

## ■ Documentação Técnica: Construção da Camada Gold (Data Mart)

**Status:** ✅ Concluído **Data:** 30/01/2026 **Objetivo:** Criar a tabela final de inteligência de mercado com indicadores pré-calculados.

### 1. Configuração do Projeto (dbt\_project.yml)

Para garantir performance, configuramos a camada Gold para ser materializada como **Tabela Física** (não View). Isso faz com que o Power BI leia os dados instantaneamente, pois o cálculo pesado já foi feito e salvo no disco.

Alteração realizada no dbt\_project.yml:

```
YAML

models:
  transformacao_btc:
    # Camada Silver (Lógica de Limpeza) = View Virtual
    silver:
      +materialized: view
      +schema: silver

    # Camada Gold (Inteligência de Negócio) = Tabela Física
    gold:
      +materialized: table
      +schema: gold
```

### 2. Estrutura de Pastas

Organização dos modelos SQL:

```
Plaintext

models/
├── bronze/
│   └── sources.yml          (Mapeamento da origem)
├── silver/
│   └── stg_bitcoin.sql     (Limpeza e Deduplicação)
└── gold/
    └── mart_bitcoin_indicadores.sql (Cálculo de Indicadores)
```

### 3. Lógica de Negócio (mart\_bitcoin\_indicadores.sql)

Criamos uma query SQL avançada utilizando **Window Functions** para gerar indicadores de Trading sem necessidade de Python.

**Transformações Aplicadas:**

1. **MMS 7D (Média Móvel Curta):** AVG das últimas 7 linhas.
2. **MMS 30D (Média Móvel Longa):** AVG das últimas 30 linhas.
3. **Price Ontem (Lag):** Função LAG para buscar o valor do dia anterior (D-1).
4. **Volatilidade Diária (%):** Cálculo percentual entre Hoje e Ontem.

Código SQL:

```
    {{ config(
      materialized='table',
      schema='gold'
    ) }}
```

```
WITH silver_data AS (
    SELECT * FROM {{ ref('stg_bitcoin') }}
),
```

```
calculos AS (
    SELECT
        updated_at,
        price_usd,
        -- Mantemos o cálculo com precisão total aqui no CTE
        AVG(price_usd) OVER(ORDER BY updated_at ROWS BETWEEN 6 PRECEDING AND
CURRENT ROW) as mms_7d,
        AVG(price_usd) OVER(ORDER BY updated_at ROWS BETWEEN 29 PRECEDING AND
CURRENT ROW) as mms_30d,
        LAG(price_usd) OVER(ORDER BY updated_at) as price_ontem
    FROM silver_data
)
```

```
SELECT
    updated_at,

    -- 1. Arredondar o Preço para 2 casas
```

```
CAST(price_usd AS NUMERIC(18,2)) as price_usd,
```

```
-- 2. Arredondar as Médias Móveis
```

```
CAST(mms_7d AS NUMERIC(18,2)) as mms_7d,
```

```
CAST(mms_30d AS NUMERIC(18,2)) as mms_30d,
```

```
-- 3. Tratamento de NULL + Arredondamento da Variação
```

```
-- COALESCE: Se o resultado for NULL, troca por 0
```

```
COALESCE(
```

```
  CAST(
```

```
    ((price_usd - price_ontem) / NULLIF(price_ontem, 0)) * 100
```

```
  AS NUMERIC(18,2)),
```

```
  0
```

```
) as variacao_pct
```

```
FROM calculos
```

```
ORDER BY updated_at DESC
```

#### 4. Execução e Carga

O modelo foi processado pelo dbt, que compilou o SQL, criou a tabela no schema gold do Supabase e inseriu os dados históricos.

##### Comando utilizado:

```
PowerShell:
```

```
dbt run --select transformacao_btc.gold.mart_bitcoin_indicadores --profiles-dir .
```

##### Resultado:

- **Objeto Criado:** Tabela postgres.public\_gold.mart\_bitcoin\_indicadores.
- **Volume:** Histórico completo processado.
- **Tempo de Execução:** ~1.8 segundos.

---

#### ✅ Projeto de Engenharia de Dados Concluído!

Você agora tem um pipeline **ELT (Extract, Load, Transform)** completo:

1. **Ingestão:** Python pega da API e joga na Bronze.

2. **Limpeza:** dbt cria a Silver (sem duplicatas).
3. **Inteligência:** dbt cria a Gold (com médias móveis).

**Próximo Passo:** Conectar o Power BI nessa tabela mart\_bitcoin\_indicadores.