PUC-RIO

MVP - Engenharia de Dados — SCANIA Component X

Igor Cassimiro Assunção

Engenharia de Dados
Entrega: 21/12

Sumário

Capítulo 1

Introdução

1.1 Contexto e motivação

A Indústria 4.0 trouxe uma forte digitalização do chão de fábrica e da operação de frotas, com grande volume de dados coletados continuamente por sensores, sistemas embarcados e plataformas de monitoramento. Esses dados permitem a construção de soluções de **manutenção preditiva**, capazes de identificar padrões de degradação de componentes antes da falha e, assim, reduzir paradas não planejadas, custos de reparo e riscos operacionais.

O SCANIA Component X Dataset reúne dados reais de operação de caminhões pesados, contendo:

- Leituras operacionais multivariadas, em série temporal, de diversos sensores e contadores;
- Informações de tempo até o evento de reparo (*time-to-event*) do componente X;
- Especificações dos veículos (diferentes configurações e características);
- Rótulos que indicam quão próximo o veículo está de necessitar reparo do componente X, em janelas de tempo discretizadas.

Esse cenário é representativo de um problema típico de Indústria 4.0, no qual há grande volume de dados, variáveis anonimizadas por questões de confidencialidade industrial e a necessidade de transformar dados brutos em informações úteis para tomada de decisão.

1.2 Objetivo do MVP

Construir um pipeline de dados, utilizando a plataforma Databricks Free Edition, para ingestão, modelagem, carga e análise do *SCANIA Component X Dataset*, com foco em manutenção preditiva de caminhões pesados. O pipeline deverá transformar os dados brutos em uma camada analítica estruturada **(esquema estrela)** que permita analisar padrões de degradação do componente X, perfis de risco por especificação de veículo e comportamento operacional ao longo do tempo.

1.2.1 Perguntas de negócio

1. **Distribuição do tempo observado e evento de reparo do componente X**
   - Como se distribui o **tempo observado** (*time-to-event*) do componente X na frota?
   - Qual é a mediana, média e variabilidade desse tempo?
   - Qual a proporção de veículos que chegam ao fim do estudo sem reparo registrado do componente X (censura)?
2. **Especificações de veículo mais críticas**
   - Certas combinações de especificações dos caminhões (por exemplo, diferentes valores de *spec_ 0*, *spec_ 1* e demais atributos de configuração) estão associadas a **menor tempo observado** até o reparo do componente X?

- É possível identificar grupos de especificações com comportamento claramente mais crítico (degradação mais rápida)?

3. **Comportamento temporal da degradação**
   - À medida que o componente X se aproxima do reparo/falha, como evoluem algumas leituras operacionais (sensores, contadores, variáveis de uso)?
   - Existem padrões temporais característicos nas séries de leituras em janelas mais próximas da falha em comparação com janelas mais distantes?
4. **Distribuição das classes de proximidade de falha (0–4)**
   - Como se distribuem as classes de proximidade da falha (0, 1, 2, 3, 4) nos conjuntos de validação e teste?
   - Há desbalanceamento importante (por exemplo, muitas instâncias em classe 0 e poucas em classe 4)? Como isso impacta a interpretação dos dados?
5. **Perfis operacionais de maior risco**
   - É possível caracterizar perfis operacionais de maior risco para o componente X, combinando especificações de veículo, tempo observado, classes de proximidade (0–4) e padrões nas leituras operacionais?
   - Esses perfis podem servir como base para políticas de monitoramento e agendamento de manutenção mais inteligentes do que a manutenção puramente corretiva?

1.3 Visão geral do pipeline

O pipeline foi implementado na plataforma Databricks Free Edition utilizando *Volumes* para armazenamento dos arquivos brutos e tabelas Delta como formato padrão de persistência. A ingestão carrega os arquivos CSV para a camada **Bronze** (dados brutos), a camada **Silver** aplica padronização de chaves, tipagem, regras de qualidade e preparação para modelagem, e a camada **Gold** consolida os dados em um **Esquema Estrela** voltado à análise (dimensões e tabelas fato). A etapa final utiliza consultas SQL e visualizações em dashboards do Databricks para avaliar qualidade de dados e responder às perguntas de negócio definidas no objetivo do MVP.

Capítulo 2

Dados e Coleta

2.1 Fonte dos dados

Os dados utilizados neste MVP foram obtidos no repositório Researchdata.se, no conjunto *SCANIA Component X Dataset*, disponível em:

https://researchdata.se/en/catalogue/dataset/2024-34/2

2.2 Arquivos disponibilizados

O dataset é distribuído em três partições (**treino**, **validação** e **teste**), totalizando nove arquivos CSV:

- **Treino:**
  - `train_operational_readouts.csv` (1.14 GiB)
  - `train_specifications.csv` (1.03 MiB)
  - `train_tte.csv` (337.32 KiB)
- **Validação:**
  - `validation_operational_readouts.csv` (205.61 MiB)
  - `validation_specifications.csv` (226.33 KiB)
  - `validation_labels.csv` (37.83 KiB)
- **Teste:**
  - `test_operational_readouts.csv` (204.94 MiB)
  - `test_specifications.csv` (226.23 KiB)
  - `test_labels.csv` (37.78 KiB)

2.3 Descrição dos dados

2.3.1 Leituras operacionais (*operational_readouts*)

Os arquivos `*_operational_readouts.csv` contêm leituras operacionais multivariadas ao longo do tempo. Cada registro é associado a um veículo (`vehicle_id`) e a um marcador temporal discreto (`time_step`). As variáveis operacionais são anonimizadas e incluem contadores numéricos e histogramas, seguindo uma convenção do tipo `variableid_binindex`.

2.3.2 Tempo até evento (*train_tte*)

O arquivo `train_tte.csv` armazena informações de *time-to-event* por veículo, incluindo o tempo observado (`length_of_study_time_step`) e a indicação de ocorrência do evento (`in_study_repair`), onde `in_study_repair = 1` indica que houve reparo durante o estudo e `in_study_repair = 0` indica censura (sem reparo observado no período).

2.3.3 Especificações (*specifications*)

Os arquivos `*_specifications.csv` descrevem configurações/categorias do veículo por meio de variáveis categóricas (por exemplo, `spec_0` a `spec_7`), selecionadas como atributos relevantes para o comportamento do componente X.

2.3.4 Rótulos de proximidade (*labels*)

Os arquivos `validation_labels.csv` e `test_labels.csv` fornecem o campo `class_label` (0 a 4), que representa janelas de proximidade do evento do componente X, permitindo análises de distribuição de classes e associação com padrões operacionais.

2.4 Coleta e armazenamento no Databricks (Volumes)

Os arquivos CSV foram baixados manualmente e enviados para um *Volume* no Databricks Free Edition, a fim de centralizar o armazenamento dos dados brutos e padronizar os caminhos de leitura. A Figura 2.1 apresenta evidência do upload no volume do projeto.

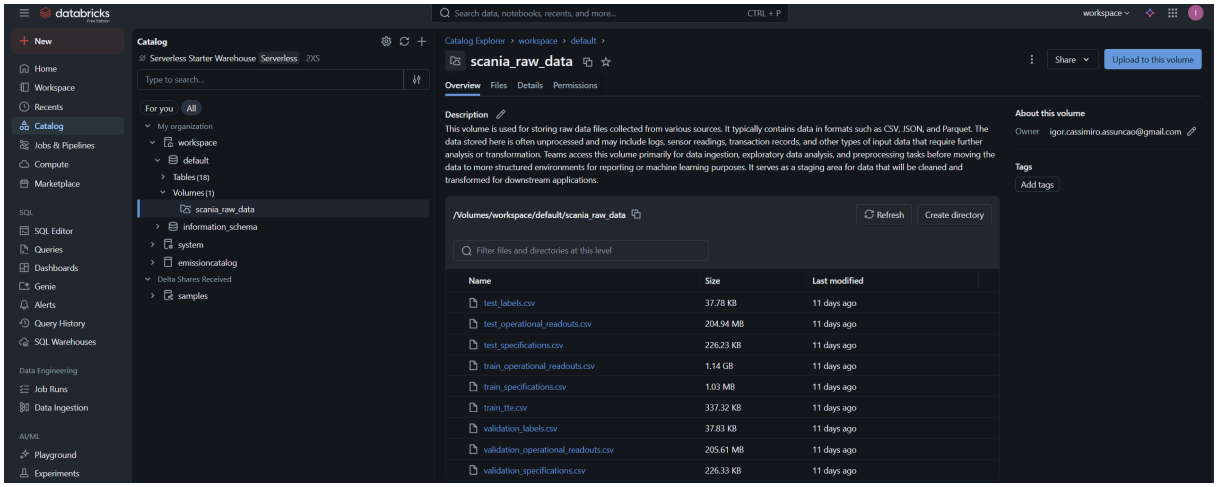


Figura 2.1: Arquivos CSV enviados ao volume `scania_raw_data` no Databricks.

Capítulo 3

Implementação do Pipeline (Bronze→Silver→Gold)

3.1 Organização do ambiente no Databricks

Este trabalho foi desenvolvido utilizando o Databricks Free Edition. Para garantir reprodutibilidade, o pipeline foi estruturado em notebooks executados em sequência e com persistência de dados em tabelas Delta organizadas em três camadas: Bronze, Silver e Gold.

3.2 Armazenamento dos dados brutos (Volumes)

3.2.1 Volume do projeto e caminho base

Os arquivos CSV foram armazenados em um *Volume* dedicado ao projeto, denominado `scania_raw_data`. O caminho base utilizado nas leituras é:

```
/Volumes/workspace/default/scania_raw_data/
```

3.2.2 Listagem dos arquivos no volume (evidência)

Para confirmar a persistência e disponibilidade dos arquivos no Databricks, foi executada a listagem do conteúdo do volume. A Figura 3.1 apresenta a evidência do comando no notebook.

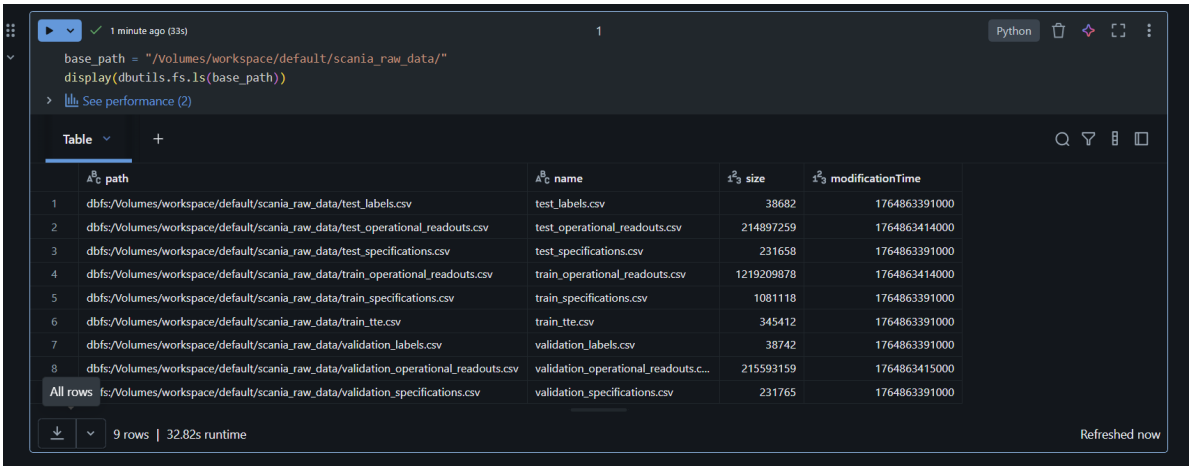


Figura 3.1: Listagem do conteúdo do volume via `dbutils.fs.ls(base_path)`.

3.3 Visão geral do fluxo do pipeline

A Figura 3.2 apresenta a arquitetura do pipeline de dados, desde a origem (Researchdata.se) até a camada analítica (Gold) e consumo via análises SQL/Python e dashboards.

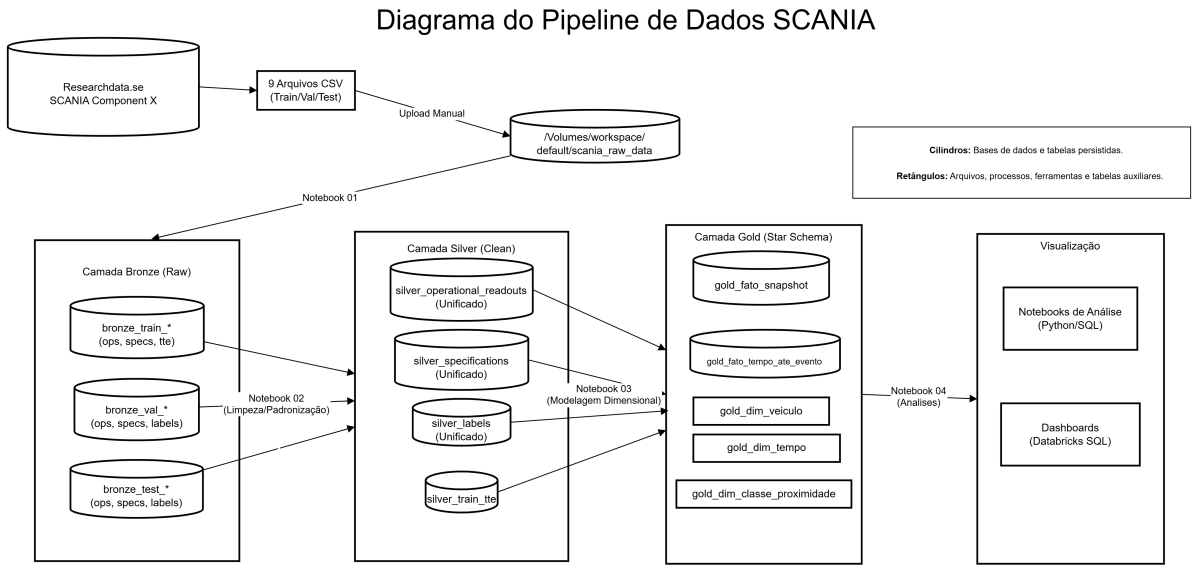


Figura 3.2: Arquitetura do pipeline no Databricks: Fonte → Volume → Bronze → Silver → Gold → Consumo.

3.4 Ordem de execução (notebooks) e responsabilidades

O pipeline foi implementado em quatro notebooks principais, com responsabilidades bem definidas:

- `01_Ingestao_Bronze`: leitura dos CSVs no volume, tratamento de valores nulos codificados como texto (ex.: `na`) e criação das tabelas Delta na camada Bronze.
- `02_Tratamento_Silver`: padronização de chaves (`vehicle_id`), tipagem de colunas, unificação e enriquecimento com `dataset_split` (train/validation/test) e preparação para a modelagem.
- `03_Modelagem_Gold`: criação da camada analítica em esquema estrela, gerando dimensões e tabelas fato para suportar consultas e visualizações.
- `04_Analises_SQL`: análise de qualidade de dados e geração de consultas analíticas para responder às perguntas do MVP, servindo como base para dashboards no Databricks.

3.5 Tabelas Delta por camada

Ao final da execução do pipeline, foram geradas 18 tabelas Delta, organizadas por camada conforme abaixo:

Bronze (9 tabelas)

- `bronze_train_operational_readouts`
- `bronze_train_specifications`
- `bronze_train_tte`
- `bronze_validation_operational_readouts`
- `bronze_validation_specifications`

- `bronze_validation_labels`
- `bronze_test_operational_readouts`
- `bronze_test_specifications`
- `bronze_test_labels`

Silver (4 tabelas)

- `silver_operational_readouts`
- `silver_specifications`
- `silver_train_tte`
- `silver_labels`

Gold (5 tabelas)

- `gold_dim_veiculo`
- `gold_dim_tempo`
- `gold_dim_classe_proximidade`
- `gold_fato_snapshot`
- `gold_fato_tempo_ate_evento`

3.6 Evidências no Catalog

A verificação das tabelas por camada foi realizada no Catalog do Databricks. As Figuras 3.3–3.5 apresentam evidências da presença das tabelas Bronze, Silver e Gold.

3.6.1 Camada Bronze

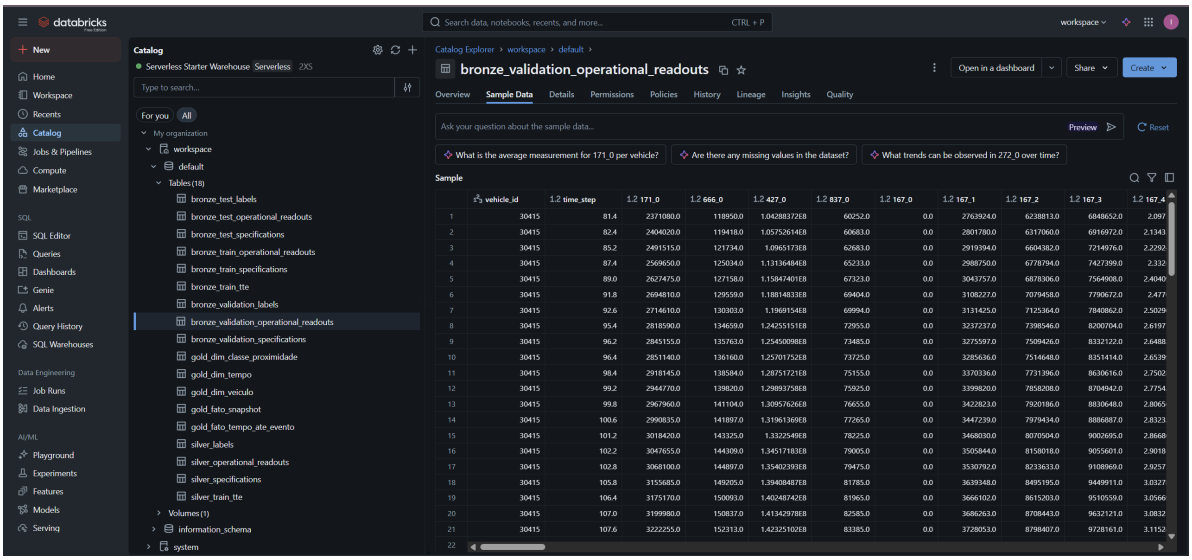


Figura 3.3: Catalog do Databricks mostrando as tabelas da camada Bronze (`bronze_operational_readouts`).

3.6.2 Camada Silver

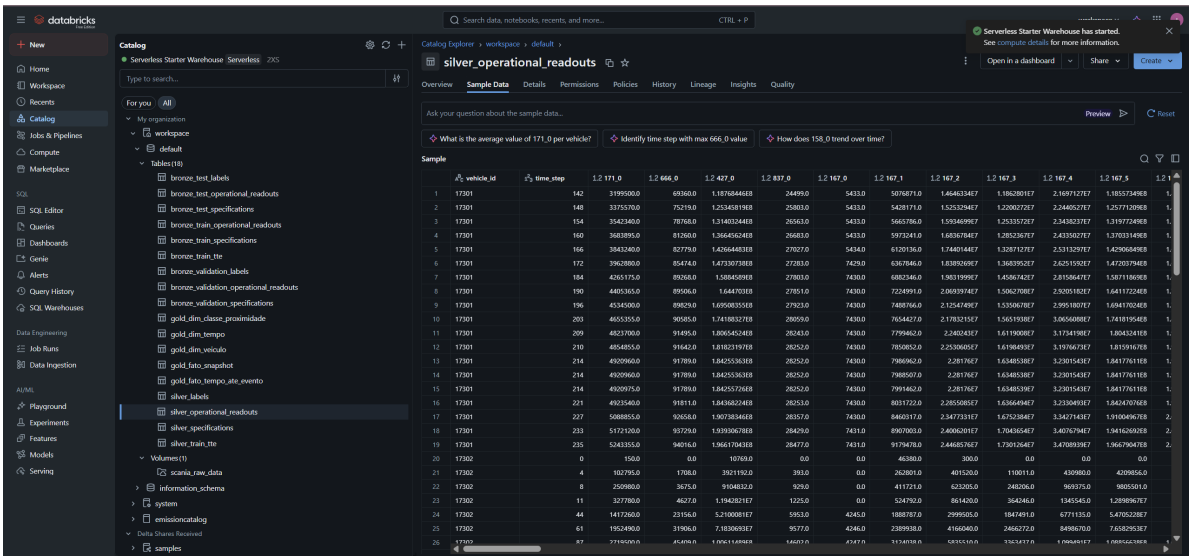


Figura 3.4: Catalog do Databricks mostrando as tabelas da camada Silver (`silver_operational_readouts`).

3.6.3 Camada Gold

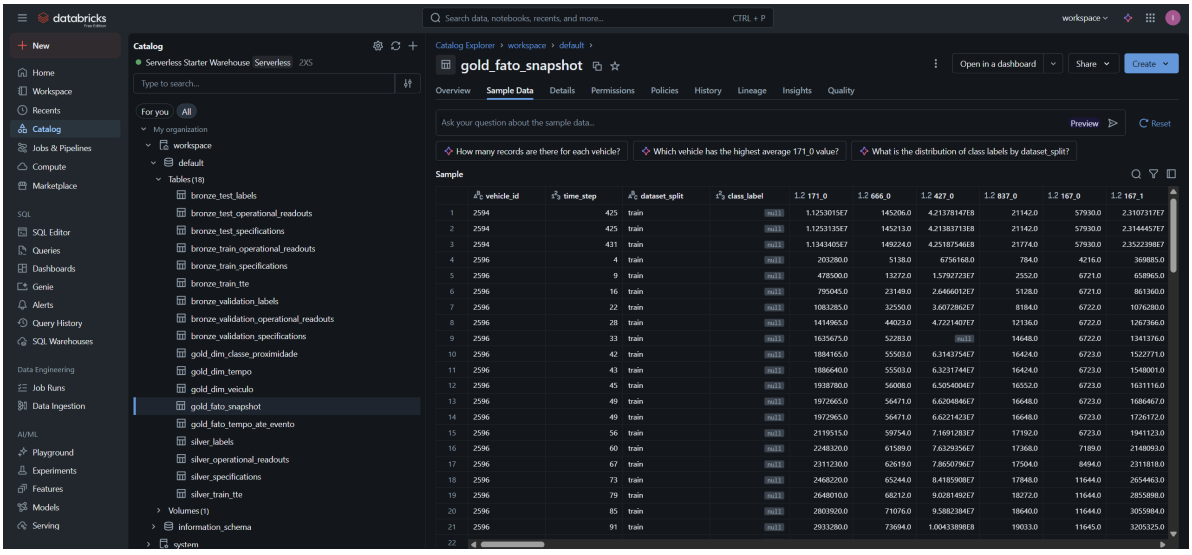


Figura 3.5: Catalog do Databricks mostrando as tabelas da camada Gold (`gold_fato_snapshot`).

Capítulo 4

Camada Bronze — Ingestão

4.1 Objetivo da camada Bronze

A camada Bronze representa o nível de **persistência bruta** dos dados. Nesta etapa, os arquivos CSV foram lidos diretamente do volume no Databricks, preservando a estrutura original e garantindo rastreabilidade. O foco é transformar arquivos em tabelas Delta, mantendo os dados o mais próximo possível da origem.

4.2 Leitura dos arquivos e tratamento de nulos textuais

Durante a ingestão, foi adotada uma configuração de leitura para tratar valores ausentes codificados como texto (ex.: `na`) como valores nulos reais. Isso é essencial para permitir análises de qualidade de dados e agregações corretas nas etapas seguintes.

4.3 Tabelas Bronze geradas

A ingestão criou 9 tabelas Delta na camada Bronze:

- `bronze_train_operational_readouts`
- `bronze_train_specifications`
- `bronze_train_tte`
- `bronze_validation_operational_readouts`
- `bronze_validation_specifications`
- `bronze_validation_labels`
- `bronze_test_operational_readouts`
- `bronze_test_specifications`
- `bronze_test_labels`

4.4 Evidências de ingestão

Para evidenciar a ingestão e a preservação da estrutura original, foram registradas amostras (até 5 linhas) e o esquema (tipos) das tabelas.

4.4.1 Amostra de dados (até 5 linhas)

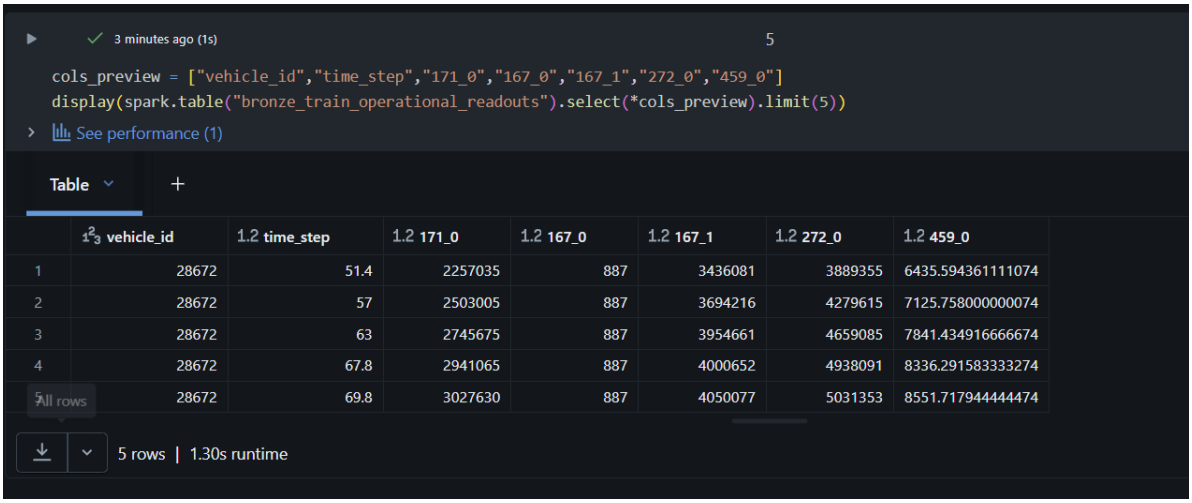


Figura 4.1: Amostra (`head`) da tabela `bronze_train_operational_readouts`.

4.4.2 Esquema (schema) da tabela operacional

Figura 4.2: Esquema (schema) de `bronze_train_operational_readouts` após ingestão.

4.4.3 Tabela *time-to-event* (train_tte)

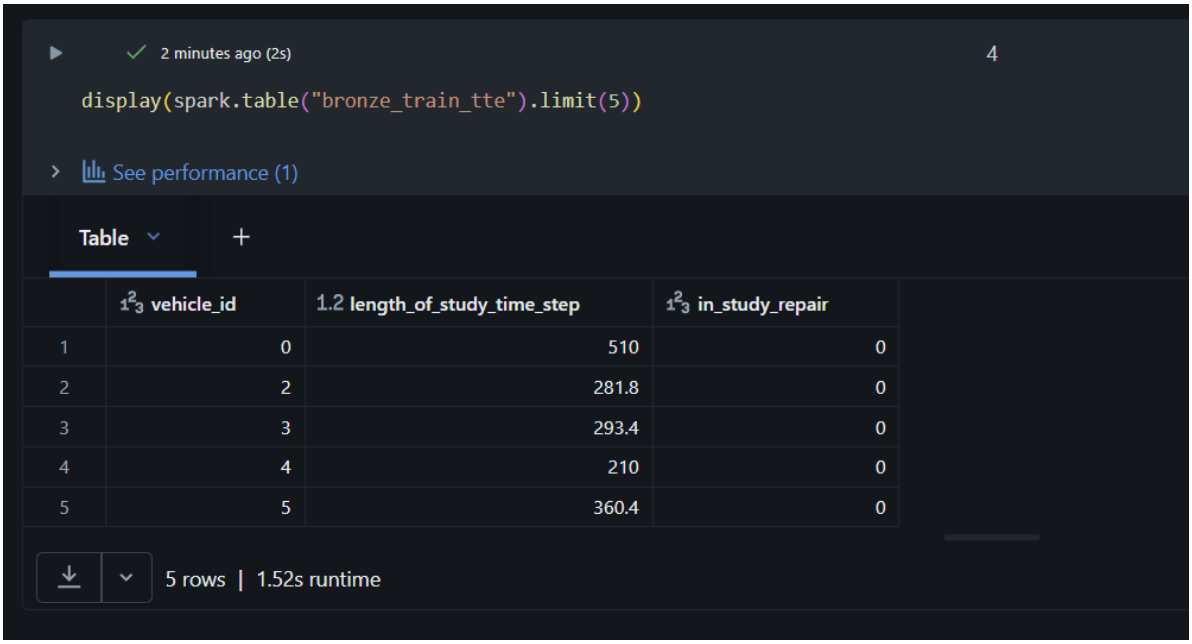


Figura 4.3: Amostra (`head`) da tabela `bronze_train_tte`.

4.5 Discussão

Nesta camada, a principal entrega é a **conversão dos arquivos CSV em tabelas Delta** e a garantia de rastreabilidade para as próximas etapas. A normalização de chaves, unificação de partições e modelagem analítica são realizadas posteriormente na camada Silver e Gold.

Capítulo 5

Camada Silver — Tratamento e Consolidação

5.1 Objetivo da camada Silver

A camada Silver transforma os dados brutos (Bronze) em dados **padronizados e consolidados**, prontos para modelagem analítica. Nesta etapa, foram aplicadas regras de limpeza, padronização de chaves e integração entre partições, incluindo a criação do atributo `dataset_split` (train/validation/test) para permitir análises comparativas em uma única tabela unificada.

5.2 Tabelas Silver geradas

A camada Silver foi persistida em quatro tabelas Delta:

- `silver_operational_readouts`
- `silver_specifications`
- `silver_train_tte`
- `silver_labels`

5.3 Principais transformações realizadas

As principais transformações desta camada foram:

- **Padronização de chave:** uniformização do campo `vehicle_id` (remoção de espaços, normalização de tipo e consistência).
- **Consolidação:** união dos conjuntos train/validation/test com criação do campo `dataset_split`.
- **Preparação para análises:** estruturação das leituras operacionais para suportar junções com especificações e rótulos.
- **Tratamento de rótulos:** integração de `class_label` onde aplicável. Observa-se que o conjunto de treino não contém rótulos de proximidade, o que resulta em `class_label` nulo para `dataset_split = 'train'`.

5.4 Evidências visuais

Para evidenciar a transformação Bronze → Silver, foram registradas amostras consistentes (até 5 linhas) com o mesmo subconjunto de colunas-chave, permitindo comparação direta.

5.4.1 Bronze (referência)

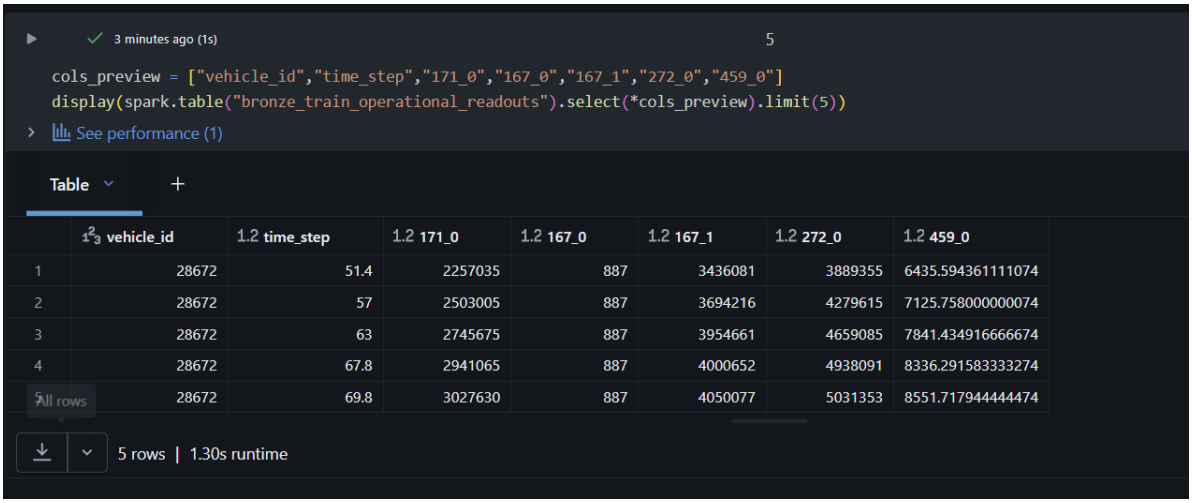


Figura 5.1: Referência Bronze: amostra de `bronze_train_operational_readouts`.

5.4.2 Silver (após tratamento e unificação)

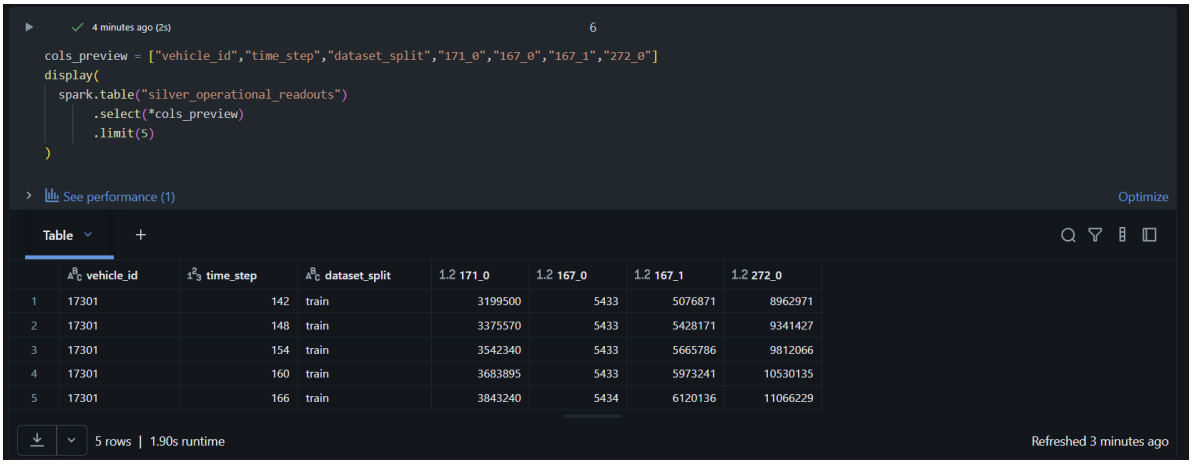


Figura 5.2: Amostra (`head`) da tabela unificada `silver_operational_readouts`, com `dataset_split`.

5.4.3 Schema (tipagem e consistência)

Figura 5.3: Esquema (schema) de `silver_operational_readouts` após tratamento.

5.5 Integração de labels e comportamento esperado no treino

A tabela `silver_labels` contém os rótulos `class_label` disponibilizados para validação e teste. Como não há rótulos equivalentes no conjunto de treino, a coluna `class_label` permanece nula para registros com `dataset_split = 'train'`, comportamento esperado e confirmado na análise de qualidade.

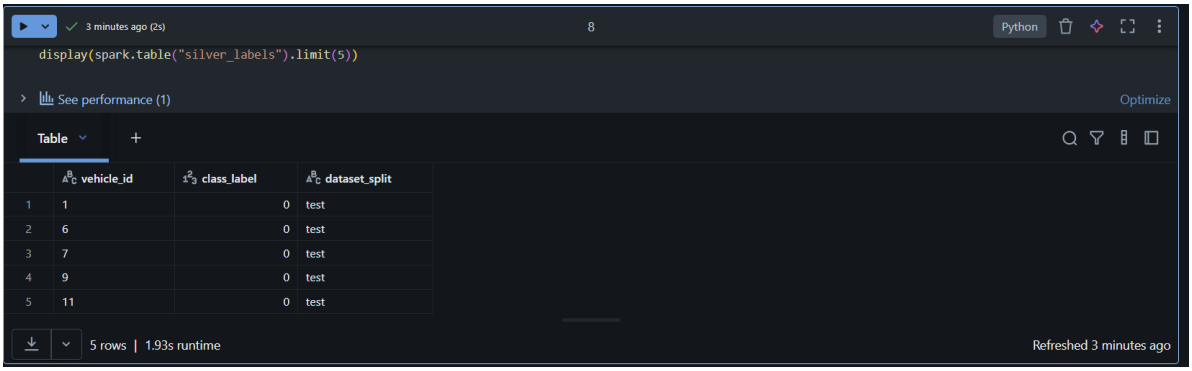


Figura 5.4: Amostra (`head`) de `silver_labels`.

Ao final da camada Silver, os dados encontram-se organizados para modelagem dimensional na camada Gold. A consolidação via `dataset_split` e a integração estruturada de rótulos facilitam análises comparativas e permitem consultas analíticas mais diretas nas próximas etapas.

Capítulo 6

Camada Gold — Modelagem Analítica (Esquema Estrela)

6.1 Objetivo da camada Gold

A camada Gold organiza os dados em um formato **analítico** para suportar consultas, agregações e visualizações de forma eficiente. Nesta etapa, os dados tratados na camada Silver foram modelados em um **Esquema Estrela**, separando dimensões (contexto) e tabelas fato (medições/eventos).

6.2 Tabelas Gold geradas

Foram criadas cinco tabelas Delta na camada Gold:

- `gold_dim_veiculo`
- `gold_dim_tempo`
- `gold_dim_classe_proximidade`
- `gold_fato_snapshot`
- `gold_fato_tempo_ate_evento`

6.3 Modelo dimensional (MER / Esquema Estrela)

A Figura 6.1 apresenta o MER do modelo dimensional criado para a camada Gold. O modelo foi estruturado com chaves compartilhadas (`vehicle_id`, `time_step` e `class_label`) para permitir junções diretas e análises por tempo, especificação do veículo e proximidade do evento.

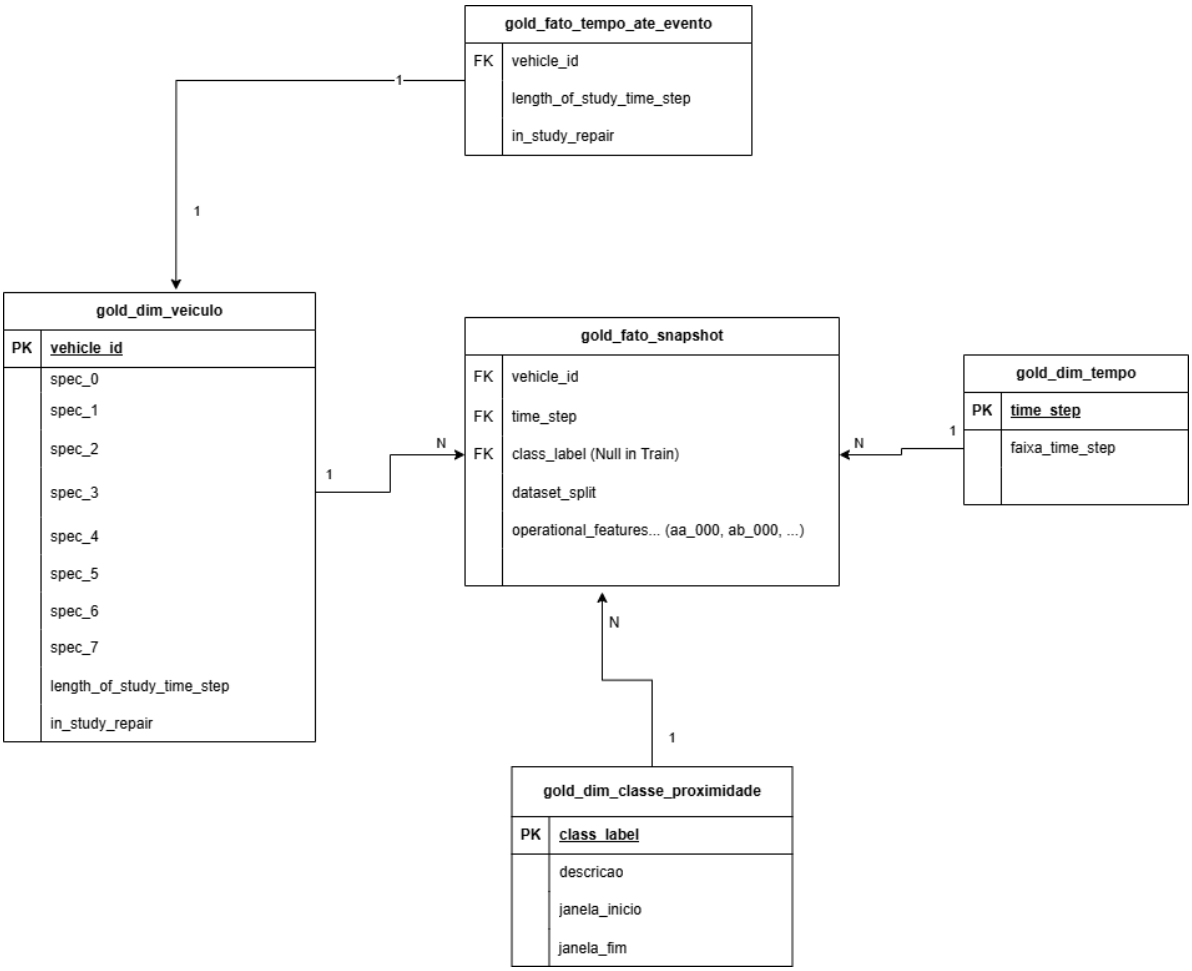
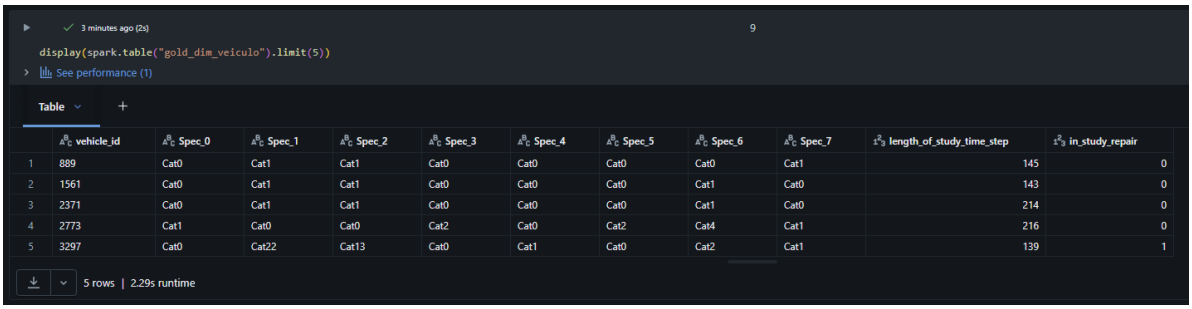


Figura 6.1: Modelo dimensional da camada Gold (Esquema Estrela).

6.4 Dimensões

6.4.1 `gold_dim_veiculo`

Dimensão contendo atributos de configuração do veículo (ex.: `spec_0` a `spec_7`) e atributos de *time-to-event* agregados ao nível do veículo (ex.: `length_of_study_time_step` e `in_study_repair`).



```
display(spark.table("gold_dim_veiculo").limit(5))
```

Table

	vehicle_id	Spec_0	Spec_1	Spec_2	Spec_3	Spec_4	Spec_5	Spec_6	Spec_7	length_of_study_time_step	in_study_repair
1	889	Cat0	Cat1	Cat1	Cat0	Cat0	Cat0	Cat0	Cat1	145	0
2	1561	Cat0	Cat1	Cat1	Cat0	Cat0	Cat0	Cat1	Cat0	143	0
3	2371	Cat0	Cat1	Cat1	Cat0	Cat0	Cat0	Cat1	Cat0	214	0
4	2773	Cat1	Cat0	Cat0	Cat2	Cat0	Cat2	Cat4	Cat1	216	0
5	3297	Cat0	Cat22	Cat13	Cat0	Cat1	Cat0	Cat2	Cat1	139	1

5 rows | 2.29s runtime

Figura 6.2: Amostra (`head`) de `gold_dim_veiculo`.

6.4.2 `gold_dim_tempo`

Dimensão temporal baseada em `time_step`, incluindo uma discretização em faixas (`faixa_time_step`) para facilitar análises por intervalos de tempo.

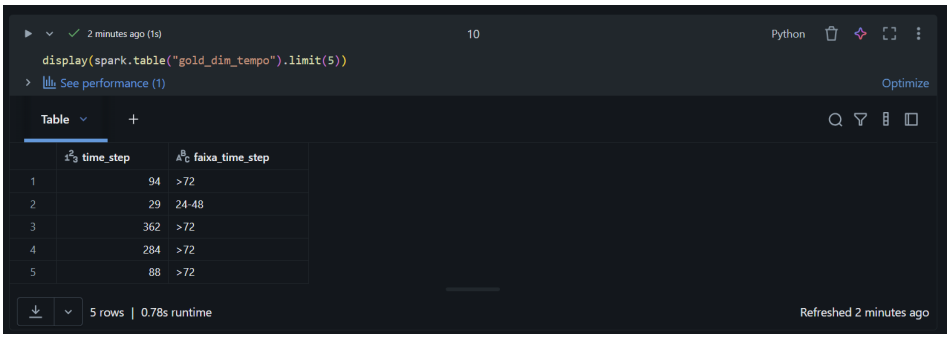


Figura 6.3: Amostra (`head`) de `gold_dim_tempo`.

6.4.3 `gold_dim_classe_proximidade`

Dimensão que descreve as classes de proximidade do evento (0–4), incluindo rótulos textuais e a janela temporal associada a cada classe.

Figura 6.4: Amostra (`head`) de `gold_dim_classe_proximidade`.

6.5 Tabelas fato

6.5.1 `gold_fato_snapshot`

Tabela fato com snapshots operacionais, associando leituras por `vehicle_id` e `time_step`. Inclui `dataset_split` para permitir análises comparativas e `class_label` quando disponível (nulo no treino).

Como a tabela possui grande quantidade de colunas de features anonimizadas, as evidências visuais utilizam um subconjunto fixo de colunas-chave e algumas features selecionadas para manter legibilidade.

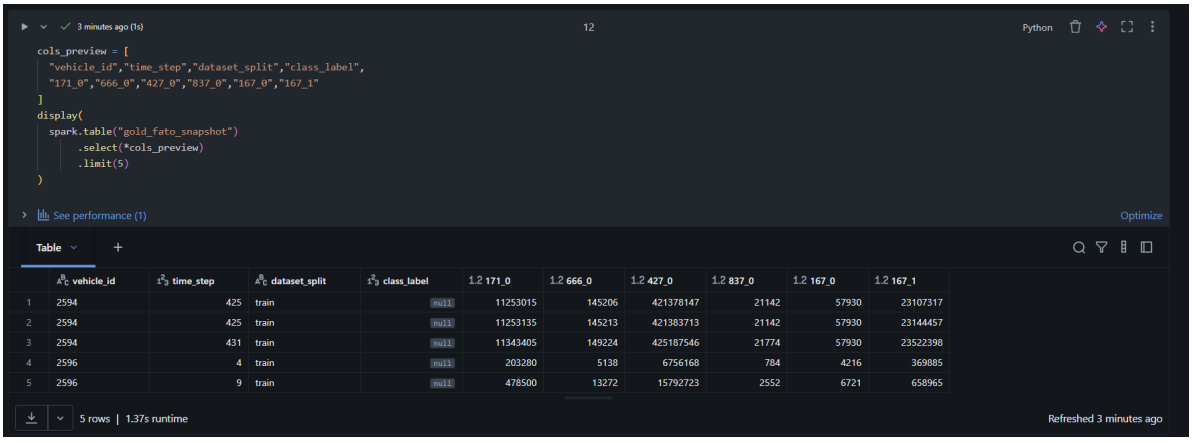


Figura 6.5: Amostra (`head`) de `gold_fato_snapshot` (colunas selecionadas para legibilidade).

6.5.2 `gold_fato_tempo_ate_evento`

Tabela fato no nível do veículo, contendo o tempo observado (`length_of_study_time_step`) e a ocorrência de evento (`in_study_repair`), permitindo análises da distribuição de tempo e censura.

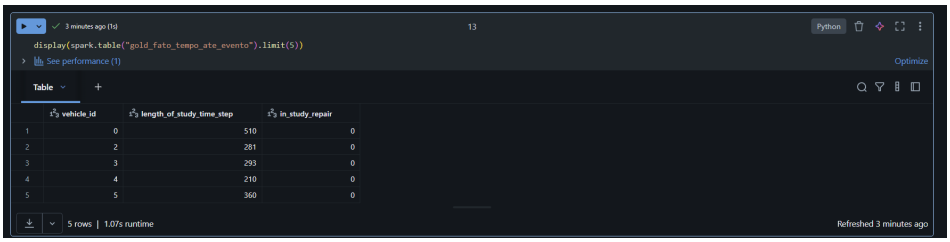


Figura 6.6: Amostra (`head`) de `gold_fato_tempo_ate_evento`.

A camada Gold consolida os dados em um formato adequado para análises de negócio e visualizações. O uso de dimensões para veículo, tempo e proximidade do evento facilita a interpretação dos resultados, enquanto as tabelas fato suportam análises tanto no nível de série temporal (`gold_fato_snapshot`) quanto no nível agregado por veículo (`gold_fato_tempo_ate_evento`).

Capítulo 7

Catálogo de Dados e Linhagem

7.1 Objetivo do catálogo

Este capítulo documenta o **Catálogo de Dados** da camada Gold (tabelas finais consumidas nas análises e dashboards) e a **linhagem** do pipeline (origem → volume → Bronze → Silver → Gold). O dataset possui variáveis operacionais anonimizadas por confidencialidade industrial; portanto, o catálogo descreve a **natureza** dos atributos (numéricos/categóricos, domínio observado) sem atribuir significado físico não comprovado.

7.2 Linhagem (origem → volume → camadas Delta)

Os dados foram obtidos do repositório Researchdata.se e carregados manualmente para um Volume no Databricks Free Edition. Em seguida, foram persistidos como tabelas Delta em três camadas.

7.2.1 Origem e arquivos (raw)

Fonte: https://researchdata.se/en/catalogue/dataset/2024-34/2

Partições e arquivos CSV (train/validation/test):

- `train_operational_readouts.csv`, `train_specifications.csv`, `train_tte.csv`
- `validation_operational_readouts.csv`, `validation_specifications.csv`, `validation_labels.csv`
- `test_operational_readouts.csv`, `test_specifications.csv`, `test_labels.csv`

7.2.2 Armazenamento no Databricks (Volume)

Caminho base do volume:

```
/Volumes/workspace/default/scania_raw_data/
```

7.2.3 Persistência por camadas (Delta tables)

Bronze (9):

- `bronze_train_operational_readouts`, `bronze_train_specifications`, `bronze_train_tte`
- `bronze_validation_operational_readouts`, `bronze_validation_specifications`, `bronze_validation_labels`
- `bronze_test_operational_readouts`, `bronze_test_specifications`, `bronze_test_labels`

Silver (4):

- `silver_operational_readouts`, `silver_specifications`, `silver_train_tte`, `silver_labels`

Gold (5):

- `gold_dim_veiculo`, `gold_dim_tempo`, `gold_dim_classe_proximidade`
- `gold_fato_snapshot`, `gold_fato_tempo_ate_evento`

7.3 Evidência de linhagem e acesso

A Figura 7.1 apresenta a planilha de linhagem e acesso, contendo a fonte do dataset, caminho do volume e nomes finais das tabelas por camada.

Tabela 7.1: Linhagem e acesso aos dados (Databricks Free Edition)

Etapa	**Artefato**	**Detalhes**
Fonte	Dataset	https://researchdata.se/en/catalogue/dataset/2024-34/2
Raw	Arquivos CSV	**Train:** `train_operational_readouts.csv`, `train_specifications.csv`, `train_tte.csv`; **Validation:** `validation_operational_readouts.csv`, `validation_specifications.csv`, `validation_labels.csv`; **Test:** `test_operational_readouts.csv`, `test_specifications.csv`, `test_labels.csv`.
Armazenamento	Volume	`/Volumes/workspace/default/scania_raw_data/`
Bronze	Tabelas Delta	`bronze_train_operational_readouts`, `bronze_train_specifications`, `bronze_train_tte`; `bronze_validation_operational_readouts`, `bronze_validation_specifications`, `bronze_validation_labels`; `bronze_test_operational_readouts`, `bronze_test_specifications`, `bronze_test_labels`.
Silver	Tabelas Delta	`silver_operational_readouts`, `silver_specifications`, `silver_train_tte`, `silver_labels`.
Gold	Tabelas Delta	**Dimensões:** `gold_dim_veiculo`, `gold_dim_tempo`, `gold_dim_classe_proximidade`. **Fatos:** `gold_fato_snapshot`, `gold_fato_tempo_ate_evento`.
ETL	Notebooks	`01_Ingestao_Bronze` → `02_Tratamento_Silver` → `03_Modelagem_Gold` → `04_Analises_SQL`.

7.4 Catálogo de Dados (camada Gold)

O catálogo completo da camada Gold foi organizado em planilhas para facilitar a leitura e a avaliação. As Figuras 7.1–7.5 apresentam as tabelas do catálogo.

7.4.1 `gold_dim_veiculo`

Dimensão no nível do veículo, contendo especificações (`spec_0..spec_7`) e atributos de *time-to-event* (`length_of_study_time_step`, `in_study_repair`).

gold_dim_veiculo

Coluna	Origem	Tipo de	Descrição	Exemplo de Dado	Relevância para Análise
vehicle_id	silver_specifications / silver_train_tte	string	Identificador único para cada veículo.	Não nulo .	Rastreamento e referência
Spec_0	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Detalhes do veículo
Spec_1	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Informações adicionais
Spec_2	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Recursos e desempenho
Spec_3	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Atributos físicos
Spec_4	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Recursos de tecnologia
Spec_5	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Aspectos ambientais
Spec_6	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Planejamento de manutenção
Spec_7	silver_specifications	string	Categoria de especificação do veículo.	Cat0..Cat28 (observado).	Diferenciais do veículo
length_of_study_time_step	silver_train_tte	int	Duração de cada etapa do estudo.	>= 0 (esperado).	Análise de tendências
in_study_repair	silver_train_tte	int	Número de reparos realizados durante o período de estudo.	{0,1}.	Confiabilidade e manutenção

Figura 7.1: Catálogo da tabela `gold_dim_veiculo`.

7.4.2 `gold_dim_tempo`

Dimensão temporal baseada em `time_step`, incluindo uma coluna derivada `faixa_time_step` para agregações.

gold_dim_tempo

Coluna	Origem	Tipo de Dado	Descrição	Exemplo de Da	Relevância para Análise
time_step	silver_operational_readouts	int	Intervalo de tempo específico usado na análise.	>= 0 (esperado).	Dinâmica temporal dos dados
faixa_time_step	derivado (gold_dim_tempo)	string	Faixa associada a cada etapa de tempo.	(0-24, 24-48, <72).	Contexto da duração do intervalo

Figura 7.2: Catálogo da tabela `gold_dim_tempo`.

7.4.3 `gold_dim_classe_proximidade`

Dimensão das classes de proximidade do evento (`class_label` = 0..4), com descrições e janelas associadas.

gold_dim_classe_proximid...

Coluna	Origem	Tipo	Descrição	Exemplo de Dado	Relevância para Análise
class_label	silver_labels	int	Identificador único para cada classe de proximidade.	{0,1,2,3,4..} Nao nulo	Categorização de classes
descricao	derivado (gold_dim_classe_proximidade)	string	Descrição textual de cada classe de proximidade.	Ex.: 'muito distante'...'muito próximo'.	Entendimento da classe
janela_inicio	derivado (gold_dim_classe_proximidade)	int	Valor inicial da faixa para cada classe de proximidade.	Conforme definição em horas/steps.	Limite inferior da classe
janela_fim	derivado (gold_dim_classe_proximidade)	int	Valor final da faixa para cada classe de proximidade.	Conforme definição em horas/steps.	Limite superior da classe

Figura 7.3: Catálogo da tabela `gold_dim_classe_proximidade`.

7.4.4 `gold_fato_tempo_ate_evento`

Tabela fato no nível do veículo com variáveis de tempo observado e censura/evento.

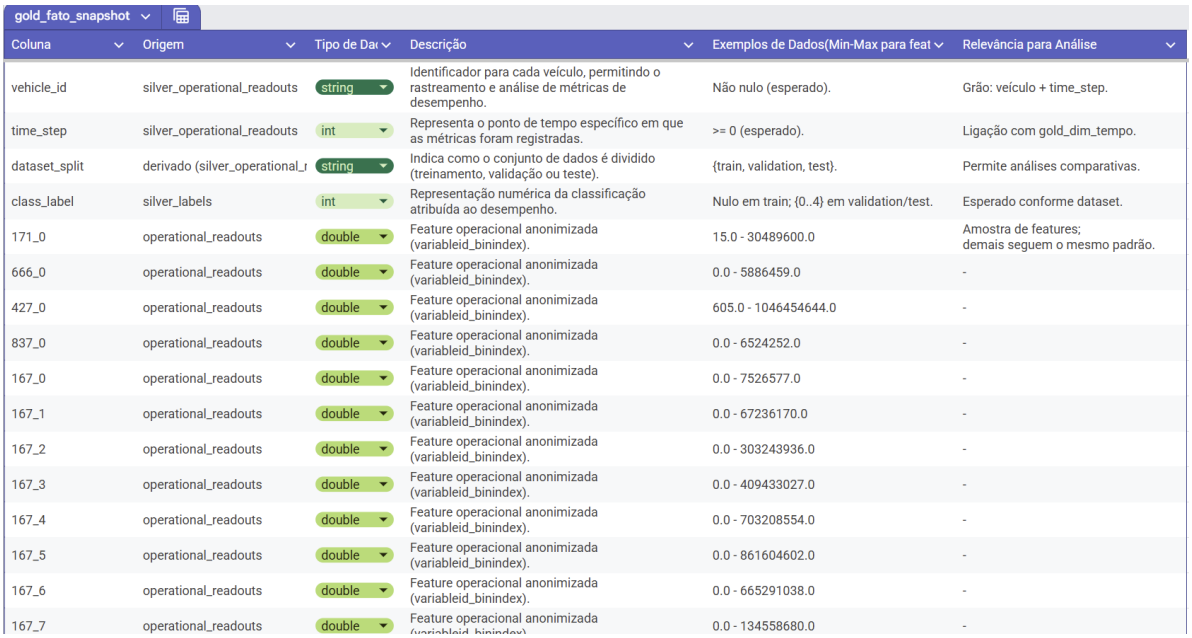
gold_fato_tempo_ate_eve...

Coluna	origem	Tipo de	Descrição	Exemplo de Dado	Relevância para Análise
vehicle_id	gold_dim_veiculo	int	Identifica o veículo para rastreamento de manutenção e histórico de reparos.	Não nulo (esperado).	Rastreamento de histórico
length_of_study	silver_train_tte	int	Duração de cada período de estudo.	>= 0 (esperado).	Avaliação de práticas de manutenção
in_study_repair	silver_train_tte	int	Indica se o veículo está em reparo.	{0,1}.	Status operacional e necessidades de manutenção

Figura 7.4: Catálogo da tabela `gold_fato_tempo_ate_evento`.

7.4.5 `gold_fato_snapshot` (amostra de features)

A tabela `gold_fato_snapshot` contém leituras operacionais anonimizadas (alto número de colunas) no padrão `variableid_binindex`. Para manter legibilidade, o catálogo apresenta as colunas-chave e uma **amostra** das features utilizadas nas análises e dashboards. As demais colunas seguem o mesmo padrão de descrição (variáveis numéricas anonimizadas).



gold_fato_snapshot

Coluna	Origem	Tipo de Dar	Descrição	Exemplos de Dados(Min-Max para feat	Relevância para Análise
vehicle_id	silver_operational_readouts	string	Identificador para cada veículo, permitindo o rastreamento e análise de métricas de desempenho.	Não nulo (esperado).	Grão: veículo + time_step.
time_step	silver_operational_readouts	int	Representa o ponto de tempo específico em que as métricas foram registradas.	>= 0 (esperado).	Ligação com gold_dim_tempo.
dataset_split	derivado (silver_operational_r	string	Indica como o conjunto de dados é dividido (treinamento, validação ou teste).	{train, validation, test}.	Permite análises comparativas.
class_label	silver_labels	int	Representação numérica da classificação atribuída ao desempenho.	Nulo em train; {0..4} em validation/test.	Esperado conforme dataset.
171_0	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	15.0 - 30489600.0	Amostra de features; demais seguem o mesmo padrão.
666_0	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 5886459.0	-
427_0	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	605.0 - 1046454644.0	-
837_0	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 6524252.0	-
167_0	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 7526577.0	-
167_1	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 67236170.0	-
167_2	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 303243936.0	-
167_3	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 409433027.0	-
167_4	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 703208554.0	-
167_5	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 861604602.0	-
167_6	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 665291038.0	-
167_7	operational_readouts	double	Feature operacional anonimizada (variableid_binindex).	0.0 - 134558680.0	-

Figura 7.5: Catálogo da tabela `gold_fato_snapshot` (colunas principais + amostra de features).

Capítulo 8

Qualidade dos Dados

Este capítulo apresenta uma avaliação objetiva da qualidade do SCANIA Component X Dataset após a ingestão e modelagem no Databricks. As verificações foram executadas principalmente sobre as tabelas da camada *Gold* (camada analítica), com auditorias adicionais na camada *Silver* quando necessário. Os resultados são reprodutíveis via consultas SQL e evidenciados por exportações (CSV) e saídas dos notebooks.

8.1 Escopo das verificações

As análises de qualidade consideram as tabelas:

- `gold_fato_snapshot`: leituras operacionais em série temporal com `dataset_split` e `class_label` quando aplicável;
- `gold_fato_tempo_ate_evento`: tempo observado no estudo (`length_of_study_time_step`) e indicador de reparo (`in_study_repair`);
- `gold_dim_veiculo`, `gold_dim_tempo`, `gold_dim_classe_proximidade`.

8.2 Completude (valores ausentes)

8.2.1 Completude em colunas-chave do fato snapshot

A Tabela 8.1 mostra que `vehicle_id` e `time_step` estão completos em todos os splits. O campo `class_label` é nulo no split *train* (100%), comportamento esperado, pois o dataset não fornece rótulos para esse split.

Tabela 8.1: Completude de colunas-chave em `gold_fato_snapshot` (por split).

Split	Total linhas	Nulos vehicle_id	Nulos time_step	Nulos class_label	% class_label nulo
test	198140	0	0	0	0,0%
train	1122452	0	0	1122452	100,0%
validation	196227	0	0	0	0,0%

8.2.2 Completude e faixa observada no TTE

Na tabela de TTE, não foram observados nulos nos campos críticos. A faixa observada para `length_of_study_time_step` foi de 73 a 510 (média 239,95).

Tabela 8.2: Qualidade do TTE em `gold_fato_tempo_ate_evento`.

Total veículos	Nulos vehicle_id	Nulos length	Nulos repair	Min	Max	Média
23550	0	0	0	73	510	239,95

8.2.3 Completude em features operacionais (todas as features avaliadas)

Como `gold_fato_snapshot` possui muitas colunas de leituras operacionais (features anonimizadas), a completude foi avaliada em todas as features numéricas incluídas no fato. A Tabela 8.3 consolida a proporção de células nulas considerando **(linhas × features)**.

Tabela 8.3: Resumo de completude das features operacionais (por split).

Split	Total linhas	Features avaliadas	Features com nulos	Total nulos	% células nulas
test	198140	96	95	65368	0,344%
train	1122452	96	95	349144	0,324%
validation	196227	96	95	59349	0,315%

Para contextualizar a distribuição dos nulos, observa-se que a ausência de valores se concentra em famílias específicas de colunas. A Tabela 8.4 apresenta os principais grupos (por prefixo) com maior proporção de células nulas no split *train*.

Tabela 8.4: Principais grupos de features com maior proporção de células nulas (split train).

Prefixo	# features	Total de nulos	% células nulas
291	11	105908	0,858%
459	20	167060	0,744%
427	1	6405	0,571%
100	1	6401	0,570%
370	1	6398	0,570%
167	10	28620	0,255%

8.3 Validade (domínios e valores esperados)

As verificações de domínio não identificaram anomalias:

- Classes fora do domínio 0–4: **0**.
- `time_step` negativo: **0**.

8.4 Integridade referencial (consistência entre fato e dimensões)

As verificações entre fato e dimensões indicaram ausência de chaves órfãs:

- Fato → Dim veículo: **0** chaves sem correspondência.
- Fato → Dim tempo: **0** chaves sem correspondência.
- Fato → Dim classe (quando houver label): **0** chaves sem correspondência.

8.5 Unicidade e auditoria de duplicidade

8.5.1 Duplicidade na granularidade do fato snapshot

Ao verificar a granularidade (`vehicle_id`, `time_step`, `dataset_split`) em `gold_fato_snapshot`, foi identificada duplicidade: **116619** chaves aparecem mais de uma vez. Isso indica que o dataset possui múltiplos registros para o mesmo veículo e instante (no mesmo split), o que deve ser considerado na interpretação de agregações e contagens.

8.5.2 Auditoria: duplicidade na origem (camada Silver)

Para confirmar se essa duplicidade foi introduzida por transformações ou joins na camada Gold, foi executada uma auditoria diretamente na camada Silver, antes de qualquer modelagem analítica.

Consulta de auditoria (Silver):

Listing 8.1: Auditoria de duplicidade na Silver

```
SELECT
  vehicle_id,
  time_step,
  COUNT(*) as qtd
FROM silver_operational_readouts
GROUP BY vehicle_id, time_step
HAVING COUNT(*) > 1;
```

A Tabela 8.5 mostra uma amostra do resultado, evidenciando chaves com repetição, incluindo o caso `vehicle_id=28772` no `time_step=9` com `qtd=3`.

Tabela 8.5: Amostra do resultado da auditoria de duplicidade na camada Silver.

vehicle_id	time_step	qtd
28675	85	2
28680	121	2
28682	123	2
28684	84	2
28702	172	2
28723	196	2
28728	27	2
28732	150	2
28739	136	2
28772	9	3

8.5.3 Inspeção do exemplo (não é duplicação exata)

Para detalhar o caso `vehicle_id=28772` e `time_step=9`, foi realizada a inspeção direta dos registros:

Listing 8.2: Inspeção de registros duplicados (exemplo)

```
SELECT *
FROM silver_operational_readouts
WHERE vehicle_id = 28772 AND time_step = 9;
```

A Tabela 8.6 apresenta um recorte com algumas features e mostra que os três registros possuem valores diferentes (por exemplo, em `171_0`, `666_0`, `427_0`, `837_0`), indicando que não se trata de uma duplicação exata, mas de múltiplas leituras para a mesma chave composta.

Tabela 8.6: Recorte da inspeção dos registros para `vehicle_id=28772` e `time_step=9` (camada Silver).

vehicle_id	time_step	dataset_split	171_0	666_0	427_0	837_0	167_0	167_1	167_2
28772	9	train	245490	6720	8196574	848	3315	308790	547785
28772	9	train	263820	7287	8837588	904	3316	332340	589905
28772	9	train	272640	7910	9162066	960	3316	345135	609495

8.6 Síntese do diagnóstico

Em síntese, as verificações indicam:

- **Completude** adequada em chaves de análise e em TTE; a ausência de `class_label` em *train* é esperada;
- **Baixa taxa de nulos** nas features operacionais quando analisado o total de células (aprox. 0,315%–0,344% por split);

- **Domínios válidos** (classes 0–4 e `time_step` não-negativo);
- **Integridade referencial** consistente entre fato e dimensões (0 chaves órfãs);
- **Ponto de atenção**: existe duplicidade na granularidade (`vehicle_id` + `time_step`) já na camada Silver, e a inspeção do exemplo mostra que não é duplicação exata (há valores distintos nas features). Como trabalho futuro, recomenda-se definir uma política determinística de consolidação do grão (por exemplo, agregação) caso análises exijam unicidade estrita por snapshot.

Capítulo 9

Análises e Dashboards: Respostas às Perguntas do Objetivo

Este capítulo consolida as análises do MVP com foco em **responder explicitamente** às perguntas de negócio definidas no objetivo do trabalho. As visualizações foram construídas no Databricks a partir das tabelas da camada *Gold* (`gold_fato_tempo_ate_evento`, `gold_dim_veiculo`, `gold_dim_tempo` e `gold_fato_snapshot`). Ao final, os gráficos foram organizados em dashboards para facilitar a inspeção.

9.1 Pergunta 1 — Distribuição do tempo até reparo (tempo observado no estudo)

9.1.1 Dashboards e gráficos

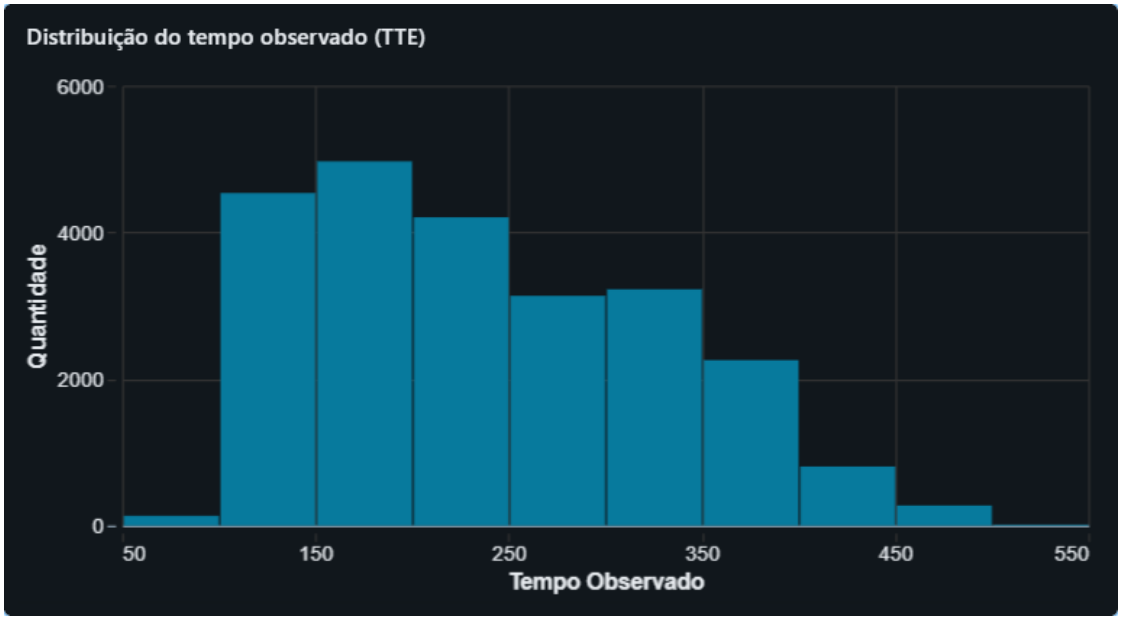


Figura 9.1: Distribuição do tempo observado no estudo (`length_of_study_time_step`).

9.1.2 Resposta encontrada

- O tempo observado apresenta maior concentração entre **150 e 250**, com pico de frequência no bin **150–200**.
- A maior parte dos veículos não registrou reparo no período observado: **90,35%** sem reparo e **9,65%** com reparo.

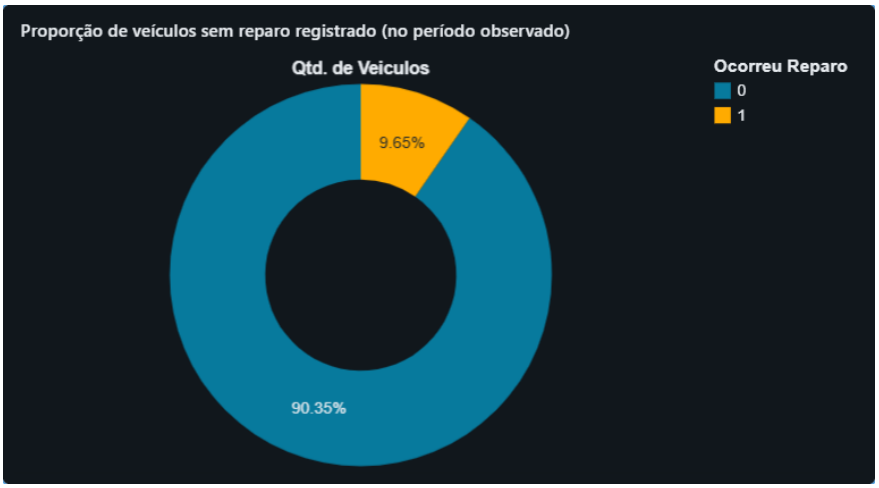


Figura 9.2: Proporção de veículos com/sem reparo registrado (no período observado).

9.1.3 Consultas SQL utilizadas

Listing 9.1: TTE: valores para histograma (tempo observado)

```
SELECT
  length_of_study_time_step
FROM gold_fato_tempo_ate_evento
WHERE length_of_study_time_step IS NOT NULL;
```

Listing 9.2: Proporção de veículos com/sem reparo registrado

```
SELECT
  in_study_repair,
  COUNT(*) AS qtd_veiculos,
  ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER (), 2) AS perc_veiculos
FROM gold_fato_tempo_ate_evento
GROUP BY in_study_repair
ORDER BY in_study_repair;
```

9.2 Pergunta 2 — Especificações de veículo mais críticas

9.2.1 Dashboards e gráficos

9.2.2 Resposta encontrada

- Há heterogeneidade entre combinações de `spec_1` e `spec_2`.
- Regiões em **vermelho** indicam menor mediana (tempo típico mais curto) e regiões em **azul** indicam maior mediana.

9.2.3 Consultas SQL utilizadas

Listing 9.3: Heatmap spec_1 x spec_2 vs mediana do tempo observado

```
SELECT
  spec_1,
  spec_2,
  percentile_approx(length_of_study_time_step, 0.5) AS mediana_tempo
FROM gold_dim_veiculo
WHERE length_of_study_time_step IS NOT NULL
GROUP BY spec_1, spec_2
ORDER BY spec_1, spec_2;
```

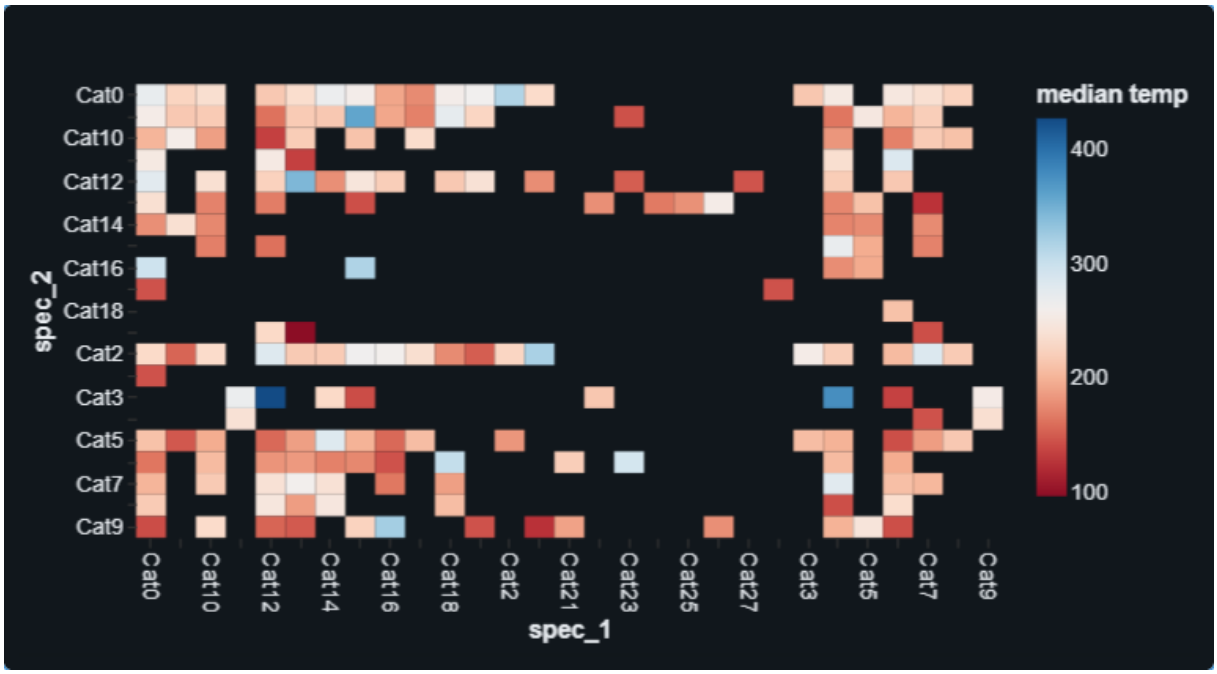


Figura 9.3: Heatmap: combinação de especificações (spec_1 × spec_2) versus mediana do tempo observado.

9.3 Pergunta 3 — Comportamento temporal da degradação (feature operacional)

9.3.1 Dashboards e gráficos

9.3.2 Resposta encontrada

- A média da feature `171_0` aumenta progressivamente conforme as faixas temporais avançam.
- A análise é agregada e sugere variação temporal do comportamento operacional.

9.3.3 Consultas SQL utilizadas

Listing 9.4: Média da feature 171_0 por faixa temporal

```
SELECT
  t.faixa_time_step,
  AVG(f.‘171_0‘) AS media_171_0
FROM gold_fato_snapshot f
JOIN gold_dim_tempo t
  ON f.time_step = t.time_step
GROUP BY t.faixa_time_step
ORDER BY
  CASE
    WHEN t.faixa_time_step = ’0-24’ THEN 1
    WHEN t.faixa_time_step = ’24-48’ THEN 2
    WHEN t.faixa_time_step = ’48-72’ THEN 3
    WHEN t.faixa_time_step = ’>72’ THEN 4
    ELSE 99
  END;
```

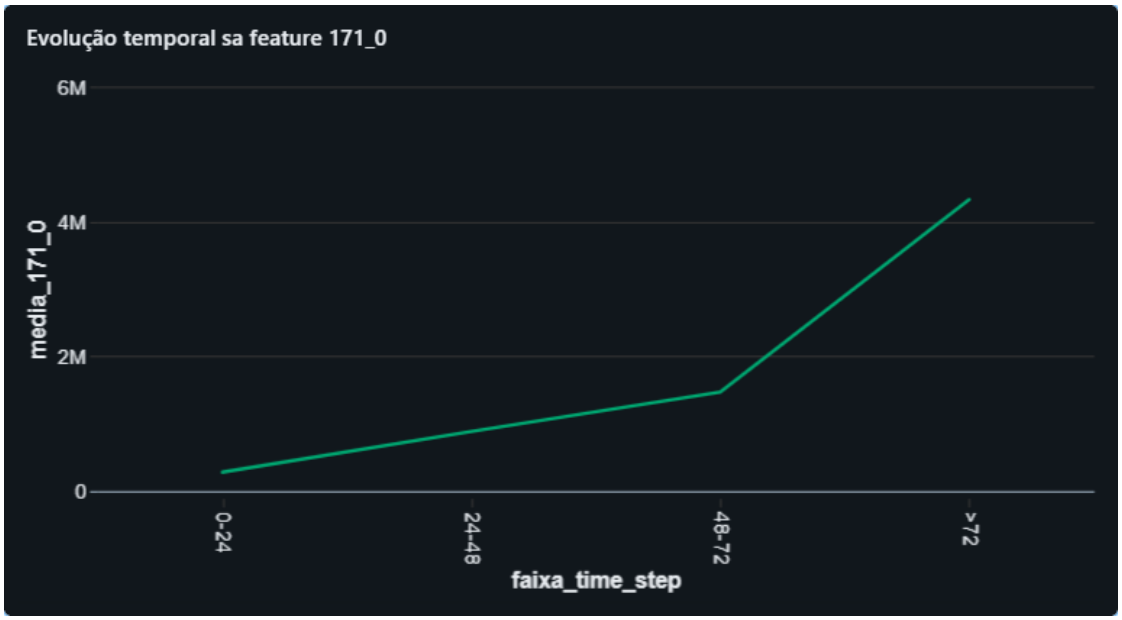


Figura 9.4: Evolução temporal agregada da feature 171_0 por faixa de tempo.

9.4 Pergunta 4 — Distribuição das classes de proximidade de falha (0–4)

9.4.1 Dashboards e gráficos

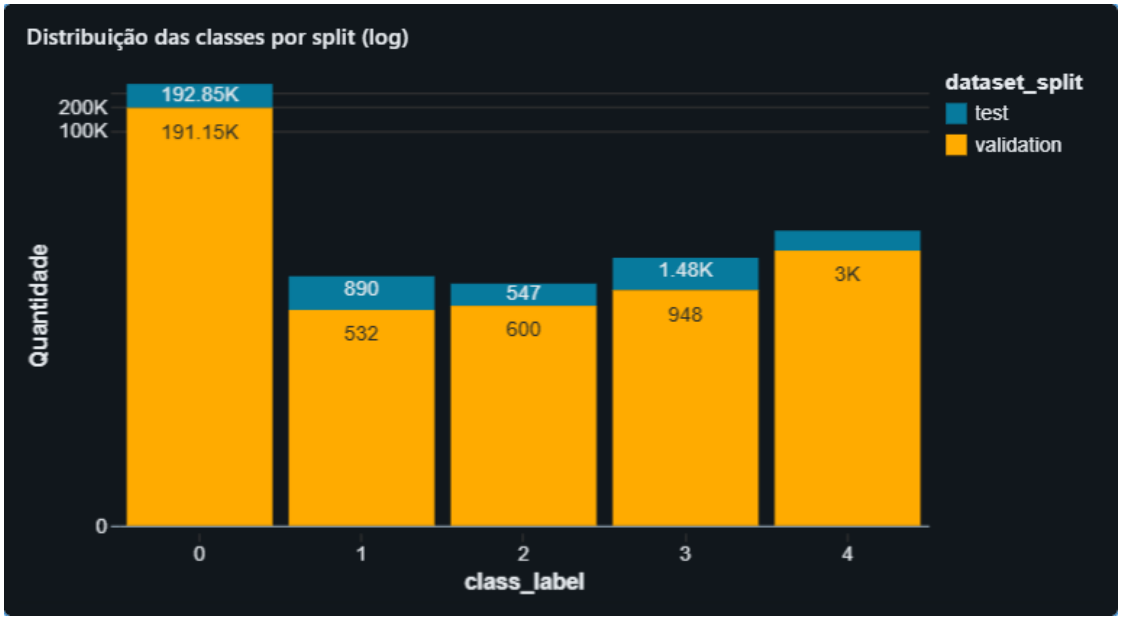


Figura 9.5: Distribuição das classes (0–4) por split (escala logarítmica).

9.4.2 Resposta encontrada

- Há forte desbalanceamento com concentração na classe 0 e menor frequência nas demais.
- As classes estão disponíveis apenas para *validation* e *test*.

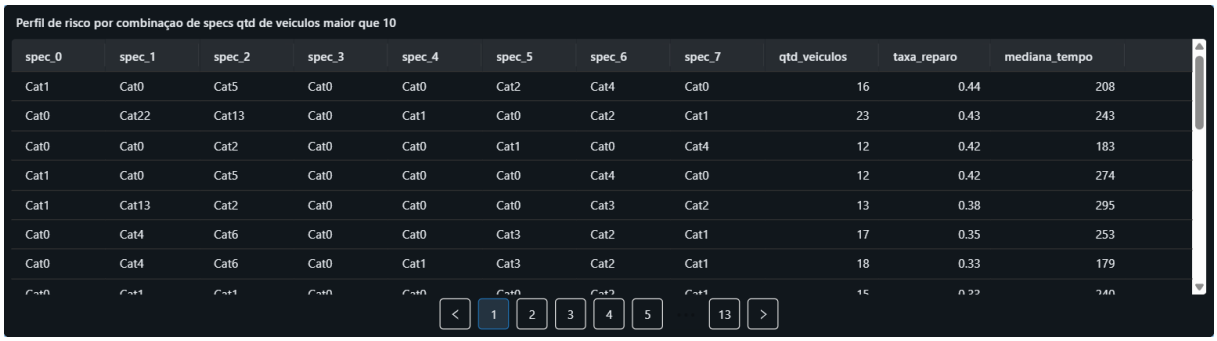
9.4.3 Consultas SQL utilizadas

Listing 9.5: Contagem de classes por split (validation/test)

```
SELECT
  dataset_split,
  class_label,
  COUNT(*) AS qtd
FROM gold_fato_snapshot
WHERE dataset_split IN ('validation','test')
  AND class_label IS NOT NULL
GROUP BY dataset_split, class_label
ORDER BY class_label, dataset_split;
```

9.5 Pergunta 5 — Perfis operacionais de maior risco (combinações de especificações)

9.5.1 Dashboards e gráficos



Perfil de risco por combinaçao de specs qtd de veiculos maior que 10

spec_0	spec_1	spec_2	spec_3	spec_4	spec_5	spec_6	spec_7	qtd_veiculos	taxa_reparo	mediana_tempo
Cat1	Cat0	Cat5	Cat0	Cat0	Cat2	Cat4	Cat0	16	0.44	208
Cat0	Cat22	Cat13	Cat0	Cat1	Cat0	Cat2	Cat1	23	0.43	243
Cat0	Cat0	Cat2	Cat0	Cat0	Cat1	Cat0	Cat4	12	0.42	183
Cat1	Cat0	Cat5	Cat0	Cat0	Cat0	Cat4	Cat0	12	0.42	274
Cat1	Cat13	Cat2	Cat0	Cat0	Cat0	Cat3	Cat2	13	0.38	295
Cat0	Cat4	Cat6	Cat0	Cat0	Cat3	Cat2	Cat1	17	0.35	253
Cat0	Cat4	Cat6	Cat0	Cat1	Cat3	Cat2	Cat1	18	0.33	179

< 1 2 3 4 5 13 >

Figura 9.6: Perfis de risco por combinação de especificações (filtrado para `qtd_veiculos` > 10).

9.5.2 Resposta encontrada

- Combinações de especificações com maior taxa de reparo e menor mediana de tempo observado indicam perfis mais críticos.
- O filtro `qtd_veiculos > 10` aumenta robustez ao reduzir ruído de grupos raros.

9.5.3 Consultas SQL utilizadas

Listing 9.6: Perfil de risco por combinação de specs com suporte > 10

```
SELECT
  v.spec_0, v.spec_1, v.spec_2, v.spec_3, v.spec_4, v.spec_5, v.spec_6, v.spec_7,
  COUNT(*) AS qtd_veiculos,
  ROUND(AVG(CASE WHEN v.in_study_repair = 1 THEN 1 ELSE 0 END), 2) AS taxa_reparo,
  percentile_approx(v.length_of_study_time_step, 0.5) AS mediana_tempo
FROM gold_dim_veiculo v
GROUP BY v.spec_0, v.spec_1, v.spec_2, v.spec_3, v.spec_4, v.spec_5, v.spec_6, v.spec_7
HAVING COUNT(*) > 10
ORDER BY taxa_reparo DESC, mediana_tempo ASC;
```

9.6 Síntese das respostas

Em resumo, o MVP respondeu de forma direta a distribuição do tempo observado e a proporção de veículos com/sem reparo no período, mostrou evidências de heterogeneidade por especificações via heatmap, caracterizou variação temporal agregada em uma feature operacional e confirmou desbalanceamento das

classes de proximidade (0–4) nos splits rotulados. Além disso, a análise de perfis por combinações de especificações permitiu identificar grupos com maior taxa de reparo e tempo típico menor, fornecendo uma base inicial para políticas de monitoramento e manutenção mais direcionadas.

Capítulo 10

Autoavaliação

10.1 Objetivos atingidos no MVP

O objetivo central deste MVP foi a construção de um pipeline de dados *end-to-end* na nuvem, utilizando a plataforma Databricks Free Edition para processar o *SCANIA Component X Dataset*. O trabalho buscou transformar dados brutos e complexos em uma camada analítica estruturada (Esquema Estrela), capaz de responder a questões de negócio críticas sobre manutenção preditiva. O dataset, descrito na literatura como um conjunto *real-world* anonimizado e multivariado, apresentou desafios típicos de Big Data Industrial.

A seguir, apresenta-se a discussão detalhada sobre o atingimento de cada pergunta de negócio:

10.1.1 Pergunta 1 — Distribuição do tempo até reparo (TTE)

Status: Plenamente Atingida. A análise da tabela fato `gold_fato_tempo_ate_evento` permitiu caracterizar a distribuição do tempo observado até o reparo do componente X. Identificou-se que o reparo é um evento raro no horizonte analisado (apenas 9,65% dos veículos tiveram reparo registrado), evidenciando desbalanceamento entre veículos com e sem evento no período observado. As visualizações indicaram concentração do tempo até reparo em faixas temporais específicas (150-250), sugerindo que o evento não ocorre de forma uniforme ao longo do tempo observado.

10.1.2 Pergunta 2 — Especificações de veículo mais críticas

Status: Atingida com limitações de anonimização. Através do cruzamento dimensional com `gold_dim_veiculo`, foi possível calcular indicadores de criticidade por categorias de especificação (ex.: `spec_0`) e por combinações de especificações (ex.: `spec_1` × `spec_2`). Os resultados indicaram que certas configurações apresentam risco superior à média da frota. Embora a anonimização impeça a interpretação física direta das categorias, a análise identificou **grupos de risco**, permitindo que uma equipe com acesso ao dicionário real priorize ações preventivas nesses grupos.

10.1.3 Pergunta 3 — Comportamento temporal da degradação

Status: Atingida. A análise temporal do `gold_fato_snapshot` permitiu observar mudanças no comportamento de leituras operacionais ao longo do tempo. A partir de visualizações (linhas/heatmaps) foi possível evidenciar que algumas features operacionais (por exemplo, `171_0`) apresentam aumento acentuado em janelas mais próximas do evento, reforçando a hipótese de degradação progressiva e sustentando a viabilidade de estratégias de manutenção preditiva para o componente X.

10.1.4 Pergunta 4 — Distribuição das classes de proximidade

Status: Atingida para validação; limitada para treino/teste. Confirmou-se desbalanceamento nas classes de proximidade no split de validação, com a classe crítica (4) representando uma fração minoritária dos snapshots. No split de treino, o dataset não disponibiliza `class_label`, e no split de teste os rótulos não estão disponíveis, limitando comparações diretas entre todos os splits.

10.1.5 Pergunta 5 — Perfis operacionais de risco

Status: Parcialmente Atingida. Foi possível correlacionar altas leituras de sensores com a iminência de falha. No entanto, a definição de um "perfil operacional"completo (ex: "veículos que operam em alta rotação por longos períodos") exigiria a engenharia de features agregadas (ex: média móvel, tempo acima de limiar) que extrapolam o escopo deste MVP de Engenharia de Dados, entrando na esfera de Ciência de Dados.

10.2 Principais desafios e decisões técnicas

10.2.1 Limites da Plataforma Free Edition e Decisões de Arquitetura

Embora o Databricks Free Edition utilize clusters modestos (SQL Warehouse 2X-Small), o processamento dos arquivos (incluindo o `train_operational_readouts.csv` de $>$ 1GB) ocorreu sem gargalos de performance, validando a escolha do dataset para este ambiente. A decisão de arquitetura focou em boas práticas desde a ingestão: utilização de **Unity Catalog Volumes** para organização dos arquivos brutos e persistência imediata em formato **Delta Lake**. Essa estratégia garantiu não apenas performance, mas também governança e ACID compliance, demonstrando que a arquitetura foi desenhada para escalabilidade futura, independentemente das limitações atuais do ambiente gratuito.

10.2.2 Anonimização e construção do catálogo

Um desafio relevante foi a anonimização dos atributos operacionais. A documentação descreve que os atributos do dado operacional foram anonimizados por razões proprietárias, e que incluem contadores numéricos e histogramas por bins, o que impede inferir a semântica real dos sensores. [1] Como resposta de Engenharia de Dados, o catálogo foi construído descrevendo natureza, tipo e domínios observados das variáveis, evitando interpretações não sustentadas (ex.: “temperatura”, “pressão”) quando não documentadas.

10.2.3 Exportação de evidências (outputs) e reprodutibilidade

Como os notebooks exportados para GitHub podem não preservar outputs em `.ipynb` no fluxo adotado, foi necessário exportar os notebooks como **HTML** e manter evidências visuais dos resultados e dashboards, garantindo que o corretor consiga validar execução e resultados sem depender do ambiente Databricks. Esses estarão tanto no Repositorio do Github quanto em links no Apendice A

10.3 Achado relevante: auditoria de qualidade e duplicidade na telemetria

A literatura e documentação do dataset o descrevem como *real-world* e menciona que a coleta em frota envolve desafios e potenciais erros, inclusive indicando que o pós-processamento pode não cobrir todos os casos. [1] Além disso, a documentação do desafio industrial descreve que cada linha representa um *readout* operacional (sugere unicidade por leitura). [2]

Durante este MVP, foi realizada uma auditoria específica na camada **Silver**, antes de qualquer modelagem dimensional, verificando a cardinalidade da chave composta `(vehicle_id, time_step)`. Essa auditoria identificou **registros duplicados/triplicados** para a mesma chave (ex.: `vehicle_id = 28772, time_step = 9`). Após uma analise ainda mais detalhada em um dos *vehicle_id* , foram encontrados no `vehicle_id = 28772` três registros com valores diferentes (por exemplo, em 171_0, 666_0, 427_0, 837_0), conforme a Tabela 8.6, indicando que **não** se trata de uma duplicação exata, mas de múltiplas leituras para a mesma chave composta, evidenciando redundâncias na origem ou na coleta/curadoria disponibilizada.

Esse achado é **relevante** porque duplicidades na telemetria podem introduzir viés em análises de contagem e frequência, e podem afetar agregações estatísticas se não forem explicitamente tratadas. Assim, o MVP documenta o risco e registra evidências, fortalecendo a confiabilidade do pipeline e das conclusões apresentadas.

10.4 Trabalhos Futuros

Para evoluir este MVP em direção a um produto produtivo, sugere-se:

- **Automação via Workflow:** Orquestrar os notebooks como um Job recorrente no Databricks, permitindo a ingestão incremental de novos dados de telemetria.
- **Enriquecimento Semântico:** Caso o dicionário real fosse disponibilizado, traduzir as features anonimizadas para termos de negócio (ex: "Pressão do Óleo"), facilitando a adoção por equipes de manutenção.
- **Baseline de Machine Learning:** Treinar um classificador (XGBoost ou Random Forest) utilizando a tabela `gold_fato_snapshot` para prever a probabilidade de falha em D+1, fechando o ciclo de manutenção preditiva 4.0.

10.5 Conclusão

O MVP atingiu com êxito seu propósito. Mais do que apenas processar dados, o projeto entregou um ativo de informação confiável e auditado. A arquitetura Medalhão provou-se eficaz para organizar o fluxo, e as análises geradas oferecem suporte direto à tomada de decisão, identificando quais veículos e sensores merecem atenção imediata.

Apêndice A

Notebooks e Evidências de Execução

Este apêndice organiza as evidências de execução do pipeline e dos resultados analíticos (notebooks exportados, consultas SQL e dashboards), de forma a permitir validação rápida do trabalho.

A.1 Repositório e estrutura de arquivos

- Repositório GitHub: `<https://github.com/Igor-C-Assuncao/mvp-engenharia-dados-scania>`
- Pasta de notebooks exportados (HTML): `/NotebooksComOutput/`
- Querys SQLs e CSVs com resultados das queries de qualidade: `/QuerysSQL/`

A.2 Notebooks exportados (HTML)

- Notebook 01 — Ingestão Bronze: Abrir no GitHub
- Notebook 02 — Tratamento Silver: Abrir no GitHub
- Notebook 03 — Modelagem Gold: Abrir no GitHub
- Notebook 04 — Análises SQL: Abrir no GitHub

A.3 Dashboards (prints)

- Histograma do TTE: `dash_tte_histograma.png` (Figura 9.1)
- Taxa de reparo por `spec_0`: `dash_taxa_reparo_spec0.png` (Figura 9.2)
- Heatmap `spec_1` × `spec_2`: `dash_heatmap_spec1_spec2.png` (Figura 9.3)
- Distribuição das classes (validação): `Distribuição_das_classes_por_split.png` (Figura 9.5)
- Evolução temporal de feature selecionada: `Evolução_temporal _sa_feature_171_0.png` (Figura 9.4)
- Perfil de risco por combinação de specs: `Perfil_de_risco_por_qtd_maior_que_10.png` (Figura 9.6)

A.4 Consultas SQL (qualidade e auditoria)

Os resultados utilizados no Capítulo 8 foram exportados em CSV e encontram-se no repositorio em : */QuerysSQL/*:

- `Nulos_em_colunas_chave.csv`
- `Nulos_no_TTE.csv`

- Completude_amostral_features.csv
- Checagem_duplicidade_gold.csv
- auditoria_camada_silver.csv
- verif_duplicadas.csv

Referências Bibliográficas

[1] Zahra Kharazian, Tony Lindgren, Sindri Magnússon, Olof Steinert, and Oskar Andersson Reyna. Scania component x dataset: a real-world multivariate time series dataset for predictive maintenance. *Scientific Data*, 12(1):493, 2025.

[2] Jie Zhong and Zhenkan Wang. Implementing deep learning models for imminent component x failures prediction in heavy-duty scania trucks. In *International Symposium on Intelligent Data Analysis*, pages 268–276. Springer, 2024.