

André Solano Ferreira Rodrigues Maiolini (19.02012-0)

Durval Consorti Soranz de Barros Santos (22.01097-0)

Leonardo Roberto Amadio (22.01300-8)

Lucas Castanho Paganotto Carvalho (22.00921-3)

Plataforma de BI para Engenharia - Análise de Motores Turbofan (C-MAPSS)

São Caetano Sul – SP

2025

André Solano Ferreira Rodrigues Maiolini (19.02012-0)

Durval Consorti Soranz de Barros Santos (22.01097-0)

Leonardo Roberto Amadio (22.01300-8)

Lucas Castanho Paganotto Carvalho (22.00921-3)

Plataforma de BI para Engenharia - Análise de Motores Turbofan (C-MAPSS)

Trabalho apresentado para aprovação na disciplina ECM401 - Banco de Dados, do Instituto Mauá de Tecnologia.

Professor: Me. Antônio Fernando Nunes Guardado

Instituto Mauá de Tecnologia
Engenharia de Computação

São Caetano Sul – SP
2025

Resumo

Este projeto propõe o desenvolvimento de uma plataforma de Business Intelligence voltada à Engenharia de Confiabilidade, utilizando o dataset C-MAPSS da NASA. O objetivo é demonstrar como dados de sensores de motores turbofan — organizados como séries temporais — podem ser transformados em indicadores que auxiliem engenheiros e gestores na avaliação de degradação, desempenho e vida útil de componentes. O foco do BI (*Business Intelligence*) não é prever falhas via aprendizado de máquina, mas monitorar, contextualizar e analisar o comportamento dos motores em diferentes cenários de teste, oferecendo uma base quantitativa para decisões de engenharia, como ajustes de projeto e definição de políticas de manutenção.

Palavras-chave: Business Intelligence. Engenharia de Confiabilidade. DataMart. C-MAPSS. Manutenção Preditiva. Aeronáutica.

Sumário

1	INTRODUÇÃO	5
1.1	IoT, Big Data e a Evolução da Manutenção	5
1.2	Objetivos	6
1.2.1	Objetivo Geral	6
1.2.2	Objetivos Específicos	6
1.3	Perguntas-Chave para o BI	7
2	METODOLOGIA E DESENVOLVIMENTO	8
2.1	Modelo Lógico Relacional (OLTP)	8
2.1.1	Tabelas do Modelo OLTP	9
2.2	Dataset NASA C-MAPSS	9
2.2.1	Modelo de Simulação	9
2.2.2	Fundamentos Termodinâmicos e Condições Operacionais	10
2.2.3	Estrutura e Cenários de Teste	12
2.2.4	Estrutura dos Dados	13
2.2.5	Pré-processamento e Modelagem Dimensional	14
2.3	Arquitetura de Dados e Stack de Tecnologias	15
2.3.1	Stack de Tecnologias Utilizada	15
2.3.2	Modelo Lógico e Dimensional (DataMart)	15
2.4	Processo de Extração, Transformação e Carga (ETL)	18
2.4.1	Camada Bronze: Ingestão de Dados Brutos	18
2.4.2	Camada Silver: Limpeza e Enriquecimento	18
2.4.3	Definição e Hipótese do KPI	19
2.4.3.1	BI como Consumidor	19
2.5	Infraestrutura e Containerização	20
2.5.1	Orquestração da Execução	20
2.6	Consultas Analíticas	21
3	RESULTADOS E ANÁLISE	22
3.1	Escala Microscópica (Diagnóstico do Motor)	22

3.2	Escala Mesoscópica (Análise Comparativa)	23
3.3	Escala Macroscópica (Visão Global)	24
	Conclusão	25
	REFERÊNCIAS	26

1 Introdução

Este projeto propõe o desenvolvimento de uma plataforma de Business Intelligence (BI) voltada à Engenharia de Confiabilidade, utilizando o dataset C-MAPSS da NASA. O objetivo é demonstrar como dados de sensores de motores turbofan — organizados como séries temporais — podem ser transformados em indicadores que auxiliem engenheiros e gestores na avaliação de degradação, desempenho e vida útil de componentes. O foco do BI não é prever falhas via aprendizado de máquina, mas monitorar, contextualizar e analisar o comportamento dos motores em diferentes cenários de teste, oferecendo uma base quantitativa para decisões de engenharia, como ajustes de projeto e definição de políticas de manutenção.

1.1 IoT, Big Data e a Evolução da Manutenção

A indústria aeronáutica moderna é uma das maiores geradoras de dados do mundo. Uma única aeronave comercial pode gerar terabytes de dados por voo, coletados por milhares de sensores embarcados em uma arquitetura complexa de Internet das Coisas (IoT) (1, 2). Esses dados representam um desafio de Big Data, mas também uma oportunidade sem precedentes para otimizar operações.

Um exemplo prático e ilustrativo do volume de dados gerados no contexto aeronáutico pode ser observado em uma aeronave Boeing 747 (2). Essa aeronave possui dois motores, sendo que cada um deles é capaz de gerar aproximadamente 20 terabytes de informações por hora de voo. Conforme ilustrado na Figura 1, em um trajeto de seis horas, estima-se a produção de cerca de 240 terabytes de dados por aeronave. Quando esse valor é extrapolado para a média de 28.537 voos comerciais diários realizados nos Estados Unidos, o resultado é um volume anual superior a 2,4 bilhões de terabytes. Esse cenário evidencia o enorme potencial das aeronaves comerciais como fontes de dados em larga escala, reforçando a importância do uso de técnicas de Big Data e Internet das Coisas (IoT) para o processamento, análise e aproveitamento dessas informações.

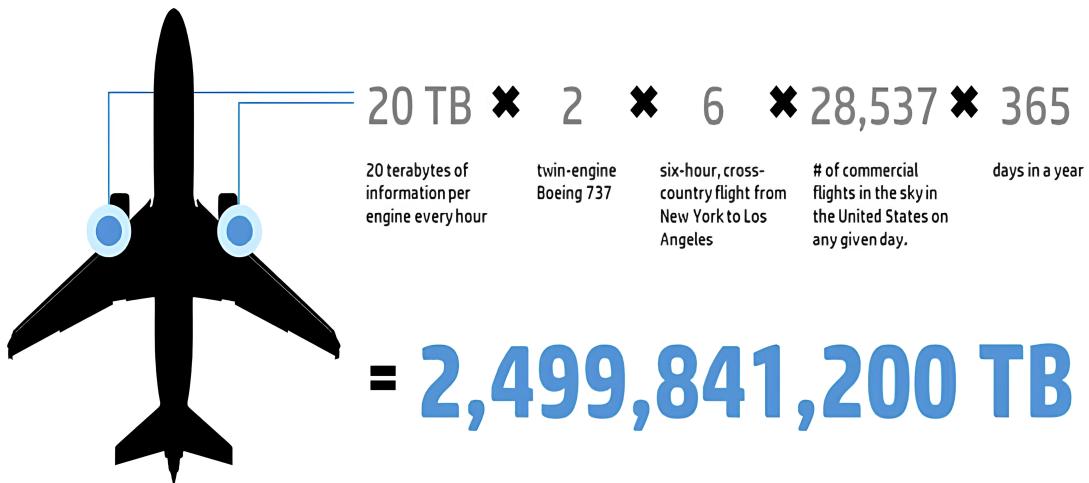


Figura 1 – Dados de sensores em voos entre Nova Iorque e Los Angeles.

Historicamente, a manutenção aeronáutica era dominada pela abordagem conservadora baseada em tempo (TBM - *Time-Based Maintenance*), onde componentes são substituídos após um número fixo de horas de voo, independentemente de sua condição real. Esta abordagem, embora segura, é extremamente cara e ineficiente - significando a troca de peças que estariam ainda adequadas para o uso.

A disponibilidade desses vastos conjuntos de dados impulsiona a transição para a Manutenção Baseada na Condição (CBM - *Condition-Based Maintenance*) e para a Manutenção Preditiva (PdM). O BI é a ferramenta-chave que permite às equipes de engenharia consumir esse volume de dados e identificar os "sintomas digitais" de degradação antes que uma falha ocorra.

1.2 Objetivos

1.2.1 Objetivo Geral

Construir um DataMart e um conjunto de dashboards de BI que consolidem dados de séries temporais do C-MAPSS em indicadores de confiabilidade, permitindo análises comparativas de degradação e suporte a decisões de engenharia.

1.2.2 Objetivos Específicos

- Integrar os quatro subconjuntos do C-MAPSS (FD001–FD004) em uma única base analítica, identificando o cenário de teste como uma dimensão de análise.

- Modelar um DataMart com foco em engenharia, contendo dimensões como Motor, Cenário de Teste, Sensor, Tempo e Condição Operacional.
- Implementar consultas analíticas e dashboards para apoiar decisões de engenharia, tais como identificação de sensores críticos, comparação entre regimes operacionais e análise de tendências de falha.

1.3 Perguntas-Chave para o BI

O sucesso da plataforma de BI será medido pela sua capacidade de responder às seguintes perguntas de engenharia:

- Qual a temperatura média global e por subnível de altitude? (*Motores operam em altitudes variadas, e a temperatura de saída da turbina muda naturalmente com a altitude.*)
- Quais são os motores mais duráveis de cada cenário? (*Identifica os motores excepcionais e aqueles problemáticos.*)
- Qual o impacto térmico incremental ao subir a potência? (*Mede a sensibilidade térmica, um motor degradado pode ter um pico térmico desproporcional ao subir a potência.*)
- Qual o desvio de performance em relação ao motor ideal? (*Realiza um Benchmarking. A query pega o motor mais "frio" (mais eficiente) do cenário e o define como o líder. Em seguida, calcula quantos graus acima desse líder cada outro motor está operando.*)
- Quais zonas de altitude causam maior estresse de rotação? (*Descobre se voar em certas altitudes força o motor a girar muito rápido para manter o empuxo. Isso pode informar pilotos ou o planejamento de voo para evitar certas altitudes, estendendo a vida útil dos componentes rotativos.*)
- Qual o ciclo de falha médio para cada configuração? (*Calcula o MTBF (Mean Time Between Failures) esperado. A query conta quantos ciclos os motores aguentaram em média para cada configuração.*)

2 Metodologia e Desenvolvimento

A construção da plataforma de Business Intelligence (BI) proposta neste trabalho seguirá as etapas clássicas de um projeto de Data Warehouse, conforme as abordagens de Kimball e Inmon, adaptadas ao contexto de engenharia de confiabilidade e ao domínio do conjunto de dados C-MAPSS. A metodologia engloba desde a compreensão do domínio e estrutura dos dados até sua modelagem dimensional e integração em um ambiente analítico, possibilitando o monitoramento da saúde de frota de motores turbofan simulados.

2.1 Modelo Lógico Relacional (OLTP)

O Modelo Lógico Relacional representa a base operacional de onde os dados do C-MAPSS seriam extraídos. Ele foi normalizado para a Terceira Forma Normal (3NF) para garantir a integridade e otimizar as transações. O script SQL que gera o OLTP pode ser encontrado na pasta 'bronze' dentro da pasta 'sql'. O diagrama, conforme gerado pelo MySQL Workbench, é apresentado na Figura 2.

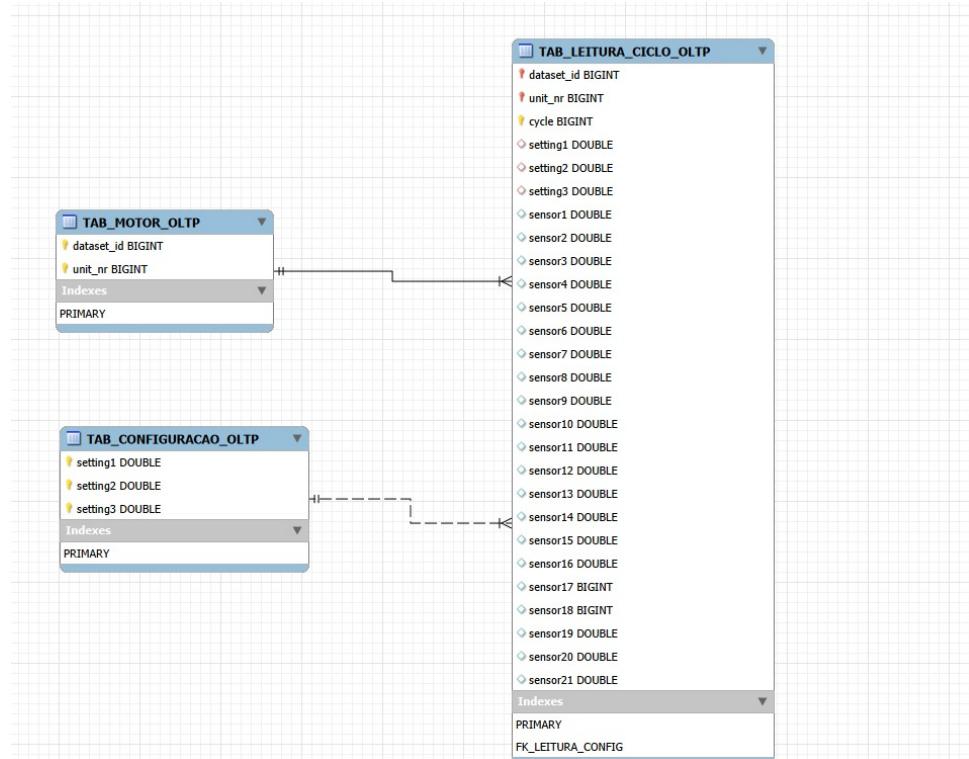


Figura 2 – Modelo Lógico Relacional (OLTP) da Base Operacional C-MAPSS

2.1.1 Tabelas do Modelo OLTP

- **TAB_MOTOR_DLTP:** Armazena dados estáticos do motor, identificado por sua chave natural (dataset_id, unit_nr).
- **TAB_CONFIGURACAO_DLTP:** Armazena as condições operacionais únicas (setting1, setting2, setting3).
- **TAB_LEITURA_CICLO_DLTP:** Tabela de fatos transacionais que registra as leituras dos sensores em um ciclo específico, referenciando as tabelas de motor e configuração.

2.2 Dataset NASA C-MAPSS

O conjunto de dados C-MAPSS (Commercial Modular Aero-Propulsion System Simulation), desenvolvido pela NASA, é amplamente reconhecido como referência para pesquisas em Prognóstico e Gerenciamento de Saúde (PHM — *Prognostics and Health Management*). Ele foi gerado por um modelo de simulação de alta fidelidade de motores turbofan comerciais, conforme descrito em (3).

O modelo considera as principais dinâmicas termodinâmicas e mecânicas do motor, permitindo simular o comportamento de seus componentes sob diferentes condições de operação e modos de falha. Esses dados possibilitam o desenvolvimento e teste de algoritmos de previsão de falhas, estimativa de vida útil restante (RUL) e suporte à manutenção preditiva.

2.2.1 Modelo de Simulação

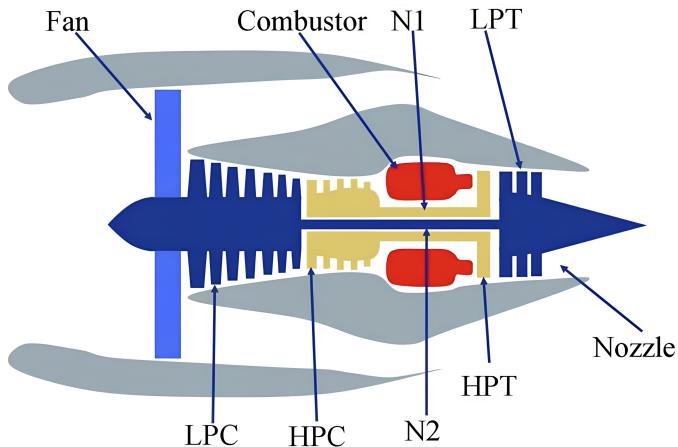


Figura 3 – Diagrama simplificado do motor turbofan utilizado na simulação.

O modelo representado na Figura 3 simula um motor turbofan de dois eixos (N1 e N2), composto por fan, compressores, câmaras de combustão e turbinas, incluindo sensores distribuídos ao longo de suas seções críticas. O processo de degradação dos componentes é modelado por meio de parâmetros de desempenho degradantes, como eficiência e vazão, que variam ao longo dos ciclos simulados.

2.2.2 Fundamentos Termodinâmicos e Condições Operacionais

O motor turbofan, assim como outros sistemas de propulsão a jato, opera segundo o **Ciclo de Brayton**, também conhecido como turbina a gás. Esse ciclo termodinâmico descreve o processo de conversão de energia química em energia mecânica e, posteriormente, em empuxo. Ele é composto por quatro etapas principais (Figura 4):

1. **Compressão (adiabático):** O ar atmosférico é comprimido pelo fan e pelos compressores de baixa (LPC) e alta pressão (HPC), elevando sua pressão e temperatura.
2. **Combustão (isobárico):** O combustível é injetado e queimado na câmara de combustão, fornecendo energia térmica ao fluxo de ar comprimido.
3. **Expansão (adiabático):** Os gases quentes resultantes passam pelas turbinas de alta (HPT) e baixa pressão (LPT), que extraem energia para acionar os compressores e o fan.
4. **Exaustão (isobárico):** O ar expandido é expelido a alta velocidade, gerando empuxo conforme a Terceira Lei de Newton.

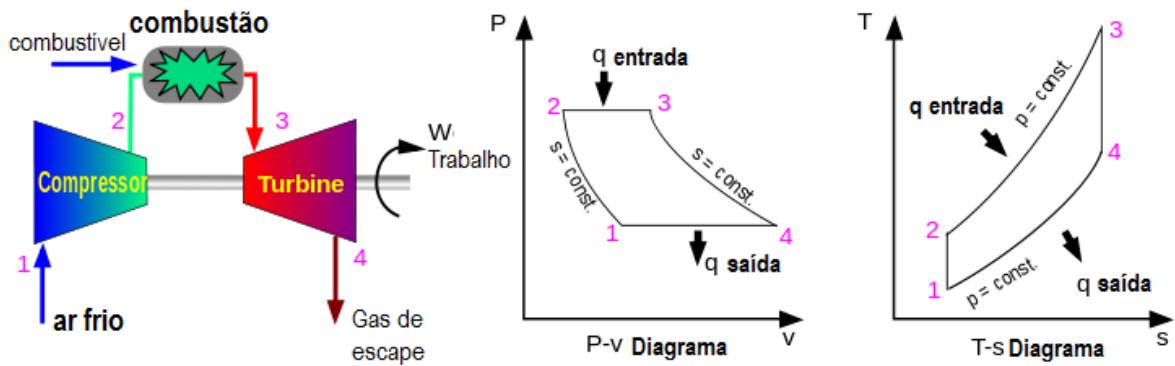


Figura 4 – Ciclo de Brayton (turbina a gás).

A eficiência e o desempenho desse ciclo são fortemente influenciados pelas **condições operacionais externas**, representadas nas variáveis de configuração do dataset C-MAPSS — **altitude, número de Mach e posição da manete de potência**.

Essas variáveis refletem o regime de voo e determinam o ponto de operação do motor:

- **Altitude:** Está relacionada à densidade do ar. Em maiores altitudes, a densidade atmosférica diminui, reduzindo a massa de ar admitida pelo fan e, consequentemente, a eficiência do ciclo.

– O Modelo de Atmosfera Padrão (ISA - International Standard Atmosphere):

Modelo utilizado para padronizar os cálculos de performance de aeronaves e motores. Este modelo matemático define como as propriedades do ar — pressão, temperatura e densidade — variam com a altitude. Assim, a altitude de operação não é apenas um valor geométrico, mas uma entrada direta para determinar o estado termodinâmico do ar que o motor irá admitir.

- **Número de Mach (Ma):** Define a razão entre a velocidade da aeronave (V) e a velocidade local do som (a), conforme $Ma = V/a$. A velocidade local do som, por sua vez, não é constante; ela depende do estado termodinâmico do ar, especificamente de sua temperatura absoluta (T), sendo calculada por:

$$a = \sqrt{\gamma RT}$$

onde γ é a razão dos calores específicos (adiabática, 1,4 para o ar) e R é a constante específica do gás (287 J/kg·K para o ar seco). O número de Mach está diretamente ligado à compressão dinâmica do ar na entrada do motor e à eficiência aerodinâmica das pás.

- **Posição da Manete de Potência (*Throttle Resolver Angle — TRA*):** Representa o comando do piloto sobre a potência de empuxo. Um aumento no ângulo de manete eleva a vazão mássica de combustível (W_f) e as temperaturas de operação, intensificando o desgaste dos componentes térmicos.

Dessa forma, as variáveis de configuração operacional não apenas contextualizam as leituras dos sensores, mas também influenciam diretamente o comportamento de degradação simulado nos diferentes cenários do C-MAPSS. Essa relação é essencial para compreender as variações observadas nos subconjuntos FD001–FD004 e justifica sua modelagem como dimensões analíticas no *Data Mart* proposto.

2.2.3 Estrutura e Cenários de Teste

O conjunto de dados C-MAPSS é composto por quatro subconjuntos — **FD001, FD002, FD003** e **FD004** —, cada um simulando uma frota de motores turbofan sob diferentes condições operacionais e modos de falha. Essa segmentação foi concebida para permitir a avaliação de algoritmos de prognóstico em diferentes níveis de complexidade e representa a principal dimensão de análise adotada neste projeto.

- **FD001:** Um único modo de falha (degradação do compressor de alta pressão — HPC) sob uma condição operacional constante.
- **FD002:** Um único modo de falha (HPC), porém sob múltiplas condições operacionais (seis regimes distintos de operação).
- **FD003:** Dois modos de falha (HPC e fan) sob condição operacional constante.
- **FD004:** Dois modos de falha (HPC e fan) sob múltiplas condições operacionais (seis regimes distintos).

Essa estrutura hierárquica torna o C-MAPSS um conjunto de dados ideal para estudos de confiabilidade e manutenção preditiva, pois permite analisar como o comportamento dos sensores e a degradação do motor variam conforme o tipo de falha e o ambiente de operação.

A integração dos quatro subconjuntos em um único *Data Mart* possibilitará análises comparativas de alto valor, permitindo observar o impacto combinado das condições de operação e dos modos de falha na vida útil dos motores.

2.2.4 Estrutura dos Dados

Cada linha do conjunto de dados representa um ciclo de operação de um motor e contém 26 colunas, classificadas em três categorias principais:

- **Identificação:** Inclui o ID do motor e o número do ciclo correspondente, que funcionam como chaves primárias para rastrear o histórico de cada unidade ao longo do tempo (ciclos de operação).
- **Configurações Operacionais (3 variáveis):** Representam o regime de voo e o contexto operacional do motor — por exemplo, altitude, número de Mach e posição do manete de potência. Essas variáveis determinam o ponto de operação no qual o motor está sendo simulado.
- **Leituras de Sensores (21 variáveis):** Capturam a resposta física e termodinâmica do motor durante cada ciclo de operação. Entre elas, destacam-se:
 - **Temperaturas (T2, T24, T30, T50):** Medem as temperaturas totais em diferentes estágios do motor, conforme a numeração do ponto de medição padrão em sistemas de propulsão aeronáutica:
 - * **T2:** Temperatura na entrada do fan (após a admissão de ar).
 - * **T24:** Temperatura após o **LPC** — *Low Pressure Compressor* (compressor de baixa pressão).
 - * **T30:** Temperatura após o **HPC** — *High Pressure Compressor* (compressor de alta pressão).
 - * **T50:** Temperatura após o **LPT** — *Low Pressure Turbine* (turbina de baixa pressão).
 - **Pressões (P2, P15, P30):** Representam pressões totais em diferentes seções:
 - * **P2:** Pressão total na entrada do fan.
 - * **P15:** Pressão na região de bypass (entre o fan e o compressor de baixa pressão).

* **P30:** Pressão total após o HPC (compressor de alta pressão).

– **Velocidades de Rotação (Nf, Nc):**

* **Nf:** Velocidade de rotação do eixo do fan — eixo de baixa pressão.

* **Nc:** Velocidade de rotação do eixo do núcleo (**core**) — eixo de alta pressão.

– **Outros Indicadores:** Incluem variáveis como:

* **Wf:** Vazão mássica de combustível (*Fuel Flow*).

* **EPR:** Razão de pressão do motor (*Engine Pressure Ratio*), definida como a razão entre a pressão total na saída e na entrada do motor.

* Razões de fluxo e parâmetros derivados de eficiência, utilizados para monitoramento de desempenho.

A análise combinada dessas medições ao longo dos ciclos permite identificar padrões sutis de degradação associados ao desgaste progressivo dos componentes do compressor e da turbina. Em particular, o acompanhamento das variáveis relacionadas ao **LPT**, **HPC** e ao **fan** é fundamental, já que ambos os componentes estão associados aos principais modos de falha simulados no C-MAPSS.

2.2.5 Pré-processamento e Modelagem Dimensional

Para viabilizar o uso dos dados na plataforma de BI, serão aplicadas etapas de preparação que incluem:

1. **Limpeza e Padronização:** Verificação de consistência e formatação das variáveis conforme seus tipos físicos e unidades de medida.
2. **Modelagem Dimensional:** Construção de um *Data Mart* voltado à análise de saúde da frota de motores, com tabelas fato (leituras por ciclo) e dimensões (motor, cenário, modo de falha, sensor e tempo ou ciclo).
3. **Integração com a Camada de BI:** Conexão do modelo ao ambiente analítico, permitindo visualizações interativas e consultas exploratórias sobre os sensores e parâmetros operacionais.

2.3 Arquitetura de Dados e Stack de Tecnologias

A construção da plataforma de BI segue uma arquitetura de dados moderna baseada no conceito **Medallion** (Bronze, Silver, Gold) - Figura 5. Esta abordagem organiza o fluxo de tratamento dos dados em etapas claras de qualidade e transformação, garantindo que os dados consumidos na camada final de BI sejam limpos, integrados e confiáveis.

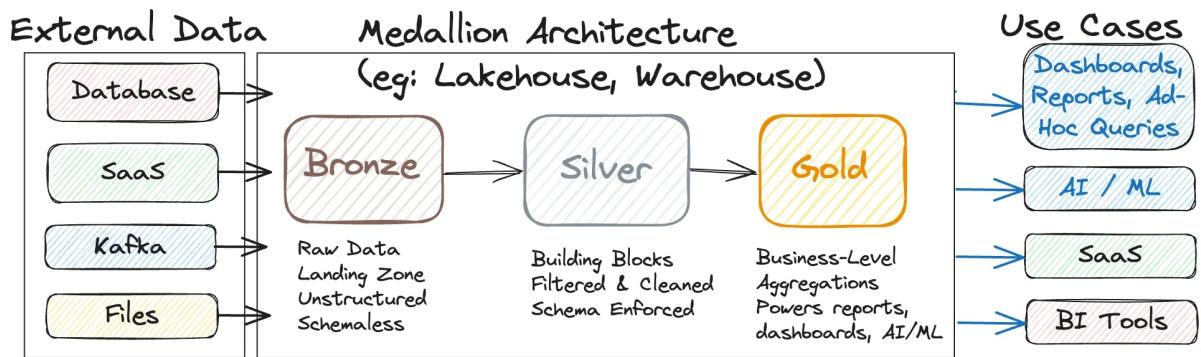


Figura 5 – Arquitetura Medalhão.

2.3.1 Stack de Tecnologias Utilizada

A implementação desta arquitetura foi realizada utilizando a seguinte stack de tecnologias:

- **Linguagem de Programação (ETL):** Python 3, com uso da biblioteca pandas e PySpark para manipulação, transformação e processamento dos dados.
- **Banco de Dados (Data Warehouse):** MySQL, um sistema de gerenciamento de banco de dados relacional (SGBDR) objeto-relacional, escolhido para armazenar as duas primeiras camadas do Data Warehouse (Bronze e Silver).
- **Ferramenta de BI (Consumo):** O destino final dos dados (Camada Gold) é ser consumido por uma ferramenta de visualização como Microsoft Power BI, onde os dashboards de engenharia serão construídos.

2.3.2 Modelo Lógico e Dimensional (DataMart)

A **Camada Silver** funcionará como o DataMart dimensional. Esta camada será modelada em um esquema *Star Schema*, otimizado para consultas analíticas de BI.

Ela será composta por:

- **Tabela Fato:** Uma tabela central (ex: Fato_Leituras) contendo as métricas principais (leituras de sensores etc.) e as chaves estrangeiras para as dimensões.
- **Tabelas de Dimensão:** Tabelas que fornecem o contexto para a análise (ex: Dim_Motor, Dim_Configuracao, Dim_Ciclo, Dim_Descricao_Sensores).

A arquitetura do Data Mart pode ser vista no diagrama ilustrado na Figura 6.

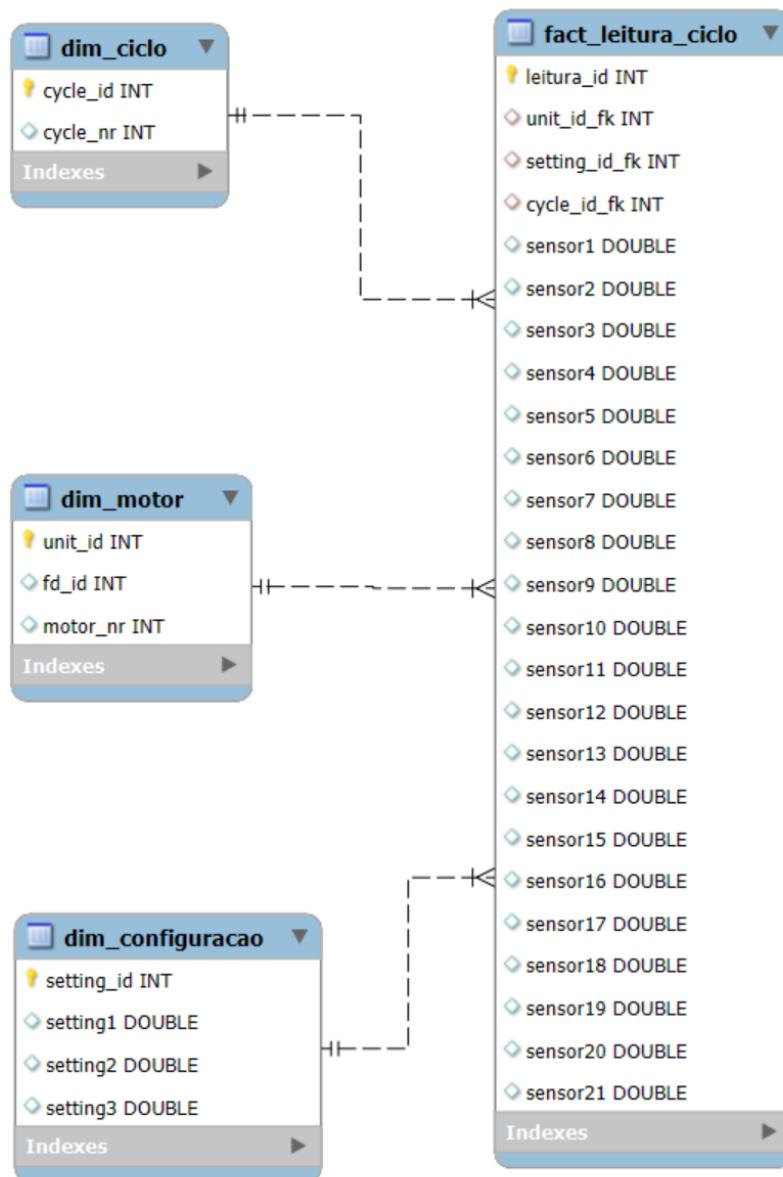


Figura 6 – Arquitetura do Data Mart.

Posteriormente, na **Camada Gold**, é adicionada a dimensão de descrição de sensores, para auxiliar na interpretação de *Business Intelligence* (BI) no *Microsoft PowerBI* - conforme reforçado na Tabela 1.

Tabela 1 – Descrição dos Sensores Utilizados no Modelo Termodinâmico

Nome_Sensor	Descrição do Sensor	Unidade
sensor1	T2 - Temp. Entrada Fan	°R
sensor2	T24 - Temp. Saída LPC	°R
sensor3	T30 - Temp. Saída HPC	°R
sensor4	T50 - Temp. Saída LPT	°R
sensor5	P2 - Pressão Entrada Fan	psia
sensor6	P15 - Pressão Bypass	psia
sensor7	P30 - Pressão Total HPC	psia
sensor8	Nf - Rotação Fan	rpm
sensor9	Nc - Rotação Core	rpm
sensor10	EPR - Razão de Pressão	-
sensor11	Ps30 - Pressão Estática HPC	psia
sensor12	phi - Razão Combustível/Ar	-
sensor13	NRf - Rotação Fan (Corrigida)	rpm
sensor14	NRc - Rotação Core (Corrigida)	rpm
sensor15	BPR - Razão de Bypass	-
sensor16	farB - Razão Combust./Ar (Queimador)	-
sensor17	htBleed - Entalpia de Sangria	BTU/lbm
sensor18	Nf_dmd - Rotação Fan (Demandada)	rpm
sensor19	PCNfR_dmd - Rotação Fan Corr. (Demandada)	rpm
sensor20	HPT - Refrigeração	lbm/s
sensor21	LPT - Refrigeração	lbm/s

Além disso, nessa mesma fase, a tabela `fact_leitura_ciclo` foi transformada, gerando linhas a partir das colunas dos diferentes sensores - isso permite que os sensores sejam filtrados nas análises, corretamente.

2.4 Processo de Extração, Transformação e Carga (ETL)

O processo de ETL é o núcleo da engenharia de dados deste projeto, responsável por mover e tratar os dados desde a fonte bruta até o DataMart analítico. A extração e a transformação inicial dos dados, fundamentais para a criação do DataMart, é realizada nessas camadas: **Bronze** e **Silver**.

2.4.1 Camada Bronze: Ingestão de Dados Brutos

A primeira etapa, correspondente à **Camada Bronze**, consistiu na extração dos dados brutos dos quatro arquivos `train_FD00X.txt` do dataset C-MAPSS. Um script em Python (`extracao.py`) foi desenvolvido para:

- Ler os arquivos de texto, que são formatados com espaços e não possuem cabeçalho.
- Atribuir os nomes corretos às 26 colunas (ID do motor, ciclo, 3 configurações operacionais e 21 sensores).
- Adicionar uma coluna `fd_id` para identificar a origem de cada registro (ex: 'FD001'), o que é essencial para a futura `Dim_Configuracao`.
- Carregar esses dados brutos, sem nenhuma outra transformação, em uma tabela de *staging* no MySQL, cujo schema foi definido no script `cria_schema.sql`.

2.4.2 Camada Silver: Limpeza e Enriquecimento

A **Camada Silver** representa a fonte de verdade limpa, validada e integrada. Nela, os dados da camada Bronze são transformados, enriquecidos e preparados para a modelagem analítica. O trabalho realizado nesta etapa (implementado em `silver_tratamento.py` e `silver_script.sql`) incluiu:

- **Cálculo do RUL:** Conforme a hipótese do projeto, o KPI central (RUL) foi calculado para os dados de treino. Isso envolveu agrupar os dados por motor para encontrar o ciclo máximo (ciclo de falha) e, em seguida, calcular o RUL para cada linha como $(\text{Ciclo_Maximo} - \text{Ciclo_Atual})$.
- **Padronização e Limpeza:** Verificação de tipos de dados e remoção de colunas que não agregam valor analítico (ex: sensores com leitura constante em todos os datasets).

- **Integração:** Consolidação dos quatro datasets (agora enriquecidos com RUL) em uma única tabela analítica na camada Silver, pronta para ser modelada na camada Gold.

2.4.3 Definição e Hipótese do KPI

A Vida Útil Remanescente, ou RUL (*Remaining Useful Life*), é o KPI central deste projeto. Ele representa uma estimativa de quantos ciclos de operação um motor ainda possui antes de atingir um ponto de falha - podendo servir como parâmetro para identificação de um estágio seguro de degradação para a manutenção.

2.4.3.1 BI como Consumidor

Na prática industrial, o RUL de um motor em operação é um valor desconhecido que precisa ser previsto. Esse prognóstico é uma tarefa clássica de Ciência de Dados e Machine Learning (ML), onde um modelo é treinado em dados históricos (como os do C-MAPSS) para aprender os padrões de degradação dos sensores que antecedem uma falha.

Para os fins deste projeto de DataMart e Business Intelligence, a premissa estabelecida é que este modelo de ML já existe e gerou a estimativa para as diferentes condições dos motores.

Nossa plataforma de BI não é responsável por criar a previsão. Ela atua na etapa seguinte: ela consome o resultado do modelo (o `RUL_Previsto`) como seu principal KPI. O foco do nosso DataMart e dashboard é, portanto:

- **Integrar:** Receber e armazenar o `RUL_Previsto` para cada motor em cada ciclo.
- **Contextualizar:** Cruzar o `RUL_Previsto` com as dimensões de engenharia (Cenário de Teste, Condições Operacionais).
- **Analizar:** Permitir que um engenheiro de confiabilidade monitore a queda do RUL e investigue os dados dos sensores que justificam aquela previsão.

Para simular o recebimento deste KPI no nosso processo de ETL, serão usados os dados de treino do C-MAPSS (onde a falha é conhecida) para analisar em função do ciclo máximo de operação - aquele onde ocorre a falha do motor simulado. Isso reduz a complexidade do modelo e das análises deste projeto.

2.5 Infraestrutura e Containerização

Para garantir a portabilidade, isolamento e reproduzibilidade do ambiente de desenvolvimento, a solução foi implementada utilizando a plataforma **Docker**. A arquitetura foi definida no padrão de microserviços, orquestrada via *Docker Compose*, segregando a camada de armazenamento (Banco de Dados) da camada de processamento (Aplicação ETL).

A infraestrutura é composta por dois serviços principais definidos no arquivo para orquestração: `docker-compose.yml`.

- **Serviço de Banco de Dados (`mysql-db`):** Utiliza a imagem oficial `mysql:8.0`. Este container é responsável pela persistência dos dados das camadas Bronze e Silver.
 - **Inicialização Automática:** Através do mapeamento de volumes no diretório, os scripts SQL de criação dos schemas (Bronze e Silver) são executados automaticamente na primeira inicialização do container.
 - **Configuração de Rede:** O serviço expõe a porta 3307 para o host, evitando conflitos com instalações locais do MySQL na porta padrão 3306.
- **Serviço de Aplicação (`etl-app`):** Container construído a partir de uma imagem Python customizada (definida no `dockerfile`). Este ambiente contém todas as dependências necessárias para a execução dos pipelines, incluindo as bibliotecas *Pandas*, *PySpark* e o conector do MySQL.
 - **Volumes:** O código-fonte (`src/`) e os dados brutos (`data/`) são mapeados do host para o container, permitindo que alterações no código sejam refletidas imediatamente sem necessidade de reconstrução da imagem (rebuild). Basta reiniciar o container para surtir as modificações.

2.5.1 Orquestração da Execução

A comunicação entre os containers é realizada através de uma rede virtual interna criada pelo Docker, permitindo que a aplicação resolva o endereço do banco de dados pelo nome do serviço (`mysql-db`), abstraindo endereços IP.

O fluxo de execução é controlado pelo script `run.sh`, que implementa uma lógica de espera (*wait logic*) para garantir que o banco de dados esteja totalmente operacional antes do início dos processos de ETL, prevenindo falhas de conexão durante a carga inicial.

2.6 Consultas Analíticas

A **Camada Gold** consolida os dados em visões de negócio, permitindo responder a perguntas de engenharia de confiabilidade. Abaixo são apresentadas as consultas analíticas fundamentais desenvolvidas para este projeto, utilizando funções de janela (*window functions*) e agregações do MySQL 8.0.

As consultas realizadas nessa camada são expostas na Tabela 2 e vão ao encontro das perguntas-chave definidas anteriormente.

Tabela 2 – Resumo das Funções Analíticas e Perguntas de Negócio Respondidas

ID	Função	Pergunta de Negócio Respondida	Dimensões Analisadas
1	ROLLUP	Qual a temperatura média global e por sub-nível de altitude?	Cenário, Altitude
2	RANK	Quem são os motores mais duráveis de cada cenário?	Cenário, Motor
3	LAG	Qual o impacto térmico incremental ao subir a potência?	Cenário, Potência (TRA)
4	FIRST_VALUE	Qual o desvio de performance em relação ao motor ideal?	Cenário, Motor
5	DENSE_RANK	Quais zonas de altitude causam maior rotação?	Cenário, Faixa de Altitude
6	ROLLUP	Qual o ciclo de falha médio para cada configuração?	Cenário, Número de Ciclo

3 Resultados e Análise (Dashboard)

A etapa final do fluxo de dados consiste na visualização das informações processadas na camada *Gold*. Para isso, foi desenvolvido um dashboard no Microsoft Power BI.

O dashboard conecta-se diretamente ao banco de dados MySQL (tabelas fatos e dimensões) e traduz as consultas analíticas apresentadas no capítulo anterior em indicadores visuais para suporte à decisão. A navegação foi estruturada em três páginas distintas, detalhadas a seguir.

3.1 Escala Microscópica (Diagnóstico do Motor)

A primeira página do painel, apresentada na Figura 7, oferece uma consciência situacional imediata sobre o estado de toda a frota de motores simulada.

O objetivo desta tela é permitir que o analista possa visualizar e até identificar anomalias a partir das séries temporais das leituras realizadas pelos sensores sobre as diferentes configurações (cenários de simulação).

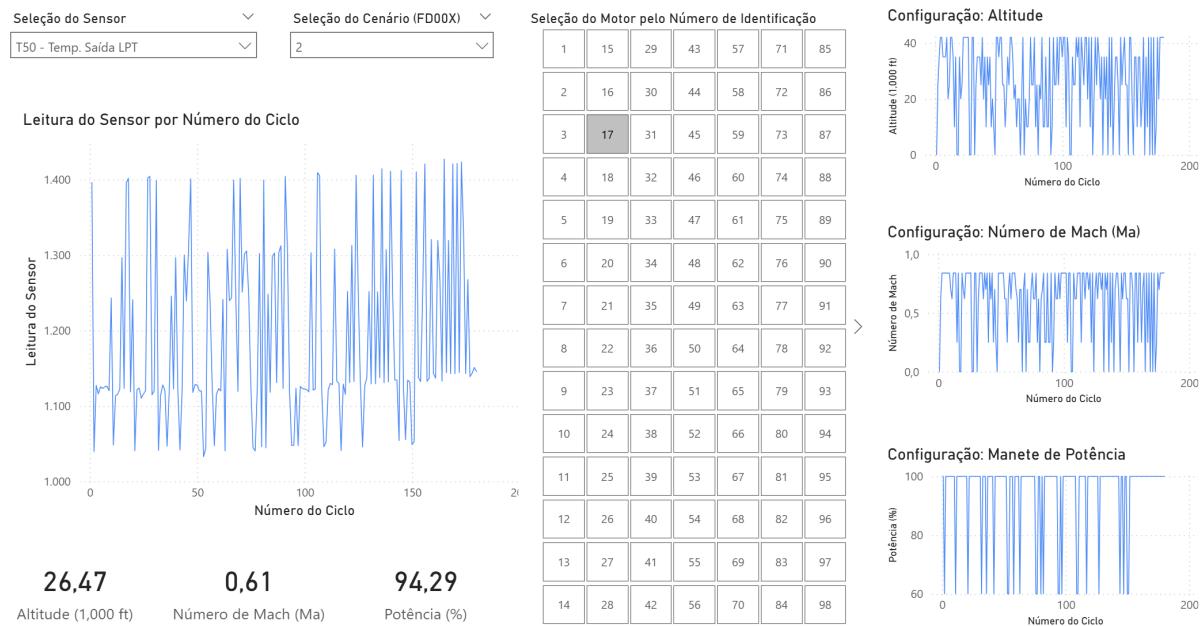


Figura 7 – Dashboard - Visão Geral da Saúde da Frota

Dentre os principais elementos visuais, vale destacar:

- Configuração do Cenário:** Exibe para cada cenário, os gráficos de linha referentes à altitude, ao número de Mach (Ma) e também a posição da manete de potência. Exibindo também nos cartões a média de cada uma dessas configurações durante o cenário;
- Seleção do Motor em Análise:** Pode-se selecionar um motor específico que foi simulado para fazer o levantamento dos dados;
- Seleção do Sensor de Interesse:** Além do motor, seleciona-se também o sensor a ser analisado - onde há uma descrição do que ele representa - permitindo que o analista monitore diferentes fenômenos nas leituras.

3.2 Escala Mesoscópica (Análise Comparativa)

A segunda página foca na análise comparativa entre os ativos. Esta visualização permite separar os motores de alta performance dos motores problemáticos. Conforme observado na Figura 8, a engenharia pode priorizar quais motores necessitam de uma atenção especial - com maior propensão a falhas etc.

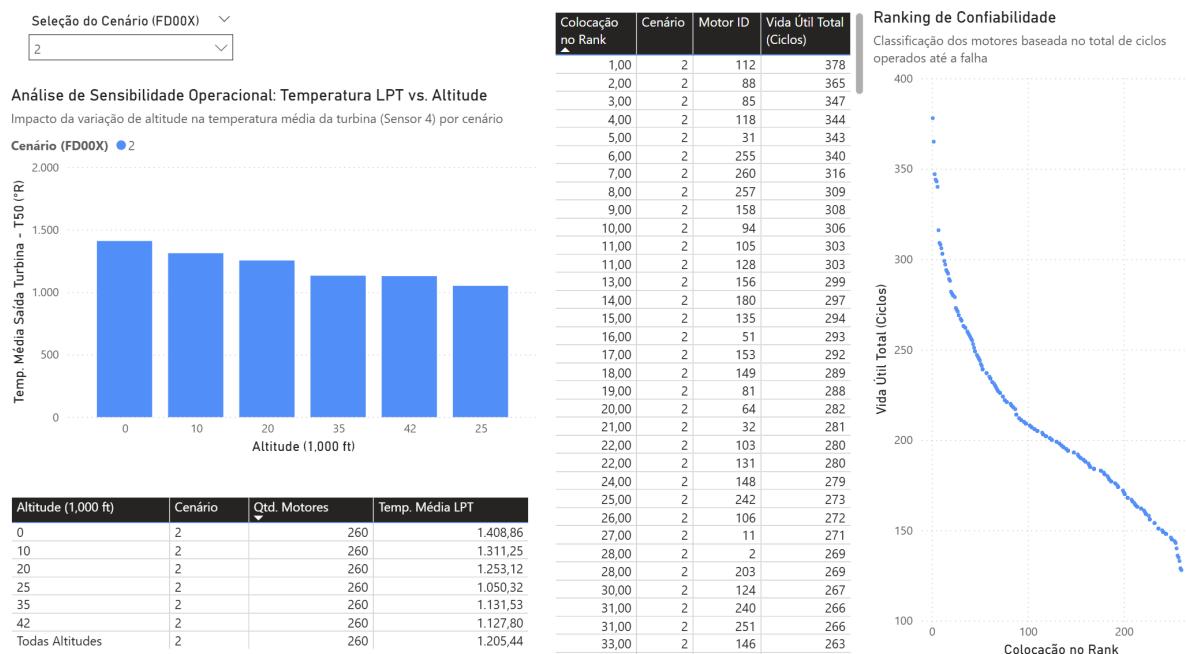


Figura 8 – Dashboard - Análise Comparativa e Ranking de Confiabilidade

O destaque dessa análise vai para o valor comparativo gerado entre os motores submetidos aos mesmos cenários de simulação.

3.3 Escala Macroscópica (Visão Global)

A terceira página, ilustrada na Figura 9, é destinada a uma visão global dos cenários aos quais os motores são expostos (via simulação). Nela, pode-se verificar como as configurações dos cenários afetam variáveis de interesse. Além disso, trás uma análise comparativa sobre o tempo de vida útil (ciclo máximo atingido na simulação) por cenário.

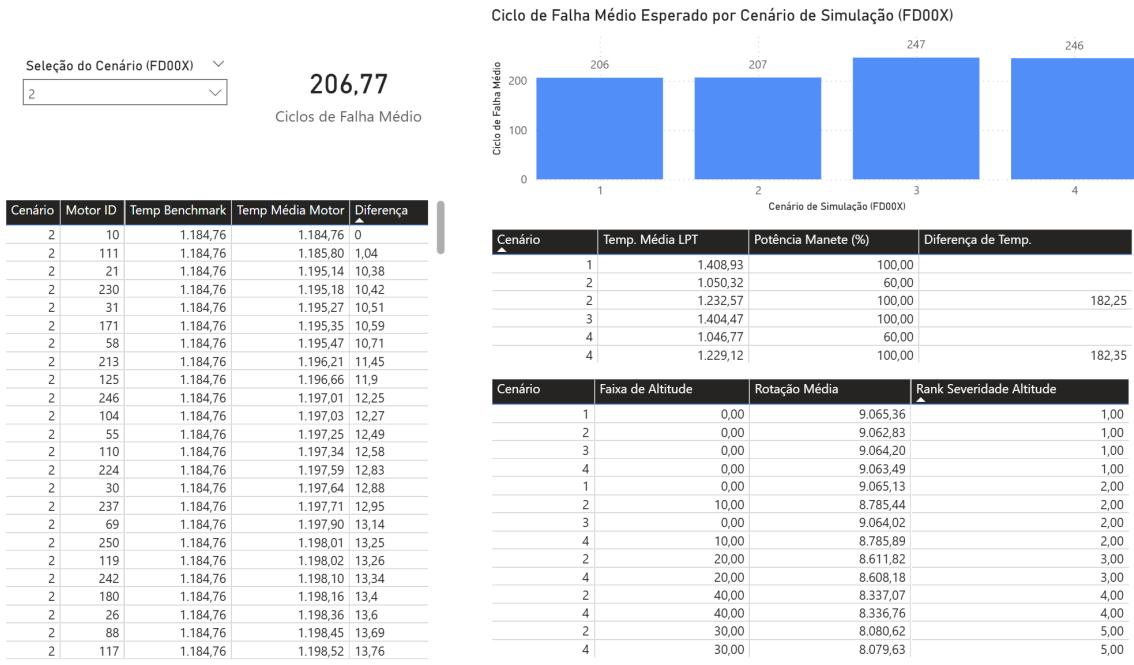


Figura 9 – Dashboard - Detalhamento Técnico dos Sensores

Conclusão

O presente trabalho atingiu seu objetivo principal ao projetar e implementar uma arquitetura de Business Intelligence completa, capaz de transformar dados brutos de sensores de motores turbofan em indicadores açãoáveis para a Engenharia de Confiabilidade.

A validação da infraestrutura containerizada (Docker) demonstrou que é possível construir um ambiente de dados reprodutível e isolado, eliminando a barreira técnica de configuração de ambiente. O fluxo de ETL proposto, estruturado nas camadas Bronze, Silver e Gold, provou-se eficaz não apenas na higienização dos dados do *dataset* C-MAPSS, mas fundamentalmente na preparação de uma base analítica performática.

A entrega final do dashboard, apresenta a visualização de dados em múltiplos níveis de granularidade que é essencial para a gestão de ativos complexos:

- A visão **Macro** permitiu a gestão logística da frota;
- A visão **Meso** facilitou a identificação de *outliers* e comparação de performance;
- A visão **Micro** habilitou o diagnóstico técnico detalhado de falhas incipientes.

Referências

- 1 ADAMOPOULOU, E.; DASKALAKIS, E. Applications and technologies of big data in the aerospace domain. *Electronics*, MDPI, v. 12, n. 10, p. 2225, 2023.
- 2 BADEA, V. E.; ZAMFIROIU, A.; BONCEA, R. Big data in the aerospace industry. *Informatica Economica*, INFOREC Association, v. 22, n. 1, p. 17–24, 2018.
- 3 SAXENA, A. et al. Damage propagation modeling for aircraft engine run-to-failure simulation. In: IEEE. *2008 international conference on prognostics and health management*. [S.I.], 2008. p. 1–9.