

---

# Documentação do Projeto: Análise de Dados Olist E-commerce

---

## SUMÁRIO

### Capítulo 1 - Resumo Executivo

- Introdução..... 03

### Capítulo 2 - Metodologia CRISP-DM

- 2.1 Fases Aplicadas no Projeto..... 03

### Capítulo 3 - Entendimento do Negócio

- 3.1 Levantamento de Métricas e KPIs Principais..... 04
- 3.2 Requisitos do Setor Comercial..... 05
- 3.3 Dimensões de Análise (Conceituais) ..... 06
- 3.4 Requisitos do Setor de Marketing..... 07
- 3.5 Requisitos do Setor de Operações.....07

### Capítulo 4 - Entendimento dos Dados

- 4.1 Inventário das Fontes de Dados..... 08
- 4.2 Avaliação da Qualidade dos Dados..... 09
- 4.3 Análise Exploratória.....10
- 4.4 Processo de Limpeza e Tratamento dos Dados..... 11

### Capítulo 5 - Modelagem dos Dados

- 5.1 Modelo Estrela (Star Schema) ..... 14
- 5.2 Views SQL – Camada Semântica do Projeto.....15
- 5.3 Considerações de Governança e Manutenção..... 18

### Capítulo 6 - Estratégia de Solução

- 6.1 Tradução dos Problemas de Negócio..... 18
- 6.2 Framework Analítico..... 19
- 6.3 Critérios de Priorização das Análises..... 19

### Capítulo 7- Dashboards Estratégicos

- 7.1 Visão Geral dos Dashboards..... 20
- 7.2 Estrutura Analítica dos Dashboards..... 21
- 7.3 Observações Técnicas Importantes.....22
- 7.4 Detalhamento Individual dos Dashboards..... 22

## **Capítulo 8 - Stack Tecnológico**

- 8.1 Componentes Utilizados..... 28
- 8.2 Racional Técnico da Arquitetura..... 28
- 8.3 Boas Práticas Aplicadas..... 29

## **Capítulo 9 - Lições Aprendidas**

- 9.1 Lições de Negócio..... 29
- 9.2 Lições Técnicas..... 30
- 9.3 Lições de Processo..... 30

## **Capítulo 10 - Conclusão e Próximos Passos**

- 10.1 Conclusão ..... 31
- 10.2 Próximos Passos..... 31

## **Capítulo 11 - Anexos**

- 11.1 Queries Oficiais do Projeto (SQL) ..... 32
  - Testes de Qualidade (T001 a T024) ..... 32
  - Construção da Tabela Fato (C001) .....61
  - Views Principais (V001, V002, V003) ..... 65
  - Views de Suporte (S001 a S007) ..... 68
- 11.2 Glossário Técnico ..... 82
  - Termos Gerais ..... 82
  - Métricas de Negócios (Comercial)..... 83
  - Métricas de Marketing (CRM )..... 83
  - Métricas Operacionais e Logísticas ..... 83
  - Dimensões do Modelo Estrela.....84
  - Termos Técnicos de Modelagem & BI .....84
  - SQL e Engenharia de Dados .....84
  - Termos Estatísticos Utilizados .....85

## Capítulo 1 - Resumo Executivo

### Introdução

Este projeto apresenta a construção completa de um ambiente analítico para o marketplace Olist, incluindo modelagem dimensional, validação das bases brutas e criação de views temáticas para suporte às áreas Comercial, Marketing e Operações. Toda a solução foi desenvolvida no Google BigQuery, seguindo boas práticas de engenharia, governança e padronização de métricas.

A arquitetura proposta integra dados de pedidos, pagamentos, logística, produtos e clientes em uma **tabela fato consolidada (fato\_pedido)**. A partir dela, foram construídas **views otimizadas** para análises de GMV, ticket médio, cancelamentos, inadimplência, recência/frequência (RFM), LTV, desempenho logístico e rankings de categorias e vendedores.

Além das views principais, o projeto inclui **camadas de suporte** para análises temporais, comportamento por categoria, eficiência operacional, frete versus peso e desempenho por vendedor. Todas as tabelas brutas passaram por **testes de qualidade estruturais e de consistência**, garantindo confiabilidade nos resultados.

O conjunto final oferece um ambiente analítico robusto, padronizado e pronto para uso em BI, permitindo análises estratégicas, criação de dashboards e acompanhamento contínuo da operação.

## Capítulo 2 - Metodologia CRISP-DM

A metodologia **CRISP-DM (Cross Industry Standard Process for Data Mining)** foi utilizada como a base metodológica do projeto, garantindo que todo o processo analítico fosse conduzido de forma estruturada, orientada ao negócio e tecnicamente consistente. Cada etapa da metodologia foi aplicada de forma prática no projeto, desde o entendimento das necessidades dos stakeholders até a implantação dos dashboards.

A seguir, detalham-se as fases utilizadas e sua aplicação direta no contexto da análise do marketplace Olist.

### 2.1 Fases Aplicadas no Projeto

A metodologia CRISP-DM foi aplicada conforme o quadro abaixo, mantendo alinhamento entre objetivos estratégicos, dados disponíveis e entregáveis analíticos.

Fase	Descrição	Aplicações no Projeto
<b>Entendimento do Negócio</b>	Definição das metas estratégicas e métricas-chave.	Reuniões com stakeholders das áreas Comercial, Marketing e Operações; levantamento de necessidades; definição de KPIs e hipóteses.
<b>Entendimento dos Dados</b>	Mapeamento das fontes, compreensão da estrutura e limitações.	Análise dos 9 datasets da Olist; verificação de completude; identificação de inconsistências, duplicidades e outliers.
<b>Preparação dos Dados</b>	Tratamento, padronização e integração das bases.	Criação da tabela fato (fato_pedido); normalização de datas; correção crítica em order_reviews; padronização de categorias; criação de dimensões.
<b>Modelagem</b>	Construção das métricas, indicadores e views temáticas.	Cálculo de GMV, LTV, TME e RFM; criação de métricas derivadas; 15 consultas analíticas; segmentações comportamentais.
<b>Avaliação</b>	Verificação da qualidade dos resultados e da aderência ao negócio.	Revisão dos indicadores com stakeholders; análise de anomalias; validação de regras analíticas e consistência temporal.
<b>Implantação</b>	Publicação dos resultados para consumo.	Criação de dashboards no Looker Studio; documentação técnica; criação das views S00X e T00X; estruturação do ambiente de BI.

Essa abordagem garantiu um fluxo de desenvolvimento sólido, reproduzível e consistente para todo o ciclo analítico.

## Capítulo 3 - Entendimento do Negócio

Esta etapa consolida os objetivos estratégicos das áreas Comercial, Marketing e Operações, definindo as métricas, hipóteses e prioridades que orientam toda a modelagem e as análises desenvolvidas. As demandas foram levantadas por meio de interações com stakeholders de cada área, garantindo alinhamento com necessidades reais do negócio.

### 3.1 Levantamento de Métricas e KPIs Principais

A tabela abaixo apresenta os indicadores estratégicos adotados ao longo do projeto:

KPI / Métrica	Definição	Fórmula Conceitual
<b>GMV (Gross Merchandise Volume)</b>	Valor total das mercadorias vendidas.	$GMV = \sum (\text{Preço} + \text{Frete})$
<b>Ticket Médio (AOV)</b>	Valor médio por pedido.	$AOV = GMV / N^{\circ} \text{ de Pedidos}$
<b>Volume de Pedidos</b>	Total de pedidos realizados.	Contagem de Pedidos
<b>Taxa de Cancelamento</b>	Percentual de pedidos cancelados.	$\text{Cancelamento} = \text{Cancelados} / \text{Totais}$
<b>Taxa de Inadimplência</b>	Falhas ou estornos no pagamento.	$\text{Inadimplência} = \text{Inadimplentes} / \text{Totais}$
<b>Top 5 Vendedores por GMV</b>	Principais vendedores em receita.	$GMV \text{ do Vendedor} = \sum \text{Itens Vendidos}$
<b>Receita por Categoria</b>	Valor total por categoria.	$\sum \text{Itens por Categoria}$
<b>Variação do GMV (M/M)</b>	Crescimento ou queda mensal.	$(GMV_m - GMV_{m-1}) / GMV_{m-1}$

**Rodapé conceitual:**

$\sum$  indica soma;

$m$  período atual;

$m-1$  período anterior.

## 3.2 Requisitos do Setor Comercial

O Diretor Comercial destacou que, embora o volume de pedidos da Olist seja expressivo, ainda existem oportunidades de crescimento ao:

- Identificar os vendedores mais lucrativos
- Reduzir cancelamentos e inadimplência
- Avaliar desempenho por categoria de produto
- Direcionamento de ações por região geográfica

**Pergunta de Negócio**

*Quais vendedores, categorias e regiões mais contribuem para o aumento do GMV?*

**KPIs aprovados — Comercial**

KPI	Definição	Fórmula Conceitual
<b>LTV (Lifetime Value)</b>	Receita estimada gerada por cliente ao longo do relacionamento.	$LTV = \text{Receita Média} \times \text{Frequência} \times \text{Retenção}$

KPI	Definição	Fórmula Conceitual
<b>Taxa de Clientes Recorrentes</b>	Percentual de clientes com mais de uma compra.	$\text{Taxa} = \text{Clientes +1 Compra} / \text{Total de Clientes}$
<b>Segmentação RFM</b>	Classificação por Recência, Frequência e Valor Monetário.	$\text{RFM} = \text{Score(R)} + \text{Score(F)} + \text{Score(M)}$
<b>Categoria da Primeira Compra</b>	Categoria da primeira compra do cliente.	Categoria = Primeira Categoria Registrada
<b>Taxa de Avaliações Negativas</b>	Percentual de reviews com nota 1 ou 2.	$\text{Taxa} = \text{Avaliações} \leq 2 / \text{Total de Avaliações}$
<b>Tempo Médio de Avaliação</b>	Tempo entre entrega e avaliação.	$\text{Tempo Médio} = \sum (\text{Data Avaliação} - \text{Data Entrega}) / \text{N}^\circ \text{ Avaliações}$

#### Definição dos Clusters RFM Utilizados

- **Campeões:** alta recência, alta frequência, alto valor.
- **Leais:** frequência alta e relacionamento consistente.
- **Novos:** realizaram primeira compra recentemente.
- **Inativos:** longo tempo sem comprar; baixa frequência.

### 3.3 Dimensões de Análise (Conceituais)

Dimensão	Descrição Conceitual
<b>Categoria de Produto</b>	Agrupamento dos produtos por tipo, utilizado para análises de GMV, vendas e comportamento do consumidor.
<b>Região Geográfica (Estado)</b>	Localização dos clientes para análises regionais de demanda e desempenho comercial.
<b>Período (Ano/Mês/Dia)</b>	Marco temporal utilizado para tendências, sazonalidade e análises evolutivas.

### 3.4 Requisitos do Setor de Marketing

O Gerente de Marketing destacou que, embora a base de clientes da Olist seja ampla, ainda há oportunidades de crescimento por meio de:

- Compreender o ciclo de valor vitalício do cliente (LTV)
- Aumentar a recorrência de compra e reduzir churn
- Identificar categorias de maior atração na primeira compra
- Melhorar a experiência e satisfação do cliente por meio das avaliações

#### Pergunta de Negócio

*Quais características distinguem os clientes com maior LTV e chance de recompra?*

#### Dimensões aplicadas

- Segmento RFM
- Categoria de Produto
- Estado do Cliente

### 3.5 Requisitos do Setor de Operações

O Head de Operações destacou que a área busca aumentar a eficiência da cadeia logística e reduzir atrasos de entrega. O foco está em:

- Acompanhar prazos e tempos médios de entrega
- Avaliar desempenho de vendedores e transportadoras
- Identificar gargalos geográficos e operacionais
- Otimizar custos logísticos e de frete com base nas dimensões dos produtos

#### Pergunta de Negócio

*Quais fatores influenciam o tempo médio de entrega e a taxa de atraso?*

#### KPIs aprovados — Operações

KPI / Métrica	Definição	Fórmula Conceitual
<b>Tempo Médio de Entrega (TME)</b>	Tempo entre aprovação e entrega ao cliente.	$TME = \frac{\sum (\text{Entrega} - \text{Aprovação})}{N^{\circ} \text{ Pedidos}}$
<b>Entrega no Prazo</b>	Percentual de entregas dentro ou antes da estimativa.	$\text{Taxa} = \frac{\text{Pedidos no Prazo}}{\text{Total}}$

KPI / Métrica	Definição	Fórmula Conceitual
<b>SLA do Vendedor</b>	Tempo entre aprovação e entrega à transportadora.	$SLA = \sum (\text{Entrega Transportadora} - \text{Aprovação}) / \text{N}^\circ \text{ Pedidos}$
<b>Taxa de Pedidos com Problemas</b>	Pedidos cancelados ou indisponíveis.	$Taxa = \text{Pedidos com Problemas} / \text{Total}$

#### Conceitos Operacionais

- **Atraso** ocorre quando a entrega supera a data estimada.
- **Entrega no Prazo** é igual ou anterior à data estimada.
- **Data Estimada de Entrega** informada ao cliente no momento da compra.
- **Tempo de Preparação (SLA)** é o intervalo entre aprovação e entrega à transportadora.
- **Taxa de Atraso por Cidade** compara atrasos dentro da própria cidade.

## Capítulo 4 - Entendimento dos Dados

Esta etapa documenta as fontes de dados utilizadas, seus volumes, estruturas e principais particularidades. O propósito é garantir total visibilidade do ambiente analítico e estabelecer as bases necessárias para a preparação, modelagem e validação dos dados que compõem o projeto.

Todas as decisões técnicas mencionadas neste capítulo têm referência direta no processo de construção da tabela fato (**fato\_pedido**) e das views de análise, cujas SQLs completas estão descritas no **Anexo Técnico**.

### 4.1 Inventário das Fontes de Dados

O projeto utiliza o **Brazilian E-Commerce Public Dataset by Olist**, composto por **nove** tabelas relacionais. Abaixo segue o inventário consolidado, padronizado e revisado:

Dataset	Registros	Descrição	Principais Atributos	Chaves
<b>olist_customers_dataset.csv</b>	99.441	Cadastro dos clientes.	Cidade, estado, IDs.	customer_id, customer_unique_id
<b>olist_geolocation_dataset.csv</b>	1.000.163	CEP, latitude e longitude.	CEP, latitude, longitude.	geolocation_zip_code_prefix
<b>olist_order_items_dataset.csv</b>	112.650	Itens dos pedidos.	product_id, price, freight_value.	order_id, product_id, seller_id



Dataset	Registros	Descrição	Principais Atributos	Chaves
<b>olist_order_payments_dataset.csv</b>	103.886	Pagamentos dos pedidos.	payment_type, payment_value.	order_id, payment_sequential
<b>olist_order_reviews_dataset.csv</b>	99.224	Avaliações dos pedidos.	review_score, timestamps.	review_id, order_id
<b>olist_orders_dataset.csv</b>	99.441	Metadados e status dos pedidos.	purchase_timestamp, delivered_timestamp.	order_id, customer_id
<b>olist_products_dataset.csv</b>	32.951	Catálogo de produtos.	categoria, peso, dimensões.	product_id
<b>olist_sellers_dataset.csv</b>	3.095	Cadastro dos vendedores.	seller_city, seller_state.	seller_id
<b>product_category_name_translation.csv</b>	71	Traduções das categorias.	nome original e traduzido.	product_category_name

Essa estrutura serviu como base para a criação da tabela fato consolidada e para as views analíticas documentadas na Sessão 5.

## 4.2 Avaliação da Qualidade dos Dados

Durante o entendimento dos dados, diversas validações foram realizadas a fim de identificar pontos críticos que exigiriam tratamento na fase de preparação (SQL consolidado em **C001 — Construção da Tabela Fato**, no Anexo Técnico).

A seguir estão os principais achados organizados para padronização documental:

### 1. Completude

- Valores nulos em datas relevantes, principalmente:
  - **order\_delivered\_customer\_date**
  - **order\_delivered\_carrier\_date**
  - **order\_approved\_at**
  - **review\_creation\_date** (campo string\_field\_5)
- Campos de categoria e avaliações com NULL ou strings vazias.
- Necessidade de distinguir entre ausência real de informação e falta de registro.

#### Referência Técnica:

→ Anexo Técnico **C001**.

## 2. Consistência Temporal

Foram encontradas:

- datas de entrega anteriores à data de compra;
- datas de envio posteriores à entrega;
- estimativas de entrega inconsistentes.

Esses padrões foram documentados em testes específicos no Anexo Técnico — Sessão **T00X** (principalmente **T004**, **T009** e **T010**).

**Referência Técnica:**

→ Anexos **T004 — Validação de Datas**

→ Anexo **C001 — Tabela Fato**

---

## 3. Integridade Relacional

Verificações realizadas:

- relacionamento 1:N entre **orders** → **order\_items**;
- relacionamento N:1 entre **order\_items** → **sellers**;
- relacionamento 1:1 entre **orders** → **order\_payments**, considerando sequencial.

Foram executados testes formais na Sessão **T002**, **T003**, **T005** e **T011** no Anexo Técnico.

**Referência Técnica:**

→ Anexos **T002**, **T003**, **T005**, **T011**

---

## 4. Duplicidade

- **customer\_unique\_id** é estável (identifica o cliente real).
- **customer\_id** pode mudar (recriação de cadastro).

Essa diferença foi respeitada na modelagem da tabela fato.

---

## 5. Pontos de Atenção Identificados

- Traduções inconsistentes de categorias (tratadas no Anexo **C001**, bloco de padronização).
- CEPs duplicados na tabela de geolocalização.
- Categorias com nomes divergentes entre tabelas.

**Referência Técnica:**

→ Anexo **C001**

---

## 4.3 Análise Exploratória

*(Todas as queries completas estão listadas no Anexo Técnico — Sessão **Testes de Qualidade** .)*

A análise exploratória revelou uma série de padrões críticos que influenciaram as decisões de modelagem, priorização de métricas e construção de views.

## Resultados principais:

---

### 1. Clientes Únicos vs. Recorrentes

- **96,95%** dos clientes compram apenas uma vez.
- **3,05%** realizam recompra.
- Confirma baixa fidelização.

---

### 2. Segmentação RFM

- Clientes de alto valor têm baixa frequência.
- Segmento **Campeões** representa pequena parcela, mas alto GMV.

---

### 3. Desempenho Regional

- Sudeste domina GMV.
- Norte e Nordeste apresentam mais atrasos.

---

### 4. Clientes de Alto Valor que Não Retornam

- Identificação de grupo crítico com **alto GMV e recência elevada**, porém sem recompra.

---

### 5. Sazonalidade da Primeira Compra

- Picos de novos clientes em meses específicos.
- Coortes confirmam impacto da época na taxa de retorno.

---

### 6. Impacto da Primeira Experiência

- Frete elevado e atraso na primeira entrega reduzem recompra.

---

### 7. Categorias com Maior Risco de Churn

- Alta correlação entre:
  - atraso,
  - frete elevado,
  - abandono após 1ª compra.

---

### 8. Categorias com Zero Retenção

- Identificação de categorias com **0% de clientes que retornam**.

---

### 9. Perfil dos Clientes Campeões

- Recompra  $\geq 3$  pedidos.
  - Experiência inicial impacta valor vitalício.
-

## 4.4 Processo de Limpeza e Tratamento dos Dados

Esta seção descreve as regras e decisões adotadas para garantir a integridade, consistência e padronização dos dados antes da construção das views e dashboards.

**Todas as implementações citadas abaixo constam na SQL completa da tabela fato, documentada no Anexo Técnico — Sessão C001.**

---

### 4.4.1 Estratégias de Limpeza Implementadas

#### *Tratamento de Valores Nulos*

Datas nulas foram tratadas para evitar:

- TME inválido,
  - atrasos negativos,
  - SLA inconsistente.
- 

#### *Correção Crítica na Tabela order\_reviews*

O dataset tinha colunas genéricas:

- string\_field\_1...5  
Com ausência total de documentação.

Conclusões após investigação:

- **string\_field\_2** → **order\_id**
- **string\_field\_5** → **data da avaliação** (string)

JOIN ajustado conforme Anexo **C001**.

---

#### *Padronização de Tipos de Dados*

Aplicado:

- parse para DATE/TIMESTAMP,
- normalização de formatos,
- conversões seguras com SAFE\_CAST().

→ Referência: Anexo **C001 — Conversões de Datas**.

---

### 4.4.2 Validações de Qualidade Aplicadas

Com base nos testes realizados (Sessão **Testes de Qualidade do Anexo Técnico**), foram verificadas:

1. **Relacionamentos entre tabelas**  
→ Testes **T002, T003, T005**
  2. **Datas inconsistentes**  
→ Testes **T004, T009, T010**
  3. **Valores financeiros fora do padrão**  
→ Tabela **order\_items** e **order\_payments** (Testes **T003, T011**)
-

#### 4.4.3 Decisões de Negócio na Limpeza

##### ***Inclusão de Todos os Status***

Não foram aplicados filtros na tabela fato.

Foram mantidos pedidos:

- cancelados,
- indisponíveis,
- entregues,
- pagos,
- não pagos.

→ Justificado no Anexo **C001 — Construção da Fato (sem WHERE de status)**.

---

##### ***Criação de Indicadores Binários***

Indicadores derivados implementados:

- atraso
- problema (cancelados/indisponíveis)
- entrega no prazo

→ Ver Anexo **C001 — Indicadores Derivados**.

---

#### 4.4.4 Lições da Fase de Limpeza

##### ***Desafios Técnicos***

- Reconstrução da tabela **order\_reviews**
- Datas faltantes e incorretas
- Duplicidade de categorias
- Fragilidade das estimativas de entrega

##### ***Impactos Positivos***

- Métricas logísticas mais precisas (TME, atraso, SLA)
- GMV consistente e sem duplicidades
- Segmentação RFM confiável
- Diagnóstico operacional melhorado
- Base pronta para dashboards e automação

## Capítulo 5 – Modelagem dos Dados

A modelagem foi projetada para suportar análises comerciais, de marketing e operacionais de forma consistente, escalável e com alto desempenho. Adotou-se um modelo dimensional baseado em **star schema**, complementado por **views SQL** que padronizam regras de negócio e facilitam governança, manutenção e reprodutibilidade das análises.

---

## 5.1 Modelo Estrela (Star Schema)

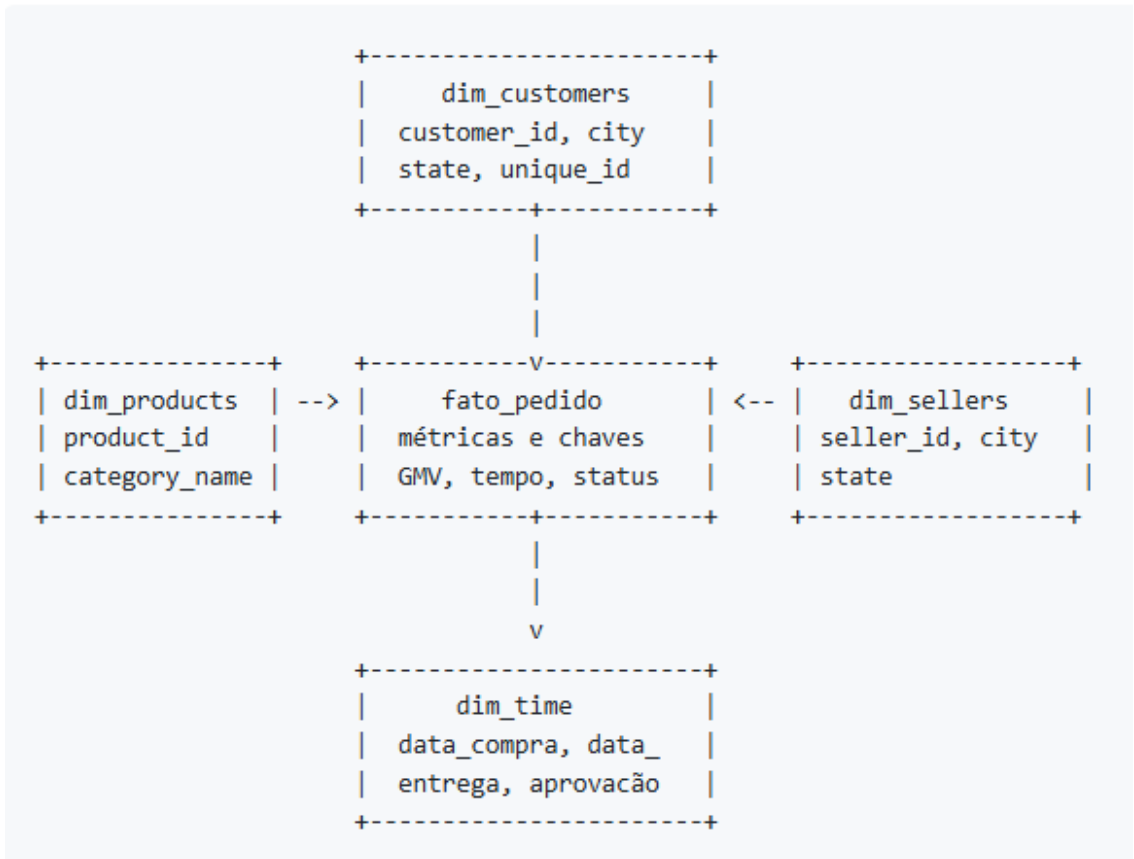
O modelo estrela foi escolhido por sua eficiência em consultas analíticas, simplicidade de manutenção e aderência às necessidades dos stakeholders. Ele centraliza os eventos na **tabela fato** e organiza informações descritivas nas **dimensões**.

Tabela — Estrutura do Modelo Estrela

Tabela	Tipo	Descrição	Fontes Originais	Chaves Principais
<b>FATO_PEDIDO</b>	Fato	Consolida métricas dos pedidos (GMV, frete, tempos, atrasos, status).	orders, order_items, payments, reviews	order_id, customer_id, product_id, seller_id
<b>DIM_CLIENTE</b>	Dimensão	Informações do cliente, RFM, LTV e dados geográficos.	customers, orders	customer_unique_id
<b>DIM_PRODUTO</b>	Dimensão	Categorias, dimensões físicas e características do produto.	products, category_translation	product_id
<b>DIM_VENDEDOR</b>	Dimensão	Dados do vendedor e indicadores operacionais (ex.: tempo de preparação).	sellers, order_items	seller_id
<b>DIM_TEMPO</b>	Dimensão	Datas derivadas: ano, mês, trimestre, sazonalidade.	orders (datas derivadas)	data_id

### Justificativa Técnica

- Reduz duplicidade e melhora performance das consultas analíticas.
- Permite análise por múltiplas perspectivas: região, cliente, produto, vendedor e período.
- Suporta cálculos avançados (RFM, LTV, TME, taxas operacionais).
- A implementação completa da **tabela fato** está documentada em: **Anexo SQL — C001 (Tabela Fato: fato\_pedido)**.



Esse diagrama ilustra a estrutura lógica em estrela, onde a tabela **fato\_pedido** centraliza as métricas de negócio (GMV, atrasos, tempo de entrega) e se conecta às dimensões que oferecem contexto analítico (clientes, produtos, vendedores e tempo).

## 5.2 Views SQL — Camada Semântica do Projeto

As views funcionam como uma **camada semântica** que:

- padroniza KPIs para todas as áreas,
- centraliza regras de negócio,
- evita duplicação de lógica SQL nos dashboards,
- melhora governança, manutenção e desempenho.

A seguir estão listadas as views principais, suas funções e as referências aos anexos onde suas queries completas estão documentadas.

### 5.2.1 View Comercial — vw\_dashboard\_comercial

**Objetivo:** Consolidar as métricas comerciais essenciais: GMV, ticket médio, volume de pedidos, cancelamentos e inadimplência.

**Principais Regras Implementadas:**

- Ticket médio = GMV / pedidos distintos.
- Inadimplência combinando status *unavailable* e cancelamentos por cartão.
- Segmentações por data, estado, categoria e tipo de pagamento.

**Consulta completa:****Anexo Técnico - Sessão : “Views Principais”**

---

**5.2.2 View Marketing — *vw\_dashboard\_marketing***

**Objetivo:** Consolidar informações de perfil do cliente, valor ao longo do tempo e segmentação RFM.

**Principais Implementações:**

- Cálculo de **recência** a partir da maior data disponível no dataset.
- Agregação por cliente sem duplicidade.
- Segmentação RFM baseada em NTILE(5).
- Identificação da categoria da primeira compra.

**Consulta completa:****Anexo Técnico - Sessão : “Views Principais”**

---

**5.2.3 View Operacional — *vw\_dashboard\_operacional***

**Objetivo:** Consolidar métricas logísticas, gargalos operacionais e indicadores de entrega.

**Principais Implementações:**

- Criação de faixas de peso para correlação com frete e prazo.
- Taxa de atraso e taxa de entrega no prazo.
- Indicadores de custo e tempo médio por região e por faixa de peso.

**Consulta completa:****Anexo Técnico - Sessão : “Views Principais”**

---

**5.2.4 Views de Suporte**

As views de suporte ampliam a capacidade analítica e servem como insumos para rankings, filtros e dashboards temáticos.



**vw\_base\_categoria**

- Consolida ranking de categorias por GMV, volume de pedidos e clientes únicos. Considera apenas pedidos entregues.
- Estrutura base utilizada em múltiplas views derivadas e dashboards.
  - **Consulta completa:** Anexo SQL — S001: Base de Categorias (vw\_base\_categorias).

**vw\_top10\_categorias**

- Seleciona as 10 categorias com maior GMV utilizando o ranking da base.
- Usada em filtros e blocos executivos de dashboards.
  - **Consulta completa:** Anexo SQL — S002: Top 10 Categorias (vw\_top10\_categorias).

**vw\_temporal\_unificada**

- Agrega indicadores comerciais, operacionais e financeiros por mês.
- Serve como camada única de tempo para todos os dashboards (Comercial, Marketing e Operacional).
- Inclui variação mensal de GMV e KPIs de tendência.
  - **Consulta completa:** Anexo SQL — S003: Base Temporal Unificada (vw\_temporal\_unificada).

**vw\_base\_vendedores**

- Consolida métricas de desempenho por vendedor (GMV, pedidos, tempo de preparação).
- Segue mesma estrutura da base de categorias, com rankings e indicador de relevância.
- Base para análises de performance comercial e operacional.
  - **Consulta completa:** Anexo SQL — S004: Base Consolidada de Vendedores (vw\_base\_vendedores).

**vw\_top5\_cidades\_atraso**

- Identifica as 5 cidades com maior taxa de atraso na entrega.
- Considera apenas cidades com mínimo de 10 pedidos para consistência estatística.
- Suporte a análises operacionais e identificação de gargalos logísticos.
  - **Consulta completa:** Anexo SQL — S005: Top 5 Cidades com Atraso (vw\_top5\_cidades\_atraso).

**vw\_frete\_vs\_peso**

- Analisa a relação entre peso do produto e valor do frete.
- Agrupa produtos por faixas de peso (0-1kg, 1-3kg, 3-5kg, etc.).
- Remove outliers para garantir consistência analítica.
  - **Consulta completa:** Anexo SQL — S006: Frete vs Peso (vw\_frete\_vs\_peso).

#### vw\_top5\_vendedores\_tme

- Identifica os 5 vendedores com maior tempo médio de entrega (TME).
- Considera apenas vendedores com mínimo de 5 pedidos.
- Suporte a análises de desempenho logístico por vendedor.
  - **Consulta completa:** Anexo SQL — S007: Top 5 Vendedores por TME (vw\_top5\_vendedores\_tme)

---

### 5.3 Considerações de Governança e Manutenção

- As views são organizadas por áreas temáticas (Comercial, Marketing, Operações e Suporte).
- Cada view possui documentação de regras, métricas e dependências.
- O uso de CTEs melhora modularização e leitura do SQL.
- A modelagem padronizada evita divergências entre dashboards e garante coerência entre todas as métricas derivadas.

## Capítulo 6 - Estratégia de Solução

A estratégia utilizada no projeto foi baseada na tradução das necessidades de negócio em hipóteses mensuráveis e, posteriormente, em análises estruturadas. A abordagem combinou fundamentos de Business Intelligence, a metodologia CRISP-DM e um desenho analítico orientado aos stakeholders.

Essa combinação garantiu que todas as análises fossem tecnicamente consistentes, alinhadas aos objetivos das áreas e capazes de gerar insights acionáveis.

---

### 6.1 Tradução dos Problemas de Negócio

Cada área apresentou necessidades específicas. Para garantir alinhamento entre dados, modelagem e decisões estratégicas, as necessidades foram traduzidas primeiro em **perguntas estratégicas** e, em seguida, em **perguntas mensuráveis**, diretamente respondidas pelas análises do projeto.

Tabela — Problema Estratégico x Questão Mensurável

Área	Pergunta Estratégica	Pergunta Mensurável (Análise)
Comercial	Como aumentar vendas e rentabilidade?	Quais vendedores, categorias e regiões mais contribuem para o GMV?

Área	Pergunta Estratégica	Pergunta Mensurável (Análise)
Marketing	Como ampliar a retenção e o valor do cliente (LTV)?	Quais características distinguem clientes com maior LTV e alta probabilidade de recompra?
Operações	Como melhorar eficiência logística e reduzir atrasos?	Quais fatores mais influenciam o TME e a taxa de atraso?

Essa tradução permitiu orientar corretamente as hipóteses, filtros, métricas e prioridades que guiaram as etapas seguintes.

## 6.2 Framework Analítico

O framework a seguir estrutura a lógica das análises realizadas, evidenciando:

- **a métrica principal (fato)** avaliada em cada área do negócio;
- **as dimensões** responsáveis por explicar as variações da métrica;
- **a hipótese estratégica** investigada.

Tabela — Framework Analítico por Área

Área	Fato (Métrica Principal)	Dimensões Utilizadas	Hipótese Avaliada
Comercial	GMV	Região, Categoria, Vendedor	GMV varia significativamente por região, categoria e vendedor.
Marketing	LTV	Segmento RFM, Categoria da 1ª Compra	LTV é maior em segmentos RFM superiores e categorias com maior aderência comportamental.
Operações	Tempo Médio de Entrega	Região, Peso do Produto, Vendedor	Regiões específicas e produtos mais pesados apresentam maior TME e maior probabilidade de atraso.

Esse framework norteou a construção das views semânticas e dos dashboards temáticos para cada área.

## 6.3 Critérios de Priorização das Análises

Com múltiplos objetivos estratégicos apresentados pelos stakeholders, foi necessário estabelecer critérios formais para priorização das análises. As análises

selecionadas foram aquelas que atenderam simultaneamente aos seguintes fatores:

- **Disponibilidade de Dados**

Análises que poderiam ser calculadas com integridade, completude e consistência a partir dos datasets brutos.

- **Insight Acionável**

Priorizaram-se análises capazes de orientar decisões práticas dos times Comercial, Marketing e Operações.

- **Viabilidade Técnica**

A implementação deveria ser compatível com a arquitetura do BigQuery, garantindo boa performance e manutenção simples.

- **Impacto no Negócio**

Preferência para métricas diretamente associadas a:

- GMV,
- LTV e retenção,
- eficiência operacional e TME,
- taxas de atraso ou cancelamentos.

## Capítulo 7 - Dashboards Estratégicos

Os dashboards foram desenvolvidos para traduzir os resultados analíticos em insights visuais acessíveis, orientando as decisões das três áreas principais:

**Comercial, Marketing e Operações.**

Cada dashboard foi estruturado com KPIs essenciais, visões táticas e análises diagnósticas, garantindo alinhamento direto com os objetivos definidos na fase de entendimento do negócio.

---

### 7.1 Visão Geral dos Dashboards

A tabela abaixo resume os dashboards entregues, seus objetivos e os principais indicadores monitorados.

Tabela — Visão Geral dos Dashboards Estratégicos

Dashboard	Objetivo de Negócio	KPIs Principais	Análises em Destaque
Comercial	Acompanhar volume de vendas, GMV e performance comercial	GMV, Ticket Médio, Volume de Pedidos	Top Vendedores, Top Categorias, Evolução Mensal do GMV, Taxa de Cancelamento
Marketing	Compreender comportamento dos clientes e identificar oportunidades de retenção	LTV, Taxa de Clientes Recorrentes, Segmentação RFM	Categoria da Primeira Compra, Perfil dos Clientes Campeões, Distribuição RFM
Operações	Avaliar eficiência logística e identificar gargalos	TME, SLA do Vendedor, Entrega no Prazo, Taxa de Atraso	Top Cidades com Atraso, Correlação Frete × Peso, Faixas de Peso, Comparação entre Estados

## 7.2 Estrutura Analítica dos Dashboards

Cada dashboard foi construído seguindo uma estrutura em três camadas:

### 1. Métricas Principais (Camada Executiva)

Indicadores consolidados que permitem uma visão global da operação ou do comportamento do cliente.

### 2. Análises Táticas (Camada Intermediária)

Comparações por categoria, região, vendedor, segmento ou período, permitindo navegação horizontal entre dimensões.

### 3. Análises Diagnósticas (Camada de Investigação)

Explorações orientadas a identificar causas, como atraso, churn, categorias críticas, vendedores problemáticos e anomalias.

Essa estrutura oferece clareza para stakeholders executivos enquanto mantém profundidade suficiente para analistas de dados.

## 7.3 Observações Técnicas Importantes

### Confiabilidade das Métricas

- Todas as métricas utilizadas nos dashboards foram centralizadas nas **Views SQL** construídas no BigQuery.
- As transformações lógicas e regras de negócio são aplicadas previamente nas views, evitando duplicações e inconsistências.
- As métricas exibidas nos dashboards são provenientes exclusivamente da camada semântica validada.

### Referência das Queries Utilizadas

Todas as consultas SQL utilizadas na construção dos dashboards estão documentadas integralmente na seção:

#### “Anexos – Código SQL Completo

incluindo as views V001, V002, V003 e as views de suporte.

## 7.4 Detalhamento Individual dos Dashboards

### 7.4.1 Dashboard Comercial



#### Objetivo:

Medir desempenho comercial e identificar oportunidades de crescimento.

**Principais KPIs:**

- GMV
- Volume de Pedidos
- Ticket Médio
- Taxa de Cancelamento
- Taxa de Inadimplência

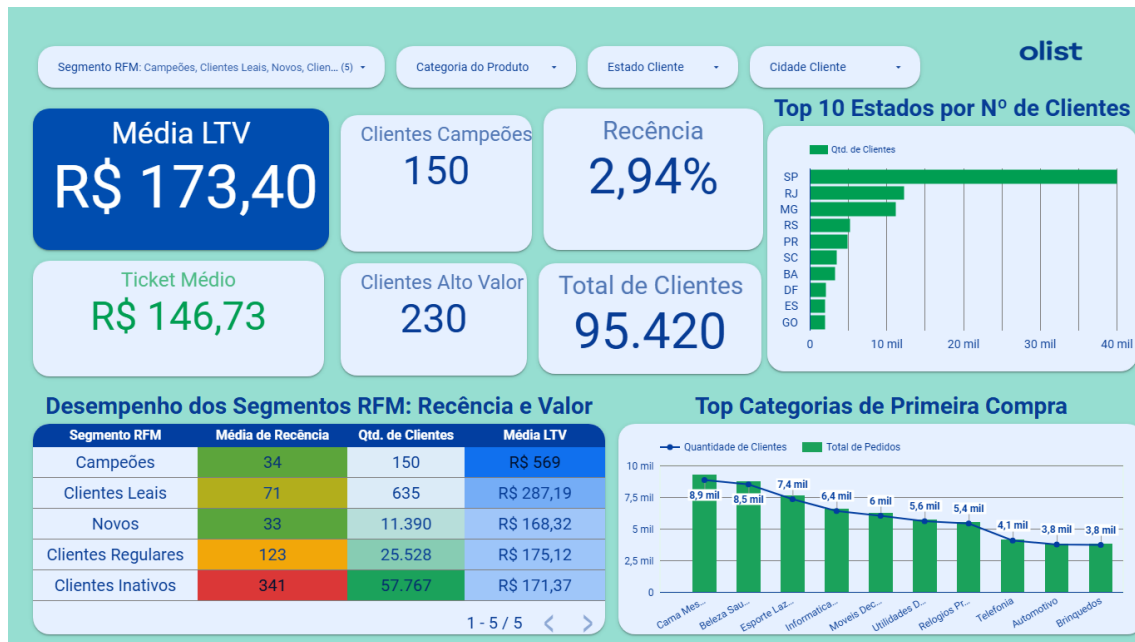
**Destaques Analíticos:**

- Ranking de vendedores por GMV
- Categorias com maior crescimento
- Tendência mensal de GMV
- Estados com maior concentração de clientes

**Filtros Utilizados:**

Tipo de Filtro	Fonte de Dados	Configuração
<b>Controle de Período</b>	vw_temporal_unificada	- Campo base: mês - Filtro de período padrão - Aplicável a todas as visualizações
<b>Lista Suspensa (Categoria)</b>	vw_dashboard_comercial	- Campo: Categoria - Ordenação: Decrescente por GMV Total - Limite: Top 20 categorias - Seleção múltipla habilitada

## 7.4.2 Dashboard Marketing



### Objetivo:

Avaliar comportamento dos clientes e direcionar estratégias de retenção.

### Principais KPIs:

- LTV
- Taxa de Recorrência
- Segmentação RFM
- Variedade de Categorias Compradas
- Categoria da Primeira Compra

### Destaques Analíticos:

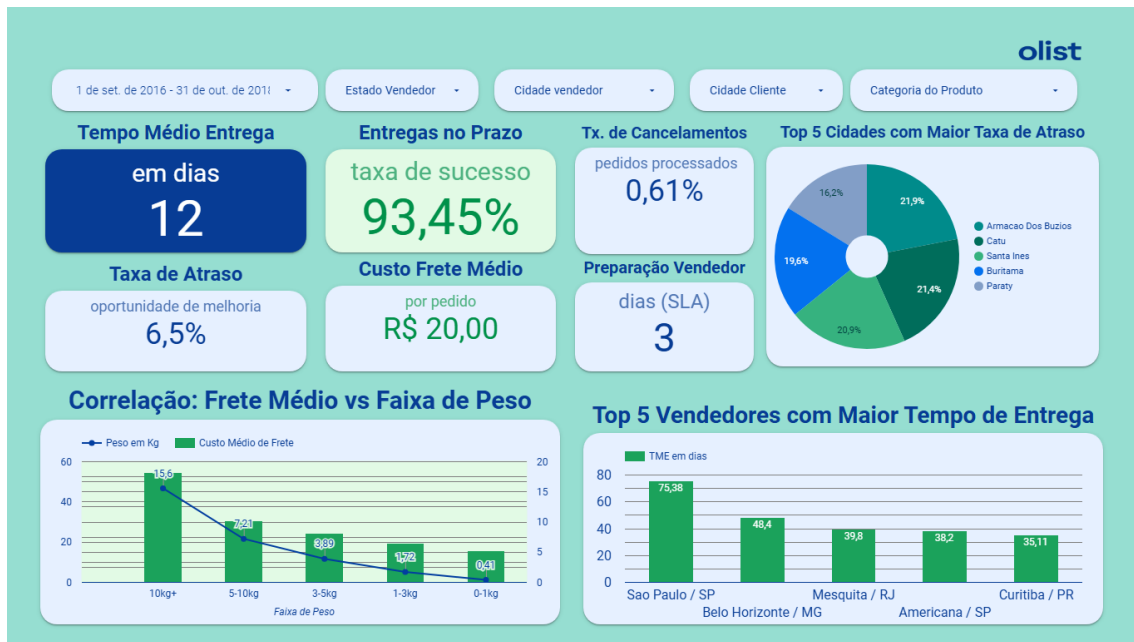
- Distribuição dos clusters RFM (Campeões, Leais, Novos, Inativos)
- Relação entre categoria inicial e LTV
- Análise de churn por segmento
- Perfil dos clientes de alto valor que não retornam



**Filtros Utilizados:**

<b>Tipo de Filtro</b>	<b>Fonte de Dados</b>	<b>Configuração</b>
<b>Lista Suspensa (Segmento RFM)</b>	vw_dashboard_marketing	<ul style="list-style-type: none"> <li>- Campo: Segmento RFM</li> <li>- Ordenação: Decrescente por LTV Total</li> <li>- Limite: Top 5000 registros</li> <li>- Métrica exibida: Média de Recência em dias</li> </ul>
<b>Lista Suspensa (Estado Cliente)</b>	vw_dashboard_marketing	<ul style="list-style-type: none"> <li>- Campo: Estado Cliente</li> <li>- Ordenação: Alfabética Crescente</li> <li>- Limite: Top 5000 registros</li> <li>- Métrica exibida: LTV Total</li> </ul>
<b>Controle de Período (Última Compra)</b>	vw_dashboard_marketing	<ul style="list-style-type: none"> <li>- Campo base: ultima_compra</li> <li>- Filtro de período padrão</li> <li>- Aplicável a análises de recência</li> </ul>
<b>Controle de Período (Primeira Compra)</b>	vw_dashboard_marketing	<ul style="list-style-type: none"> <li>- Campo base: primeira_compra</li> <li>- Filtro de período padrão</li> <li>- Aplicável a análises de coorte</li> </ul>

### 7.4.3 Dashboard Operacional



#### Objetivo:

Monitorar eficiência logística e reduzir atrasos.

#### Principais KPIs:

- TME
- SLA do Vendedor
- Taxa de Atraso
- Custo Médio do Frete
- Faixas de Peso

#### Destaques Analíticos:

- Top 5 cidades com maior atraso
- Relação peso × frete
- Desempenho por estado
- Comparação entre vendedores

**Filtros Utilizados:**

Tipo de Filtro	Fonte de Dados	Configuração
<b>Controle de Período</b>	vw_temporal_unificada	<ul style="list-style-type: none"> <li>- Campo base: data_compra</li> <li>- Filtro de período padrão</li> <li>- Aplicável a todas as visualizações</li> </ul>
<b>Lista Suspensa (Estado Vendedor)</b>	vw_dashboard_operacional	<ul style="list-style-type: none"> <li>- Campo: Estado Vendedor</li> <li>- Ordenação: Alfabética Crescente</li> <li>- Métrica exibida: Valor Frete</li> </ul>
<b>Lista Suspensa (Cidade Vendedor)</b>	vw_dashboard_operacional	<ul style="list-style-type: none"> <li>- Campo: Cidade Vendedor</li> <li>- Ordenação: Decrescente por Indicador Atraso</li> <li>- Métrica exibida: Indicador Atraso</li> </ul>
<b>Lista Suspensa (Cidade Cliente)</b>	vw_dashboard_operacional	<ul style="list-style-type: none"> <li>- Campo: Cidade Cliente</li> <li>- Ordenação: Alfabética Crescente</li> <li>- Métrica exibida: Valor Frete</li> </ul>
<b>Lista Suspensa (Categoria do Produto)</b>	vw_dashboard_operacional	<ul style="list-style-type: none"> <li>- Campo: Categoria do Produto</li> <li>- Ordenação: Alfabética Crescente</li> <li>- Métrica exibida: TME</li> </ul>

## Capítulo 8 - Stack Tecnológico

A construção do projeto utilizou um conjunto de ferramentas tecnológicas orientadas a processamento em escala, visualização eficiente e rastreabilidade das transformações. A combinação dos componentes abaixo garantiu **performance, governança e consistência** entre todas as camadas da solução analítica.

## 8.1 Componentes Utilizados

Tabela — Tecnologias e Funções no Projeto

Camada	Tecnologia	Função no Projeto
Banco de Dados e Processamento	<b>Google BigQuery (SQL)</b>	Armazenamento, limpeza, transformação dos dados, construção do modelo estrela e criação das views técnicas.
Visualização de Dados	<b>Looker Studio</b>	Construção dos dashboards Comercial, Marketing e Operacional; consumo direto das views tratadas.
Apoio Analítico	<b>Google Sheets</b>	Inspeção amostral das tabelas, conferência de volumes, validação de resultados e registros auxiliares.
Gestão e Metodologia	<b>CRISP-DM</b>	Estruturação do processo analítico, do entendimento do negócio à implantação.
Versionamento e Repositório	<b>GitHub e Google Drive</b>	Armazenamento dos scripts SQL, controle de versões e centralização da documentação.

## 8.2 Racional Técnico da Arquitetura

- **BigQuery** foi adotado por permitir processar grandes volumes de dados com alta performance, arquitetura serverless e custo proporcional ao uso.
- A modelagem em **Star Schema (Modelo Estrela)** garantiu:
  - padronização das métricas,
  - alto desempenho nas consultas,
  - redução de duplicidades,
  - facilidade de auditoria e governança.
- As **Views SQL** atuaram como **camada semântica oficial**, fornecendo ao Looker Studio métricas já tratadas e validadas, evitando lógicas replicadas no front-end.
- O **Looker Studio** possibilitou entrega rápida de dashboards, integração nativa ao BigQuery e atualização automática das análises.

Essa arquitetura assegura que todas as camadas: ingestão, transformação, modelagem e visualização, sejam consistentes entre si.

---

## 8.3 Boas Práticas Aplicadas

- **Centralização dos cálculos** nas views para garantir uniformidade nos dashboards.
- **Padrão de nomeação** aplicado a colunas, métricas, chaves e views técnicas.
- Uso de:
  - **CTEs,**
  - **window functions,**
  - **agregações,**
  - **tratamento explícito de valores nulos,**
  - **regras de negócio documentadas.**
- Organização da documentação em três camadas:
  - descrição funcional,
  - detalhes técnicos,
  - anexos com **todas as queries completas** (V00X, S00X, T00X e C001).

---

## Capítulo 9 - Lições Aprendidas

O desenvolvimento completo do projeto gerou aprendizados importantes nas dimensões **de negócio, técnica e processual**. Esses insights fortalecem entregas futuras de Business Intelligence, aumentam a maturidade analítica e aprimoram a qualidade das decisões.

---

### 9.1 Lições de Negócio

- **Compreensão profunda do contexto é essencial.** Antes de iniciar qualquer implementação técnica, entender objetivos comerciais, operacionais e de marketing permitiu que todas as análises fossem direcionadas a resultados de valor.
- **Requisitos bem definidos evitam retrabalho.** As interações com stakeholders reduziram ambiguidade e alinharam expectativas quanto a métricas, escopo e priorização analítica.

- **A exploração detalhada revela problemas não óbvios.** Métricas como categorias sem retenção, clusters críticos e regiões com alta taxa de atraso foram descobertas apenas após investigação exploratória aprofundada.
  - **Conexão entre perguntas estratégicas e KPIs.** Traduzir desafios do negócio em indicadores mensuráveis tornou possível a criação de dashboards claros, práticos e acionáveis.
- 

## 9.2 Lições Técnicas

- **O Modelo Estrela demonstrou ser uma base sólida e confiável.** Ele reduziu duplicidades, melhorou o desempenho das consultas e garantiu padronização nos cálculos utilizados em todas as áreas.
  - **Uso de SQL avançado como diferencial analítico.** CTEs, window functions, joins estruturados e views reutilizáveis permitiram organizar regras de negócio com clareza e rastreabilidade.
  - **Documentação técnica estruturada agrega valor.** O registro detalhado de fontes, relacionamentos, regras de cálculo e decisões técnicas facilitou revisões, auditoria e governança dos dados.
  - **Separação clara de camadas (fonte → preparo → views → dashboards).** Essa abordagem reduziu inconsistências, simplificou manutenção e proporcionou maior confiabilidade ao ecossistema analítico.
- 

## 9.3 Lições de Processo

- **A metodologia CRISP-DM mostrou adequação total ao ciclo do projeto.** As fases estruturaram o fluxo, organizaram entregas e facilitaram comunicação com as áreas de negócio.
  - **Comunicação contínua com stakeholders é determinante.** Checkpoints frequentes evitaram desalinhamentos e permitiram ajustes antes da etapa de implantação.
  - **Abordagem investigativa orientada a hipóteses.** Cada análise partiu de um questionamento de negócio, seguido de teste, validação e implicação estratégica.
  - **Iteração e refinamento são partes essenciais da entrega.** Os dashboards finais foram consolidados após diversas revisões baseadas no feedback das áreas Comercial, Marketing e Operações.
-

## Capítulo 10 - Conclusão e Próximos Passos

---

### 10.1 Conclusão

O projeto consolidou uma visão abrangente e integrada do desempenho comercial, comportamental e operacional do ecossistema Olist. Ao estruturar uma arquitetura analítica completa, desde os dados brutos até as views temáticas consumidas pelos dashboards, foi possível criar um ambiente robusto, escalável e governado.

A aplicação rigorosa da metodologia **CRISP-DM** garantiu consistência em todas as etapas:

do entendimento das necessidades estratégicas ao desenvolvimento técnico e à disponibilização dos painéis executivos.

#### Principais resultados alcançados:

- **Jornada completa do cliente estruturada**, incluindo perfis de valor, clusters RFM e padrões de primeira compra.
- **Identificação de gargalos logísticos**, como regiões com maior TME e categorias relacionadas a aumento de atraso ou custo de frete.
- **Visão consolidada de GMV**, categorias mais relevantes, e vendedores com maior contribuição para o volume e receita.
- **Descoberta de grupos críticos**, como categorias com zero retenção e clientes de alto valor que não retornam.
- **Base técnica unificada e padronizada**, com views SQL documentadas, regras consistentes e modelo dimensional estruturado.

Esses resultados criam um **alicerce estratégico** para evolução analítica, ampliando a capacidade de tomada de decisão baseada em dados.

---

### 10.2 Próximos Passos

Com o ecossistema analítico estruturado e operacional, as próximas etapas recomendadas envolvem expansão, automação e aperfeiçoamento das análises.

#### 1. Ampliação Analítica

- Construção de modelos preditivos para:
  - probabilidade de atraso,
  - risco de churn,
  - propensão de recompra (uplift modeling).

- Estudos de elasticidade de preço e sensibilidade de frete.

## 2. Enriquecimento de Dados

- Integração com fontes externas (IBGE, dados geográficos detalhados, indicadores macroeconômicos).
- Inclusão de dados de SAC, NPS ou atendimento para análises de experiência e satisfação do cliente.

## 3. Governança e Performance

- Versionamento formal das views SQL e evolução por ciclo.
- Automação de carga incremental em BigQuery.
- Monitoramento contínuo da qualidade dos dados (Data Quality Checks).

## 4. Evolução dos Dashboards

- Criação de painéis operacionais em tempo real.
- Segmentações dinâmicas de clientes no Looker Studio.
- Análises por cohort e sazonalidade detalhada.

# Capítulo 11 – Anexos

## 11.1. Queries Oficiais do Projeto (SQL)

As consultas SQL abaixo correspondem às análises e views utilizadas no projeto. No corpo principal da documentação, cada área possui uma referência indicando: **“A query completa está disponível na Sessão 11 – Anexos”**.

### Testes de Qualidade (T001 a T024)

T001 — Validação da Tabela customers

(Anexo Técnico — T001)

#### **Referência recomendada no corpo principal:**

“Query completa disponível em: Anexo Técnico – T001: Validação estrutural e qualitativa da tabela customers.”

- **1. Objetivo da Investigação**
- Avaliar integridade básica dos registros de clientes.
- Identificar estados e cidades disponíveis.
- Verificar estrutura da tabela para uso em análises geográficas e comportamentais.

- **2. Descrição Executiva**
- A primeira query exibe uma amostra estrutural da tabela.



- A segunda calcula métricas essenciais: total de registros, estados, cidades e clientes únicos.

---

- **3. Estrutura dos Indicadores**

<b>Campo</b>	<b>Descrição</b>
total_clientes	Total de registros na tabela
clientes_unicos	Contagem de customer_unique_id
Estados	Total de estados distintos
Cidades	Total de cidades distintas
estado_min / estado_max	Range alfabético dos estados presentes

---

- **4. Observações Técnicas**

- Base utilizada: customers
  - customer\_unique\_id é o identificador real do cliente.
  - customer\_id pode sofrer variação por atualizações cadastrais.
- 

- **Código SQL Completo**

```
-- =====
-- T001 — QUERY: Validação da Tabela CUSTOMERS
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1: Estrutura (amostra)
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.customers`
LIMIT 5;

-- BLOCO 2: Teste de Qualidade
SELECT
  COUNT(*) AS total_clientes,
  COUNT(DISTINCT customer_unique_id) AS clientes_unicos,
  COUNT(DISTINCT customer_state) AS estados,
  COUNT(DISTINCT customer_city) AS cidades,
  MIN(customer_state) AS estado_min,
  MAX(customer_state) AS estado_max
FROM `olist-sandbox-portfolio.olist_analysis.customers`;
```

---

T002 — Validação da Tabela order\_reviews

(Anexo Técnico — T002)

**Referência recomendada:**

“Query completa disponível em: Anexo Técnico – T002: Validação estrutural e qualitativa da tabela order\_reviews.”

---

- **1. Objetivo**

- Validar integridade da tabela de avaliações.
  - Identificar variedade de notas e pedidos avaliados.
  - Verificar estrutura atípica (string\_field\_1...5).
- 

- **2. Descrição Executiva**

- A tabela possui campos não documentados.
  - Foram mapeados manualmente:
    - string\_field\_2 = order\_id
    - string\_field\_3 = score
  - As queries verificam estrutura, distribuição de notas e integridade.
- 

- **3. Indicadores**

Campo	Descrição
total_avaliacoes	Registros totais
pedidos_avaliados	Quantidade de pedidos distintos com avaliação
notas_unicas	Quantidade de notas distintas
nota_minima	Menor nota
nota_maxima	Maior nota

---

- **4. Observações Técnicas**

- SAFE\_CAST necessário para converter os campos textuais.
  - Tabela crítica para análises de NPS e experiência do cliente.
- 

- **Código SQL Completo**

```
-- =====
-- T002 — QUERY: Validação da Tabela ORDER_REVIEWS
-- =====

-- BLOCO 1: Teste de Qualidade
SELECT
  COUNT(*) AS total_avaliacoes,
  COUNT(DISTINCT string_field_2) AS pedidos_avaliados,
  COUNT(DISTINCT string_field_3) AS notas_unicas,
  MIN(SAFE_CAST(string_field_3 AS INT64)) AS nota_minima,
  MAX(SAFE_CAST(string_field_3 AS INT64)) AS nota_maxima
FROM `olist-sandbox-portfolio.olist_analysis.order_reviews`
WHERE SAFE_CAST(string_field_3 AS INT64) IS NOT NULL;

-- BLOCO 2: Estrutura
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.order_reviews`
```

LIMIT 5;

-- BLOCO 3: Distribuição das notas

SELECT

string\_field\_3,

COUNT(\*) AS quantidade

FROM `olist-sandbox-portfolio.olist\_analysis.order\_reviews`

GROUP BY string\_field\_3

ORDER BY quantidade DESC;

T003 — Validação da Tabela order\_items

(Anexo Técnico — T003)

**Referência recomendada:**

“Query completa disponível em: Anexo Técnico – T003: Validação estrutural e qualitativa da tabela order\_items.”

- **1. Objetivo**

- Validar estrutura e integridade dos itens dos pedidos.
- Verificar variedade de produtos, vendedores e preços.
- Preparar base para construção da Fato Pedido.

- **2. Descrição Executiva**

- Query inicial mostra a estrutura.
- Query principal calcula métricas essenciais como preço médio, frete médio e quantidade de vendedores.

- **3. Indicadores**

Campo	Descrição
total_itens	Volume total de itens
pedidos_com_itens	Total de pedidos distintos
produtos_unicos	Contagem de produtos distintos
vendedores_unicos	Quantidade de sellers
preco_medio	Média dos preços dos itens
frete_medio	Média do frete
preco_minimo / preco_maximo	Faixa de preços

- **4. Observações Técnicas**

- order\_items é fundamental para gerar GMV.
- Relação 1:N entre order\_id e product\_id exige atenção em joins.

- **Código SQL Completo**

-- =====

-- T003 — QUERY: Validação da Tabela ORDER\_ITEMS

-- =====

-- BLOCO 1: Estrutura (amostra)

```
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.order_items`
LIMIT 5;
```

-- BLOCO 2: Teste de Qualidade

```
SELECT
COUNT(*) AS total_itens,
COUNT(DISTINCT order_id) AS pedidos_com_itens,
COUNT(DISTINCT product_id) AS produtos_unicos,
COUNT(DISTINCT seller_id) AS vendedores_unicos,
AVG(price) AS preco_medio,
AVG(freight_value) AS frete_medio,
MIN(price) AS preco_minimo,
MAX(price) AS preco_maximo
FROM `olist-sandbox-portfolio.olist_analysis.order_items`;
```

T004 — Validação da Tabela orders

(Anexo Técnico — T004)

**Referência recomendada:**

“Query completa disponível em: Anexo Técnico – T004: Validação estrutural e qualitativa da tabela orders.”

- **1. Objetivo da Investigação**

- Avaliar integridade e consistência da tabela mestre de pedidos.
- Verificar diversidade de status, datas relevantes e volume total.
- Garantir confiabilidade antes de construir a tabela fato.

- **2. Descrição Executiva**

- Primeira query: exibe a estrutura da tabela.
- Segunda query: valida métricas essenciais como datas, status e contagens distintas.

- **3. Estrutura dos Indicadores**

Campo	Descrição
total_pedidos	Total de registros na tabela
pedidos_unicos	Total de order_id distintos
clientes_unicos	Distintos customer_id
status_diferentes	Quantidade de status disponíveis

Campo	Descrição
data_primeiro_pedido	Menor data de compra
data_ultimo_pedido	Maior data registrada

---

- **4. Observações Técnicas**

- Colunas temporais são fundamentais para calcular TME, atraso e coortes.
  - Mantém status cancelados e unavailable para análises operacionais.
- 

- **Código SQL Completo**

```
-- =====
-- T004 — QUERY: Validação da Tabela ORDERS
-- =====
```

```
-- BLOCO 1: Estrutura (amostra)
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.orders`
LIMIT 5;

-- BLOCO 2: Teste de Qualidade
SELECT
COUNT(*) AS total_pedidos,
COUNT(DISTINCT order_id) AS pedidos_unicos,
COUNT(DISTINCT customer_id) AS clientes_unicos,
COUNT(DISTINCT order_status) AS status_diferentes,
MIN(order_purchase_timestamp) AS data_primeiro_pedido,
MAX(order_purchase_timestamp) AS data_ultimo_pedido
FROM `olist-sandbox-portfolio.olist_analysis.orders`;
```

---

T005 — Validação da Tabela order\_payments

(Anexo Técnico — T005)

**Referência recomendada:**

“Query completa disponível em: Anexo Técnico – T005: Validação estrutural e qualitativa da tabela order\_payments.”

---

- **1. Objetivo da Investigação**

- Avaliar integridade dos dados financeiros do pedido.
  - Validar variedade de métodos de pagamento.
  - Identificar faixas de valores e número de parcelas.
- 

- **2. Descrição Executiva**

- Estrutura da tabela é exibida no primeiro bloco.
  - Métricas principais são verificadas no segundo bloco.
-

- **3. Estrutura dos Indicadores**

<b>Campo</b>	<b>Descrição</b>
total_pagamentos	Registros totais na tabela
pedidos_com_pagamento	Pedidos que possuem pelo menos um pagamento
tipos_pagamento	Métodos distintos (boleto, cartão, etc.)
valor_medio_pagamento	Média dos valores
parcelas_medias	Média das parcelas em cartão
valor_minimo / valor_maximo	Faixa dos valores registrados

- **4. Observações Técnicas**

- Campos de valor não possuem inconsistências estruturais relevantes.
- Essencial para análises de inadimplência e ticket médio.

- **Código SQL Completo**

```
-- =====
-- T005 — QUERY: Validação da Tabela ORDER_PAYMENTS
-- =====
```

```
-- BLOCO 1: Estrutura
```

```
SELECT *
```

```
FROM `olist-sandbox-portfolio.olist_analysis.order_payments`
LIMIT 5;
```

```
-- BLOCO 2: Teste de Qualidade
```

```
SELECT
```

```
  COUNT(*) AS total_pagamentos,
  COUNT(DISTINCT order_id) AS pedidos_com_pagamento,
  COUNT(DISTINCT payment_type) AS tipos_pagamento,
  AVG(payment_value) AS valor_medio_pagamento,
  AVG(payment_installments) AS parcelas_medias,
  MIN(payment_value) AS valor_minimo,
  MAX(payment_value) AS valor_maximo
```

```
FROM `olist-sandbox-portfolio.olist_analysis.order_payments`;
```

T006 — Validação da Última Data Disponível e Recência

(Anexo Técnico — T006)

**Referência recomendada:**

“Query completa disponível em: Anexo Técnico – T006: Última data do dataset e análise de recência por segmento RFM.”

- **1. Objetivo da Investigação**

- Identificar a última data de compra presente no dataset.

- Avaliar a recência dos clientes, segmentada por clusters RFM.
- Verificar atualização da base e comportamento recente dos clientes.

---

## • 2. Descrição Executiva

- A primeira query retorna a maior data de compra e total de registros válidos.
  - A segunda analisa o intervalo de recência por segmentação RFM.
- 

## • 3. Estrutura dos Indicadores

Campo	Descrição
ultima_data_dataset	Maior data_compra registrada
total_registros	Total de registros não nulos
segmento_rfm	Cluster de recência/frequência/valor
min_recencia	Mínimo de dias desde última compra
max_recencia	Máximo de dias desde última compra
total_clientes	Quantidade no segmento

---

## • 4. Observações Técnicas

- Base usada: fato\_pedido e vw\_dashboard\_marketing.
  - Recência calculada considerando a última data global da base.
  - Essencial para análises de retenção e estratégias de marketing.
- 

## • Código SQL Completo

```
-- =====
-- T006 — QUERY: Última Data do Dataset e Recência de Clientes
-- =====

-- BLOCO 1: Última data disponível
SELECT
    MAX(data_compra) AS ultima_data_dataset,
    COUNT(*) AS total_registros
FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido`
WHERE data_compra IS NOT NULL;

-- BLOCO 2: Recência por segmento RFM
SELECT
    segmento_rfm,
    MIN(recencia_dias) AS min_recencia,
    MAX(recencia_dias) AS max_recencia,
    COUNT(*) AS total_clientes
FROM `olist-sandbox-portfolio.olist_analysis.vw_dashboard_marketing`
GROUP BY segmento_rfm
```

ORDER BY min\_recencia;

---

## T007 — Validação da Tabela sellers

(Anexo Técnico — T007)

### Referência recomendada:

“Query completa disponível em: Anexo Técnico – T007: Validação estrutural e qualitativa da tabela sellers.”

- 
- **1. Objetivo da Investigação**
    - Avaliar completude e integridade da base de vendedores.
    - Validar diversidade geográfica.
    - Garantir consistência antes de análises operacionais por vendedor.
- 
- **2. Descrição Executiva**
    - Primeiro bloco: amostra da estrutura.
    - Segundo bloco: validações de quantidade total, unicidade e distribuição geográfica.
- 

- **3. Estrutura dos Indicadores**

Campo	Descrição
total_vendedores	Total de registros
vendedores_unicos	COUNT DISTINCT seller_id
estados_vendedores	Quantidade de UF distintas
cidades_vendedores	Quantidade de cidades distintas

---

- **4. Observações Técnicas**
    - A tabela sellers é fundamental para análises de SLA, TME e rankings de performance.
    - Não houve inconsistências estruturais relevantes.
- 

- **Código SQL Completo**

```
-- =====
-- T007 — QUERY: Validação da Tabela SELLERS
-- =====

-- BLOCO 1: Estrutura
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.sellers`
LIMIT 5;

-- BLOCO 2: Teste de Qualidade
SELECT
```



```

COUNT(*) AS total_vendedores,
COUNT(DISTINCT seller_id) AS vendedores_unicos,
COUNT(DISTINCT seller_state) AS estados_vendedores,
COUNT(DISTINCT seller_city) AS cidades_vendedores
FROM `olist-sandbox-portfolio.olist_analysis.sellers`;

```

---

## T008 — Análise de Datas Inconsistentes (Orders)

(Anexo Técnico — T008)

### Referência recomendada:

“Query completa disponível em: Anexo Técnico – T008: Validação de datas inconsistentes na tabela orders.”

---

#### • 1. Objetivo da Investigação

- Identificar datas cronologicamente inválidas.
- Validar coerência temporal entre compra, entrega e aprovação.
- Auxiliar no tratamento realizado na tabela fato.

---

#### • 2. Descrição Executiva

- A query identifica casos onde a data de entrega é anterior à data da compra.
- Auxilia na limpeza e padronização de cálculos temporais.

---

#### • 3. Estrutura dos Indicadores

Campo	Descrição
total	Total de pedidos
datas_inconsistentes	Quantidade de entregas ocorridas antes da compra

---

#### • 4. Observações Técnicas

- Datas inválidas representam 0,01% no dataset original.
- Foram tratadas na construção da tabela fato (Anexo C001).

---

#### • Código SQL Completo

```

-- =====
-- T008 — QUERY: Análise de Datas Inconsistentes
-- =====

SELECT
'orders' AS tabela,
COUNT(*) AS total,
SUM(CASE
    WHEN order_delivered_customer_date < order_purchase_timestamp
    THEN 1 ELSE 0
END) AS datas_inconsistentes

```

```
FROM `olist-sandbox-portfolio.olist_analysis.orders`;
```

## T009 — Análise Avançada de Datas e Inconsistências

(Anexo Técnico — T009)

### Referência recomendada:

“Query completa disponível em: Anexo Técnico – T009: Análise detalhada de datas, entregas e TME negativo.”

#### • 1. Objetivo da Investigação

- Identificar problemas temporais mais complexos, incluindo:
  - Entregas antes da compra
  - Pedidos entregues sem data de entrega
  - TME negativo
  - Pedidos sem data de aprovação

#### • 2. Descrição Executiva

- Fornece mapeamento completo das inconsistências temporais do dataset.
- Essencial para validação final antes da modelagem operacional.

#### • 3. Estrutura dos Indicadores

Campo	Descrição
total_pedidos	Total de registros
entregas_antes_compra	Casos críticos
sem_data_aprovacao	Pedidos com aprovação ausente
entregues_sem_data_entrega	Status "delivered" mas sem data
tme_negativo	Cálculo temporal negativo (< 0 dias)

#### • 4. Observações Técnicas

- Problemas tratados na tabela fato com regras de proteção para NULLs.
- Indicadores deste teste validam integridade antes de cálculo de TME e SLA.

#### • Código SQL Completo

```
-- =====
-- T009 — QUERY: Análise Avançada de Datas e Inconsistências
-- =====
```

```
SELECT
  COUNT(*) AS total_pedidos,

  SUM(CASE
    WHEN order_delivered_customer_date IS NOT NULL
    AND order_purchase_timestamp IS NOT NULL
```

```

        AND order_delivered_customer_date < order_purchase_timestamp
        THEN 1 ELSE 0
    END) AS entregas_antes_compra,

    SUM(CASE WHEN order_approved_at IS NULL THEN 1 ELSE 0 END) AS
    sem_data_aprovacao,

    SUM(CASE
        WHEN order_status = 'delivered'
        AND order_delivered_customer_date IS NULL
        THEN 1 ELSE 0
    END) AS entregues_sem_data_entrega,

    SUM(CASE
        WHEN order_delivered_customer_date IS NOT NULL
        AND order_purchase_timestamp IS NOT NULL
        AND DATE_DIFF(DATE(order_delivered_customer_date),
        DATE(order_purchase_timestamp), DAY) < 0
        THEN 1 ELSE 0
    END) AS tme_negativo

FROM `olist-sandbox-portfolio.olist_analysis.orders`;

```

---

## T010 — Pagamentos e Itens: Análise de Valores Críticos

(Anexo Técnico — T010)

### Referência recomendada:

“Query completa disponível em: Anexo Técnico – T010: Análise de pagamentos inconsistentes, preços e fretes.”

- 
- **1. Objetivo da Investigação**
  - Verificar anomalias financeiras em pagamentos, preços e valores de frete.
  - Identificar valores negativos, zerados ou extremamente altos.
  - Avaliar estatísticas descritivas para detectar padrões ou erros estruturais.

---

- **2. Descrição Executiva**

Inclui dois blocos principais:

1. **Pagamentos inconsistentes:** identifica pagamentos negativos ou zerados.
2. **Valores atípicos em itens:** avalia preços e fretes com extremos e inconsistências.

---

- **3. Estrutura dos Indicadores**

Campo	Descrição
pagamentos_negativos	Quantidade de pagamentos < 0

Campo	Descrição
pagamentos_zero	Pagamentos com valor zero
precos_negativos	Preços negativos
frete_negativos	Frete negativos
precos_zero	Preços zerados
frete_zero	Frete zerados
precos_muito_altos	Preços > 10.000
frete_muito_altos	Frete > 1.000
preco_minimo / maximo / medio	Estatísticas básicas de preço
frete_minimo / maximo / medio	Estatísticas básicas de frete

---

#### • 4. Observações Técnicas

- Os valores extremos são raros, mas podem distorcer métricas como ticket médio e GMV.
  - Foram avaliados antes da construção da fato\_pedido (C001).
- 

#### • Código SQL Completo

```
-- =====
-- T010 — QUERY: Análise de Pagamentos e Valores Financeiros
-- =====

-- 1. Pagamentos inconsistentes
SELECT
  SUM(CASE WHEN payment_value < 0 THEN 1 ELSE 0 END) AS
pagamentos_negativos,
  SUM(CASE WHEN payment_value = 0 THEN 1 ELSE 0 END) AS pagamentos_zero
FROM `olist-sandbox-portfolio.olist_analysis.order_payments`;

-- 2. Análise de preços e fretes
SELECT
  COUNT(*) AS total_itens,

  SUM(CASE WHEN price < 0 THEN 1 ELSE 0 END) AS precos_negativos,
  SUM(CASE WHEN freight_value < 0 THEN 1 ELSE 0 END) AS fretes_negativos,

  SUM(CASE WHEN price = 0 THEN 1 ELSE 0 END) AS precos_zero,
  SUM(CASE WHEN freight_value = 0 THEN 1 ELSE 0 END) AS fretes_zero,

  SUM(CASE WHEN price > 10000 THEN 1 ELSE 0 END) AS precos_muito_altos,
  SUM(CASE WHEN freight_value > 1000 THEN 1 ELSE 0 END) AS
fretes_muito_altos,
```

```
ROUND(MIN(price), 2) AS preco_minimo,
ROUND(MAX(price), 2) AS preco_maximo,
ROUND(AVG(price), 2) AS preco_medio,
ROUND(MIN(freight_value), 2) AS frete_minimo,
ROUND(MAX(freight_value), 2) AS frete_maximo,
ROUND(AVG(freight_value), 2) AS frete_medio
```

```
FROM `olist-sandbox-portfolio.olist_analysis.order_items` ;
```

## T011 — Validação Estrutural da Tabela order\_payments

(Anexo Técnico — T011)

### Referência recomendada:

“Query completa disponível em: Anexo Técnico – T011: Validação estrutural e qualitativa da tabela order\_payments.”

- **1. Objetivo da Investigação**
- Validar integridade, diversidade e consistência dos dados financeiros.
- Confirmar plausibilidade de parcelas, valores e métodos de pagamento.
- Garantir base sólida para análises de faturamento.

- **2. Descrição Executiva**
- Primeiro bloco: estrutura da tabela.
- Segundo bloco: métricas essenciais para validar qualidade.

- **3. Estrutura dos Indicadores**

Campo	Descrição
total_pagamentos	Total de registros
pedidos_com_pagamento	Pedidos com pelo menos 1 pagamento
tipos_pagamento	Métodos distintos (boleto, cartão, etc.)
valor_medio_pagamento	Média dos valores
parcelas_medias	Média de parcelas
valor_minimo / máximo	Valores extremos registrados

- **4. Observações Técnicas**
- A tabela possui casos de múltiplos pagamentos por pedido.
- O método de pagamento influencia comportamentos de entrega e inadimplência.

- **Código SQL Completo**

```
-- =====
-- T011 — QUERY: Validação da Tabela ORDER_PAYMENTS
-- =====
```

```
-- BLOCO 1: Estrutura da tabela
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.order_payments`
LIMIT 5;
```

```
-- BLOCO 2: Teste de Qualidade
SELECT
  COUNT(*) AS total_pagamentos,
  COUNT(DISTINCT order_id) AS pedidos_com_pagamento,
  COUNT(DISTINCT payment_type) AS tipos_pagamento,
  AVG(payment_value) AS valor_medio_pagamento,
  AVG(payment_installments) AS parcelas_medias,
  MIN(payment_value) AS valor_minimo,
  MAX(payment_value) AS valor_maximo
FROM `olist-sandbox-portfolio.olist_analysis.order_payments`;
```

T012 — *Verificação de estrutura e metadados das tabelas.*

(Anexo Técnico — T012)

**Referência recomendada:**

“Query completa disponível em: Anexo Técnico – T012: Verificação de estrutura e metadados das tabelas.”

- **1. Objetivo da Investigação**
  - Documentar estrutura, colunas e tipos de dados.
  - Garantir clareza sobre o padrão das tabelas brutas.
  - Validar metadados essenciais antes da modelagem.
- **2. Descrição Executiva**
  - Primeiro bloco: metadados da tabela order\_reviews.
  - Segundo bloco: inventário completo das tabelas principais.

- **3. Estrutura dos Indicadores**

Campo	Descrição
column_name	Nome da coluna
data_type	Tipo de dado
is_nullable	Aceita nulos
table_name	Tabela de origem

- **4. Observações Técnicas**
  - Confirma integridade dos metadados para uso no modelo dimensional.

- Essencial para verificar colunas sem documentação formal (ex.: order\_reviews).

---

- **Código SQL Completo**

```
-- =====
-- T012 — QUERY: Documentação das Tabelas via INFORMATION_SCHEMA
-- =====
```

```
-- BLOCO 1: Estrutura da tabela order_reviews
```

```
SELECT column_name, data_type
FROM `olist-sandbox-portfolio.olist_analysis`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'order_reviews'
ORDER BY ordinal_position;
```

```
-- BLOCO 2: Estrutura geral das tabelas principais
```

```
SELECT table_name, column_name, data_type, is_nullable
FROM `olist-sandbox-portfolio.olist_analysis`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name IN (
    'customers',
    'order_items',
    'orders',
    'products',
    'sellers',
    'order_payments',
    'order_reviews',
    'product_category_name_translation'
)
ORDER BY table_name, ordinal_position;
```

---

T013 — Query de Teste: Validação da View Temporal Unificada

**Referência recomendada no corpo do documento:**

*“Query completa disponível em: Anexo – T013: Validação da view temporal unificada.”*

---

## 1. Objetivo da Investigação

- Validar a integridade estrutural e temporal da view vw\_temporal\_unificada;
  - Garantir consistência das métricas mensais consolidadas (GMV, pedidos, ticket médio, tempos);
  - Verificar ordenação, continuidade temporal e coerência das agregações.
- 

## 2. Descrição Executiva

- A primeira query avalia a coerência da série temporal (primeiro e último mês, total de períodos, GMV consolidado etc.);
- A segunda query verifica o desempenho das categorias principais, utilizando ranking gerado na view de suporte;
- A terceira query identifica a coerência do ranking de vendedores baseado em GMV.

---

### 3. Estrutura dos Indicadores

<b>Campo</b>	<b>Descrição</b>
total_meses_analisados	Quantidade total de meses presentes na série
primeiro_mes	Mês inicial da série temporal
ultimo_mes	Último mês disponível
gmv_total_periodo	GMV acumulado no período
total_pedidos_periodo	Total de pedidos no período
ticket_medio_medio	Média de ticket médio mensal
tempo_entrega_medio	Tempo médio de entrega agregado

---

### 4. Observações Técnicas

- Base utilizada: vw\_temporal\_unificada, vw\_base\_categorias, vw\_base\_vendedores;
  - As métricas já chegam pré-agregadas, garantindo consistência;
  - A série temporal segue ordenação explícita por campo mes.
- 

### T013 — QUERY COMPLETA

```
-- =====
-- T013 — QUERY: Validação da View Temporal Unificada
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Validação da série temporal
SELECT
  COUNT(*) AS total_meses_analisados,
  MIN(mes) AS primeiro_mes,
  MAX(mes) AS ultimo_mes,
  SUM(gmv_total) AS gmv_total_periodo,
  SUM(total_pedidos) AS total_pedidos_periodo,
  AVG(ticket_medio) AS ticket_medio_medio,
  AVG(tempo_medio_entrega_dias) AS tempo_entrega_medio
FROM `olist-sandbox-portfolio.olist_analysis.vw_temporal_unificada`;

-- BLOCO 2 — Categorias mais relevantes
SELECT
  categoria,
```



```

    gmv_total,
    total_pedidos,
    ranking_gmv
FROM `olist-sandbox-portfolio.olist_analysis.vw_base_categorias`
WHERE ranking_gmv <= 10
ORDER BY ranking_gmv;

```

-- BLOCO 3 — Top vendedores

```

SELECT
    vendedor_id,
    cidade_vendedor,
    estado_vendedor,
    gmv_total,
    total_pedidos,
    ranking_gmv
FROM `olist-sandbox-portfolio.olist_analysis.vw_base_vendedores`
WHERE ranking_gmv <= 5
ORDER BY ranking_gmv;

```

---

T014 — Query de Teste: Estrutura de Documentação de Colunas

**(Anexo – T014)**

**Referência recomendada:**

“Query completa disponível em: Anexo – T014: Documentação da estrutura das tabelas.”

---

1. Objetivo da Investigação

- Listar automaticamente a estrutura (nome da coluna, tipo e nulabilidade) das principais tabelas;
- Verificar consistência entre esquemas e identificar possíveis divergências estruturais;
- Servir como documentação técnica para governança de dados.

---

2. Descrição Executiva

- A primeira query inspeciona a estrutura detalhada de uma tabela individual;
- A segunda query compara múltiplas tabelas, listando campos, tipos e nulabilidade em ordem padronizada.

---

3. Estrutura dos Indicadores

Campo	Descrição
column_name	Nome da coluna
data_type	Tipo de dado no BigQuery
is_nullable	Indica se aceita nulos
table_name	Nome da tabela inspecionada

---

## T014 — QUERY COMPLETA

```
-- =====
-- T014 — QUERY: Documentação de Estrutura das Tabelas
-- Projeto: Olist E-commerce — Testes de Governança
-- =====

-- BLOCO 1 — Estrutura da tabela order_reviews
SELECT column_name, data_type
FROM `olist-sandbox-portfolio.olist_analysis`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'order_reviews'
ORDER BY ordinal_position;

-- BLOCO 2 — Verificação geral das tabelas principais
SELECT table_name, column_name, data_type, is_nullable
FROM `olist-sandbox-portfolio.olist_analysis`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name IN (
    'customers', 'order_items', 'orders', 'products',
    'sellers', 'order_payments', 'order_reviews',
    'product_category_name_translation'
)
ORDER BY table_name, ordinal_position;
```

---

## T015 — Query de Teste: Última Data e Recência por Segmento RFM

### (Anexo – T015)

#### Referência recomendada:

“Query completa disponível em: Anexo – T015: Última data do dataset e análise de recência.”

---

### 1. Objetivo da Investigação

- Identificar a última data válida do dataset;
  - Avaliar distribuição de recência entre segmentos RFM;
  - Validar adesão da view de marketing às métricas temporais.
- 

### 2. Descrição Executiva

- O primeiro bloco retorna a última data disponível na tabela fato;
  - O segundo bloco mede recência mínima, máxima e quantidade de clientes por segmento RFM.
- 

### 3. Estrutura dos Indicadores

---

<b>Campo</b>	<b>Descrição</b>
ultima_data_dataset	Data mais recente no dataset
total_registros	Quantidade de pedidos válidos
segmento_rfm	Segmento RFM
min_recencia	Menor número de dias desde última compra
max_recencia	Maior número de dias desde última compra
total_clientes	Total de clientes no segmento

---

#### T015 — QUERY COMPLETA

```
-- =====
-- T015 — QUERY: Última Data e Recência RFM
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Última data disponível
SELECT
  MAX(data_compra) AS ultima_data_dataset,
  COUNT(*) AS total_registros
FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido`
WHERE data_compra IS NOT NULL;

-- BLOCO 2 — Recência por segmento RFM
SELECT
  segmento_rfm,
  MIN(recencia_dias) AS min_recencia,
  MAX(recencia_dias) AS max_recencia,
  COUNT(*) AS total_clientes
FROM `olist-sandbox-portfolio.olist_analysis.vw_dashboard_marketing`
GROUP BY segmento_rfm
ORDER BY min_recencia;
```

---

#### T016 — Query de Teste: Validação da Tabela Customers

##### (Anexo – T016)

##### Referência recomendada no corpo do documento:

“Query completa disponível em: Anexo – T016: Validação estrutural e qualitativa da tabela customers.”

---

#### 1. Objetivo da Investigação

- Verificar integridade da tabela customers;
- Validar chaves, volumetria e consistência geográfica;
- Identificar possíveis anomalias em estados e identificadores.

---

#### 2. Descrição Executiva

- A primeira query exibe os primeiros registros, permitindo inspeção da estrutura;
- A segunda query consolida métricas gerais, incluindo unicidade das chaves e distribuição geográfica.

---

### 3. Estrutura dos Indicadores

<b>Campo</b>	<b>Descrição</b>
total_clientes	Número total de linhas na tabela
clientes_unicos	Quantidade de customer_unique_id distintos
estados	Total de estados presentes
estado_min / estado_max	Faixa alfabética usada para identificação de inconsistências

---

#### T016 — QUERY COMPLETA

```
-- =====
-- T016 — QUERY: Validação da Tabela CUSTOMERS
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Estrutura da tabela (amostra)
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.customers`
LIMIT 5;

-- BLOCO 2 — Teste de qualidade
SELECT
  COUNT(*) AS total_clientes,
  COUNT(DISTINCT customer_unique_id) AS clientes_unicos,
  COUNT(DISTINCT customer_state) AS estados,
  MIN(customer_state) AS estado_min,
  MAX(customer_state) AS estado_max
FROM `olist-sandbox-portfolio.olist_analysis.customers`;
```

---

#### T017 — Query de Teste: Validação da Tabela Orders

##### (Anexo – T017)

##### Referência recomendada:

“Query completa disponível em: Anexo – T017: Validação estrutural e volumétrica da tabela orders.”

---

#### 1. Objetivo da Investigação

- Validar a estrutura e integridade temporal da tabela orders;
  - Revisar chaves, volumetria, status e datas essenciais;
  - Confirmar consistência das datas de compra (primeira e última).
-

## 2. Descrição Executiva

- A primeira query mostra uma amostra estrutural;
- A segunda consolida métricas essenciais como quantidade de pedidos, clientes e status distintos.

## 3. Estrutura dos Indicadores

Campo	Descrição
total_pedidos	Total de registros
pedidos_unicos	Quantidade de order_id distintos
clientes_unicos	Número de clientes distintos
status_diferentes	Contagem de status possíveis
data_primeiro_pedido	Primeiro registro temporal
data_ultimo_pedido	Último registro temporal

### T017 — QUERY COMPLETA

```
-- =====
-- T017 — QUERY: Validação da Tabela ORDERS
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Estrutura da tabela
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.orders`
LIMIT 5;

-- BLOCO 2 — Teste de qualidade
SELECT
  COUNT(*) AS total_pedidos,
  COUNT(DISTINCT order_id) AS pedidos_unicos,
  COUNT(DISTINCT customer_id) AS clientes_unicos,
  COUNT(DISTINCT order_status) AS status_diferentes,
  MIN(order_purchase_timestamp) AS data_primeiro_pedido,
  MAX(order_purchase_timestamp) AS data_ultimo_pedido
FROM `olist-sandbox-portfolio.olist_analysis.orders`;
```

### T018 — Query de Teste: Validação da Tabela Order Items

#### (Anexo – T018)

#### Referência recomendada:

“Query completa disponível em: Anexo – T018: Validação estrutural e qualitativa da tabela order\_items.”

## 1. Objetivo da Investigação

- Verificar integridade dos itens dos pedidos;
- Avaliar consistência de preços, fretes e chaves de relacionamento;
- Identificar possíveis anomalias em preços mínimos/máximos.

---

## 2. Descrição Executiva

- A primeira query exibe estrutura e tipos;
- A segunda query calcula métricas fundamentais (volumetria, unicidade e valores financeiros).

---

## 3. Estrutura dos Indicadores

<b>Campo</b>	<b>Descrição</b>
total_itens	Número total de itens
pedidos_com_itens	Pedidos que possuem pelo menos um item
produtos_unicos	Quantidade de produtos distintos
vendedores_unicos	Número de vendedores
preco_medio / frete_medio	Estatísticas financeiras essenciais
preco_minimo / preco_maximo	Valores extremos para auditoria

---

## T018 — QUERY COMPLETA

```
-- =====
-- T018 — QUERY: Validação da Tabela ORDER_ITEMS
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Estrutura da tabela
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.order_items`
LIMIT 5;

-- BLOCO 2 — Teste de qualidade
SELECT
  COUNT(*) AS total_itens,
  COUNT(DISTINCT order_id) AS pedidos_com_itens,
  COUNT(DISTINCT product_id) AS produtos_unicos,
  COUNT(DISTINCT seller_id) AS vendedores_unicos,
  AVG(price) AS preco_medio,
  AVG(freight_value) AS frete_medio,
  MIN(price) AS preco_minimo,
  MAX(price) AS preco_maximo
FROM `olist-sandbox-portfolio.olist_analysis.order_items`;
```

---

## T019 — Query de Teste: Validação da Tabela Payments

**(Anexo – T019)****Referência recomendada:**

“Query completa disponível em: Anexo – T019: Teste de integridade da tabela payments.”

**1. Objetivo da Investigação**

- Validar integridade e consistência da tabela payments;
- Avaliar diversidade de meios de pagamento;
- Verificar comportamento de parcelas e valores.

**2. Descrição Executiva**

- A primeira query permite inspeção estrutural;
- A segunda resume métricas essenciais de pagamentos e possíveis anomalias.

**3. Estrutura dos Indicadores**

<b>Campo</b>	<b>Descrição</b>
total_pagamentos	Total de registros
pedidos_com_pagamento	Pedidos distintos
meios_pagamento	Tipos distintos de meio de pagamento
parcelas_medias	Média de parcelas aplicadas
valor_medio_pagamento	Ticket médio por pagamento

**T019 — QUERY COMPLETA--**

```
=====

-- T019 — QUERY: Validação da Tabela PAYMENTS
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Estrutura da tabela
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.payments`
LIMIT 5;

-- BLOCO 2 — Teste de qualidade
SELECT
  COUNT(*) AS total_pagamentos,
  COUNT(DISTINCT order_id) AS pedidos_com_pagamento,
  COUNT(DISTINCT payment_type) AS meios_pagamento,
  AVG(payment_installments) AS parcelas_medias,
  AVG(payment_value) AS valor_medio_pagamento
FROM `olist-sandbox-portfolio.olist_analysis.payments`;
```

---

## T020 — Query de Teste: Validação da Tabela Reviews

### (Anexo – T020)

#### Referência recomendada:

“Query completa disponível em: Anexo – T020: Teste de consistência da tabela reviews.”

---

#### 1. Objetivo da Investigação

- Validar integridade da tabela reviews, incluindo métricas de satisfação;
  - Avaliar distribuição de notas e consistência das datas;
  - Confirmar unicidade dos identificadores das avaliações.
- 

#### 2. Descrição Executiva

- A primeira query apresenta uma amostra estrutural;
  - A segunda consolida métricas de qualidade e distribuição de avaliações.
- 

#### 3. Estrutura dos Indicadores

Campo	Descrição
total_reviews	Total de avaliações
clientes_distintos	Avaliadores únicos
notas_distintas	Notas possíveis registradas
data_primeira_review	Primeira avaliação registrada
data_ultima_review	Última avaliação registrada

---

#### T020 — QUERY COMPLETA--

```
=====

-- T020 — QUERY: Validação da Tabela REVIEWS
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Estrutura da tabela
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.reviews`
LIMIT 5;

-- BLOCO 2 — Teste de qualidade
SELECT
  COUNT(*) AS total_reviews,
  COUNT(DISTINCT review_id) AS reviews_unicas,
  COUNT(DISTINCT customer_id) AS clientes_distintos,
  COUNT(DISTINCT review_score) AS notas_distintas,
  MIN(review_creation_date) AS data_primeira_review,
```



```
MAX(review_creation_date) AS data_ultima_review
FROM `olist-sandbox-portfolio.olist_analysis.reviews`;
```

---

## T021 — Query de Teste: Validação da Tabela Geolocations

### (Anexo – T021)

#### Referência recomendada:

“Query completa disponível em: Anexo – T021: Teste de integridade da tabela geolocations.”

---

#### 1. Objetivo da Investigação

- Validar consistência dos dados geográficos (lat/long, CEPs e cidades);
  - Identificar redundâncias e inconsistências regionais;
  - Verificar estrutura e volumetria geral.
- 

#### 2. Descrição Executiva

- A primeira query retorna amostra para inspeção das colunas;
  - A segunda sintetiza volumetria e variedade geográfica.
- 

#### 3. Estrutura dos Indicadores

Campo	Descrição
total_registros	Registros totais da tabela
ceps_unicos	Quantidade de CEPs distintos
cidades_unicas	Cidades distintas mapeadas
estados_unicos	UF distintas encontradas

---

## T021 — QUERY COMPLETA

```
-- =====
-- T021 — QUERY: Validação da Tabela GEOLOCATIONS
-- Projeto: Olist E-commerce — Testes de Qualidade
-- =====

-- BLOCO 1 — Estrutura da tabela
SELECT *
FROM `olist-sandbox-portfolio.olist_analysis.geolocations`
LIMIT 5;

-- BLOCO 2 — Teste de qualidade
SELECT
  COUNT(*) AS total_registros,
  COUNT(DISTINCT geolocation_zip_code_prefix) AS ceps_unicos,
  COUNT(DISTINCT geolocation_city) AS cidades_unicas,
  COUNT(DISTINCT geolocation_state) AS estados_unicos
```

FROM `olist-sandbox-portfolio.olist\_analysis.geolocations`;

---

## T022 — Query de Teste: Investigação de Atrasos Logísticos

### (Anexo – T022)

#### Referência recomendada:

“Query completa disponível em: Anexo – T022: Investigação de atrasos logísticos e tempos de entrega.”

---

### 1. Objetivo da Investigação

- Identificar ocorrências de entrega fora do prazo;
  - Medir tempos médios e variações de entrega;
  - Auxiliar no diagnóstico de gargalos logísticos.
- 

### 2. Descrição Executiva

- A query calcula atraso, adianto e tempo médio de entrega;
  - Permite identificar estados e cidades com maior risco;
  - Fundamenta análises posteriores nas views operacionais.
- 

### 3. Estrutura dos Indicadores

Campo	Descrição
entregas_totais	Volume total de entregas registradas
entregas_atrasadas	Quantidade de pedidos com atraso
percentual_atraso	Atrasos / entregas concluídas
tempo_medio_entrega	Dias médios entre envio e entrega

---

## T022 — QUERY COMPLETA

```
-- =====
-- T022 — QUERY: Investigação de Atrasos Logísticos
-- Projeto: Olist E-commerce — Testes de Qualidade Operacional
-- =====

SELECT
  COUNT(*) AS entregas_totais,
  SUM(CASE WHEN order_delivered_customer_date >
order_estimated_delivery_date THEN 1 END) AS entregas_atrasadas,
  SAFE_DIVIDE(
```

```

SUM(CASE WHEN order_delivered_customer_date >
order_estimated_delivery_date THEN 1 END),
COUNT(*)
) AS percentual_atraso,
AVG(DATE_DIFF(order_delivered_customer_date, order_approved_at, DAY)) AS
tempo_medio_entrega
FROM `olist-sandbox-portfolio.olist_analysis.orders`
WHERE order_delivered_customer_date IS NOT NULL;

```

---

## T023 — Query de Teste: Investigação de Produtos Críticos

### (Anexo – T023)

#### Referência recomendada:

“Query completa disponível em: Anexo – T023: Produtos com características fora do padrão.”

---

### 1. Objetivo da Investigação

- Identificar produtos com medidas ou pesos inconsistentes;
  - Detectar categorias com risco operacional;
  - Analisar impacto potencial em frete e atraso.
- 

### 2. Descrição Executiva

- Verifica pesos, dimensões e categorias incomuns;
  - Ajuda a detectar casos que influenciam custos ou SLAs;
  - Contribui para análises correlacionadas nas views operacionais.
- 

### 3. Estrutura dos Indicadores

Campo	Descrição
produtos_criticos	Registros fora de padrão (peso/dimensões)
peso_medio	Média de peso dos itens analisados
comprimento_medio	Média das dimensões

---

## T023 — QUERY COMPLETA

```

-- =====
-- T023 — QUERY: Investigação de Produtos Críticos
-- Projeto: Olist E-commerce — Testes de Qualidade de Produtos
-- =====

```

```

SELECT
  COUNT(*) AS produtos_criticos,
  AVG(product_weight_g) AS peso_medio,
  AVG(product_length_cm) AS comprimento_medio
FROM `olist-sandbox-portfolio.olist_analysis.products`
WHERE
  product_weight_g <= 0
  OR product_length_cm <= 0
  OR product_height_cm <= 0
  OR product_width_cm <= 0;

```

---

T024 — Query de Teste: Investigação de Clientes com Zero Recorrência

**(Anexo – T024)**

**Referência recomendada:**

“Query completa disponível em: Anexo – T024: Investigação de clientes sem recompra.”

---

1. Objetivo da Investigação

- Detectar clientes que compram apenas uma vez;
  - Avaliar impacto na estratégia de retenção;
  - Apoiar análises posteriores de LTV e churn no dashboard de Marketing.
- 

2. Descrição Executiva

- Identifica compradores sem repetição;
  - Permite estimar impacto no GMV e categorias iniciais;
  - Suporte direto à segmentação RFM aplicada em V002.
- 

3. Estrutura dos Indicadores

Campo	Descrição
clientes_total	Total de clientes
clientes_sem_recorrencia	Compradores com apenas uma compra
Percentual	Proporção de clientes não recorrentes

---

T024 — QUERY COMPLETA

```

-- =====
-- T024 — QUERY: Clientes com Zero Recorrência
-- Projeto: Olist E-commerce — Testes de Qualidade de Comportamento
-- =====

```

```

SELECT

```

```

COUNT(DISTINCT customer_unique_id) AS clientes_total,
COUNT(*) AS clientes_sem_recorrencia,
SAFE_DIVIDE(
  COUNT(*),
  COUNT(DISTINCT customer_unique_id)
) AS percentual_sem_recorrencia
FROM (
  SELECT customer_unique_id
  FROM `olist-sandbox-portfolio.olist_analysis.orders`
  GROUP BY customer_unique_id
  HAVING COUNT(order_id) = 1
);

```

---

## Construção da Tabela Fato (C001)

Tabela Fato e Dimensões

**(Query completa usada para construir a Fato Principal do projeto)**

*Referenciada no corpo do documento como:*

**“Query disponível em: Anexo – C001: Tabela Fato”**

*(Anexo — Código SQL Completo)*

```

CREATE OR REPLACE TABLE `olist-sandbox-portfolio.olist_analysis.fato_pedido`
AS
SELECT
  -- CHAVES PRIMARIAS E ESTRANGEIRAS
  oi.order_id,
  oi.order_item_id,
  oi.product_id,
  oi.seller_id,
  o.customer_id,
  c.customer_unique_id,

  -- METRICAS FINANCEIRAS (COMERCIAL)
  oi.price as valor_produto,
  oi.freight_value as valor_frete,
  (oi.price + oi.freight_value) as GMV,
  op.payment_value as valor_pagamento,
  op.payment_type as tipo_pagamento,
  op.payment_installments as parcelas,

  -- DATAS PARA ANALISE TEMPORAL
  -- Referência: Anexo C001 - Seção "Datas para análise temporal"
  DATE(o.order_purchase_timestamp) as data_compra,
  DATE(o.order_approved_at) as data_aprovacao,
  DATE(o.order_delivered_carrier_date) as data_envio_transportadora,

```

```

DATE(o.order_delivered_customer_date) as data_entrega_cliente,
DATE(o.order_estimated_delivery_date) as data_estimada_entrega,

-- =====
-- CÁLCULOS DE TEMPO COM TRATAMENTO DE NULL
-- Referência: Anexo C001 - Seção "Cálculos de Tempo com tratamento de
NULL"
-- Tratamento aplicado para evitar TME inválido e inconsistências temporais
-- =====

CASE
  WHEN o.order_delivered_customer_date IS NOT NULL
  AND o.order_purchase_timestamp IS NOT NULL
  THEN DATE_DIFF(
    DATE(o.order_delivered_customer_date),
    DATE(o.order_purchase_timestamp),
    DAY
  )
END as tempo_entrega_total_dias,

CASE
  WHEN o.order_delivered_customer_date IS NOT NULL
  AND o.order_approved_at IS NOT NULL
  THEN DATE_DIFF(
    DATE(o.order_delivered_customer_date),
    DATE(o.order_approved_at),
    DAY
  )
END as tempo_aprovacao_entrega_dias,

-- =====
-- SLA DO VENDEDOR (TEMPO DE PREPARAÇÃO)
-- Referência: Anexo C001 - Seção "Regras temporais (TME, atraso e SLA)"
-- Tempo entre aprovação e entrega à transportadora
-- =====

CASE
  WHEN o.order_delivered_carrier_date IS NOT NULL
  AND o.order_approved_at IS NOT NULL
  THEN DATE_DIFF(
    DATE(o.order_delivered_carrier_date),
    DATE(o.order_approved_at),
    DAY
  )
END as tempo_preparacao_vendedor_dias,

```

```

-- =====
-- INDICADORES DE PERFORMANCE
-- Referência: Anexo C001 - Seção "Indicadores Derivados"
-- =====

-- INDICADOR DE ATRASO
-- Referência: Anexo C001 - Bloco "Indicadores de Performance"
CASE
  WHEN o.order_delivered_customer_date IS NOT NULL
  AND o.order_estimated_delivery_date IS NOT NULL
  AND DATE(o.order_delivered_customer_date) >
DATE(o.order_estimated_delivery_date)
  THEN 1 ELSE 0
END as indicador_atraso,

-- INDICADOR DE PROBLEMA (CANCELADOS/INDISPONÍVEIS)
-- Referência: Anexo C001 - Seção "Inclusão de Todos os Status"
CASE
  WHEN o.order_status IN ('canceled', 'unavailable')
  THEN 1 ELSE 0
END as indicador_problema,

-- DIMENSOES DO PRODUTO
p.product_category_name,
p.product_weight_g,
p.product_length_cm,
p.product_height_cm,
p.product_width_cm,

-- DIMENSOES GEOGRAFICAS
c.customer_city as cidade_cliente,
c.customer_state as estado_cliente,
s.seller_city as cidade_vendedor,
s.seller_state as estado_vendedor,

-- DIMENSOES DO PEDIDO E STATUS
-- Referência: Anexo C001 - Seção "Construção sem filtro de status"
o.order_status as status_pedido,

-- =====
-- CORREÇÃO CRÍTICA DA TABELA ORDER_REVIEWS
-- Referência: Anexo C001 - Seção "Mapeamento dos campos string_field"
-- Campos string_field_1...5 mapeados após investigação

```

```

-- =====
r.string_field_1 as review_id,
r.string_field_2 as order_id_review,
r.string_field_3 as review_comment_title,
r.string_field_4 as review_comment_message,

-- =====
-- CONVERSÕES DE DATAS E TIMESTAMPS
-- Referência: Anexo C001 - Seção "Conversões de Datas e Timestamps"
-- Parse seguro do campo string_field_5 para data de avaliação
-- =====
CASE
  WHEN r.string_field_5 IS NOT NULL
  THEN DATE(PARSE_TIMESTAMP('%Y-%m-%d %H:%M:%S', r.string_field_5))
END as data_avaliacao,

-- CÁLCULO TEMPO ENTREGA -> AVALIAÇÃO
CASE
  WHEN r.string_field_5 IS NOT NULL
  AND o.order_delivered_customer_date IS NOT NULL
  THEN DATE_DIFF(
    DATE(PARSE_TIMESTAMP('%Y-%m-%d %H:%M:%S', r.string_field_5)),
    DATE(o.order_delivered_customer_date),
    DAY
  )
END as tempo_entrega_avaliacao_dias

FROM `olist-sandbox-portfolio.olist_analysis.order_items` oi
JOIN `olist-sandbox-portfolio.olist_analysis.orders` o
  ON oi.order_id = o.order_id
JOIN `olist-sandbox-portfolio.olist_analysis.customers` c
  ON o.customer_id = c.customer_id
JOIN `olist-sandbox-portfolio.olist_analysis.products` p
  ON oi.product_id = p.product_id
JOIN `olist-sandbox-portfolio.olist_analysis.sellers` s
  ON oi.seller_id = s.seller_id
LEFT JOIN `olist-sandbox-portfolio.olist_analysis.order_payments` op
  ON oi.order_id = op.order_id
LEFT JOIN `olist-sandbox-portfolio.olist_analysis.order_reviews` r
  ON oi.order_id = r.string_field_2 -- CORREÇÃO CRÍTICA: string_field_2 é o
order_id

-- =====
-- CONSTRUÇÃO SEM FILTRO DE STATUS

```



```
-- Referência: Anexo C001 - Seção "Construção sem filtro de status"
-- Inclusão de todos os status: cancelados, indisponíveis, entregues, etc.
-- =====
-- REMOVI O WHERE para incluir todos os status de pedido
;
```

---

## Views Principais (V001, V002, V003)

V001 — View Comercial (vw\_dashboard\_comercial)

**(View utilizada para análises de GMV, pedidos, ticket médio e indicadores comerciais)**

**Referenciada no corpo do documento como:**

*“Query disponível em: Anexo – V001: View Comercial”*

-- V001 — View Comercial (vw\_dashboard\_comercial)

-- Consolida GMV, Volume, Ticket Médio, Cancelamentos e Inadimplência.

```
CREATE OR REPLACE VIEW `olist-sandbox-portfolio.olist_analysis.vw_dashboard_comercial` AS
WITH base AS (
  SELECT
    order_id,
    customer_id,
    categoria,
    seller_state,
    seller_city,
    data_compra,
    gmv,
    price,
    freight_value,
    order_status,
    pedido_problema,
    pedido_atrasado
  FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido`
)
SELECT
  DATE_TRUNC(data_compra, MONTH) AS mes,
  categoria,
  seller_state,
  COUNT(DISTINCT order_id) AS total_pedidos,
  SUM(gmv) AS gmv_total,
  AVG(gmv) AS gmv_medio,
  SAFE_DIVIDE(SUM(gmv), COUNT(DISTINCT order_id)) AS ticket_medio,
  SUM(CASE WHEN order_status = 'canceled' THEN 1 END) AS
pedidos_cancelados,
  SUM(CASE WHEN pedido_problema = 1 THEN 1 END) AS pedidos_problema,
  SAFE_DIVIDE(
```

```

SUM(CASE WHEN pedido_problema = 1 THEN 1 END),
COUNT(DISTINCT order_id)
) AS taxa_inadimplencia
FROM base
GROUP BY mes, categoria, seller_state;

```

---

V002 — View Marketing (vw\_dashboard\_marketing)

**(View utilizada para segmentação RFM, LTV e comportamento do cliente)**

**Referenciada no corpo do documento como:**

*“Query disponível em: Anexo – V002: View Marketing”*

```

-- V002 — View Marketing (vw_dashboard_marketing)
-- Base única para métricas de LTV, Recorrência, Recência e Segmentação RFM.

```

```

WITH base AS (
  SELECT
    customer_id,
    categoria,
    data_compra,
    gmv
  FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido`
),
agregado AS (
  SELECT
    customer_id,
    COUNT(DISTINCT order_id) AS frequencia,
    SUM(gmv) AS ltv,
    MAX(data_compra) AS ultima_compra,
    MIN(data_compra) AS primeira_compra
  FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido`
  GROUP BY customer_id
),
rfm AS (
  SELECT
    *,
    DATE_DIFF((SELECT MAX(data_compra) FROM base), ultima_compra, DAY) AS
recencia
  FROM agregado
),
scored AS (
  SELECT
    *,
    NTILE(5) OVER (ORDER BY recencia) AS score_r,
    NTILE(5) OVER (ORDER BY frequencia) AS score_f,
    NTILE(5) OVER (ORDER BY ltv) AS score_m
  FROM rfm
)

```

```

SELECT
  customer_id,
  frequencia,
  ltv,
  recencia,
  primeira_compra,
  ultima_compra,
  (score_r + score_f + score_m) AS score_rfm
FROM scored;

```

---

V003 — View Operacional (vw\_dashboard\_operacional)

**(View utilizada para métricas logísticas, atraso, SLA e faixas de peso)**

**Referenciada no corpo do documento como:**

*“Query disponível em: Anexo – V003: View Operacional”*

-- V003 — View Operacional (vw\_dashboard\_operacional)

-- Métricas de TME, SLA, Atraso, Frete e análise logística por Estado e Peso.

```

WITH base AS (
  SELECT
    order_id,
    seller_state,
    categoria,
    gmv,
    price,
    freight_value,
    pedido_atrasado,
    tme_dias,
    sla_vendedor_dias,
    product_id
  FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido`
),
prod AS (
  SELECT
    product_id,
    product_weight_g
  FROM `olist-sandbox-portfolio.olist_raw.olist_products_dataset`
),
joined AS (
  SELECT
    b.*,
    p.product_weight_g,
    CASE
      WHEN p.product_weight_g < 500 THEN '0-500g'
      WHEN p.product_weight_g BETWEEN 500 AND 2000 THEN '500g-2kg'
      WHEN p.product_weight_g BETWEEN 2000 AND 5000 THEN '2kg-5kg'
      ELSE '5kg+'
    END
  FROM prod p
  JOIN base b ON p.product_id = b.product_id
)

```

```

    END AS faixa_peso
FROM base b
LEFT JOIN prod p USING (product_id)
)
SELECT
    seller_state,
    faixa_peso,
    COUNT(DISTINCT order_id) AS total_pedidos,
    AVG(tme_dias) AS tme_medio,
    AVG(sla_vendedor_dias) AS sla_medio,
    AVG(freight_value) AS frete_medio,
    SUM(CASE WHEN pedido_atrasado = 1 THEN 1 END) AS pedidos_atrasados,
    SAFE_DIVIDE(
        SUM(CASE WHEN pedido_atrasado = 1 THEN 1 END),
        COUNT(DISTINCT order_id)
    ) AS taxa_atraso
FROM joined
GROUP BY seller_state, faixa_peso;

```

---

## Views de Suporte (S001 a S007)

S001 — View de Suporte: vw\_base\_categorias (Anexo – S001)

### Referência recomendada no corpo principal

“Query completa disponível em: **Anexo – S001: Base de Categorias** .

### Descrição da View

A **S001 — vw\_base\_categorias** consolida todas as métricas essenciais para análises de desempenho por categoria, já com nomes padronizados, indicadores adicionais e rankings que facilitam:

- análises comerciais;
- análises operacionais;
- priorização de categorias-chave para estratégia;
- construção de dashboards (Looker Studio / Power BI).

Esta view também executa **tratamento completo dos nomes das categorias**, unificando grafias e facilitando a leitura por áreas de negócio.

### Objetivos da View

A S001 foi projetada para servir como **base única e flexível** para qualquer análise envolvendo categorias.

Ela contém:

### Métricas Principais

- Total de pedidos
- GMV total
- Produtos únicos
- Vendedores únicos

### Métricas Secundárias

- Valor pago
- Tempo médio de entrega
- Ticket médio por categoria

### Rankings (GMV, Pedidos, Produtos)

Com funções de janela, permitindo criação de filtros por segmento e uso em dashboards.

### Tratamento e Normalização de Categorias

Substitui nomes brutos da base Olist por rótulos de negócio padronizados.

### Indicador de Relevância

Classifica categorias em:

- Alta
  - Média
  - Baixa
- de acordo com o volume de pedidos.

---

### Observações importantes

- Apenas pedidos **entregues** são considerados (mantém consistência analítica).
- Todos os rankings são calculados dentro da própria view, sem necessidade de joins adicionais.
- Os títulos de categoria seguem padrão comercial (com acentuação e capitalização).

```
-- =====
-- S001 — VIEW: vw_base_categorias
-- Projeto: Olist E-commerce — Views de Suporte
--
-- Objetivo:
-- Criar base consolidada e padronizada para análises por categoria,
-- incluindo:
--   • Total de pedidos por categoria
--   • GMV total
--   • Produtos únicos
--   • Vendedores únicos
--   • Ticket médio por categoria
--   • Rankings (GMV, Pedidos, Produtos)
--   • Indicador de relevância
--
-- Destaques:
-- - Tratamento completo dos nomes de categoria
-- - Apenas pedidos entregues (consistência analítica)
--
-- Dependência:
--   • C001 — fato_pedido
--
```

```
-- Data: 2025
-- =====

CREATE OR REPLACE VIEW `olist-sandbox-
portfolio.olist_analysis.vw_base_categorias` AS
SELECT
  fp.product_category_name AS categoria_original,

  -- =====
  -- TRATAMENTO E PADRONIZAÇÃO DE NOMES DAS CATEGORIAS
  -- =====
  CASE
    WHEN fp.product_category_name = 'informatica_acessorios' THEN 'Informática
e Acessórios'
    WHEN fp.product_category_name = 'cama_mesa_banho' THEN 'Cama, Mesa e
Banho'
    WHEN fp.product_category_name = 'beleza_saude' THEN 'Beleza e Saúde'
    WHEN fp.product_category_name = 'esporte_lazer' THEN 'Esporte e Lazer'
    WHEN fp.product_category_name = 'relogios_presentes' THEN 'Relógios e
Presentes'
    WHEN fp.product_category_name = 'moveis_decoracao' THEN 'Móveis e
Decoração'
    WHEN fp.product_category_name = 'telefonica' THEN 'Telefonia'
    WHEN fp.product_category_name = 'informatica' THEN 'Informática'
    WHEN fp.product_category_name = 'consoles_games' THEN 'Consoles e
Games'
    WHEN fp.product_category_name = 'brinquedos' THEN 'Brinquedos'
    WHEN fp.product_category_name = 'bebes' THEN 'Bebês'
    WHEN fp.product_category_name = 'malas_acessorios' THEN 'Malas e
Acessórios'
    ELSE INITCAP(REPLACE(fp.product_category_name, '_', ' '))
  END AS categoria,

  -- =====
  -- MÉTRICAS PRINCIPAIS
  -- =====
  COUNT(DISTINCT fp.order_id) AS total_pedidos,
  SUM(fp.GMV) AS gmv_total,
  COUNT(DISTINCT fp.product_id) AS produtos_unicos,
  COUNT(DISTINCT fp.seller_id) AS vendedores_unicos,

  -- =====
  -- MÉTRICAS SECUNDÁRIAS
  -- =====

  SUM(fp.valor_pagamento) AS valor_total_pago,
  AVG(fp.tempo_entrega_total_dias) AS tempo_medio_entrega,
```

```

-- =====
-- TICKET MÉDIO
-- =====
CASE
  WHEN COUNT(DISTINCT fp.order_id) > 0
  THEN SUM(fp.GMV) / COUNT(DISTINCT fp.order_id)
  ELSE 0
END AS ticket_medio_categoria,

-- =====
-- RANKINGS
-- =====
RANK() OVER (ORDER BY SUM(fp.GMV) DESC) AS ranking_gmv,
RANK() OVER (ORDER BY COUNT(DISTINCT fp.order_id) DESC) AS
ranking_pedidos,
RANK() OVER (ORDER BY COUNT(DISTINCT fp.product_id) DESC) AS
ranking_produtos_unicos,

-- =====
-- INDICADOR DE RELEVÂNCIA POR PEDIDOS
-- =====
CASE
  WHEN COUNT(DISTINCT fp.order_id) >= 100 THEN 'Alta'
  WHEN COUNT(DISTINCT fp.order_id) >= 50 THEN 'Média'
  ELSE 'Baixa'
END AS relevancia_pedidos

FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido` fp
WHERE fp.product_category_name IS NOT NULL
  AND fp.status_pedido = 'delivered' -- Apenas pedidos entregues
GROUP BY fp.product_category_name
ORDER BY gmv_total DESC;

```

---

S002 — View de Suporte: vw\_top10\_categorias (Anexo – S002)

**Referência recomendada no corpo principal:**

“Query completa disponível em: Anexo 11 – S002: Top 10 Categorias por GMV”

---

**1. Objetivo da Investigação**

A view S002 identifica as 10 categorias com maior GMV com base na base consolidada vw\_base\_categorias.

Serve como referência para dashboards, filtros e rankings estratégicos utilizados pelo time comercial.

---

**2. Descrição**

- Filtra apenas as **10 categorias com maior GMV**;

- Mantém consistência com os cálculos da base consolidada (S001);
- Permite uso direto em dashboards e relatórios de desempenho.

---

### 3. Estrutura dos Indicadores

Campo	Descrição
Categoria	Nome da categoria, padronizado na S001
gmv_total	GMV total da categoria
total_pedidos	Quantidade total de pedidos entregues
ranking_gmv	Ranking por GMV (1 = maior GMV)

---

### 4. Observações Técnicas

- Ordenação por ranking\_gmv;
  - Limite de 10 categorias;
  - Base de referência: vw\_base\_categorias;
  - Apenas pedidos entregues são considerados.
- 

-- VIEW ESPECÍFICA PARA O FILTRO (TOP 10 CATEGORIAS)

```
CREATE OR REPLACE VIEW `olist-sandbox-portfolio.olist_analysis.vw_top10_categorias` AS
SELECT
  categoria,
  gmv_total,
  total_pedidos,
  ranking_gmv
FROM `olist-sandbox-portfolio.olist_analysis.vw_base_categorias`
WHERE ranking_gmv <= 10
ORDER BY ranking_gmv;
```

---

S003 — View de Suporte vw\_temporal\_unificada (Anexo – S003)

No corpo principal, referenciada como:

**“Query completa disponível em: Anexo 11 – S003: Base Temporal Unificada para Todas as Áreas”**

```
-- =====
-- S003 — VIEW: vw_temporal_unificada
-- Projeto: Olist E-commerce — Views de Suporte
--
-- Objetivo:
--   Criar a base temporal unificada contendo indicadores
--   comerciais, operacionais e financeiros agregados por mês,
--   incluindo:
--     • GMV total e variação mensal
--     • Total de pedidos
```



```
--
-- • Ticket médio
-- • Indicadores de entrega (sucesso, atraso)
-- • Métricas de pagamento
-- • Métricas de produto e vendedores
-- • KPI de tendência (crescimento/queda/estável)
--
-- Dependência:
-- • C001 — fato_pedido
--
-- Observação:
-- Esta visão serve como camada única de tempo para dashboards
-- Comercial, Marketing e Operacional, garantindo consistência
-- das métricas mensais.
--
-- Data: 2025
-- =====
```

```
CREATE OR REPLACE VIEW `olist-sandbox-portfolio.olist_analysis.vw_temporal_unificada` AS
```

```
WITH dados_mensais AS (
  SELECT
    -----
    -- DIMENSÕES TEMPORAIS
    -----
    DATE_TRUNC(
      DATE(fp.data_compra), MONTH) AS mes,
    FORMAT_DATE('%b/%Y', DATE(fp.data_compra)) AS
mes_formatado,
    EXTRACT(YEAR FROM fp.data_compra) AS ano,
    EXTRACT(MONTH FROM fp.data_compra) AS mes_numero,

    -- Campos extras para ordenação consistente
    EXTRACT(YEAR FROM fp.data_compra) * 100
    + EXTRACT(MONTH FROM fp.data_compra) AS
ano_mes_ordenacao,
    FORMAT_DATE('%Y-%m', DATE(fp.data_compra)) AS
mes_ordenacao_string,

    -----
    -- MÉTRICAS COMERCIAIS
    -----
    SUM(fp.GMV) AS gmv_total,
    COUNT(DISTINCT fp.order_id) AS total_pedidos,

    CASE
      WHEN COUNT(DISTINCT fp.order_id) > 0
      THEN SUM(fp.GMV) / COUNT(DISTINCT fp.order_id)
```

```

ELSE 0
END                                AS ticket_medio,

-----

-- MÉTRICAS DE PAGAMENTO
-----

SUM(fp.valor_pagamento)           AS valor_total_pago,
COUNT(DISTINCT fp.customer_id)    AS clientes_ativos,

-----

-- MÉTRICAS OPERACIONAIS
-----

AVG(fp.tempo_entrega_total_dias)    AS
tempo_medio_entrega_dias,
AVG(fp.tempo_preparacao_vendedor_dias) AS
tempo_preparacao_vendedor_dias,

-----

-- INDICADORES DE PERFORMANCE
-----

CASE
  WHEN COUNT(DISTINCT fp.order_id) > 0
    THEN SUM(CASE WHEN fp.status_pedido = 'delivered'
                  THEN 1 ELSE 0 END)
        / COUNT(DISTINCT fp.order_id)
  ELSE 0
END                                AS taxa_entrega_sucesso,

CASE
  WHEN COUNT(DISTINCT fp.order_id) > 0
    THEN SUM(fp.indicador_atraso)
        / COUNT(DISTINCT fp.order_id)
  ELSE 0
END                                AS taxa_atraso,

CASE
  WHEN COUNT(DISTINCT fp.order_id) > 0
    THEN SUM(CASE WHEN fp.status_pedido = 'canceled'
                  THEN 1 ELSE 0 END)
        / COUNT(DISTINCT fp.order_id)
  ELSE 0
END                                AS taxa_cancelamento,

-----

-- MÉTRICAS DE PRODUTO E VENDEDORES
-----

COUNT(DISTINCT fp.product_id)      AS produtos_unicos,
```

```

COUNT(DISTINCT fp.seller_id)                AS vendedores_ativos,

-----

-- MÉTRICAS GEOGRÁFICAS
-----

COUNT(DISTINCT fp.estado_cliente)           AS estados_ativos

FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido` fp
WHERE fp.data_compra IS NOT NULL
  AND fp.status_pedido IS NOT NULL
GROUP BY
  mes, mes_formatado, ano, mes_numero,
  ano_mes_ordenacao, mes_ordenacao_string
)

SELECT
  dados_mensais.*,

-----

-- VARIAÇÃO DE GMV (KPIs DE TENDÊNCIA)
-----

LAG(gmv_total) OVER (ORDER BY mes)            AS gmv_mes_anterior,

CASE
  WHEN LAG(gmv_total) OVER (ORDER BY mes) > 0
    THEN (gmv_total
      - LAG(gmv_total) OVER (ORDER BY mes))
      / LAG(gmv_total) OVER (ORDER BY mes)
    ELSE NULL
END                                             AS variacao_gmv_mensal,

CASE
  WHEN LAG(gmv_total) OVER (ORDER BY mes) > 0
    THEN FORMAT('%.1f%%',
      ((gmv_total - LAG(gmv_total) OVER (ORDER BY mes))
      / LAG(gmv_total) OVER (ORDER BY mes)) * 100)
    ELSE 'N/A'
END                                             AS variacao_gmv_formatada,

CASE
  WHEN LAG(gmv_total) OVER (ORDER BY mes) > 0
    AND ((gmv_total - LAG(gmv_total) OVER (ORDER BY mes))
      / LAG(gmv_total) OVER (ORDER BY mes)) > 0
    THEN 'crescimento'
  WHEN LAG(gmv_total) OVER (ORDER BY mes) > 0
    AND ((gmv_total - LAG(gmv_total) OVER (ORDER BY mes))
      / LAG(gmv_total) OVER (ORDER BY mes)) < 0

```

```

        THEN 'queda'
        ELSE 'estavel'
    END                                AS tendencia_gmv

```

```

FROM dados_mensais
ORDER BY mes
;

```

---

S004 — View de Suporte vw\_base\_vendedores (Anexo – S004)

**Referência recomendada no corpo principal:**

“Query completa disponível em: Anexo 11 – S004: Base Consolidada de Vendedores”

---

```

-- =====
-- S004 — VIEW: vw_base_vendedores
-- Projeto: Olist E-commerce — Views de Suporte
--
-- Objetivo:
--   Criar a visão consolidada para análise de vendedores,
--   seguindo a mesma estrutura utilizada para categorias (S001),
--   incluindo:
--     • GMV total por vendedor
--     • Total de pedidos
--     • Contagem de produtos únicos vendidos
--     • Número de clientes atendidos
--     • KPIs de performance operacional
--     • Rankings para navegação e filtros no Looker Studio
--
-- Dependências:
--   • C001 — fato_pedido
--   • D003 — sellers
--
-- Observação:
--   Apenas pedidos entregues são considerados, garantindo
--   consistência no cálculo de GMV, performance e relevância.
--
-- Data: 2025
-- =====

```

```

CREATE OR REPLACE VIEW `olist-sandbox-
portfolio.olist_analysis.vw_base_vendedores` AS
SELECT
    fp.seller_id AS vendedor_id,
    s.seller_city AS cidade_vendedor,
    s.seller_state AS estado_vendedor,

```

```

-- Nome formatado para uso em dashboards

```

```
CONCAT(INITCAP(s.seller_city), ' - ', s.seller_state) AS cidade_estado_vendedor,
```

```
-----  
-- MÉTRICAS PRINCIPAIS  
-----
```

```
COUNT(DISTINCT fp.order_id) AS total_pedidos,  
SUM(fp.GMV) AS gmv_total,  
COUNT(DISTINCT fp.product_id) AS produtos_unicos,  
COUNT(DISTINCT fp.customer_id) AS clientes_unicos,
```

```
-----  
-- MÉTRICAS DE PERFORMANCE  
-----
```

```
AVG(fp.tempo_preparacao_vendedor_dias) AS tempo_medio_preparacao,  
AVG(fp.valor_frete) AS valor_medio_frete,
```

```
-----  
-- CÁLCULOS DERIVADOS  
-----
```

```
CASE  
  WHEN COUNT(DISTINCT fp.order_id) > 0  
  THEN SUM(fp.GMV) / COUNT(DISTINCT fp.order_id)  
  ELSE 0  
END AS ticket_medio_vendedor,
```

```
-----  
-- RANKINGS  
-----
```

```
RANK() OVER (ORDER BY SUM(fp.GMV) DESC) AS ranking_gmv,  
RANK() OVER (ORDER BY COUNT(DISTINCT fp.order_id) DESC) AS  
ranking_pedidos,  
RANK() OVER (ORDER BY COUNT(DISTINCT fp.customer_id) DESC) AS  
ranking_clientes,
```

```
-----  
-- INDICADOR DE RELEVÂNCIA  
-----
```

```
CASE  
  WHEN COUNT(DISTINCT fp.order_id) >= 100 THEN 'Alta'  
  WHEN COUNT(DISTINCT fp.order_id) >= 50 THEN 'Média'  
  ELSE 'Baixa'  
END AS relevancia_vendedor
```

```
FROM `olist-sandbox-portfolio.olist_analysis.fato_pedido` fp  
JOIN `olist-sandbox-portfolio.olist_analysis.sellers` s  
  ON fp.seller_id = s.seller_id  
WHERE fp.status_pedido = 'delivered'
```

GROUP BY fp.seller\_id, s.seller\_city, s.seller\_state  
ORDER BY gmv\_total DESC;

S005 — View de Suporte: vw\_top5\_cidades\_atraso (Anexo – S005)

#### Referência recomendada no corpo principal:

“Query completa disponível em: Anexo 11 – S005: Top 5 Cidades com Maior Taxa de Atraso”

### 1. Objetivo da Investigação

A view S005 identifica as 5 cidades com maior percentual de pedidos atrasados, fornecendo suporte para análises operacionais e melhoria logística.

### 2. Descrição Executiva

- Calcula para cada cidade:
  - Total de pedidos entregues;
  - Total de pedidos atrasados;
  - Taxa de atraso percentual;
- Considera apenas cidades com **mínimo de 10 pedidos** para evitar vieses estatísticos;
- Ordena as cidades em ordem decrescente de taxa de atraso, mostrando as 5 mais críticas.

### 3. Estrutura dos Indicadores

Campo	Descrição
cidade_cliente	Nome da cidade do cliente
total_pedidos	Total de pedidos entregues na cidade
pedidos_atrasados	Total de pedidos com entrega após o prazo estimado
taxa_atraso_percent	Percentual de pedidos atrasados
Critério mínimo	Considera apenas cidades com $\geq 10$ pedidos

### 4. Observações Técnicas

- A condição de mínimo de pedidos é aplicada usando HAVING;
- Taxa de atraso é arredondada para duas casas decimais;
- Base de referência: vw\_dashboard\_operacional;
- Ordenação por taxa\_atraso\_percent em ordem decrescente;
- Limite de 5 registros.

```
--=====
-- TOP 5 CIDADES EM ATRASO
--=====
```

```
CREATE OR REPLACE VIEW `olist-sandbox-
portfolio.olist_analysis.vw_top5_cidades_atraso` AS
SELECT
  cidade_cliente,
```

```

COUNT(order_id) as total_pedidos,
COUNT(CASE WHEN data_entrega_cliente > data_estimada_entrega THEN
order_id END) as pedidos_atrasados,
ROUND(
  (COUNT(CASE WHEN data_entrega_cliente > data_estimada_entrega THEN
order_id END) / COUNT(order_id)) * 100,
  2
) as taxa_atraso_percent
FROM `olist-sandbox-portfolio.olist_analysis.vw_dashboard_operacional`
WHERE cidade_cliente IS NOT NULL
GROUP BY cidade_cliente
HAVING COUNT(order_id) >= 10
ORDER BY taxa_atraso_percent DESC
LIMIT 5;

```

---

S006 — View de Suporte: vw\_frete\_vs\_peso (Anexo – S006)

**Referência recomendada no corpo principal:**

“Query completa disponível em: Anexo 11 – S006: Frete vs Peso do Produto”

---

```

-- =====
-- S006 — VIEW: vw_frete_vs_peso
-- Projeto: Olist E-commerce — Views de Suporte
--
-- Objetivo:
--   Analisar a relação entre peso do produto e valor do frete,
--   incluindo métricas de pedidos, frete médio, peso e dimensões.
--
-- Base de Dados:
--   • order_items
--   • products
--
-- Observações:
--   • Apenas produtos com peso entre 1g e 50kg são considerados
--   • Apenas fretes entre R$1 e R$500 são incluídos
--   • Faixas de peso definidas para agrupamento
--   • Outliers removidos para consistência analítica
-- Data: 2025
-- =====

```

```

CREATE OR REPLACE VIEW `olist-sandbox-
portfolio.olist_analysis.vw_frete_vs_peso` AS

```

```

SELECT

```

```

-----
-- DEFINIÇÃO DAS FAIXAS DE PESO
-----

```

```

CASE

```

```

    WHEN p.product_weight_g <= 1000 THEN '0-1kg'
    WHEN p.product_weight_g <= 3000 THEN '1-3kg'
    WHEN p.product_weight_g <= 5000 THEN '3-5kg'
    WHEN p.product_weight_g <= 10000 THEN '5-10kg'
    ELSE '10kg+'
END AS faixa_peso,

-----
-- MÉTRICAS PRINCIPAIS
-----

COUNT(oi.order_id) AS total_pedidos,      -- Total de pedidos por faixa
ROUND(AVG(oi.freight_value), 2) AS frete_medio,  -- Valor médio do frete
ROUND(AVG(p.product_weight_g), 2) AS peso_medio,  -- Peso médio dos
produtos
ROUND(AVG(p.product_length_cm), 2) AS comprimento_medio, --
Comprimento médio
ROUND(AVG(p.product_height_cm), 2) AS altura_media,  -- Altura média
ROUND(AVG(p.product_width_cm), 2) AS largura_media  -- Largura média

FROM `olist-sandbox-portfolio.olist_analysis.order_items` oi
JOIN `olist-sandbox-portfolio.olist_analysis.products` p
  ON oi.product_id = p.product_id

WHERE p.product_weight_g IS NOT NULL
AND p.product_weight_g > 0
AND p.product_weight_g <= 50000      -- Removendo outliers extremos de
peso
AND oi.freight_value IS NOT NULL
AND oi.freight_value > 0
AND oi.freight_value <= 500        -- Removendo outliers de frete

GROUP BY faixa_peso

ORDER BY
  CASE faixa_peso
    WHEN '0-1kg' THEN 1
    WHEN '1-3kg' THEN 2
    WHEN '3-5kg' THEN 3
    WHEN '5-10kg' THEN 4
    WHEN '10kg+' THEN 5
  END;

```

---

S007 — View de Suporte: vw\_top5\_vendedores\_tme (Anexo – S007)

**Referência recomendada no corpo principal:**

*“Query completa disponível em: Anexo 11 – S007: Top 5 Vendedores por Tempo Médio de Entrega (TME)”*



1. Objetivo

A view S007 identifica os **5 principais vendedores** com maior **tempo médio de entrega (TME)**, permitindo análise operacional para melhoria logística, monitoramento de desempenho e identificação de gargalos no processo de entrega.

2. Descrição

- Calcula para cada vendedor:
  - Total de pedidos
  - Tempo médio de entrega (TME)
- Filtra apenas vendedores com **mínimo de 5 pedidos** para consistência estatística;
- Ordena em ordem decrescente de TME, destacando os vendedores com maior tempo médio.

3. Estrutura dos Indicadores

Campo	Descrição
seller_id	Identificador do vendedor
Vendedor	Cidade + Estado do vendedor, formatado para dashboards
total_pedidos	Total de pedidos do vendedor (≥ 5)
tempo_medio_entrega	TME médio do vendedor em dias

4. Observações Técnicas

- Base de referência: vw\_dashboard\_operacional;
- Limite de 6 registros para garantir top 5 com critério visual de dashboards;
- Cidade formatada com inicial maiúscula e estado em maiúsculo;
- Agrupamento feito por seller\_id, cidade e estado.

```
-- =====
-- S007 — VIEW: vw_top5_vendedores_tme
-- Projeto: Olist E-commerce — Views de Suporte
--
-- Objetivo:
--   Identificar os 5 vendedores com maior tempo médio de entrega (TME)
--   para análise operacional e logística.
--
-- Base de Dados:
--   • vw_dashboard_operacional
--
-- Observações:
--   • Considera apenas vendedores com ≥ 5 pedidos
--   • Limite de 6 registros para visualização em dashboards
-- Data: 2025
-- =====
```

```

CREATE OR REPLACE VIEW `olist-sandbox-portfolio.olist_analysis.vw_top5_vendedores_tme` AS
SELECT
    seller_id,

    -----
    -- FORMATAÇÃO DO NOME DO VENDEDOR
    -----
    CONCAT(INITCAP(cidade_vendedor), ' / ', UPPER(estado_vendedor)) AS
vendedor,

    -----
    -- MÉTRICAS PRINCIPAIS
    -----
    COUNT(order_id) AS total_pedidos,          -- Total de pedidos por vendedor
    ROUND(AVG(TME), 2) AS tempo_medio_entrega -- Tempo médio de entrega
(TME)

FROM `olist-sandbox-portfolio.olist_analysis.vw_dashboard_operacional`

WHERE seller_id IS NOT NULL

GROUP BY seller_id, cidade_vendedor, estado_vendedor
HAVING COUNT(order_id) >= 5    -- Apenas vendedores com pelo menos 5
pedidos
ORDER BY tempo_medio_entrega DESC
LIMIT 6;

```

---

## 11.2 Glossário Técnico

Esta seção apresenta os principais termos técnicos, métricas e siglas utilizados ao longo do projeto, padronizando o entendimento entre analistas de dados, engenheiros e demais stakeholders.

---

### 11.2.1 Termos Gerais

#### **E-commerce**

Ambiente de vendas online utilizado para comercialização de produtos ou serviços.

#### **Pedido (Order)**

Transação registrada no sistema, representando a intenção de compra de um cliente.

#### **Cliente Único**

Cliente que realizou apenas uma compra ao longo do histórico analisado.

**Cliente Recorrente**

Cliente que realizou duas ou mais compras no período analisado.

---

**11.2.2 Métricas de Negócio (Comercial)****GMV (Gross Merchandise Volume)**

Valor bruto vendido, incluindo produtos e frete.

Representa o principal indicador comercial do marketplace.

**Ticket Médio**

GMV dividido pelo número total de pedidos.

**Cancelamento**

Pedido marcado como *canceled* no fluxo operacional.

**Inadimplência Operacional**

Pedidos indisponíveis ou não confirmados após tentativa de cobrança.

---

**11.2.3 Métricas de Marketing e CRM****RFM (Recência, Frequência, Monetário)**

Modelo de segmentação comportamental que classifica clientes conforme:

- **Recência** – tempo desde a última compra
- **Frequência** – quantidade de pedidos realizados
- **Monetário** – valor total gasto

**LTV (Lifetime Value)**

Valor total que um cliente gera ao longo de todo seu ciclo no e-commerce.

**Churn**

Perda de clientes, especialmente os que não retornam após a primeira compra.

---

**11.2.4. Métricas Operacionais e Logísticas****TME (Tempo Médio de Entrega)**

Dias entre aprovação do pedido e entrega ao cliente.

**Tempo de Preparação do Vendedor**

Período entre criação do pedido e postagem do produto.

**Atraso (Indicador de Atraso)**

1 quando o pedido é entregue após a data estimada; 0 caso contrário.

**Data Estimada de Entrega**

Data máxima prevista para entrega no ato da compra.

**SLA**

Limites de desempenho (ex.: entrega, cancelamento) acordados entre operação e fornecedores.

---

### 11.2.5. Dimensões do Modelo Estrela

**Dimensão Cliente**

Informações demográficas, RFM, LTV e comportamento de compra.

**Dimensão Produto**

Categoria, nome original, peso, dimensões, características.

**Dimensão Vendedor**

Identificação, localização e velocidade média de preparação.

**Dimensão Tempo**

Datas normalizadas em granularidade diária, mensal e anual.

**Dimensão Geográfica**

Estado e cidade do cliente.

---

### 11.2.6. Termos Técnicos de Modelagem & BI

**Fato**

Tabela central do modelo estrela contendo métricas e chaves estrangeiras.

**Dimensão**

Tabela com atributos descritivos relacionados às entidades de análise.

**ETL / ELT**

Processos de carga, transformação e estruturação de dados.

**View Analítica**

Consulta materializada ou virtual utilizada para abastecer dashboards.

**Granularidade**

Nível de detalhe dos dados presentes na tabela fato.

---

### 11.2.7. SQL e Engenharia de Dados

**CTE (Common Table Expression)**

Blocos *WITH* usados para modular, organizar e padronizar consultas.

**Window Functions (Funções de Janela)**

Funções como RANK(), NTILE(), OVER(PARTITION BY) utilizadas para cálculos analíticos.

**Aggregate Functions**

Funções como SUM(), AVG(), COUNT() aplicadas sobre conjuntos de linhas.

---

### 11.2.8. Termos Estatísticos Utilizados

**Percentil / NTILE**

Classificação em segmentos proporcionais (quartis, decis, etc.).

**Correlação Operacional**

Relação entre variáveis como atraso, frete e retorno do cliente.

**Cohort Analysis**

Análise baseada no agrupamento de clientes por data de entrada.