5. Sua estrutura de grade contínua permite análise de padrões de superfície, identificação de sistemas de alta e baixa pressão, previsão de condições meteorológicas e integração com dados de voo. Também pode treinar modelos de IA para previsão de demanda, detecção de congestionamentos, otimização de rotas e previsão de atrasos.

4.7.5.1 Análise Estatística

O dataset é adequado para análise de padrões de superfície, estudo de sistemas de alta e baixa pressão, correlação com dados de aeroportos próximos e análise de rotas reais.

4.7.5.2 Modelos Preditivos

O NCEP Superfície pode ser usado para prever demanda de tráfego, detectar congestionamentos, prever atrasos de voo, identificar padrões de risco e treinar modelos de IA para previsão de demanda, detecção de conflito, otimização de rotas e previsão de atrasos.

4.7.5.3 Integração com Bases de Dados Meteorológicas

A combinação com METAR, INMET, ERA5, NOTAM, dados de voo, dados de aeroportos e dados de tráfego aéreo permite inúmeras aplicações, como previsão de atrasos e otimização de rotas. Recomenda-se integrar com METAR, INMET e ERA5, validar padrões sinóticos, correlacionar com dados de voo e usar para previsão de demanda. Para IA, recomenda-se usar como base de modelos preditivos, integrar com dados de voo, validar padrões sinóticos e treinar modelos para previsão de atrasos e congestionamentos e otimização de rotas. Para monitoramento, recomenda-se monitorar a completude estrutural, validade física, consistência espaço-temporal e a coerência sinótica de forma contínua.

4.8 METAR

O dataset contém observações horárias ou sub-horárias relacionadas a temperatura, vento, visibilidade, pressão, cobertura de nuvens e fenômenos meteorológicos.

Esses dados são essenciais para previsão de atraso, modelagem de risco operacional, análise de impacto climático na malha aérea e integração com modelos de Inteligência Artificial aplicados à aviação. O foco da avaliação é garantir integridade estrutural, coerência física e consistência temporal das observações.

4.8.1 Visão Geral dos Dados

Cada registro representa uma observação meteorológica associada a uma estação (aeródromo) em um instante específico. Trata-se de dados com granularidade por estação e timestamp, tipicamente horária, contendo dados contínuos (numéricos), categóricos (texto) e temporais.

A estrutura conceitual pode ser agrupada em: (1) identificação da estação, (2) temporalidade da observação, (3) temperatura e umidade, (4) vento, (5) pressão, (6) visibilidade, (7) nuvens, (8) precipitação e gelo, (9) eventos especiais e (10) mensagem METAR bruta.

4.8.2 Dicionário de Dados

4.8.2.1 Identificação da Estação

O campo `station` identifica a estação meteorológica com 3 ou 4 caracteres, geralmente correspondendo ao código ICAO do aeródromo. O campo `valid` registra o timestamp da observação meteorológica.

4.8.2.2 Temperatura e Umidade

Os campos `tmpc` e `tmpf` representam a temperatura do ar em Celsius e Fahrenheit, respectivamente. Os campos `dwpc` e `dwpf` representam a temperatura do ponto de orvalho. O campo `relh` registra a umidade relativa em percentual, e `feel` representa a temperatura aparente (wind chill ou heat index).

4.8.2.3 Vento

O campo `drct` representa a direção do vento em graus a partir do norte verdadeiro. Os campos `sknt` e `sped` representam a velocidade média do vento em nós. Os campos `gust` e `gustmph` representam a rajada de vento em nós e milhas por hora. Os campos `peak_wind_gust`, `peak_wind_drct` e `peak_wind_time` registram a rajada máxima, sua direção e horário.

4.8.2.4 Pressão Atmosférica

O campo `alti` representa a pressão altimétrica em polegadas de mercúrio (inHg). O campo `mslp` representa a pressão ao nível médio do mar em milibares.

4.8.2.5 Visibilidade e Fenômenos

O campo `vsby` registra a visibilidade horizontal em milhas. O campo `wxcodes` contém códigos de fenômenos meteorológicos presentes (ex.: chuva, neblina, trovoada).

4.8.2.6 Precipitação

Os campos `p01i` e `p01m` representam a precipitação acumulada na última hora em polegadas e milímetros, respectivamente.

4.8.2.7 Cobertura de Nuvens

Os campos `skyc1` a `skyc4` representam a cobertura de nuvens por camada (ex.: FEW, SCT, BKN, OVC). Os campos `skyl1` a `skyl4` representam a altitude das camadas de nuvens em pés.

4.8.2.8 Gelo e Neve

Os campos `ice_accretion_1hr`, `ice_accretion_3hr` e `ice_accretion_6hr` registram o acúmulo de gelo no período em polegadas. O campo `snowdepth` registra a profundidade de neve acumulada.

4.8.2.9 Campo Original

O campo `metar` contém a observação bruta no formato textual METAR, permitindo reconstrução e auditoria da decodificação estruturada. O campo `dt_insercao` registra o timestamp de ingestão no banco.

4.8.3 Dimensões de Qualidade Avaliadas

O dataset METAR exige avaliação sob dimensões estruturais e físicas, pois representa observações atmosféricas sujeitas a restrições naturais.

As dimensões avaliadas são completude, validade física, consistência temporal e conformidade meteorológica.

4.8.3.1 Completude

Campos críticos incluem `station`, `valid`, `tmpc` ou `tmpf`, `drct`, `sknt`, `vsby` e `alti`. A ausência desses campos compromete análises operacionais.

4.8.3.2 Validade Física

Valores devem respeitar limites plausíveis: temperatura em $[-80, +60]$ °C, umidade relativa em $[0, 100]\%$, direção do vento em $[0, 360]°$, velocidade do vento ≥ 0, visibilidade ≥ 0 e pressão dentro de faixa plausível.

4.8.3.3 Consistência Temporal

O campo `valid` deve ser crescente por estação e `peak_wind_time` deve ser $\leq$ `valid`. Não devem existir múltiplos registros idênticos por estação e timestamp.

4.8.3.4 Conformidade Meteorológica

O ponto de orvalho não pode exceder a temperatura, a velocidade de rajada deve ser $\geq$ velocidade média do vento, e se `wxcodes` indicar precipitação, `p01i` deve ser positivo. A altitude das camadas de nuvem deve ser crescente.

4.8.4 Regras e Métricas de Qualidade

As regras garantem integridade estrutural, coerência física e consistência meteorológica. As validações estão organizadas em: regras básicas, regras físicas e meteorológicas, e regras cross-field e temporais.

4.8.4.1 Regras Básicas

Devem estar preenchidos `station`, `valid`, `tmpc` ou `tmpf`, `drct`, `sknt`, `vsby` e `alti`. Os valores devem ser tipados corretamente: `valid` como timestamp, campos numéricos como números, `skyc*` como texto. A padronização exige que `station` possua 3 ou 4 caracteres alfanuméricos, `drct` esteja entre 0 e 360°, `relh` entre 0 e 100%, e velocidades de vento sejam ≥ 0.

4.8.4.2 Regras Físicas e Meteorológicas

O ponto de orvalho deve ser $\leq$ temperatura em ambas as unidades (`dwpc` $\leq$ `tmpc`). A umidade relativa deve estar em $[0, 100]\%$. Pressão altimétrica deve estar em faixa plausível (ex.: 25–32 inHg) e `mslp` em 850–1100 mb. A rajada deve ser $\geq$ velocidade média do vento, `peak_wind_gust` $\geq$ `gust`, e `peak_wind_time` $\leq$ `valid`. Visibilidade deve ser ≥ 0, e quando `wxcodes` indica neblina (FG), `vsby` deve ser reduzida. Alturas de camadas de nuvem devem ser crescentes (`skyl1` $\leq$ `skyl2` $\leq$ `skyl3` $\leq$ `skyl4`) e `skyc*` deve pertencer ao conjunto {FEW, SCT, BKN, OVC, CLR}.

4.8.4.3 Regras Cross-field e Temporais

Não deve existir mais de um registro com `station` + `valid` (duplicidade). Se `wxcodes` indicar chuva (RA), neve (SN) ou tempestade (TS), `p01i` deve ser positivo. Se acúmulo de gelo for registrado, temperatura deve estar próxima ou abaixo de 0°C. O campo `valid` deve ser crescente por `station`, e quando `metar` estiver presente, valores estruturados devem ser coerentes com a mensagem textual.

4.8.4.4 Classificação de Severidade

Violações são classificadas em: alta severidade (ex.: dew point > temperatura, inversão temporal, pressão fora de faixa extrema), média severidade (ex.: rajada incoerente, camadas de nuvem fora de ordem) e baixa severidade (ex.: ausência de campos não críticos, divergência leve entre unidades).

4.8.4.5 Indicadores e Métricas

A qualidade deve ser monitorada continuamente por representar observações utilizadas em decisões operacionais críticas.

4.8.4.6 Indicadores de Validade Física

Violação de faixa física mede percentuais de valores fora de limites (ex.: `tmpc` fora de $[-80, +60]$ °C, `relh` fora de $[0, 100]\%$, `drct` fora de $[0, 360]°$, `sknt` < 0). Violação cross-field mede registros onde `dew_point` > `temperatura`, `gust` < `sknt` ou `skyl` não crescente.

4.8.4.7 Indicadores Temporais

Duplicidade temporal mede proporção de registros duplicados considerando `station` + `valid`. Intervalo de observação mede diferença entre observações consecutivas, sinalizando desvios do intervalo esperado.

4.8.4.8 Indicadores Meteorológicos Derivados

Intensidade de vento avalia distribuição de `sknt`, `gust` e `peak_wind_gust`. Condições de baixa visibilidade medem percentual com `vsby` abaixo de limiar operacional.

4.8.5 Potencial Analítico e Aplicações em IA

O dataset METAR representa observações meteorológicas estruturadas e temporalmente indexadas por estação, constituindo base crítica para análises operacionais e IA no domínio aeronáutico. Por refletir condições atmosféricas reais, possui elevado valor explicativo quando integrado a dados de voo, permitindo modelagem causal entre condições meteorológicas e desempenho operacional.

4.8.5.1 Adequação para Análise Estatística

O dataset é adequado para análise de frequência de eventos meteorológicos severos, distribuição de vento, estudo de visibilidade crítica, monitoramento de pressão e identificação de sazonalidade climática. A granularidade temporal permite análises por hora, dia ou estação.

4.8.5.2 *Aplicações de Machine Learning*

Quando integrado a dados de voo, permite modelar atraso de decolagem, atraso de chegada, risco de cancelamento e tempo adicional de táxi. Features potenciais incluem velocidade e rajada de vento, direção, visibilidade, cobertura de nuvens, fenômenos severos e gelo acumulado. Modelos podem classificar condições em normal, atenção, risco elevado e risco crítico. Técnicas como Isolation Forest e séries temporais com LSTM detectam anomalias meteorológicas e leituras anômalas de sensores.

4.8.5.3 *Integração com Dados Operacionais*

O potencial máximo emerge quando integrado com dados de eventos operacionais, regulatórios e de malha aérea. Isso permite modelagem causal clima atraso, quantificação de impacto meteorológico por aeroporto e estimativa de custo operacional associado a eventos severos.

4.8.5.4 *Riscos para Inteligência Artificial*

Erros de medição geram padrões artificiais, valores fisicamente impossíveis distorcem modelos, intervalos temporais irregulares prejudicam séries temporais e unidades inconsistentes comprometem aprendizado. Modelos treinados sobre dados inconsistentes podem superestimar ou subestimar impacto climático.

4.8.5.5 *Avaliação Consolidada*

Tabela 4.2: Adequação do Dataset METAR para Aplicações

Aplicação	**Adequação**	**Condição**
Análise Meteorológica Descritiva	Muito Alta	Validação física aplicada
Previsão de Atraso	Muito Alta	Integração com dados de voo
Classificação de Risco	Alta	Definição clara de limiares
Detecção de Anomalias	Alta	Histórico consistente
Modelagem de Impacto Operacional	Muito Alta	Integração multi-dataset

4.9 OpenSky

O dataset contém informações dinâmicas de posição, altitude, velocidade e estado da aeronave ao longo do tempo.

Esses dados são essenciais para análise de fluxo aéreo, modelagem de complexidade operacional, estudo de rotas reais, previsão de conflito ou congestionamento e treinamento de modelos de IA baseados em trajetória.

4.9.1 Visão Geral dos Dados

Cada registro representa o estado instantâneo de uma aeronave em determinado timestamp. Trata-se de dados dinâmicos de rastreamento ADS-B, com granularidade no estado da aeronave por timestamp, dimensão espacial em latitude, longitude e altitude, dimensão temporal contínua com frequência elevada (segundos).

A estrutura conceitual pode ser agrupada em: (1) identificação da aeronave, (2) dimensão temporal, (3) posição espacial, (4) movimento (velocidade e direção) e (5) estado operacional.

4.9.2 Dicionário de Dados

4.9.2.1 Identificação da Aeronave

O campo `icao24` é o identificador único hexadecimal da aeronave. O campo `callsign` contém a identificação do voo. O campo `indicat` pode conter informações adicionais de identificação.

4.9.2.2 Temporalidade

Os campos `time` e `hour` registram o timestamp da observação e a hora da observação, respectivamente. Os campos `lastposupdate` e `lastcontact` registram a última atualização de posição e último contato, respectivamente.

4.9.2.3 Posição

Os campos `lat` e `lon` representam a latitude e longitude da aeronave. Os campos `baroaltitude` e `geoaltitude` representam altitude barométrica e geométrica.

4.9.2.4 Movimento

O campo `velocity` representa a velocidade da aeronave. O campo `heading` (ou `true_track`) representa a direção do deslocamento. O campo `vertrate` representa a taxa de variação vertical (subida ou descida).

4.9.2.5 Estado Operacional

O campo `onground` indica se a aeronave está em solo (1) ou em voo (0). O campo `squawk` contém o código transponder SSR. O campo `spi` indica se o sinal de identificação especial está ativo (1 para emergência). O campo `alert` indica se há alerta de emergência. Os campos `departure` e `arrival` registram os aeródromos de partida e chegada.

4.9.2.6 Controle de Ingestão

O campo `dt_insercao` registra o timestamp de ingestão no banco.

4.9.3 Dimensões de Qualidade Avaliadas

O dataset OpenSky exige avaliação estrutural, espacial e dinâmica, pois representa movimento real de aeronaves. As dimensões avaliadas são completude, validade física, consistência espaço-temporal e conformidade aeronáutica.

4.9.3.1 Completude

Campos críticos incluem `icao24`, `time`, `lat`, `lon`, `velocity` e `heading`. A ausência compromete rastreamento e análise de fluxo.

4.9.3.2 Validade Física

Valores devem respeitar limites plausíveis: latitude em $[-90, 90]$, longitude em $[-180, 180]$, velocidade ≥ 0, direção em $[0, 360]^\circ$ e altitude ≥ -1000 m.

4.9.3.3 Consistência Espaço-Temporal

As trajetórias devem ser contínuas e saltos bruscos devem ser investigados. A velocidade deve ser compatível com altitude e distância percorrida.

4.9.3.4 Conformidade Aeronáutica

A altitude deve respeitar níveis de voo, a direção deve ser consistente com velocidade e posição, o `squawk` deve ser válido e a origem da posição deve ser confiável.

4.9.4 Regras e Métricas de Qualidade

As regras asseguram integridade estrutural, validade física e consistência dinâmica dos dados.

4.9.4.1 Regras Básicas

Devem estar preenchidos `icao24`, `time`, `lat`, `lon`, `velocity` e `heading`. O campo `time` deve ser timestamp válido, `lat` e `lon` devem respeitar limites geográficos, `velocity` deve ser ≥ 0, `heading` deve estar em $[0, 360]°$ e altitude deve ser ≥ -1000 m. Não deve haver duplicidade para a combinação `icao24` + `time` + `lat` + `lon`.

4.9.4.2 Regras Físicas

Posição deve ter latitude em $[-90, 90]$, longitude em $[-180, 180]$ e altitude ≥ -1000 m, com geométrica $\geq$ barométrica. Movimento deve ter velocidade ≥ 0 e direção em $[0, 360]°$. Estado deve ter `squawk` em faixa válida (0000–7777), `spi` em {0, 1} e `alert` em {0, 1}.

4.9.4.3 Regras de Consistência Espaço-Temporal

Saltos maiores que 100 km devem ser sinalizados, e variações de altitude maiores que 10.000 pés devem ser investigadas. Velocidade deve ser compatível com distância percorrida no intervalo e `vertrate` com mudanças de altitude. Aeronaves em voo devem ter altitude > 1000 pés, em solo devem ter `velocity` aproximadamente 0, e em cruzeiro `vertrate` deve ser aproximadamente 0.

4.9.4.4 Regras de Conformidade Aeronáutica

Altitude deve respeitar níveis de voo (FL) com variações em múltiplos de 1000 pés. O `spi` = 1 indica emergência, e `squawk` = 7500, 7600, 7700 indicam emergência.

4.9.4.5 Métricas de Qualidade

As métricas avaliam completude, validade física, consistência espaço-temporal e conformidade aeronáutica.

4.9.4.6 Validade Física

Monitorar percentuais de `lat`, `lon`, `velocity` e `altitude` válidos.

4.9.4.7 Consistência Espaço-Temporal

Avaliar percentuais de trajetórias contínuas, velocidade plausível e coerência de voo.

4.9.4.8 Conformidade Aeronáutica

Monitorar percentuais de altitude em conformidade, `squawk` válido e origem de posição confiável.

4.9.5 Potencial Analítico e Aplicações em IA

O dataset OpenSky é valioso para análise de fluxo aéreo e modelagem de complexidade operacional, especialmente quando combinado com dados de aeroportos e METAR. Sua estrutura de trajetórias dinâmicas permite análise de padrões de tráfego, identificação de congestionamentos, previsão de atrasos e estudo de rotas reais.

4.9.5.1 Adequação para Análise Estatística

O dataset é adequado para análise de densidade de tráfego, estudo de padrões de altitude e velocidade, identificação de congestionamentos, análise de rotas reais e correlação com dados de aeroportos próximos.

4.9.5.2 Adequação para Modelos Preditivos

O OpenSky pode ser usado para prever demanda de tráfego, detectar congestionamentos, prever atrasos de voo e identificar padrões de risco. Modelos de IA podem ser treinados para previsão de demanda, detecção de conflito, otimização de rotas e previsão de atrasos.

4.9.5.3 Integração com Outros Datasets

A combinação com METAR, NOTAM, dados de voo, dados de aeroportos e dados de tráfego aéreo permite criar modelos híbridos que combinam precisão observacional com cobertura espacial, preveem atrasos de voo, otimizam rotas e melhoram a segurança do tráfego aéreo. Recomenda-se padronizar unidades de medida, implementar validação automática pós-ingestão (near-real-time), adicionar metadados de cobertura e monitorar degradação sistêmica. Para IA, recomenda-se combinar com METAR e NOTAM, criar índices de complexidade de tráfego e desenvolver modelos preditivos de atraso. Para monitoramento, recomenda-se usar gráficos P para violações, com controle por `icao24`, região geográfica e altitude, além de alertas para degradação sistêmica.

4.10 SIROS

O conjunto de dados compreende o ecossistema de informações estruturadas que regem o ciclo de vida das operações aéreas. Ele integra variáveis críticas que abrangem desde a identificação do operador até as especificações técnicas de frota e infraestrutura aeroportuária.

4.10.1 Visão Geral dos Dados

Diferente de uma simples listagem, o dataset funciona como um **snapshot do planejamento aeropolítico**, consolidando dados de entidades operacionais, malha logística, cronogramas e capacidade de oferta.

4.10.1.1 Unidade de Análise

A unidade fundamental (atômica) deste dataset é a **Etapa de Voo**. Cada registro no sistema não representa apenas um evento isolado, mas sim um elo dentro do plane-

jamento comercial ou operacional da companhia.

4.10.1.2 Natureza e Tipologia

- **Granularidade:** Etapa Individual de Voo.
- **Tipologia:** Dataset estruturado composto por dados categóricos (identificadores ICAO), temporais (timestamps) e numéricos (capacidade).
- **Orientação Temporal:** Focado no **Planejamento e Operação Prevista**.
- **Rastreabilidade:** Todos os registros imutáveis acompanhados por `dt_insercao`.

4.10.2 Dicionário de Dados

Os principais campos de referência do SIROS incluem identificadores de voo e etapa (`numero_voo`, `numero_etapa`), operador aéreo (`sigla_icao_empresa_aerea`), origem e destino (`sigla_icao_aeroporto_origem`, `sigla_icao_aeroporto_destino`), temporalidade planejada e de referência (`data_partida_prevista_utc`, `data_chegada_prevista_utc`, `data_referencia_utc`) e rastreabilidade de ingestão (`dt_insercao`).

4.10.3 Dimensões de Qualidade Avaliadas

A qualidade deste conjunto de dados é avaliada por meio de dimensões estruturadas, amplamente reconhecidas na literatura de Governança e Engenharia de Dados, adaptadas ao contexto específico de planejamento e oferta de voos.

4.10.3.1 Completude

Mensura a densidade informacional do dataset. Campos críticos obrigatórios:

- `numero_voo`, `numero_etapa`, `sigla_icao_empresa_aerea`
- `sigla_icao_aeroporto_origem`, `sigla_icao_aeroporto_destino`
- `data_partida_prevista_utc`, `data_chegada_prevista_utc`
- `data_referencia_utc`, `dt_insercao`

4.10.3.2 Validade

Assegura conformidade com padrões sintáticos e restrições de domínio da indústria aeronáutica:

- Códigos ICAO devem possuir exatamente 4 caracteres alfanuméricos maiúsculos.
- `quantidade_assentos_previstos` deve ser inteiro > 0 para voos comerciais de passageiros.
- `numero_etapa` deve ser inteiro ≥ 1.
- Timestamps devem seguir padrão ISO 8601.

4.10.3.3 Consistência Temporal

Valida a harmonia cronológica entre diferentes marcos de tempo:

$$data\_partida\_prevista\_utc \leq data\_chegada\_prevista\_utc$$

$$data\_referencia\_utc \leq data\_partida\_prevista\_utc$$

$$dt\_insercao \geq data\_referencia\_utc$$

Inversões cronológicas indicam falhas de integração ou erros de fuso horário.

4.10.3.4 Conformidade Semântica

Assegura que os registros respeitem as ontologias e relações lógicas do domínio aeronáutico:

- Todo registro deve estar vinculado a um operador aéreo certificado.
- `quantidade_assentos_previstos` deve estar contida no intervalo técnico homologado para `sigla_icao_modelo_aeronave`.
- `sigla_icao_aeroporto_origem` $\neq$ `sigla_icao_aeroporto_destino` (exceto voos específicos).

4.10.4 Regras e Métricas de Qualidade

4.10.4.1 Validações de Integridade Básica

Atributo	Regra	Descrição Técnica
Mandatórios	`NOT NULL`	Campos: `numero_voo`, `numero_etapa`, `sigla_icao_empresa_aerea`, `sigla_icao_aeroporto_origem`, `sigla_icao_aeroporto_destino`, `data_partida_prevista_utc`, `data_chegada_prevista_utc`, `data_referencia_utc`, `dt_insercao`.
Códigos ICAO	REGEX $[A-Z]\{4\}$	Aeroportos de origem e destino devem possuir exatamente 4 caracteres alfabéticos maiúsculos.
Empresa ICAO	REGEX $[A-Z]\{3,4\}$	`sigla_icao_empresa_aerea` deve possuir 3 ou 4 caracteres alfabéticos maiúsculos.
Etapa	≥ 1	`numero_etapa` deve ser inteiro estritamente positivo.
Assentos	> 0	`quantidade_assentos_previstos` deve ser inteiro positivo para voos comerciais.
Timestamps	ISO 8601	Formato `YYYY-MM-DDTHH:MM:SSZ`.

Tabela 4.3: Validações de Integridade para SIROS

4.10.4.2 Consistência Temporal - Sequência Lógica

A validade de uma etapa de voo é definida pela ordem cronológica dos marcos operacionais:

$$data\_referencia\_utc \leq data\_partida\_prevista\_utc < data\_chegada\_prevista\_utc$$

Além disso:

$$dt\_insercao \geq data\_referencia\_utc$$

Qualquer violação sinaliza erro estrutural ou falha de integração sistêmica.

4.10.4.3 Validações Cross-field

- **Coerência de Rota:** `sigla_icao_aeroporto_origem` $\neq$ `sigla_icao_aeroporto_destino`, exceto em casos específicos com flag de classificação.

- **Compatibilidade Equipamento-Oferta:** `quantidade_assentos_previstos` deve estar dentro do *range* de configuração homologado para `sigla_icao_modelo_aeronave`.
- **Unicidade da Chave Composta:** Não deve existir mais de um registro para a combinação: `data_referencia_utc` + `sigla_icao_empresa_aerea` + `numero_voo` + `numero_etapa`.
- **Relacionamento Tipo de Voo e Oferta:**
  Se `tipo_de_voo` = *SOBREVOOS OU TRASLADOS OPERACIONAIS INTERNACIONAL* $\rightarrow$ `quantidade_assentos_previstos` deve ser 0. Se for *REGULAR DE PASSAGEIROS INTERNACIONAL* $\rightarrow$ deve ser > 0.

4.10.5 Potencial Analítico e Aplicações em IA

4.10.5.1 Adequação para Análise Estatística

O dataset é adequado para:

- Cálculo de oferta e análise de mercado.
- Inteligência de malha e conectividade entre aeroportos.
- Segmentação por operador e tipo de voo.
- Estudos de sazonalidade operacional.

Pilares da confiabilidade estatística: agregações corretas, consistência de contagem e integridade geográfica.

4.10.5.2 Adequação para Planejamento e Inteligência de Mercado

Aplicações em inteligência de mercado:

- Dinâmica de rede (expansão/retração de rotas).
- Análise de frota e eficiência.
- Market Share de oferta.

A presença de `quantidade_assentos_previstos` transforma lista de voos em modelo de capacidade, permitindo estimar capacidade agregada, densidade de oferta e benchmark competitivo.

4.10.5.3 Adequação para Machine Learning

Previsão de Demanda e Ocupação: Com enriquecimento externo, o dataset permite prever demanda esperada por rota, ocupação provável e elasticidade de oferta.

Engenharia de Features: Features de alto valor preditivo:

- **Rota:** Identifica padrões geográficos e hubs.
- **Operador:** Captura efeito de marca.
- **Modelo de Aeronave:** Define teto de oferta.
- **Número da Etapa:** Indica perna única ou trilho longo.
- **Timestamps:** Permite extração de componentes cíclicos (hora, dia, sazonalidade).

Otimização de Malha: Permite simulações de redistribuição de capacidade, ajuste de alocação e identificação de ineficiências.

Detecção de Anomalias: Suporta modelos não supervisionados para identificar capacidades incoerentes, rotas atípicas e combinações raras.

4.10.5.4 Adequação Consolidada

Aplicação	Adequação	Condição
Análise Descritiva	Muito Alta	Controle de duplicidade e filtragem de nulos.
Inteligência de Mercado	Muito Alta	Validação de capacidade e códigos ICAO.
Previsão de Demanda	Moderada a Alta	Necessita enriquecimento externo.
Otimização de Malha	Moderada	Consistência temporal absoluta.
Modelos de Anomalia	Alta	Saneamento cross-field prévio.

Tabela 4.4: Adequação para IA e Analytics do SIROS

4.11 TaticFlow

4.11.1 Visão Geral dos Dados

A análise acompanha todo o voo, desde o início no portão e manobra de saída (*push-back*) até a chegada e posicionamento final no destino (*in-block*). O objetivo é garantir que os dados, que combinam horários estimados e registros confirmados quase em tempo real, sejam confiáveis o suficiente para suportar operações críticas.

4.11.1.1 Pilares de Qualidade

A metodologia de validação foi desenhada para garantir que o *dataset* atenda aos seguintes requisitos fundamentais:

- **Consistência Cronológica:** Verificação da sucessão lógica de eventos, garantindo que a linha do tempo respeite a física do setor aeronáutico (ex: o horário de *take-off* deve ser obrigatoriamente posterior ao *pushback*).
- **Integridade Estrutural:** Validação da conformidade do esquema, tratamento de valores ausentes e garantia de tipagem adequada para processamento de alto desempenho.
- **Coerência Semântica:** Alinhamento dos dados com as regras de negócio e terminologias do domínio da aviação, garantindo que o dado digital reflita com precisão a operação em solo e em voo.
- **Prontidão Analítica:** Certificação de que o dado possui o refinamento necessário para alimentar modelos de Inteligência Artificial, algoritmos preditivos (ETA/ETD) e *dashboards* de eficiência operacional.

4.11.1.2 Escopo de Atuação

O escopo técnico desta análise abrange validações multidimensionais, incluindo:

1. **Validações Estruturais:** Verificação de *null constraints* e duplicidade de chaves primárias de voo.
2. **Validações Temporais:** Análise de *timestamps* para detecção de anomalias ou inversões de fluxo.

3. **Validações Cross-field:** Cruzamento de dados entre diferentes fontes para garantir a unicidade da informação.

A prioridade absoluta reside na **integridade da linha do tempo**, tratando-a como o ativo principal para a reconstrução histórica e otimização da malha aérea.

4.11.2 Dicionário de Dados

O conjunto de dados é composto por registros de eventos operacionais de alta fidelidade, onde a unidade fundamental de análise é o **voo individual**. Trata-se de um *dataset* estruturado de natureza predominantemente temporal (séries de eventos), projetado para capturar a discrepância e a correlação entre o planejamento operacional e a execução real.

Os dados são originados de sistemas críticos de monitoramento de controle de tráfego aéreo e plataformas de gestão operacional, garantindo uma visão fidedigna do fluxo de movimentação aérea.

4.11.2.1 Arquitetura Lógica da Informação

Para fins de análise e processamento, os atributos do *dataset* estão organizados em cinco domínios conceituais:

1. **Identificação do Voo:** Chaves alfanuméricas (como o `callsign`) que permitem o rastreio único e a vinculação com planos de voo específicos.
2. **Localização e Infraestrutura:** Metadados sobre aeródromos de origem, destino e posições de pátio (*stands*), fornecendo o contexto geográfico da operação.
3. **Características da Aeronave:** Atributos técnicos do equipamento, essenciais para validar performance e compatibilidade com a infraestrutura aeroportuária.
4. **Linha do Tempo Operacional:** O núcleo do *dataset*, contendo os *timestamps* de marcos críticos (como *pushback*, decolagem, pouso e corte de motores).
5. **Controle e Inserção:** Metadados de auditoria que registram o momento da entrada do dado no sistema, garantindo a rastreabilidade do fluxo de informação.

4.11.2.2 Fluxo Conceitual do Dado

O ciclo de vida do dado dentro deste *dataset* espelha a operação física da aeronave. Ele inicia-se com a intenção de voo (dados estimados), evolui através de eventos discretos capturados por sensores e *input* humano, e encerra-se com a consolidação dos horários reais de chegada e ocupação de posição.

Esta estrutura permite a reconstrução completa da jornada operacional, sendo um ativo fundamental para o cálculo de KPIs de pontualidade (*On-Time Performance*) e análise de gargalos em solo.

4.11.3 Dimensões de Qualidade Avaliadas

A integridade deste *dataset* é sustentada pela **Consistência Temporal**, uma dimensão crítica onde o valor analítico reside na precisão da ordenação e nos *deltas* (intervalos) entre eventos operacionais.

Diferente de repositórios estáticos, este conjunto de dados documenta uma **sequência causal de eventos físicos** (ex: *pushback* $\rightarrow$ *taxi* $\rightarrow$ *take-off*). Consequentemente, anomalias na cronologia não apenas invalidam o registro individual, mas corrompem métricas agregadas de eficiência e inviabilizam o treinamento de modelos de *Machine Learning* baseados em séries temporais.

Para assegurar a confiabilidade analítica, a qualidade é auditada sob quatro dimensões integradas:

4.11.3.1 Completude

Mensura a densidade do dado, garantindo que campos mandatórios para a reconstrução da jornada do voo estejam populados. A ausência de marcos críticos (como o *timestamp* de decolagem) é tratada como uma quebra na continuidade do ativo de dados.

4.11.3.2 Validade

Verifica se os atributos aderem aos padrões técnicos e domínios estritos da aviação civil:

- **Identificadores:** Conformidade com padrões ICAO/IATA.

- **Tipagem:** Rigor na formatação de campos temporais (ISO 8601) e coordenadas geográficas.
- **Domínio:** Validação de códigos de resposta de transponder (SSR) e categorias de aeronaves.

4.11.3.3 Consistência Temporal

Esta camada garante que a cronologia dos eventos respeite a física do domínio aeronáutico. O motor de regras de qualidade impede:

- **Inversões Lógicas:** Eventos de chegada registrados antes de eventos de partida.
- **Intervalos Impossíveis:** Tempos de *taxi* ou subida que fogem aos limites operacionais da aeronave.
- **Sobreposição de Estados:** Uma única aeronave operando múltiplos trechos simultâneos.

4.11.3.4 Conformidade Semântica

Assegura a harmonia entre as entidades do ecossistema. Um registro é considerado semanticamente íntegro apenas quando a relação entre o voo, a aeronave utilizada, a infraestrutura aeroportuária e o tipo de missão permanece coerente com as regras de negócio vigentes.

4.11.4 Regras e Métricas de Qualidade

4.11.4.1 Validações de Integridade e Padronização

Esta camada assegura que o dado está completo e em conformidade com os padrões internacionais da aviação (ICAO/IATA).

Atributo	Regra	Descrição Técnica
Mandatórios	`NOT NULL`	Campos críticos: `id`, `callsign`, `adep`, `ades`, `createdat`, `dt_insercao`.
Aeródromos	`REGEX [A-Z]{4}`	Códigos ICAO devem possuir exatamente 4 caracteres alfabéticos maiúsculos.
Transponder	`OCTAL (4 digits)`	O código SSR deve respeitar o padrão octal (0-7), contendo 4 dígitos.
Aeronave	`Standard ICAO`	O campo `acfttype` deve seguir a tabela de designadores de tipos de aeronaves da ICAO.
Auditoria	`createdat <= dt_insercao`	A criação do registro no sistema de origem deve preceder a persistência no banco.

Tabela 4.5: Validações de Integridade para TaticFlow

4.11.4.2 Consistência Temporal

Qualquer inversão nos *timestamps* abaixo sinaliza uma corrupção na telemetria ou erro de processamento.

Sequência Lógica Obrigatória:

$$EOBT \leq WPUSH \leq CPUSH \leq WTAXI \leq TAXI \leq HOLD$$

$$HOLD \leq CRWY \leq CDEP \leq DEP \leq ETA \leq ARR \leq CPOS$$

Principais Checkpoints de Fluxo:

- **Acionamento:** O horário estimado de saída ($EOBT$) deve preceder o *pushback* real ($CPUSH$).
- **Movimentação:** O início do táxi ($TAXI$) deve ocorrer após a liberação do *pushback*.
- **Decolagem:** A confirmação de pista livre ($CDEP$) deve anteceder o pouso (ARR).
- **Encerramento:** O horário de pouso (ARR) deve preceder a ocupação da posição final ($CPOS$).

4.11.4.3 Validações Cross-field e Regras de Negócio

Estas regras validam a semântica entre diferentes campos para evitar cenários impossíveis no mundo real:

- **Geolocalização:** O aeródromo de partida (`adep`) deve ser obrigatoriamente diferente do aeródromo de destino (`ades`).
- **Coerência de Infraestrutura:** A pista utilizada (`runway`) deve pertencer à localidade informada no plano de voo.
- **Tipagem de Operação:** O tipo de voo (`flighttype`) deve ser compatível com as regras de voo (IFR/VFR) e com a aeronave escalada (`acfttype`).
- **Eventos:** O gatilho de `eventtype` deve possuir obrigatoriamente um *timestamp* correspondente preenchido.

4.11.4.4 Indicadores e Métricas

A robustez deste *dataset* é monitorada através de métricas de integridade estrutural e temporal. Dado que este conjunto de dados sustenta modelos de ML para predição de atrasos e otimização de malha, qualquer degradação nos indicadores abaixo impacta diretamente a confiabilidade das análises de eficiência aeroportuária.

4.11.4.5 Métricas Operacionais Derivadas

Indicador	Fórmula	Objetivo Analítico
Taxi-Out Time	$DEP - CPUSH$	Identificar gargalos de solo e ineficiência de pista.
Departure Delay	$DEP - EOBT$	Alimentar modelos preditivos de pontualidade.
Arrival Delay	$ARR - ETA$	Avaliar a precisão do planejamento e aderência à malha.
Gate Occupancy	$CPOS - ARR$	Medir tempo entre pouso e liberação de posição.

Tabela 4.6: Métricas Operacionais Derivadas do TATICFLOW

4.11.5 Potencial Analítico e Aplicações em IA

Este *dataset* representa o ciclo operacional completo de um voo, contendo tanto estimativas quanto eventos confirmados ao longo da execução real. Diferentemente de datasets puramente planejados, permite análise causal e modelagem temporal baseada em eventos sequenciais.

4.11.5.1 Adequação para Análise Operacional

O dataset é adequado para análises de:

- Eficiência de *pushback* e táxi.
- Tempo de ocupação de pista.
- Aderência ao planejamento (EOBT vs DEP).
- Análise de congestionamento por aeroporto.
- Identificação de gargalos operacionais.

A presença de múltiplos *timestamps* permite decompor o voo em fases: *off-block*, *pushback*, táxi, entrada em pista, decolagem, enrota, chegada e ocupação de posição.

4.11.5.2 Adequação para Modelagem Temporal

Possíveis variáveis derivadas:

- Taxi Time = $DEP - CPUSH$
- Departure Delay = $DEP - EOBT$
- Arrival Delay = $ARR - ETA$
- Gate Occupancy Time = $CPOS - ARR$

Essas variáveis são fundamentais para previsão de atraso, estimativa de tempo de táxi, modelagem de *turnaround* e otimização de fluxo aeroportuário.

4.11.5.3 Aplicações de Machine Learning

Previsão de Atraso de Decolagem: *Features* potenciais: `adep`, `runway`, `flighttype`, `acfttype`, `equipment`, horário do dia, histórico médio de táxi. Alvo: `delay_departure` = `dep` − `eobt`.

Previsão de Taxi Time: Alvo: $DEP - CPUSH$. Modelos baseados em regressão com histórico, congestionamento temporal e *features* categóricas (`runway`, aeroporto, horário).

Detecção de Anomalias Operacionais: Aplicação de Isolation Forest, Autoencoders ou modelos baseados em desvio padrão histórico para detectar táxi excessivo, atrasos fora do padrão e inversões temporais não capturadas por regras.

4.11.5.4 *Riscos para IA*

Este dataset é extremamente sensível a erros temporais. Inversões cronológicas geram padrões impossíveis. Duplicidade de eventos distorce distribuições. Timestamps inconsistentes produzem atrasos negativos.

Modelos treinados sem validação temporal podem aprender ruído estrutural, padrões artificiais e relações causalmente incorretas.

4.11.5.5 *Adequação Consolidada*

Aplicação	**Adequação**	**Condição**
Análise Operacional	Muito Alta	Consistência temporal rigorosa.
Previsão de Atraso	Muito Alta	*Features* históricas complementares.
Previsão de Taxi	Muito Alta	Controle de *outliers*.
Detecção de Anomalia	Alta	Regras básicas aplicadas.
Modelagem de *Turnaround*	Alta	Integração com dados de *gate*.

Tabela 4.7: Adequação para IA e Analytics do TATICFLOW

4.12 VRA

Os dados são capturados sob a perspectiva da Agência Nacional de Aviação Civil (ANAC) e integram o sistema de monitoramento operacional de voos civis brasileiros, garantindo conformidade com padrões ICAO/IATA e regras de relatório obrigatório. O dataset permite análises de:

- Pontualidade operacional.
- Regularidade por empresa aérea.
- Oferta de assentos e segmentação de mercado.
- Operações domésticas versus internacionais.
- Desempenho regulatório e conformidade normativa.

4.12.1 *Visão Geral dos Dados*

Cada registro representa uma etapa de voo operada por uma empresa aérea, contendo identificação da empresa, do voo, tipo de operação, aeroportos envolvidos, horários previstos e realizados, capacidade ofertada e situação operacional.

- **Granularidade:** Etapa individual de voo.
- **Tipo:** Dados estruturados, temporais e categóricos.
- **Contexto:** Regulatório e operacional.
- **Atualização:** Controlada por `dt_insercao`.

4.12.2 Dicionário de Dados

Os campos estão agrupados em seis categorias funcionais:

1. **Identificação e Operador:** Empresa e número de voo.
2. **Classificação Regulatória:** Códigos de tipo de operação.
3. **Infraestrutura:** Aeródromos de origem e destino.
4. **Temporalidade:** Horários previstos e realizados.
5. **Situação e Justificativa:** Status operacional e justificativas.
6. **Controle e Auditoria:** Metadados de inserção e rastreabilidade.

4.12.3 Dimensões de Qualidade Avaliadas

Este dataset exige avaliação integrada entre estrutura regulatória, execução temporal e coerência operacional. A confiabilidade analítica depende não apenas da presença de dados, mas da coerência entre tipo de linha, execução real e capacidade ofertada.

4.12.3.1 Completude

Campos críticos que devem estar sempre preenchidos:

- `id`, `sg_empresa_icao`, `nr_voo`
- `sg_icao_origem`, `sg_icao_destino`
- `dt_partida_prevista`, `dt_chegada_prevista`
- `dt_referencia`, `dt_insercao`

A ausência desses campos compromete análises de regularidade e mercado.

4.12.3.2 Validade

Verificação de conformidade com padrões técnicos e domínios estritos da aviação civil:

- Códigos ICAO devem possuir 4 letras maiúsculas.
- `nr_assentos_ofertados` deve ser inteiro ≥ 0.
- `cd_di` deve pertencer ao conjunto $\{0, 2, 3, 4, 6, 7, 9, D\}$.
- `cd_tipo_linha` deve pertencer ao conjunto $\{N, C, I, G\}$.
- Datas devem ser válidas e coerentes.

4.12.3.3 Consistência Temporal

Regras principais para a ordenação de eventos:

- $dt\_partida\_prevista \leq dt\_chegada\_prevista$
- $dt\_partida\_real \leq dt\_chegada\_real$
- $dt\_referencia \leq dt\_partida\_prevista$
- $dt\_insercao \geq dt\_referencia$

Inversões cronológicas indicam erro estrutural ou falha de integração.

4.12.3.4 Conformidade Regulatória e Semântica

Validação cruzada entre campos para garantir coerência normativa:

- `cd_tipo_linha` deve ser coerente com localidade dos aeródromos (doméstico vs internacional).
- `nr_assentos_ofertados` deve ser compatível com equipamento informado.
- `ds_situacao_partida` e `ds_situacao_chegada` devem ser coerentes com presença de horários reais.

4.12.4 Regras e Métricas de Qualidade

As regras descritas nesta seção têm como objetivo garantir integridade estrutural, coerência regulatória e consistência temporal entre planejamento e execução da etapa de voo.

4.12.4.1 Validações de Integridade e Padronização

Atributo	**Regra**	**Descrição Técnica**
Mandatórios	`NOT NULL`	Campos: `id`, `sg_empresa_icao`, `nr_voo`, `sg_icao_origem`, `sg_icao_destino`, `dt_partida_prevista`, `dt_chegada_prevista`, `dt_referencia`, `dt_insercao`.
Códigos ICAO	REGEX $A-Z\{4\}$	`sg_icao_origem` e `sg_icao_destino` devem possuir exatamente 4 caracteres alfabéticos maiúsculos.
Empresa ICAO	REGEX $A-Z\{3,4\}$	`sg_empresa_icao` deve possuir 3 ou 4 caracteres alfabéticos maiúsculos.
Código DI	Domínio	`cd_di` $\in \{0,2,3,4,6,7,9,D\}$.
Tipo de Linha	Domínio	`cd_tipo_linha` $\in \{N,C,I,G\}$.
Assentos	≥ 0	`nr_assentos_ofertados` deve ser inteiro não negativo.
Auditoria Temporal	Ordem Cronológica	`dt_referencia` $\leq$ `dt_insercao`.

Tabela 4.8: Validações de Integridade para VRA

4.12.4.2 Consistência Temporal - Sequência Lógica

A validade de uma etapa de voo é definida pela ordem cronológica dos marcos operacionais.

$$dt\_partida\_prevista \leq dt\_chegada\_prevista$$

$$dt\_partida\_real \leq dt\_chegada\_real$$

Além disso:

$$dt\_referencia \leq dt\_partida\_prevista$$

$$dt\_insercao \geq dt\_referencia$$

Qualquer violação sinaliza erro estrutural ou falha de integração sistêmica.

4.12.4.3 Validações Cross-field e Regulatórias

- **Coerência de Rota:** `sg_icao_origem` $\neq$ `sg_icao_destino`, exceto em casos específicos devidamente justificados.
- **Tipo de Linha vs Aeródromos:**
  - Se `cd_tipo_linha` $\in \{N, C\} \rightarrow$ ambos aeródromos devem estar no Brasil.
  - Se `cd_tipo_linha` $\in \{I, G\} \rightarrow$ ao menos um aeródromo deve estar fora do Brasil.
- **Capacidade vs Equipamento:** `nr_assentos_ofertados` deve ser compatível com `sg_equipamento_icao`. Valores extremos configuram erro de cadastro.
- **Situação vs Execução:**
  - Se `ds_situacao_partida` indicar cancelamento $\rightarrow$ `dt_partida_real` deve estar ausente.
  - Se `ds_situacao_chegada` indicar realizado $\rightarrow$ `dt_chegada_real` deve estar presente.
- **Duplicidade de Etapa:** Não deve existir mais de um registro com idênticos: `sg_empresa_icao`, `nr_voo`, `dt_referencia`, `cd_di`.

4.12.5 Potencial Analítico e Aplicações em IA

Este dataset integra informações regulatórias, planejamento operacional e execução real da etapa de voo, constituindo uma base robusta para análises de desempenho e aplicações de Inteligência Artificial.

4.12.5.1 Adequação para Análise Estatística

O dataset é altamente adequado para:

- Cálculo de pontualidade por empresa.
- Análise de regularidade por tipo de linha.
- Comparação doméstico versus internacional.
- Avaliação de oferta de assentos por rota.

- Estudo de sazonalidade operacional.

As variáveis temporais permitem calcular: - Atraso médio de partida por empresa e aeroporto. - Percentual de voos dentro da tolerância regulatória. - Distribuição de atrasos por horário.

4.12.5.2 Aplicações de Machine Learning

Previsão de Atraso de Partida: *Features* potenciais: `sg_empresa_icao`, `sg_icao_origem`, `sg_icao_destino`, `cd_tipo_linha`, horário do dia, dia da semana, histórico médio de atraso. Alvo: `delay_partida` = `dt_partida_real` − `dt_partida_prevista`.

Previsão de Atraso de Chegada: `delay_chegada` = `dt_chegada_real` − `dt_chegada_prevista`. Aplicações: antecipar impactos em conexões, estimar risco de perda de slot.

Classificação de Regularidade: Classificar voos em: pontual, atraso leve, atraso moderado, atraso severo. Usado para benchmarking entre empresas e análise de eficiência operacional.

Modelagem de Mercado e Capacidade: Com `nr_assentos_ofertados` e `cd_tipo_linha`, modelos podem prever oferta futura, detectar rotas sub ou superdimensionadas, e simular redistribuição de capacidade.

4.12.5.3 Riscos para IA

Apesar do alto potencial, alguns riscos devem ser considerados:

- Duplicidades geram viés estatístico.
- Classificações regulatórias inconsistentes distorcem segmentações.
- Atrasos negativos ou inversões temporais comprometem aprendizado.
- Dados ausentes de execução real reduzem robustez preditiva.

Modelos treinados sobre dados estruturalmente imperfeitos podem aprender padrões artificiais, reduzindo generalização e explicabilidade.

4.12.5.4 Adequação Consolidada

Aplicação	Adequação	Condição
Análise de Pontualidade	Muito Alta	Validação temporal aplicada.
Benchmarking Regulatório	Alta	Coerência normativa garantida.
Previsão de Atraso	Alta	Histórico consistente.
Modelagem de Capacidade	Muito Alta	Controle de duplicidade.
Classificação de Regularidade	Alta	Dados completos de execução.

Tabela 4.9: Adequação para IA e Analytics do VRA

4.13 Waypoints

Waypoints representam pontos fixos no espaço geográfico que compõem aerovias, procedimentos SID/STAR e rotas planejadas. Esses dados são essenciais para modelagem de trajetórias, construção de grafos de navegação aérea, análise de convergência de fluxos, cálculo de distância e complexidade, bem como aplicações de Inteligência Artificial baseadas em estrutura de rede.

4.13.1 Visão Geral dos Dados

Cada registro representa um ponto fixo de navegação com coordenadas geográficas associadas. O dataset possui natureza estrutural (geoespacial), com dimensão espacial definida por latitude e longitude, e dimensão vertical representada por altitude mínima e máxima quando presente.

4.13.2 Dicionário de Dados

O dicionário de dados dos waypoints está organizado em três categorias principais: identificação, localização e propriedades operacionais.

Para identificação, cada waypoint possui um identificador único (`ident`) que pode representar um FIX ou NAVAID. A localização é definida pelas coordenadas geográficas de `latitude` e `longitude`, expressas em graus decimais. O campo `beginposition` contem a data em que aquele waypoint entrou em vigor com aquele identificador e reespectivas coordenadas.

4.13.3 Dimensões de Qualidade Avaliadas

O dataset de Waypoints representa a infraestrutura estrutural da navegação aérea. Diferentemente de datasets dinâmicos, sua qualidade impacta diretamente a integridade de grafos de navegação, rotas planejadas e modelos de complexidade operacional. As dimensões avaliadas consideram tanto a integridade geoespacial quanto a coerência estrutural da malha aérea, abrangendo completude, validade geográfica, consistência estrutural, conformidade aeronáutica e integridade topológica.

4.13.3.1 Completude

A completude avalia a presença dos campos essenciais necessários para representação espacial e identificação do waypoint. Os campos críticos incluem `ident` (ou nome do fixo), `latitude` e `longitude`. A ausência desses campos inviabiliza a construção de grafos e o uso em algoritmos de roteamento, comprometendo toda a cadeia de análise estrutural.

4.13.3.2 Validade Geográfica

Esta dimensão garante que as coordenadas representem posições plausíveis no espaço terrestre. Os critérios estabelecem que a latitude deve estar no intervalo $[-90, 90]$, a longitude no intervalo $[-180, 180]$, as coordenadas não podem ser simultaneamente nulas, e não deve haver inversão entre latitude e longitude. Waypoints com coordenadas inválidas comprometem cálculos de distância, rotas e projeções geográficas, propagando erros para sistemas downstream.

4.13.3.3 Consistência Estrutural

A consistência estrutural avalia a integridade interna do dataset como base de dados estruturada. Os critérios incluem a unicidade do `ident`, ausência de duplicidade de coordenadas com identificadores distintos quando não justificado, e ausência de registros incompletos. Inconsistências estruturais propagam erros para grafos de navegação e embeddings de rota, comprometendo a confiabilidade de análises subsequentes.

4.13.3.4 Conformidade Aeronáutica

A conformidade aeronáutica garante aderência às convenções e padrões do domínio aeronáutico internacional. Os critérios verificam se a nomenclatura é compatível com o padrão ICAO, onde fixes RNAV geralmente possuem 5 letras, se há coerência com a FIR ou região declarada, e se existe compatibilidade com aerovias e procedimentos SID/STAR quando aplicável. Essa dimensão assegura que o waypoint represente uma entidade válida no espaço aéreo operacional, facilitando a interoperabilidade entre sistemas.

4.13.3.5 Integridade Topológica

A integridade topológica avalia a capacidade do dataset de sustentar modelagem em grafo. Os critérios analisam a conectividade entre waypoints associados a aerovias, a ausência de nós isolados quando parte de rede declarada, e a coerência de continuidade geográfica entre pontos sequenciais. Esta dimensão é essencial para construção de grafos de navegação, cálculo de complexidade operacional, clusterização de fluxos e geração de embeddings estruturais, constituindo a base para análises avançadas de rede.

4.13.4 Regras e Métricas de Qualidade

As regras desta seção garantem integridade geoespacial, consistência estrutural e robustez topológica do dataset de Waypoints. Como este dataset sustenta a modelagem de rotas, grafos de navegação e análise de complexidade operacional, qualquer inconsistência pode propagar erros para sistemas downstream. As regras estão organizadas em cinco categorias: básicas, geográficas, estruturais, topológicas e de consistência com trajetórias.

4.13.4.1 Regras Básicas

As regras básicas estabelecem os requisitos fundamentais de integridade. O campo `ident` não pode ser nulo, latitude e longitude são obrigatórios, e o `ident` deve ser único para o mesmo `beginposition` . Não são permitidos registros completamente duplicados, e todos os campos numéricos devem respeitar a tipagem correta, garantindo a consistência de processamento em sistemas analíticos.

4.13.4.2 Regras Geográficas

As regras geográficas asseguram a plausibilidade espacial dos waypoints. A latitude deve estar no intervalo $[-90, 90]$ e a longitude no intervalo $[-180, 180]$. Coordenadas não podem ser simultaneamente $(0, 0)$, não é permitida a troca de latitude por longitude, e a distância mínima entre waypoints distintos não deve ser zero, evitando sobreposição geográfica inadequada.

4.13.4.3 Regras Estruturais

As regras estruturais validam a coerência interna dos atributos. Waypoints com mesmo nome não devem possuir múltiplas localizações exceto se explicitamente diferenciados pelo campo `beginposition`. Não é permitido waypoint com identificador vazio, e campos de tipo ou categoria devem pertencer a domínio controlado, garantindo padronização semântica.

4.13.4.4 Regras Topológicas

As regras topológicas asseguram a integridade da rede de navegação. Waypoints associados a aerovias devem possuir conexão válida no grafo, não são permitidos nós isolados quando parte de rede declarada, a ordem sequencial de waypoints em aerovia deve respeitar continuidade geográfica, e não são permitidos saltos geográficos incoerentes dentro da mesma aerovia, preservando a lógica operacional da malha aérea.

4.13.4.5 Regras de Consistência com Trajetórias

As regras de consistência com trajetórias garantem alinhamento com dados dinâmicos. Todo waypoint utilizado em rota real deve existir no dataset, o map-matching não deve associar aeronave a waypoint inexistente, a sequência de waypoints em rota deve minimizar distância cumulativa, e waypoints consecutivos em trajetória real devem estar dentro de raio plausível, viabilizando análises de aderência ao plano de voo.

4.13.5 Potencial Analítico e Aplicações em IA

O dataset de Waypoints constitui a base estrutural da malha de navegação aérea. Diferentemente de datasets temporais ou meteorológicos, sua relevância está na de-

finição da topologia do espaço aéreo. Sob a perspectiva de Inteligência Artificial, os waypoints representam nós de um grafo dirigido, onde aerovias e rotas constituem arestas, permitindo modelagem avançada baseada em teoria de grafos, aprendizado estrutural e análise de complexidade operacional.

4.13.5.1 Modelagem em Grafo

Waypoints podem ser representados como um grafo $G = (V, E)$, onde V representa o conjunto de nós (waypoints) e E o conjunto de arestas (aerovias ou conexões). Essa modelagem permite cálculo de caminhos mínimos, análise de conectividade, detecção de gargalos e identificação de hubs estruturais. A representação em grafo viabiliza a aplicação de algoritmos clássicos como Dijkstra, A* e algoritmos de fluxo máximo para otimização de rotas.

4.13.5.2 Integração com Trajetórias Reais

Quando integrado ao OpenSky, o dataset de waypoints permite operações avançadas de análise de trajetória. O map-matching de trajetórias ADS-B possibilita identificar a aderência ao plano de voo, detectar desvios operacionais e calcular a diferença entre distância real percorrida e distância planejada. Esta integração é fundamental para avaliação de eficiência operacional e detecção de anomalias com baixa latência.

4.13.5.3 Complexidade Operacional

A combinação de grau do nó, frequência de utilização, convergência de fluxos e densidade local permite estimar métricas de complexidade estrutural do espaço aéreo. Waypoints com alta centralidade e alto fluxo tendem a representar pontos de convergência, gargalos operacionais e regiões críticas de tráfego. Esta análise é essencial para planejamento de capacidade e gestão de fluxo de tráfego aéreo.

4.13.5.4 Aplicações de Machine Learning

O dataset é altamente adequado para técnicas avançadas de aprendizado de máquina. GNN podem capturar padrões espaciais complexos na topologia da rede. Node2Vec permite gerar embeddings estruturais preservando propriedades de vizinhança e comunidade. Técnicas de clusterização identificam grupos de rotas com características similares. A detecção de comunidades na malha aérea revela estruturas operacionais emergentes. Finalmente, modelos preditivos podem antecipar

congestionamentos baseados em padrões históricos de utilização combinados com a estrutura de rede.

4.13.5.5 Riscos para IA

Diversos fatores podem comprometer a qualidade de modelos baseados em waypoints. A duplicidade de nós distorce a estrutura do grafo, introduzindo redundância artificial. Waypoints isolados prejudicam a qualidade de embeddings por não capturarem informação de vizinhança. Coordenadas incorretas afetam diretamente o cálculo de distância e métricas espaciais. A conectividade incompleta reduz a robustez do modelo, limitando sua capacidade de generalização e produzindo previsões inconsistentes em regiões mal conectadas da rede.

4.13.5.6 Resumo de Adequação

A tabela a seguir sintetiza a adequação do dataset para diferentes aplicações de inteligência artificial:

Tabela 4.10: Adequação do Dataset Waypoints para Aplicações de IA

Aplicação	**Adequação**	**Condição**
Modelagem de Grafo	Muito Alta	Integridade topológica garantida
Embeddings Estruturais	Muito Alta	Ausência de duplicidade
Análise de Complexidade	Alta	Integração com fluxo real
Previsão de Congestionamento	Alta	Dados dinâmicos integrados

O dataset Waypoints representa o esqueleto estrutural do espaço aéreo brasileiro. Sua qualidade determina diretamente a robustez de grafos de navegação, a precisão de embeddings de rota e a confiabilidade de modelos de complexidade operacional.

Quando integrado a dados dinâmicos provenientes do OpenSky e dados operacionais de outras fontes, transforma-se em uma plataforma completa para modelagem estrutural do tráfego aéreo. A combinação de informação estática (topologia) com informação dinâmica (fluxo) permite análises de alta fidelidade sobre padrões de tráfego, eficiência de rotas e capacidade do espaço aéreo.

Sob a perspectiva de Inteligência Artificial aplicada à gestão de tráfego aéreo, este dataset é fundamental para sistemas baseados em grafos, análise de emergência ope-

racional e modelagem de convergência de fluxos. A manutenção de alta qualidade deste dataset é essencial não apenas para operações diárias, mas também para o desenvolvimento de sistemas preditivos e de suporte à decisão baseados em IA, contribuindo para a evolução contínua da eficiência e segurança do sistema de aviação civil brasileiro.

4.14 Processo de Validação e Qualidade de Dados

4.14.1 Visão Geral e Filosofia

O **AirData Data Check** implementa um **agente especializado em qualidade de dados** que opera com uma filosofia clara: dados confiáveis sustentam decisões seguras. Este agente executa validações sistemáticas e automatizadas em tempo real após cada ciclo de ingestão, garantindo que cada registro, cada coluna e cada valor sejam submetidos a uma bateria rigorosa de testes de qualidade antes de serem disponibilizados para análise.

A abordagem central do sistema é fundamentada em sete pilares metodológicos: *completude* (os dados estão presentes e não nulos?), *validade* (os dados seguem formatos, domínios e padrões regulatórios esperados?), *consistência* (os dados são coerentes interna e externamente, inclusive entre sistemas distintos?), *acurácia* (os valores representam fielmente a realidade operacional?), *pontualidade* (os dados estão disponíveis dentro da janela temporal útil para decisão?), *unicidade* (existem duplicidades ou redundâncias indevidas?) e *rastreabilidade* (é possível auditar a origem, histórico e eventuais alterações?).

Esse conjunto integrado garante que cada dataset possa ser utilizado com confiança em aplicações de inteligência artificial, análise regulatória, governança institucional e tomada de decisão operacional em ambiente ATM.

4.14.2 Estratégia de Validação Multi-Camadas

O processo de validação opera em três camadas hierárquicas e complementares. Na primeira camada, o sistema executa **checks globais** que avaliam a sanidade estrutural do dataset completo, incluindo a detecção de linhas completamente duplicadas e a identificação de colunas inteiramente vazias. Esses checks são executados uma única vez por arquivo e fornecem uma visão panorâmica da integridade estrutural dos dados.

Na segunda camada, o agente executa **checks por coluna** que se aplicam universalmente independentemente do tipo de dado. Esse conjunto inclui análises de nulidade (percentual de valores ausentes), cardinalidade (distribuição de valores únicos), e a identificação de colunas com valor constante, que geralmente indicam problemas na coleta ou processamento.

A terceira camada implementa **checks especializados por tipo de variável**. Aqui, cada coluna é classificada semanticamente (numérico, data/hora, texto, categórico, identificador) e submetida a validações específicas para seu domínio. Por exemplo, colunas numéricas são testadas para outliers estatísticos usando o método IQR (Interquartile Range), enquanto colunas de data/hora são validadas quanto a datas futuras implausíveis ou datas anteriores a 2000, que podem indicar erros de parsing ou entrada.

4.14.3 Tipologia Completa de Quality Checks

4.14.3.1 Checks Globais

Linhas Duplicadas (linhas_duplicadas). O sistema detecta registros inteiramente idênticos no dataset, situação que pode indicar problemas no processo de ETL ou coletas acidentais duplicadas. Cada linha duplicada detectada gera um alerta de severidade `WARNING`, pois embora não inviabilize o uso imediato dos dados, pode distorcer análises estatísticas e modelos preditivos.

Coluna Vazia (coluna_vazia). Quando uma coluna apresenta valores nulos em 100% de seus registros, o sistema emite um alerta de severidade `CRITICAL`. Colunas completamente vazias representam perda total de informação e frequentemente indicam falhas graves na integração ou mapeamento de dados.

4.14.3.2 Checks de Completude

Nulos Críticos (nulos_criticos). Quando uma coluna apresenta mais de 50% de valores nulos, o sistema considera a situação crítica. A severidade `CRITICAL` reflete que a coluna perdeu sua utilidade prática para a maioria das análises, comprometendo significativamente a confiabilidade dos insights derivados.

Nulos Altos (nulos_altos). Colunas com 20% a 50% de valores ausentes recebem severidade `ERROR`. Esse nível de nulidade ainda permite análises, mas requer tratamento cuidadoso (imputação, remoção controlada) e deve ser documentado em qualquer relatório analítico derivado.

Nulos Moderados (nulos). Percentuais de nulidade entre 5% e 20% acionam alertas de severidade `WARNING`. Esse nível é geralmente gerenciável, mas merece atenção em análises longitudinais ou quando a coluna é essencial para modelagem.

Strings Vazias (strings_vazias). O sistema detecta valores que, embora não sejam tecnicamente NULL, contêm apenas strings vazias ("") ou whitespace. Esses casos recebem severidade `WARNING` pois representam ausência de informação disfarçada.

4.14.3.3 Checks Numéricos

Outliers Estatísticos (outliers). Utilizando o método IQR (Interquartile Range), o sistema identifica valores que se desviam significativamente da distribuição central. Valores abaixo de $Q_1 - 1.5 \times IQR$ ou acima de $Q_3 + 1.5 \times IQR$ são marcados como outliers. Quando a proporção de outliers é inferior a 10% do total, o sistema emite alertas de severidade `INFO`, considerando que outliers podem ser legítimos e relevantes para análise de fenômenos extremos.

Valores Negativos (valores_negativos). Em colunas onde valores negativos são semanticamente inválidos (altitude acima do solo, número de assentos, duração de voo), o sistema emite alertas de severidade `WARNING`. Esse check é contextual e adaptável às regras de domínio de cada base de dados.

4.14.3.4 Checks Temporais

Datas Futuras (datas_futuras). Registros com timestamps posteriores à data/hora da execução do agente recebem severidade `WARNING`. Embora datas futuras possam ser legítimas em contextos de planejamento (planos de voo, slots ATFM), elas frequentemente indicam erros de timezone, sincronização de relógios ou problemas de parsing.

Datas Antigas (datas_antigas). Registros com datas anteriores a 1 de janeiro de 2000 são sinalizados com severidade `INFO`. Essa regra captura casos comuns de parsing incorreto (como interpretação de anos de dois dígitos) ou valores default (epoch 1970).

4.14.3.5 Checks de Texto

Strings Muito Curtas (strings_muito_curtas). Valores textuais com menos de 2 caracteres podem indicar truncamento, abreviações incorretas ou problemas de encoding. O sistema emite alertas de severidade `INFO` para rastreamento.

Strings Muito Longas (strings_muito_longas). Campos textuais excedendo 500 caracteres recebem severidade `WARNING`. Textos excessivamente longos podem indicar

concatenação indevida, falta de normalização ou problemas de integração de sistemas.

4.14.3.6 Checks de Distribuição

Coluna Constante (coluna_constante). Quando uma coluna apresenta cardinalidade igual a 1 (apenas um valor único em todos os registros não-nulos), o sistema emite severidade `ERROR`. Colunas constantes não agregam informação e frequentemente indicam problemas de configuração ou coleta.

Cardinalidade Alta (cardinalidade_alta). Quando mais de 90% dos valores são únicos, o sistema emite severidade `INFO`. Esse padrão é esperado em identificadores, mas pode ser problemático em colunas categóricas, indicando falta de normalização ou inconsistência de entrada.

4.14.4 Sistema de Severidades e Interpretação

O framework de qualidade emprega quatro níveis de severidade cuidadosamente calibrados:

INFO representa achados puramente informativos que não comprometem análises. Exemplos incluem outliers estatísticos legítimos ou datas antigas em datasets históricos. Esses alertas servem para documentação e rastreabilidade, mas não penalizam significativamente o score de qualidade.

WARNING indica situações que requerem atenção e podem comprometer análises específicas, mas não inviabilizam o uso geral dos dados. Linhas duplicadas, valores negativos inesperados e strings muito longas se enquadram nesta categoria.

ERROR sinaliza problemas relevantes que comprometem a confiabilidade dos dados para muitas aplicações. Colunas com 20-50% de nulidade ou colunas constantemente com um único valor recebem esta classificação.

CRITICAL indica problemas que inviabilizam o uso dos dados sem correção prévia. Colunas inteiramente vazias ou colunas com mais de 50% de valores ausentes são consideradas críticas.

4.14.5 Metodologia de Scoring e Quantificação de Qualidade

O **AirData Data Check** emprega um sistema de scoring inteligente e proporcional que traduz problemas de qualidade detectados em uma métrica numérica compreensível: um score de 0 a 100, onde 100 representa qualidade perfeita.

A metodologia de cálculo reconhece que a gravidade de um problema é relativa ao tamanho do dataset. Detectar 3 valores críticos em 100.000 registros indica excelente qualidade geral (impacto de 0,003%), enquanto os mesmos 3 problemas em 100 registros representam impacto muito maior (3%). Essa proporcionalidade é central para avaliações justas e contextualizadas.

4.14.5.1 Fórmula Matemática de Scoring

O cálculo do score opera em três etapas:

Etapa 1: Ponderação de Problemas. Cada problema detectado recebe um peso proporcional à sua severidade:

$$\text{pontos\_problema} = (n_{\text{CRITICAL}} \times 2.0) + (n_{\text{ERROR}} \times 1.0) + (n_{\text{WARNING}} \times 0.5) \tag{4.1}$$

Problemas `INFO` não contribuem para o score, pois são puramente informativos. Note que problemas `CRITICAL` contam duplamente em relação a `ERROR`, refletindo sua maior gravidade.

Etapa 2: Cálculo do Percentual de Erro. O total de pontos é normalizado pelo número total de registros do dataset:

$$\text{percentual\_erro} = \left(\frac{\text{pontos\_problema}}{\text{total\_registros}} \right) \times 100 \tag{4.2}$$

Etapa 3: Conversão em Score de Qualidade. O score final é calculado subtraindo o percentual de erro de 100:

$$\text{score} = \max(0, 100 - \text{percentual\_erro}) \tag{4.3}$$

A função max garante que scores nunca sejam negativos, mesmo em datasets severamente comprometidos.

4.14.5.2 Exemplos Práticos de Cálculo

Exemplo 1: Dataset de Alta Qualidade

Considere um dataset com 100.000 registros onde foram detectados apenas 3 problemas de severidade `WARNING` (por exemplo, 3 outliers estatísticos em variáveis numéricas).

$$\begin{aligned}\text{pontos\_problema} &= (0 \times 2.0) + (0 \times 1.0) + (3 \times 0.5) = 1.5 \\ \text{percentual\_erro} &= \left(\frac{1.5}{100000}\right) \times 100 = 0.0015\% \\ \text{score} &= 100 - 0.0015 = 99.9985 \approx 100.0\end{aligned}$$

Interpretação: Score de 99.9985 indica qualidade praticamente perfeita. Os outliers detectados são estatisticamente esperados e não comprometem análises.

Exemplo 2: Dataset com Problemas Moderados

Dataset de 200 registros com 5 problemas `CRITICAL` (múltiplas colunas vazias ou com >50% nulidade), 10 `ERROR`, e 30 `WARNING`.

$$\begin{aligned}\text{pontos\_problema} &= (5 \times 2.0) + (10 \times 1.0) + (30 \times 0.5) = 35.0 \\ \text{percentual\_erro} &= \left(\frac{35.0}{200}\right) \times 100 = 17.5\% \\ \text{score} &= 100 - 17.5 = 82.5\end{aligned}$$

Interpretação: Score de 82.5 sinaliza qualidade comprometida. O dataset requer limpeza e correção antes de uso em aplicações críticas. Análises derivadas devem ser interpretadas com cautela.

Exemplo 3: Dataset Extremamente Problemático

Dataset de 100 registros com 20 problemas `CRITICAL`, 10 `ERROR`, e 5 `WARNING`.

$$\begin{aligned}\text{pontos\_problema} &= (20 \times 2.0) + (10 \times 1.0) + (5 \times 0.5) \\ &= 40 + 10 + 2.5 \\ &= 52.5\end{aligned}$$

$$\begin{aligned}\text{percentual\_erro} &= \left(\frac{52.5}{100}\right) \times 100 \\ &= 52.5\%\end{aligned}$$

$$\begin{aligned}\text{score} &= 100 - 52.5 \\ &= 47.5\end{aligned}$$

Interpretação: Score de 47.5 indica qualidade extremamente baixa e alto risco operacional. O dataset não é adequado para uso produtivo sem correções estruturais imediatas, especialmente na resolução dos problemas `CRITICAL`.

4.15 Interface Web do **AirData** Data Check

O **AirData Data Check** disponibiliza uma interface web completa e intuitiva para acesso, análise e validação de qualidade dos dados aeronáuticos. Desenvolvida com tecnologias modernas (FastAPI, HTML5, CSS3 e JavaScript), a plataforma oferece funcionalidades para exploração de dados, execução de consultas SQL personalizadas, visualização de métricas de qualidade e exportação de resultados.

4.15.1 Arquitetura da Interface

A interface web opera como um cliente que se comunica com a API REST do backend, executando operações assíncronas e apresentando resultados de forma dinâmica. O sistema é acessível através do endereço `data.airdata.ita.br` e não requer instalação de software adicional, operando integralmente no navegador do usuário.

4.15.2 Página Inicial e Navegação Principal

A Figura 4.7 apresenta o menu principal de navegação, oferecendo acesso centralizado a todas as bases de dados disponíveis no sistema. O design responsivo adapta-se a diferentes resoluções de tela, mantendo a usabilidade em dispositivos desktop, tablets e celulares.

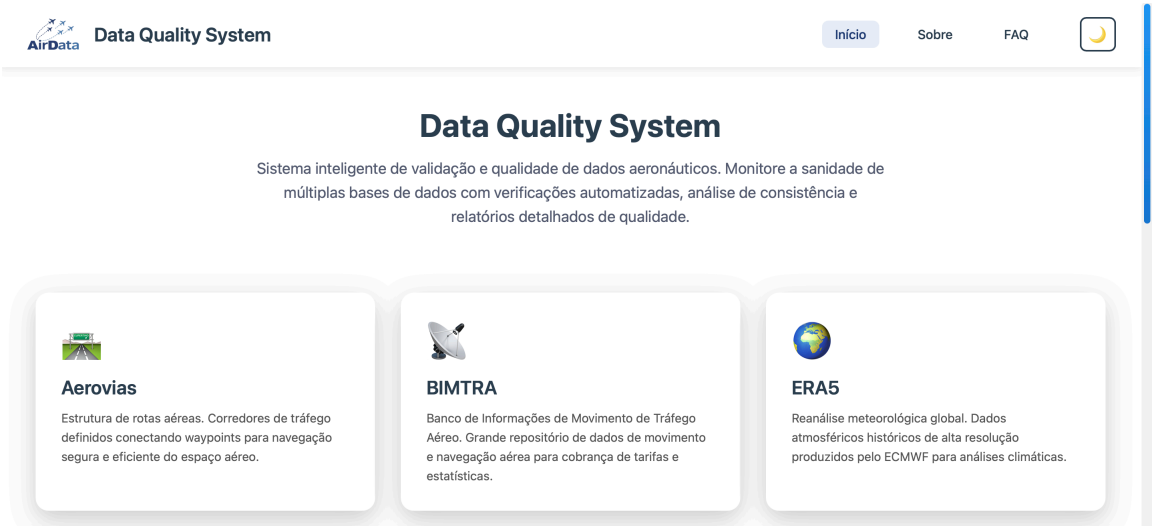

Figura 4.7: Tela inicial do **AirData** Data Check exibindo o menu de navegação principal com acesso às 18 bases de dados integradas.

4.15.3 Interface de Análise de Base de Dados

Ao selecionar uma base de dados específica, o usuário é direcionado para a interface principal de análise (Figura 4.8). Esta interface concentra todas as funcionalidades essenciais em uma única tela, organizada em seções lógicas.

Figura 4.8: Interface principal de análise da base BIMTRA, exemplificando o layout padrão utilizado para todas as bases de dados do sistema.

4.15.4 Consultas SQL Integradas

Uma das funcionalidades mais poderosas da plataforma é a capacidade de executar consultas SQL personalizadas diretamente no banco de dados PostgreSQL, sem necessidade de ferramentas externas. Esta funcionalidade está disponível em todas as interfaces de base de dados.

4.15.4.1 Execução de Queries

O usuário pode digitar consultas SQL completas no campo de busca, utilizando a sintaxe padrão PostgreSQL. Exemplos de consultas suportadas:

```
-- Consulta simples
SELECT * FROM airdata.bimtra LIMIT 100
```

```
-- Consulta com filtros
SELECT adpartida, addestino, dhmovreal
FROM airdata.bimtra
WHERE dhmovreal >= '2025-01-01'

-- Consulta com agregações
SELECT adpartida, COUNT(*) as total_voos
FROM airdata.bimtra
GROUP BY adpartida
ORDER BY total_voos DESC
LIMIT 10
```

Ao executar a consulta (pressionando o botão ”Executar Consulta”), o sistema:

1. Valida a sintaxe SQL
2. Executa a query no banco de dados via conexão SSH segura
3. Realiza análise automática de qualidade dos resultados
4. Calcula estatísticas descritivas para todas as colunas
5. Apresenta os dados em formato tabular interativo

4.15.4.2 Visualização de Resultados SQL

Os resultados das consultas SQL são apresentados na interface principal (Figura 4.9), com organização em abas para facilitar a navegação entre diferentes aspectos dos dados.

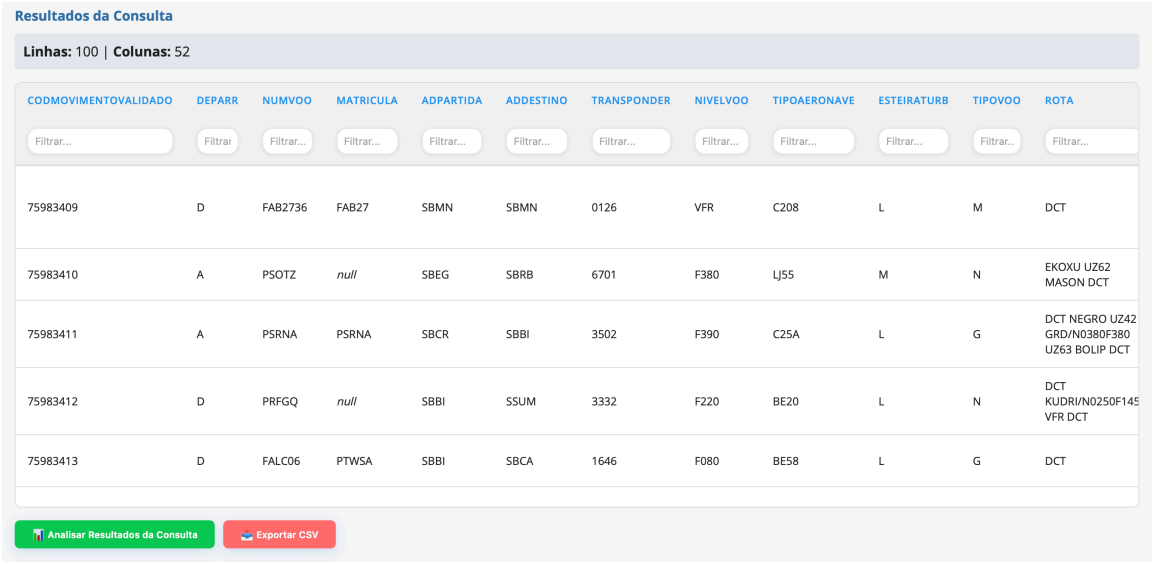

Figura 4.9: Visualização dos resultados de uma consulta SQL customizada, apresentando dados tabulares com paginação e scroll horizontal para colunas extensas.

A tabela de resultados oferece:

- **Visualização Paginada**: Exibição dos dados para consulta em formato de tabela
- **Scroll Horizontal**: Permite navegação em datasets com muitas colunas
- **Exportação *Comma-Separated Values* (CSV)**: Botão dedicado para download dos resultados completos

4.15.4.3 Exportação para CSV

A funcionalidade de exportação permite salvar os resultados de qualquer consulta SQL em formato CSV, compatível com Excel, Python, R e outras ferramentas de análise. O processo de exportação:

1. Mantém a formatação original dos dados sem truncamento
2. Inclui cabeçalhos com nomes das colunas
3. Utiliza codificação UTF-8 para suporte a caracteres especiais
4. Gera arquivo com timestamp no nome (`query_results_YYYYMMDD_HHMMSS.csv`)
5. Inicia download automático no navegador

Esta funcionalidade é essencial para integração com ferramentas externas e workflows de ciência de dados.

4.15.5 Painéis de Score de Qualidade

Após uma execução de consulta SQL, o sistema exibe painéis de KPI com métricas agregadas de qualidade (Figura 4.10).

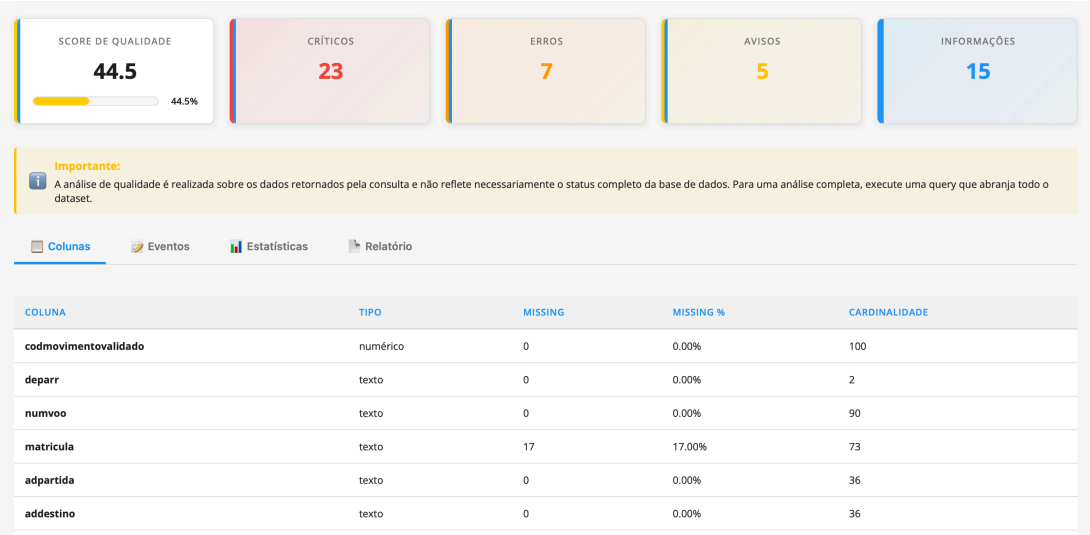

Figura 4.10: Painéis de KPI exibindo score de qualidade e contagem de problemas por severidade, com código de cores para identificação visual rápida.

4.15.5.1 Card de Score de Qualidade

O card principal apresenta o score global de qualidade (0-100), calculado pela fórmula documentada:

$$\text{pontos\_problema} = (\text{CRITICAL} \times 2.0) + (\text{ERROR} \times 1.0) + (\text{WARNING} \times 0.5) \quad (4.4)$$

$$\text{score} = \max\left(0, 100 - \left(\frac{\text{pontos\_problema}}{\text{total\_linhas}} \times 100\right)\right) \quad (4.5)$$

O card inclui barra de progresso visual com código de cores:

- Vermelho (0-40%): Qualidade crítica, dados não recomendados para uso
- Amarelo (40-70%): Qualidade aceitável, requer atenção
- Verde (70-100%): Qualidade boa a excelente, dados confiáveis

4.15.5.2 Cards de Severidade

Quatro cards adicionais apresentam contagens de problemas por nível de severidade:

- **Críticos** (vermelho): Problemas graves que inviabilizam uso dos dados
- **Erros** (laranja): Problemas significativos que comprometem análises
- **Avisos** (amarelo): Problemas leves que requerem atenção
- **Info** (azul): Observações informativas sem impacto crítico

4.15.6 Aba de Catálogo da Base de Dados

A aba Catálogo (Figura 4.11) apresenta informações sobre o crescimento e evolução da base de dados no PostgreSQL, oferecendo visão temporal da expansão do dataset ao longo do tempo.

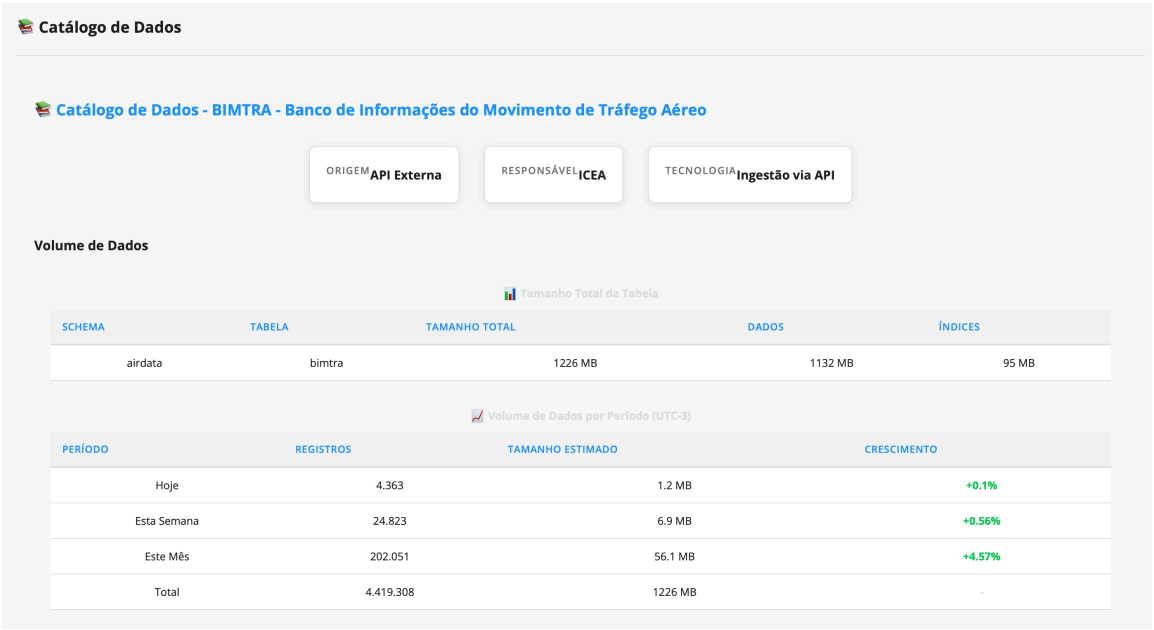

Figura 4.11: Aba de Catálogo exibindo métricas de crescimento da base de dados, incluindo evolução temporal do volume de registros e estatísticas de inserção.

O catálogo apresenta as seguintes métricas de crescimento:

- **Volume Total de Registros**: Contagem atual de linhas armazenadas na tabela PostgreSQL

- **Evolução Temporal**: Número de registros na base ao longo do tempo
- **Tamanho em Disco**: Espaço físico ocupado pela tabela e seus índices no PostgreSQL
- **Crescimento**: Cálculo de crescimento baseado no histórico

Esta aba é fundamental para compreensão da maturidade do dataset, planejamento de capacidade de armazenamento e monitoramento da saúde dos processos de coleta de dados.

4.15.7 Aba de Eventos de Qualidade

A aba "Eventos" (Figura 4.12) lista todos os problemas de qualidade identificados pelos 14 checks de validação, apresentados em ordem de severidade decrescente.

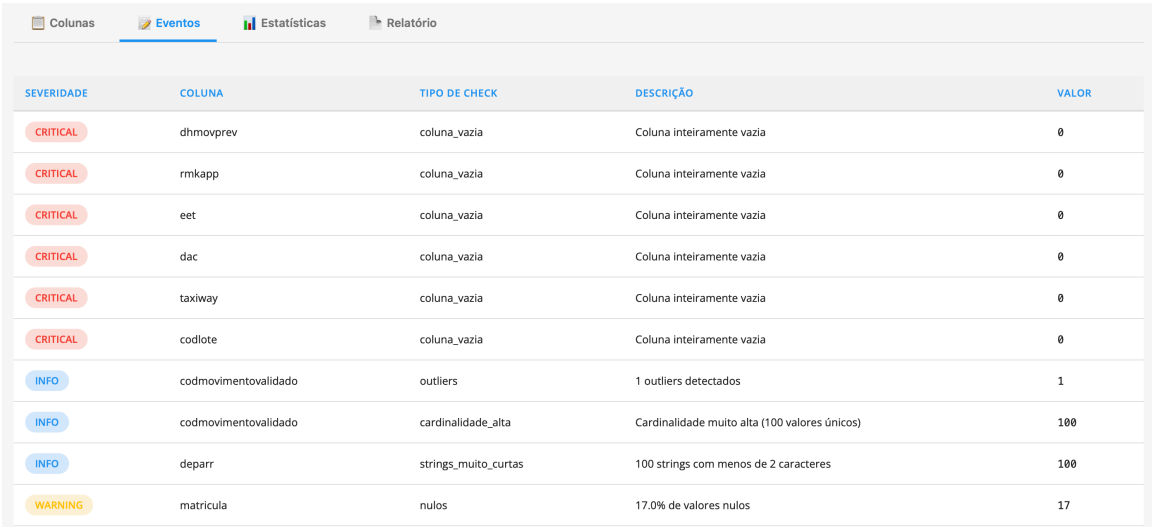

Figura 4.12: Aba de Eventos listando todos os problemas de qualidade detectados, organizados por severidade com descrições detalhadas e valores de exemplo.

Cada evento apresenta:

- **Badge de Severidade**: Cor indicando criticidade (CRITICAL, ERROR, WARNING, INFO)
- **Coluna Afetada**: Nome da coluna onde o problema foi detectado
- **Tipo de Check**: Identificador do check que detectou o problema (`nulos_altos`, `outliers`, etc.)

- **Descrição Detalhada**: Explicação do problema em português
- **Valor de Exemplo**: Amostra do dado problemático quando aplicável

4.15.8 Aba de Estatísticas e Visão Geral da Qualidade

A aba Estatísticas (Figura 4.13) apresenta um painel consolidado com métricas gerais do dataset e visualizações gráficas voltadas à análise de qualidade e estrutura dos dados.

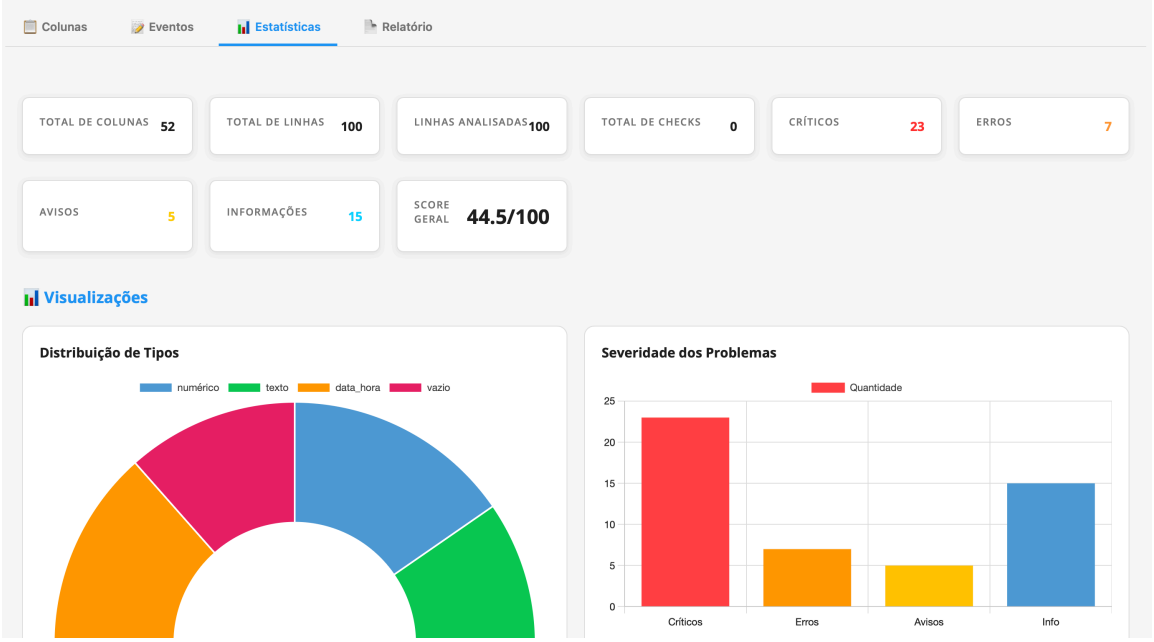

Figura 4.13: Aba de Estatísticas exibindo métricas gerais do dataset, score de qualidade e gráficos de distribuição e severidade.

4.15.8.1 Métricas Gerais do Dataset

O painel superior apresenta os principais indicadores estruturais e de qualidade:

- **Total de Colunas**
- **Total de Linhas**
- **Linhas Analisadas**
- **Total de Checks**
- **Críticos**

- **Erros**
- **Avisos**
- **Informações**
- **Score Geral**

O **Score Geral** sintetiza o nível de qualidade do dataset com base na severidade dos problemas identificados, permitindo avaliação rápida do risco operacional.

4.15.8.2 Visualizações Disponíveis

A aba disponibiliza gráficos que facilitam a interpretação dos dados:

- **Distribuição de Tipos**: gráfico de pizza representando a proporção de tipos de dados no banco (inteiro, decimal, texto, data, etc.).
- **Severidade dos Problemas**: gráfico de barras exibindo a quantidade de ocorrências por nível de severidade (Críticos, Erros, Avisos e Informações).
- **Dados Faltantes por Coluna**: gráfico que evidencia a quantidade ou percentual de valores nulos por coluna, auxiliando na identificação de problemas de completude.
- **Cardinalidade**: visualização da quantidade de valores distintos por coluna, permitindo identificar colunas com alta variabilidade ou possíveis identificadores únicos.

Essa aba fornece uma visão executiva da qualidade e estrutura do dataset, combinando indicadores numéricos consolidados com visualizações gráficas que apoiam a análise diagnóstica.

4.15.9 Aba de Relatório de Checks

A aba Relatório (Figura 4.14) apresenta visão consolidada de todos os 14 checks de qualidade executados, exibindo contadores agregados para o dataset completo.

BIMTRA - Banco de Informações do Movimento de Tráfego Aéreo - Tipos de Checks

TIPO DE CHECK	DESCRIÇÃO	SEVERIDADE
linhas_duplicadas	Linhas completamente duplicadas	WARNING
coluna_vazia	Coluna inteiramente vazia (100% nulos)	CRITICAL
nulos_criticos	Mais de 50% de valores nulos	CRITICAL
nulos_altos	20-50% de valores nulos	ERROR
nulos	5-20% de valores nulos	WARNING
strings_vazias	Valores vazios ou apenas whitespace	WARNING
outliers	Outliers estatísticos detectados (IQR)	INFO
valores_negativos	Valores negativos em colunas numéricas	WARNING
datas_futuras	Datas posteriores à data atual	WARNING
datas_antigas	Datas anteriores a 2000	INFO
strings_muito_curtas	Strings com menos de 2 caracteres	INFO
strings_muito_longas	Strings com mais de 500 caracteres	WARNING
coluna_constante	Apenas 1 valor único (coluna constante)	ERROR
cardinalidade_alta	Cardinalidade muito alta (>90% valores únicos)	INFO

Figura 4.14: Aba de Relatório mostrando contadores agregados para cada um dos 14 tipos de checks de qualidade implementados no sistema.

O relatório organiza os checks em categorias:

- **Checks Estruturais**: `linhas_duplicadas`, `coluna_vazia`, `coluna_constante`
- **Checks de Nulos**: `nulos_criticos`, `nulos_altos`, `nulos`
- **Checks de Texto**: `strings_vazias`, `strings_muito_curtas`, `strings_muito_longas`
- **Checks Numéricos**: `outliers`, `valores_negativos`
- **Checks Temporais**: `datas_futuras`, `datas_antigas`
- **Checks de Distribuição**: `cardinalidade_alta`

Para cada check, exibe-se:

- Nome do check
- Descrição do tipo de check
- Indicador visual da severidade

4.15.10 Aba de Logs de Execução

A aba Logs (Figura 4.15) apresenta um painel gerencial com o histórico das execuções da coleta da base usando o Airflow, permitindo monitoramento operacional e acompanhamento de desempenho.

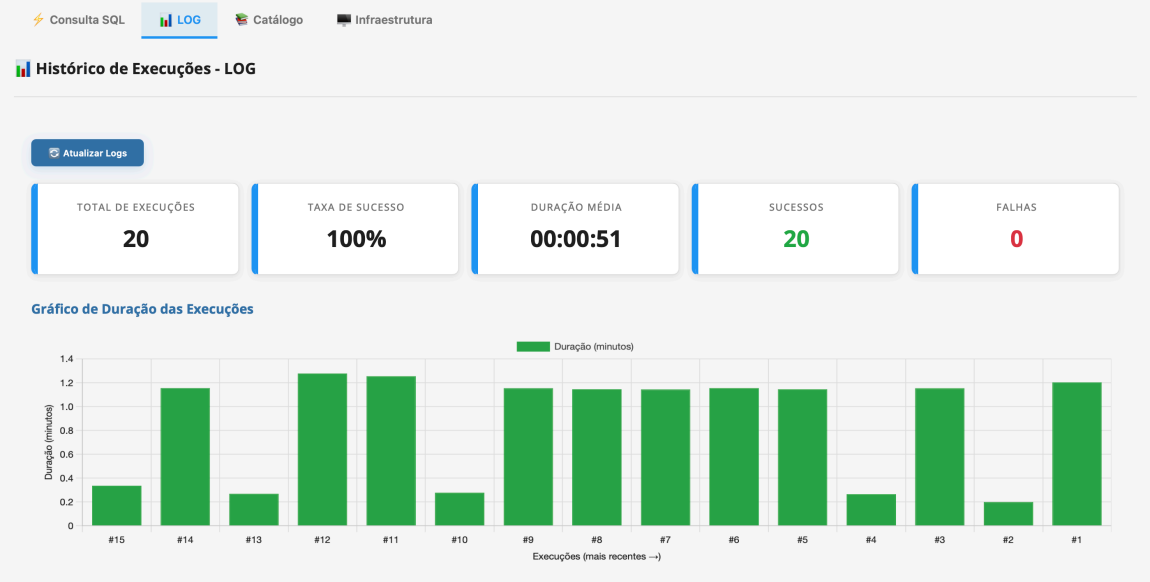

Figura 4.15: Aba de Logs exibindo métricas consolidadas das execuções, gráfico de duração e tabela com histórico recente.

4.15.10.1 Indicadores Gerais de Execução

No topo da aba são apresentados indicadores resumidos de desempenho do processo:

- **Total de Execuções**: quantidade total de validações realizadas.
- **Taxa de Sucesso**: percentual de execuções concluídas com sucesso.
- **Duração Média**: tempo médio gasto por execução.
- **Sucessos**: número absoluto de execuções finalizadas sem erro.
- **Falhas**: número de execuções que apresentaram erro.

Esses indicadores permitem avaliar estabilidade, confiabilidade e eficiência do pipeline de validação.

4.15.10.2 Visualizações Disponíveis

A aba inclui recursos gráficos e tabelares para análise detalhada:

- **Gráfico de Duração das Execuções**: visualização temporal da duração de cada execução, permitindo identificar variações de desempenho ou possíveis degradações ao longo do tempo.
- **Últimas Execuções**: tabela com o histórico recente, contendo colunas como:
  - **Estado** (ex.: sucesso ou falha)
  - **Tipo** (ex.: execução agendada ou manual)
  - **Início**
  - **Fim**
  - **Duração**

Essa aba tem foco em observabilidade operacional, fornecendo transparência sobre o comportamento do sistema de validação e suporte à análise de incidentes ou auditorias internas.

4.15.11 Visualização Geoespacial de Dados OpenSky

Para a base de dados OpenSky (dados de rastreamento de aeronaves via ADS-B), o sistema oferece funcionalidade adicional de visualização geoespacial em mapa interativo (Figura 4.16).

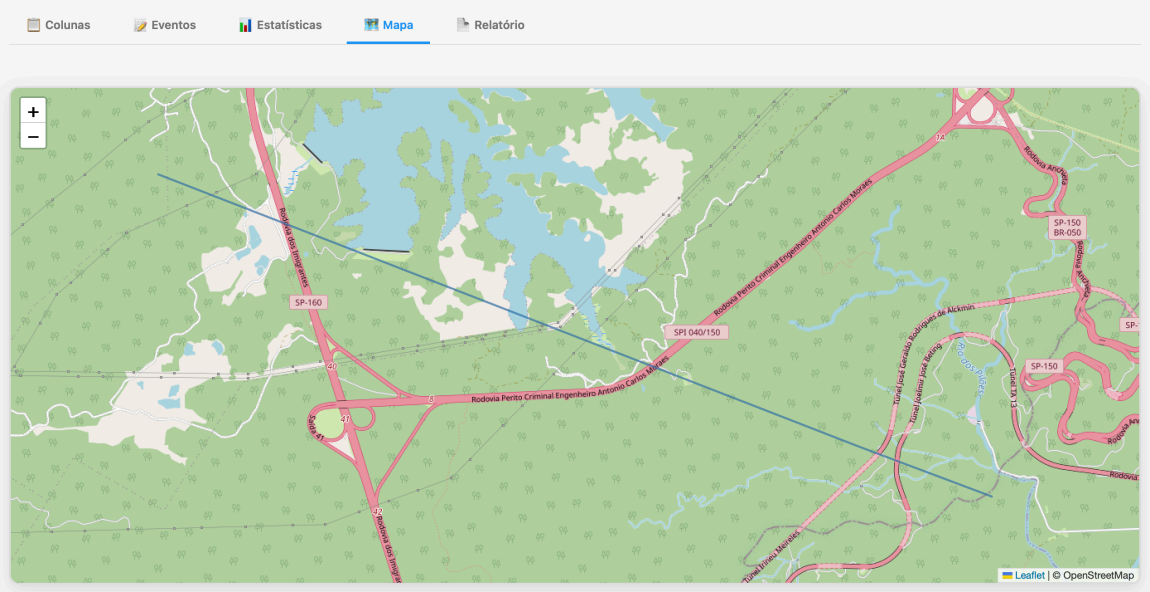

Figura 4.16: Mapa interativo de visualização de dados OpenSky, exibindo trajetórias de aeronaves com informações contextuais.

4.15.11.1 Funcionalidades do Mapa

O mapa geoespacial implementa:

- **Renderização de Trajetórias**: Plotagem de posições sequenciais de aeronaves formando rotas de voo
- **Marcadores Interativos**: Pontos clicáveis exibindo informações detalhadas (call-sign, altitude, velocidade)
- **Filtros Temporais**: Seleção de períodos específicos para análise de tráfego
- **Zoom e Navegação**: Controles para exploração geográfica interativa

Esta funcionalidade é exclusiva para dados com componente geoespacial (latitude/-longitude) e oferece capacidades analíticas avançadas para estudos de fluxo de tráfego aéreo.

4.15.12 Infraestrutura da Máquina

A aba Infraestrutura (Figura 4.17) apresenta métricas operacionais do servidor em tempo real, permitindo monitoramento direto dos recursos computacionais utilizados pela plataforma.

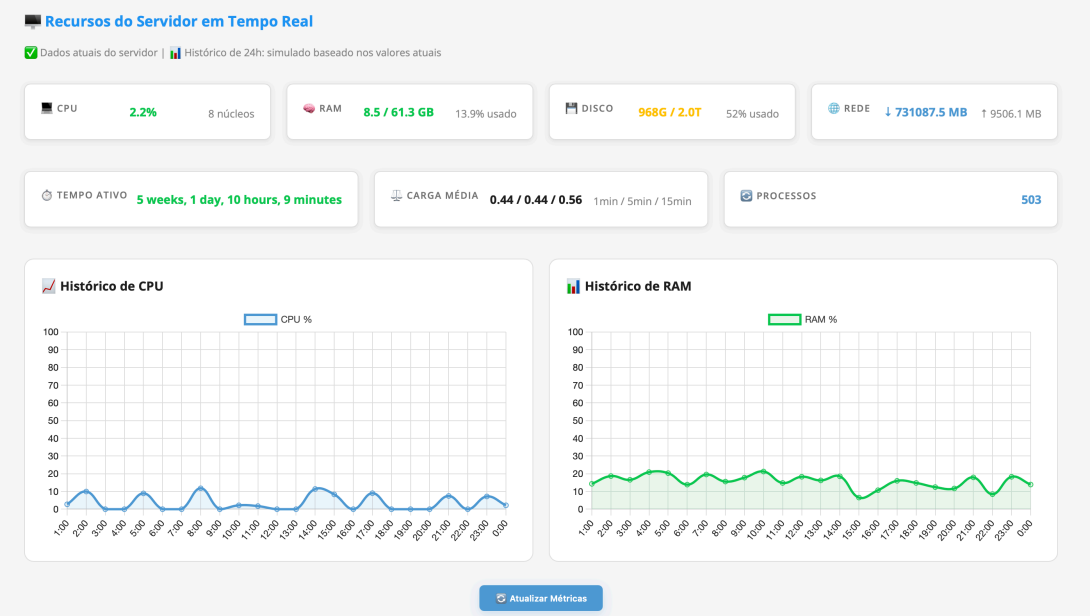

Figura 4.17: Aba de Infraestrutura exibindo métricas atuais do servidor e gráficos históricos de utilização de recursos.

4.15.12.1 Métricas do Servidor em Tempo Real

O painel principal exibe indicadores atualizados automaticamente, incluindo:

- **CPU**: percentual de utilização atual e quantidade de núcleos disponíveis.
- **RAM**: volume total de memória, quantidade utilizada e percentual de uso.
- **Disco**: capacidade total, espaço utilizado e percentual de ocupação.
- **Rede**: volume total de dados trafegados (download e upload).
- **Tempo Ativo**: tempo decorrido desde o último reinício do servidor.
- **Carga Média**: média de carga nos últimos 1, 5 e 15 minutos.
- **Processos**: número total de processos ativos no sistema.

Essas métricas permitem avaliar saúde, estabilidade e capacidade operacional do ambiente.

4.15.12.2 Histórico de Utilização

Além dos dados instantâneos, a aba apresenta visualizações históricas para análise de tendência:

- **Histórico de CPU**: gráfico temporal mostrando variações de uso do processador ao longo das últimas 24 horas.
- **Histórico de RAM**: gráfico temporal exibindo o consumo de memória no mesmo período.

Os gráficos permitem identificar picos de utilização, possíveis gargalos de desempenho e padrões de carga do sistema. Esta aba é voltada ao monitoramento operacional da infraestrutura, oferecendo visibilidade contínua sobre o comportamento dos recursos do servidor.

4.15.13 Responsividade e Compatibilidade

A interface foi desenvolvida seguindo princípios de design responsivo, adaptando-se automaticamente a diferentes resoluções de tela:

- **Desktop** (≥1920px): Layout completo com todas as funcionalidades visíveis
- **Laptop** (1366-1920px): Layout otimizado com scroll vertical mínimo
- **Tablet** (768-1366px): Menu colapsável e tabelas com scroll horizontal

A plataforma foi testada e é compatível com:

- Google Chrome 90+ (recomendado)
- Mozilla Firefox 88+
- Microsoft Edge 90+
- Safari 14+ (macOS)

4.15.14 Segurança e Controle de Acesso

O sistema implementa múltiplas camadas de segurança:

- **Túnel SSH**: Todas as conexões ao banco de dados utilizam túnel SSH criptografado

- **Autenticação PostgreSQL**: Credenciais armazenadas de forma segura no backend
- **Rate Limiting**: Limite de requisições por IP para prevenir abuso

O acesso à plataforma é restrito à rede institucional ou requer VPN para acesso externo.

4.15.15 Performance e Otimizações

A interface implementa diversas otimizações para garantir performance:

- **Virtual Scrolling**: Renderização incremental de grandes tabelas
- **Lazy Loading**: Carregamento assíncrono de abas sob demanda
- **Compressão Gzip**: Redução de payload HTTP em 70-80%
- **Navegação**: Opção de navegação com teclado ou mouse
- **Modo Noturno**: Opção de modo noturno que altera contraste das cores para melhor visualização sob baixa luminosidade

Estas otimizações permitem análise fluida de datasets com milhões de registros.

4.15.16 Casos de Uso Práticos

A interface web suporta diversos workflows analíticos:

4.15.16.1 Workflow 1: Validação Rápida de Coleta

1. Acessar base de dados desejada no menu principal
2. Fazer busca SQL nos últimos 100 registros
3. Clicar em ”Analisar”
4. Verificar score de qualidade nos KPIs
5. Revisar eventos críticos na aba de Eventos
6. Exportar CSV para auditoria

4.15.16.2 *Workflow 2: Análise Exploratória com SQL*

1. Acessar base de dados no menu
2. Digitar query SQL customizada no campo de busca
3. Executar consulta
4. Analisar estatísticas descritivas na aba Estatísticas
5. Identificar outliers na aba de Eventos
6. Exportar resultados em CSV para análise externa

4.15.16.3 *Workflow 3: Monitoramento de Qualidade Temporal*

1. Executar consulta SQL com filtro temporal (últimos 7 dias)
2. Comparar score de qualidade com período anterior
3. Analisar tendências de problemas no relatório de checks
4. Exportar CSVs para análise de séries temporais
5. Gerar relatório executivo para stakeholders

5 Considerações Finais

O Produto II do projeto **AirData** consolida a transição de um conjunto de bases de dados aeronáuticos e meteorológicos isolados para um Sistema de Integração de Dados automatizado. Ao longo do presente relatório, foram apresentados os componentes que compõem a plataforma, evidenciando que a construção de um sistema analítico confiável para a aviação civil requer, além do armazenamento de informações, orquestração de processos, representação semântica do domínio, mecanismos de consulta inteligente e garantia sistemática de qualidade dos dados.

A arquitetura desenvolvida está organizada em quatro componentes principais, descritos nos capítulos anteriores:

- **Infraestrutura ETL (Capítulo 1):** O ambiente de produção foi implantado na máquina Lessonia, utilizando o Apache Airflow como orquestrador de pipelines e o PostgreSQL como Data Warehouse. Essa infraestrutura é responsável pela extração, transformação e carga contínua dos dados, com versionamento e escalabilidade assegurados pelo fluxo de desenvolvimento adotado.
- **Ontologia AirData (Capítulo 2):** Foi desenvolvida uma ontologia OWL para representação formal dos conceitos do domínio aeronáutico. A ontologia, submetida a validações sintáticas e lógicas automatizadas, estabelece um vocabulário padronizado que relaciona aeronaves, aeródromos e regras operacionais, garantindo interoperabilidade com padrões internacionais da ICAO e possibilitando enriquecimento semântico dos dados.
- **AirData RAG System (Capítulo 3):** Foi implementado um sistema de consulta baseado em arquitetura RAG (*Retrieval-Augmented Generation*), executado localmente com modelos LLM via Ollama e busca vetorial no Qdrant. O sistema permite a interpretação de normas aeronáuticas (ICAs, RBACs, RBHAs) e fornece respostas fundamentadas em fontes oficiais verificáveis, com filtragem por vigência temporal.
- **AirData Data Check (Capítulo 4):** Foi desenvolvido um subsistema de validação e monitoramento de qualidade de dados. O sistema opera sobre 18 bases de dados (incluindo VRA, BIMTRA, SIROS, METAR e ERA5), aplicando regras

de completude, validade física, consistência temporal e conformidade aeronáutica. Os resultados são disponibilizados por meio de um painel web interativo e de um sistema de escore quantitativo de qualidade.

5.1 Estágio Atual de Integração

Os quatro componentes descritos foram desenvolvidos e validados de forma independente nesta fase do projeto. A infraestrutura ETL e o Data Check já operam de maneira integrada: o Airflow (Capítulo 1) executa a extração e carga dos dados, e o Data Check (Capítulo 4) realiza a validação de integridade física e temporal sobre as bases ingeridas.

A Ontologia (Capítulo 2) e o sistema RAG (Capítulo 3), por sua vez, foram implementados como subsistemas autônomos. A Ontologia fornece a representação semântica do domínio, e o sistema RAG permite consultas em linguagem natural sobre a base normativa indexada. Embora ambos os componentes estejam operacionais, sua conexão ao pipeline de orquestração e ao fluxo de validação de dados será objeto do Produto III (Sistema de Enriquecimento de Dados e Protótipo de Consultas), no qual estão previstos o enriquecimento semântico dos dados integrados, a correlação de eventos entre bases distintas e a disponibilização de uma interface unificada de consulta por voo.

5.2 Próximos Passos

Com a infraestrutura de extração estabilizada, o conhecimento normativo indexado e as métricas de qualidade de dados em operação, a plataforma atingiu sua maturidade estrutural. O Produto III (Sistema de Enriquecimento de Dados e Protótipo de Consultas) dará continuidade ao desenvolvimento com foco nos seguintes eixos:

- A integração entre a Ontologia, o sistema RAG e o pipeline ETL, viabilizando o enriquecimento semântico dos dados e a correlação de eventos entre bases distintas.
- A implementação de um protótipo de interface de consultas que permita a visualização unificada dos eventos associados a um determinado voo.
- A expansão da ontologia para cobertura de todos os conceitos das 18 bases de dados integradas.

- A evolução da arquitetura RAG, incluindo o *fine-tuning* supervisionado dos modelos de linguagem.

Em síntese, o Produto II constitui um sistema de dados integrado, auditável e preparado para subsidiar as etapas subsequentes de modelagem preditiva e inteligência aplicada à aviação civil.