# Dicionário de Dados - Camada Bronze (Raw Data)

Este documento descreve a estrutura e os metadados das fontes de dados na Camada Bronze. Esta camada serve como o repositório inicial para todos os dados brutos, exatamente como foram ingeridos, antes de qualquer limpeza ou transformação.

## 1. Posições dos Ônibus (API SPTrans)

Fonte: API /Posicoes da SPTrans.

Atualização: A cada 2 minutos, via processo no NiFi.

Propósito: Capturar o estado da frota de ônibus em tempo quase real. Este é o dado primário para todos os pipelines, tanto de lote (batch) quanto de tempo real (streaming).

### 1.1. Armazenamento em Data Lake (MinIO)

* **Localização:** s3a://bronze/sptrans/posicao/
* **Formato:** JSON
* **Particionamento:** Os arquivos são organizados em pastas por ano, mes, dia e hora (UTC) para otimizar as leituras do pipeline de lote.
* **Estrutura do JSON:** O arquivo JSON possui uma estrutura aninhada.

| **Campo** | **Nível** | **Tipo** | **Descrição** |
| --- | --- | --- | --- |
| hr | Raiz | Texto | A hora da consulta à API (ex: "20:15"). |
| l | Raiz | Array | Um array de objetos, onde cada objeto representa uma **linha de ônibus** que possui veículos em operação. |
| c | Em l | Texto | Letreiro completo da linha (ex: "8000-10"). |
| cl | Em l | Número | Código identificador numérico da linha. |
| vs | Em l | Array | Um array de objetos, onde cada objeto representa um **veículo (ônibus)** daquela linha. |
| p | Em vs | Número | Prefixo do veículo, seu identificador único. |
| a | Em vs | Booleano | true se o veículo é acessível para pessoas com deficiência. |
| ta | Em vs | Texto | Timestamp (em UTC) da última atualização da posição do veículo. |
| py | Em vs | Número | Coordenada de Latitude da posição do veículo. |
| px | Em vs | Número | Coordenada de Longitude da posição do veículo. |

### 1.2. Armazenamento em Barramento de Eventos (Kafka)

* **Tópico:** sptrans\_posicoes\_raw
* **Formato da Mensagem:** O corpo da mensagem é o mesmo arquivo JSON bruto descrito acima.
* **Propósito:** Servir como um buffer de alta performance para alimentar os pipelines de streaming em tempo real, permitindo que múltiplos consumidores leiam os mesmos dados de forma independente.

## 2. Dados Cadastrais das Linhas (GTFS)

Fonte: Arquivo routes.txt do feed estático GTFS (General Transit Feed Specification).

Atualização: Manual ou em baixa frequência. Os arquivos são colocados no MinIO.

Propósito: Servir como uma fonte de dados de enriquecimento (dimensão) para associar os códigos das linhas aos seus nomes completos, tornando os relatórios mais legíveis.

* **Localização:** s3a://bronze/gtfs/routes.txt
* **Formato:** CSV (Comma-Separated Values) com cabeçalho.

### Estrutura das Colunas

| **Nome da Coluna** | **Tipo de Dado** | **Descrição** |
| --- | --- | --- |
| route\_id | Texto | Identificador único da linha, que corresponde ao letreiro\_linha nos dados da API. |
| agency\_id | Número | Identificador da agência de transporte. |
| route\_short\_name | Texto | Um nome curto para a linha (geralmente o mesmo que route\_id). |
| route\_long\_name | Texto | O nome completo e descritivo da linha (ex: "Term. Lapa - Pça. Ramos de Azevedo"). |
| route\_type | Número | Tipo de transporte (ex: 3 para ônibus). |
| route\_color | Texto | Cor associada à linha (código hexadecimal). |
| route\_text\_color | Texto | Cor do texto para ser usado sobre a route\_color. |