



Fatec Ipiranga "Pastor Enéas Tognini"

Construção de Data Warehouse com ETL e Mineração de Dados Aplicados ao Delivery Projeto Integrador III

Versão 11.0

**BRUNO FERREIRA RAMOS
HELENA BATOCCHIO PINTO FLAUSINO
ISABELA REIKDAL BOMBONATTO LUGO
VITORIA CRISTINA DA SILVA COUTINHO
GABRIELA FERNANDEZ MENACHO**

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Histórico da Revisão

Data	Versão	Descrição	Autor
24/fev/2025	1.0	Tarefa 01	Grupo todo
08/abr/2025	2.0	Tarefa 06	Grupo todo
14/abr/2025	3.0	Tarefa 06 - Correções	Grupo Todo
22/abr/2025	4.0	Tarefa 06 - Correções	Grupo todo
06/mai/2025	5.0	Tarefa 08 – Gestão de Projetos	Grupo todo
12/mai/2025	6.0	Tarefa 09 – Análise de Dados com funções estatísticas	Grupo todo
19/mai/2025	7.0	Tarefa 09 - continuação	Grupo todo
26/mai/2025	8.0	Tarefa 06 - Correções	Grupo todo
17/jun/2025	9.0	Tarefa 10 - Dashboard	Grupo todo
23/jun/2025	10.0	Tarefa Análise de Dados - Programação de Banco de Dados	Grupo todo
25/jun/2025	11.0	Correções	Grupo todo

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

SUMÁRIO

1. Introdução	5
2. Objetivos da Empresa	6
2.1. Missão, Visão e Valores.....	6
2.2. Pesquisa de Mercado.....	6
2.3. Questionamentos e Insights.....	7
3. Base de dados utilizada.....	9
3.1 Descrição dos conjuntos de dados.....	10
3.2 Descrição das transformações dos dados	14
4. Gestão de Projetos	19
4.1 Estrutura Analítica do Projeto (EAP).....	19
5. Análise de Dados e Funções Estatísticas	20
6. Programação de Banco de Dados	35
7. Conclusão	37
8. Cronograma	38
9. Referências Bibliográficas.....	39
10. APÊNDICE A – Gestão de Projetos.....	40

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

ÍNDICE DE IMAGENS

Figura 1- Modelo Lógico Data Warehouse	9
Figura 2- Modelo Lógico Tabela Channels.....	10
Figura 3- Modelo Lógico Tabela Deliveries	10
Figura 4- Modelo Lógico Tabela Drivers	11
Figura 5- Modelo Lógico Tabela Hubs.....	11
Figura 6- Modelo Lógico Tabela Payments.....	11
Figura 7- Modelo Lógico Tabela Stores	12
Figura 8- Modelo Lógico Tabela Orders	13
Figura 9- Modelo EAP	19
Figura 10- Saída do código em Python mostrando qual filial tem o maior número de pedidos	20
Figura 11- Gráfico de top 10 lojas com mais pedidos.....	21
Figura 12- Saída do código em Python mostrando a duração média do ciclo de entrega	22
Figura 13- Gráfico de top 20 lojas com maior tempo médio de entrega	24
Figura 14- Saída do código em Python mostrando a porcentagem de pedidos cancelados e não cancelados de lojas com planos de fidelidade	25
Figura 15- Saída do código em Python mostrando a porcentagem de pedidos cancelados e não cancelados de lojas sem planos de assinatura.....	26
Figura 16- Saída do código em Python mostrando a porcentagem de pedidos em lojas com planos de assinatura e lojas sem planos de assinatura	Erro! Indicador não definido.
Figura 17- Gráfico de distribuição de pedidos de lojas com ou sem assinatura.....	28
Figura 18- Saída do código em Python mostrando a tabela de contingência e resultados do teste Qui-quadrado para a taxa de entregas bem-sucedidas por hub.....	31
Figura 19- Saída do código em Python mostrando resultados da frequência para entregas entregues e canceladas por hub	32
Figura 20- Gráfico de distribuição de entregas bem-sucedidas e não realizadas por hub	33
Figura 21- Dashboard	34
 Código 1- Processo ETL em Pyyhon (Configurações e Extrações)	16
Código 2- Processo ETL em Python (Transformações).....	17
Código 3-Processo ETL em Python (Carregamento no PostgreeSQL).....	17
Código 4-Processo ETL em Python (Encerramento).....	18
Código 5- Código em Python para descobrir qual filial tem o maior número de pedidos	20
Código 6- Código em Python para identificar as 10 lojas com mais pedidos.....	21
Código 7- Código em Python para identificar a duração média do ciclo de entrega	22
Código 8- Código do gráfico	23
Código 9- Código em Python para identificar porcentagem de pedidos não cancelados e cancelados quando a loja tem planos de assinatura	25
Código 10- Código em Python para identificar porcentagem de pedidos não cancelados e cancelados quando a loja sem planos de assinatura.....	26
Código 11- Código em Python para identificar se existem mais pedidos feitos em lojas com plano de assinatura ou em lojas sem plano de assinatura	27
Código 12- Código e saída no python da análise estatística	29
Código 13- Código e saída Boxplot.....	29
Código 14- Código em Python para analisar a diferença estatística na taxa de entregas bem-sucedidas entre hubs usando o teste Qui-quadrado	30
Código 15- Código em PL pg/QGL para analisar a soma das entregas por mês.....	35
Código 16- Código em PL pg/QGL para garantir a atualização na tabela stores	36
 Tabela 1- Cronograma-.....	38
Tabela 2- Declaração do Problema.....	40
Tabela 3- Declaração do Produto.....	40
Tabela 4 - Declaração dos Recursos	40

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

1. Introdução

O presente projeto integrador tem como objetivo o desenvolvimento de um Data Warehouse e a aplicação de técnicas de Mineração de Dados (Data Mining), utilizando recursos de Inteligência Artificial e Estatística, com a finalidade de gerar informações em níveis estratégicos para a tomada de decisão. A proposta contempla a compreensão e a aplicação prática de técnicas de ETL (Extração, Transformação e Carga), com uso de programação voltada a bancos de dados relacionais, além da construção de modelos dimensionais e da aplicação de métodos analíticos que possibilitem a descoberta de padrões e tendências relevantes.

A metodologia de ETL consiste na integração de dados em três etapas fundamentais: extração de informações de fontes diversas, transformação desses dados em um formato apropriado para análise, e carregamento em um repositório central — o Data Warehouse. Esse processo é essencial para consolidar dados e permitir análises eficazes que auxiliem na geração de conhecimento e no apoio à gestão estratégica.

Este projeto será aplicado ao EMPÓRIO QUEL, um pequeno comércio localizado no bairro do Jabaquara, em São Paulo, especializado na venda de produtos alimentícios selecionados. A loja se destaca pelo compromisso com a qualidade, oferecendo itens frescos e cuidadosamente escolhidos, incluindo temperos naturais, queijos artesanais, doces tradicionais, castanhas e amendoins. A partir da construção do Data Warehouse e da implementação de *dashboards* analíticos, será possível obter *insights* valiosos sobre o desempenho do negócio, contribuindo para uma gestão mais eficiente e orientada por dados.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

2. Objetivos da Empresa

O objetivo da EMPÓRIO QUEL é manter um atendimento de qualidade e acessível, sempre priorizando a proximidade com nossos clientes. Para melhorar ainda mais seus serviços, estão trabalhando para implementar um sistema *online* que permitirá registrar e gerenciar nossos produtos, vendas e valores de forma eficiente. Essa iniciativa visa otimizar o controle do estoque e as operações diárias, garantindo maior transparência e agilidade tanto para nós quanto para nossos clientes.

Pretendemos incluir algumas metas específicas com a implementação do novo sistema, como reduzir o tempo de processamento de pedidos ou aumentar a precisão do controle de estoque. Isso pode ajudar a comunicar claramente os benefícios da modernização.

2.1. Missão, Visão e Valores

A visão da administração do EMPÓRIO QUEL é torná-lo uma referência na comunidade, reconhecido pela qualidade excepcional dos nossos produtos e atendimento, para ser o destino preferido da clientela para comprar alimentos artesanais e produtos de alta qualidade.

A missão é oferecer aos clientes uma seleção única de produtos frescos, artesanais e locais, criando um ambiente acolhedor e amigável. Priorizam a qualidade e o atendimento personalizado e buscam crescer e expandir para alcançar um público ainda maior.

Os valores que guiam a empresa incluem o compromisso com a qualidade, garantindo excelência em todos os produtos e serviços que oferecem; a integridade, assegurando transparência e ética em todas as relações comerciais; e o atendimento, focado em oferecer um serviço atencioso, personalizado e de alta qualidade, com preços justos e bom atendimento, criando um ambiente onde cada cliente se sente em casa. Valorizado a confiança e a proximidade com os clientes.

2.2. Pesquisa de Mercado

O mercado de empórios e mercearias tem experimentado um crescimento significativo nos últimos anos, impulsionado pela demanda dos consumidores por produtos frescos, de qualidade e com características artesanais. A crescente preocupação com a alimentação saudável e a origem dos alimentos tem favorecido esse segmento, onde pequenos negócios se destacam pela oferta personalizada e pela atenção aos detalhes.

Além disso, mais de 80% dos produtores de alimentos artesanais e tradicionais recomendam essa atividade para quem ainda não está no segmento, segundo pesquisa da Confederação da Agricultura e Pecuária do Brasil (CNA). Entre abril e junho, a CNA constatou que 64% dos participantes da pesquisa já produzem alimentos artesanais, enquanto 38% ainda não trabalham com esses produtos, mas demonstram interesse. A maior parte desses produtores está concentrada na região Sudeste, com 35,5% dos entrevistados.

O setor tem se mostrado promissor, com uma taxa de crescimento anual composta global de 28% nos últimos cinco anos, conforme aponta a pesquisa Innova Market Insights citada pela Ingredients Network. Esse crescimento reflete o aumento da aceitação dos consumidores e o apoio de empresas que investem em produtos artesanais por meio de fusões, aquisições ou desenvolvimento interno.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

A combinação de produtos artesanais, a conexão com as tendências de mercado e a busca por alimentos saudáveis posicionam o EMPÓRIO QUEL de forma estratégica nesse cenário em expansão. Nesse contexto, o EMPÓRIO QUEL surge como uma opção diferenciada para os consumidores que buscam não apenas produtos de alta qualidade, mas também uma experiência de compra única e personalizada.

O EMPÓRIO QUEL, localizado em São Paulo, é uma microempresa que se especializa na venda de produtos alimentícios naturais e artesanais. Fundada em 2017, a empresa se destaca por sua vasta gama de produtos, que inclui desde grãos, carnes secas, massas, temperos artesanais, e doces artesanais. A qualidade dos produtos, com foco em ingredientes naturais e processos artesanais, é um dos principais atrativos.

A forte presença nas redes sociais tem sido uma ferramenta eficaz, permitindo uma comunicação direta com seus consumidores e a divulgação eficiente de seus produtos. Além disso, o serviço de *delivery* amplia o alcance, facilitando o acesso dos clientes aos produtos e contribuindo para a fidelização. No entanto, o EMPÓRIO QUEL enfrenta desafios devido à concorrência com o Empório Stillus, um estabelecimento localizado nas proximidades e que compartilha um perfil de cliente semelhante.

Embora essa concorrência direta represente uma ameaça, ela também oferece oportunidades para se diferenciar e explorar seus pontos fortes, como a variedade e a qualidade dos produtos artesanais, para atrair e reter clientes. A ênfase na experiência de compra personalizada e no atendimento especializado pode ser um diferencial importante. Além disso, o investimento em marketing digital é fundamental para o estabelecimento, com a criação de sites e a utilização de redes sociais para aumentar a visibilidade e atrair novos clientes.

2.3. Questionamentos e Insights

Inicialmente, procuramos identificar qual "hub" (ponto de distribuição ou área de atendimento) apresenta o menor tempo médio entre o recebimento do pedido e sua expedição para entrega. Também é importante compreender como esse tempo varia ao longo das diferentes faixas horárias do dia. Essa análise detalhada possibilita a detecção de gargalos operacionais específicos do contexto de *delivery* e a identificação dos períodos de maior eficiência, informações cruciais para aprimorar a logística de um empório.

Além disso, é fundamental determinar quais hubs registram as maiores taxas de atraso nas entregas e quais fatores estão mais fortemente correlacionados com esses atrasos. No contexto de um empório, isso pode incluir variáveis como o volume de pedidos, a distância média de entrega, o dia da semana e até mesmo a modalidade de entrega (motocicleta, bicicleta, carro). Para isso, torna-se necessário calcular a proporção de entregas atrasadas por hub e relacioná-la com as variáveis operacionais mencionadas, empregando técnicas de análise de correlação ou regressão.

Outro aspecto crucial é a avaliação do custo médio por entrega em cada hub e sua relação com o volume de pedidos e a área de cobertura atendida. Para um empório, essa análise pode ajudar a definir estratégias de precificação de entrega, identificar áreas de maior rentabilidade e otimizar rotas de entrega. Essa análise demanda o levantamento dos custos operacionais totais de cada hub e sua divisão pelo número de entregas realizadas, permitindo uma avaliação da eficiência em função da escala e da abrangência territorial.

Quanto à cobertura e capacidade dos hubs, é importante identificar aqueles que atendem

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

à maior área geográfica e calcular a densidade de pedidos por quilômetro quadrado atendido. Essa análise possibilita destacar os hubs com maior concentração de demanda, informação valiosa para o planejamento de expansão ou otimização da área de atendimento do empório.

Também se investiga a ocorrência de sobrecarga em determinados hubs, ou seja, se o volume de pedidos excede a média histórica, e se essa sobrecarga impacta o tempo de entrega. Para tal, pode-se utilizar o desvio em relação à média (z-score) para detectar picos de demanda e verificar seu efeito no tempo de processamento.

Buscamos estabelecer se existe alguma relação entre o tempo de entrega e a taxa de devoluções, e a categoria do produto entregue. Essa análise comparativa dos indicadores logísticos entre diferentes categorias de produtos visa identificar padrões de desempenho e desafios específicos associados a cada categoria, como, por exemplo, as exigências logísticas mais complexas para produtos perecíveis, que demandam entregas mais rápidas e cuidados especiais com a temperatura.

Outro ponto relevante é a análise do comportamento dos hubs em datas especiais. O objetivo é avaliar o desempenho de cada hub nesses períodos de alta demanda e identificar aqueles que demonstram maior capacidade de adaptação e manutenção da estabilidade logística. Para um empório, essa análise é crucial para o planejamento de ações promocionais e para garantir a qualidade do serviço em períodos de pico.

Por fim, almejamos determinar quais hubs apresentam maior previsibilidade no volume de pedidos ao longo das semanas, ou seja, aqueles com menor variância e maior padrão de comportamento. A utilização de séries temporais e métricas de variabilidade se mostra útil nesse sentido, permitindo identificar os hubs com comportamento mais estável e, assim, otimizar o planejamento logístico e a alocação de recursos do empório.

Além disso, investiga-se a influência do desempenho logístico do hub na taxa de recompra dos clientes, buscando associações entre a eficiência operacional (tempo de entrega, atrasos, devoluções) e o comportamento do cliente, de modo a avaliar se a qualidade do serviço prestado impacta na fidelização e na satisfação do cliente do empório.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

3. Base de dados utilizada

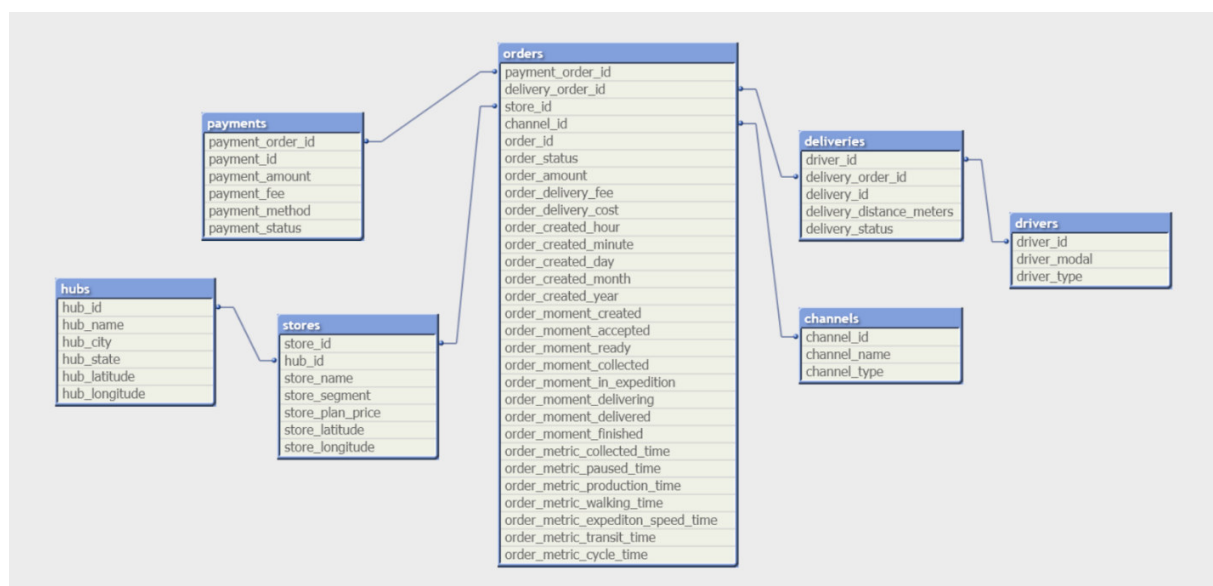
Para este projeto, será utilizada uma base de dados pública disponibilizada na plataforma Kaggle, intitulada "Delivery Center: Food & Goods orders in Brazil". Essa escolha se justifica pelo fato de o EMPÓRIO QUEL ainda não dispor de um volume de dados suficientemente robusto para a construção de um Data Warehouse, o que exige uma grande quantidade de registros para garantir a geração de análises consistentes e estratégicas. A base está acessível por meio do seguinte *link*: <https://www.kaggle.com/datasets/nosbielcs/brazilian-delivery-center>.

A base em questão se mostrou adequada ao escopo do projeto por conter dados relacionados a entregas de alimentos, o que dialoga diretamente com a proposta do EMPÓRIO QUEL. Ainda que os produtos não sejam exatamente os mesmos, o modelo de operação – baseado em pedidos e entregas no setor alimentício – apresenta forte compatibilidade com a realidade da empresa.

A fonte dos dados é o *Delivery Center*, uma plataforma que conecta lojistas a marketplaces, promovendo um ecossistema integrado de vendas e entregas no varejo brasileiro, tanto de produtos alimentícios. Os registros abrangem o período de 2020 a 2021.

O conjunto de dados contempla informações detalhadas sobre pedidos, pagamentos, lojas, entregas, canais de venda, entregas e centros logísticos, oferecendo uma estrutura rica e diversificada para a construção do *Data Warehouse* e a realização das análises pretendidas.

Figura 1- Modelo Lógico Data Warehouse



Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

3.1 Descrição dos conjuntos de dados

O conjunto de dados utilizado neste projeto é intitulado "Delivery Center: Food & Goods orders in Brazil", disponível publicamente na plataforma Kaggle. Ele reúne informações detalhadas sobre o processo de pedidos e entregas de bens e alimentos no Brasil, provenientes da plataforma Delivery Center, que atua como integradora entre lojistas e marketplaces.

A base é composta por múltiplos arquivos em formato .csv, que representam diferentes aspectos do ecossistema logístico e comercial. Entre os principais dados disponíveis, destacam-se:

Figura 2– Modelo Lógico Tabela Channels

Channels:



Possui informações sobre os canais de venda que é a identificação da origem do pedido (por exemplo, aplicativo, site, etc..).

Fonte: (Própria,2025)

Deliveries:

Figura 3– Modelo Lógico Tabela Deliveries



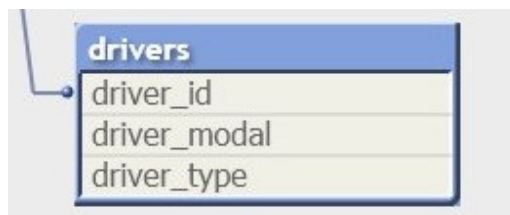
Onde constam informações sobre as entregas realizadas como o tempo de coleta e entrega, distância percorrida, status da entrega e avaliações.

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Drivers:

Figura 4– Modelo Lógico Tabela Drivers



Este *dataset* possui registros relacionados aos entregadores, como tempo de serviço e número de entregas realizadas.

Fonte: (Própria,2025)

Hubs:

Figura 5– Modelo Lógico Tabela Hubs



Encontram-se informações com a localização e atuação dos centros responsáveis pelo gerenciamento e distribuição dos pedidos do Delivery Center. Entenda que os Hubs são os centros de distribuição dos pedidos e é dali que saem as entregas.

Fonte: (Própria,2025)

Payments:

Figura 6– Modelo Lógico Tabela Payments



Nesta tabela, existem informações como detalhes sobre os métodos de pagamento utilizados e os valores pagos ao Delivery Center.

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Stores:

Figura 7– Modelo Lógico Tabela Stores



Este *dataset* possui informações sobre os lojistas como dados cadastrais das lojas participantes, incluindo localização. Eles utilizam a Plataforma do Delivery Center para vender seus itens (good e/ou food) nos marketplaces.

Fonte: (Própria,2025)

Orders:

Figura 8– Modelo Lógico Tabela Orders

orders
payment_order_id
delivery_order_id
store_id
channel_id
order_id
order_status
order_amount
order_delivery_fee
order_delivery_cost
order_created_hour
order_created_minute
order_created_day
order_created_month
order_created_year
order_moment_created
order_moment_accepted
order_moment_ready
order_moment_collected
order_moment_in_expedition
order_moment_delivering
order_moment_delivered
order_moment_finished
order_metric_collected_time
order_metric_paused_time
order_metric_production_time
order_metric_walking_time
order_metric_expediton_speed_time
order_metric_transit_time
order_metric_cycle_time

Esta tabela traz informações, como identificador do pedido, data e hora da criação, status, valor total e identificadores de clientes e lojas sobre as vendas processadas através da plataforma do Delivery Center.

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

3.2 Descrição das transformações dos dados

O objetivo é a construção de um Data Warehouse (DW) a partir de uma base de dados composta por arquivos CSV, representando as operações de um empório, incluindo pedidos, entregas, pagamentos, lojas, hubs e canais de venda.

O modelo de dados implementado foi o Esquema em Floco de Neve (Snowflake Schema). Neste modelo, uma única tabela fato principal (fato_orders) é relacionada a múltiplas tabelas dimensionais (dim_drivers, dim_deliveries, dim_channels, dim_payments, dim_stores, dim_hubs) de forma normalizada. Esta estrutura promove maior controle de consistência, evita redundância e melhora a organização das análises de dados.

O pipeline de ETL foi dividido em três etapas fundamentais:

- **Extract (Extração):** os dados são extraídos de arquivos .csv organizados em uma pasta local, com tratamento de codificação adequada para leitura correta de caracteres especiais.
- **Transform (Transformação):** cada dataset passa por uma função específica de transformação, que inclui padronização de colunas e valores, conversão de tipos, tratamento de dados ausentes, limpeza textual e adequação às necessidades do modelo dimensional.
- **Load (Carga):** os dados transformados são carregados para o banco PostgreSQL utilizando SQLAlchemy e o método `pandas.to_sql()` com o parâmetro `if_exists='replace'`, sobrescrevendo as tabelas no banco de destino.

O processo de Transformação, etapa intermediária e fundamental dentro do fluxo de ETL, é responsável por tratar, limpar e padronizar os dados brutos extraídos, adequando-os à modelagem proposta no DW. Esta etapa garante que os dados sejam consistentes, coerentes e prontos para consumo analítico.

A seguir, detalhamos as principais decisões de transformação tomadas por tabela:

Tabela drivers

Contém dados sobre os entregadores. As colunas relevantes incluem:

- **driver_id:** identificador único do entregador. Não será alterado pois atua como chave primária e estrangeira.
- **driver_modal:** meio de transporte utilizado, padronizado com `.upper().strip()` e validado para conter apenas valores esperados (BIKER, MOTOBOY).
- **driver_type:** tipo de vínculo do entregador (freelancer ou operador logístico), também padronizado.

Tabela channels

Relaciona os canais de origem do pedido. Suas transformações incluem:

- Remoção de espaços em `channel_name`.
- Padronização de `channel_type` com `.upper().strip()` garantindo valores como OWN e MARKETPLACE.

Tabela deliveries

Abrange informações logísticas das entregas:

- **delivery_distance_meters:** convertido com `pd.to_numeric(..., errors='coerce')` para

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

permitir cálculos e análises de distância.

- driver_id: valores nulos substituídos por "SEM_ENTREGADOR", representando retiradas presenciais pelo cliente.
- delivery_status: padronização textual com .upper().strip().

Tabela payments

Trata os pagamentos realizados:

- payment_amount e payment_fee: convertidos para tipo NUMERIC com duas casas decimais.
- payment_method e payment_status: padronizados para caixa alta e sem espaços extras.

Tabela stores

Registra as lojas parceiras do empório:

- store_plan_price: valores nulos mantidos e campo convertido para monetário.
- store_latitude e store_longitude: convertidos para NUMERIC(10,8) com precisão geográfica.
- store_segment: valores como FOOD ou GOOD padronizados para garantir integridade nas análises.

Tabela hubs

Armazena informações dos centros de distribuição:

- Campos de localização (hub_latitude, hub_longitude) convertidos com precisão.
- hub_state e hub_city normalizados para caixa alta e sem espaços desnecessários.

Tabela orders (Fato)

Principal tabela de análise do modelo:

- Contém valores monetários como order_amount, order_delivery_fee e order_delivery_cost, convertidos para NUMERIC(10,2).
- order_moment_created: convertido para tipo datetime, com errors='coerce' para evitar interrupções por dados inválidos (registrados como NaT e posteriormente gravados como NULL no banco).
- Outras colunas derivadas de data foram descartadas por não serem necessárias neste escopo analítico.

Essa estrutura permite uma análise eficiente ao cruzar métricas de pedidos com entregas, canais, pagamentos, localizações e tipos de loja.

Todas essas transformações foram aplicadas de forma sistemática e automatizada no script ETL, detalhado na próxima seção deste documento.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Etapas detalhadas e script Python com comentários:

Código 1- Processo ETL em Pyyhon (Configurações e Extrações)

```

1  import pandas as pd
2  from sqlalchemy import create_engine
3  import os
4
5  # -----
6  # CONFIGURAÇÃO
7  # -----
8  # define o caminho local dos arquivos CSV e os parâmetros de conexão com o banco PostgreSQL
9  CAMINHO_CSV = r'D:\Fatec\03.2025\Projeto Integrador III\tabelas csv'
10 DB_CONFIG = {
11     'usuario': 'postgres',
12     'senha': '1234',
13     'host': 'localhost',
14     'porta': '5432',
15     'banco': '20251_fatec_pi_iii'
16 }
17
18 # -----
19 # EXTRACT
20 # -----
21 # função para ler arquivos CSV utilizando codificação latin1 para evitar erros de acentuação
22 # retorna um DataFrame com os dados brutos de cada arquivo
23 def ler_csv(nome_arquivo):
24     caminho_completo = os.path.join(CAMINHO_CSV, nome_arquivo)
25     df = pd.read_csv(caminho_completo, encoding='latin1')
26     return df
27
28 # -----
29 # TRANSFORM
30 # -----
31 # padroniza os nomes das colunas removendo espaços, colocando letras minúsculas e usando underscore
32 def padronizar_colunas(df):
33     df.columns = [col.lower().strip().replace(' ', '_') for col in df.columns]
34     return df
35
36 # transforma a tabela de entregadores: padroniza os textos e filtra os modais esperados
37 def transformar_drivers(df):
38     df['driver_modal'] = df['driver_modal'].str.upper().str.strip()
39     df['driver_type'] = df['driver_type'].str.upper().str.strip()
40     df = df[df['driver_modal'].isin(['BIKER', 'MOTOBOY'])] # mantém apenas registros válidos
41     return df

```

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Código 2– Processo ETL em Python (Transformações)

```

42
43 # transforma a tabela de canais de venda: normaliza os textos
44 def transformar_channels(df):
45     df['channel_name'] = df['channel_name'].str.strip()
46     df['channel_type'] = df['channel_type'].str.upper().str.strip()
47     return df
48
49 # transforma a tabela de entregas: converte valores, trata dados ausentes e padroniza o status
50 def transformar_deliveries(df):
51     df['delivery_distance_meters'] = pd.to_numeric(df['delivery_distance_meters'], errors='coerce').round(2)
52     df['driver_id'] = df['driver_id'].fillna('SEM_ENTREGADOR')
53     df['delivery_status'] = df['delivery_status'].str.upper().str.strip()
54     return df
55
56 # transforma a tabela de pagamentos: converte valores e padroniza texto
57 def transformar_payments(df):
58     df['payment_amount'] = pd.to_numeric(df['payment_amount'], errors='coerce').round(2)
59     df['payment_fee'] = pd.to_numeric(df['payment_fee'], errors='coerce').round(2)
60     df['payment_method'] = df['payment_method'].str.upper().str.strip()
61     df['payment_status'] = df['payment_status'].str.upper().str.strip()
62     return df
63
64 # transforma a tabela de lojas: converte coordenadas com alta precisão e valores financeiros
65 def transformar_stores(df):
66     df['store_plan_price'] = pd.to_numeric(df['store_plan_price'], errors='coerce').round(2)
67     df['store_latitude'] = pd.to_numeric(df['store_latitude'], errors='coerce').round(8)
68     df['store_longitude'] = pd.to_numeric(df['store_longitude'], errors='coerce').round(8)
69     df['store_segment'] = df['store_segment'].str.upper().str.strip()
70     return df
71
72 # transforma os hubs: trata coordenadas geográficas e padroniza localização
73 def transformar_hubs(df):
74     df['hub_latitude'] = pd.to_numeric(df['hub_latitude'], errors='coerce').round(8)
75     df['hub_longitude'] = pd.to_numeric(df['hub_longitude'], errors='coerce').round(8)
76     df['hub_state'] = df['hub_state'].str.upper().str.strip()
77     df['hub_city'] = df['hub_city'].str.strip()
78     return df

```

Fonte: (Própria,2025)

Código 3–Processo ETL em Python (Carregamento no PostgreSQL)

```

79
80 # transforma a tabela fato de pedidos: converte valores e datas, remove colunas derivadas
81 def transformar_orders(df):
82     df['order_amount'] = pd.to_numeric(df['order_amount'], errors='coerce').round(2)
83     df['order_delivery_fee'] = pd.to_numeric(df['order_delivery_fee'], errors='coerce').round(2)
84     df['order_delivery_cost'] = pd.to_numeric(df['order_delivery_cost'], errors='coerce').round(2)
85     df['order_moment_created'] = pd.to_datetime(df['order_moment_created'], errors='coerce') # NaT vira NULL
86     df['order_status'] = df['order_status'].str.upper().str.strip()
87     colunas_remove = [col for col in df.columns if 'created_' in col or ('moment_' in col and col != 'order_moment_created')]
88     df = df.drop(columns=[col for col in colunas_remove if col in df.columns])
89     return df
90
91 # -----
92 # LOAD
93 # -----
94 # cria engine de conexão com o banco de dados PostgreSQL
95 def conectar_postgres():
96     config = DB_CONFIG
97     return create_engine(f"postgresql+psycopg2://{config['usuario']}:{config['senha']}@{config['host']}:{config['porta']}/{config['banco']}")
98
99 # carrega o DataFrame para o PostgreSQL, substituindo a tabela existente
100 def salvar_tabela(df, nome_tabela, engine):
101     df.to_sql(nome_tabela, con=engine, if_exists='replace', index=False)
102     print(f"Tabela '{nome_tabela}' carregada com sucesso.")
103

```

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Código 4-Processo ETL em Python (Encerramento)

```

104 # -----
105 # MAIN
106 # -----
107 # função principal que coordena todas as etapas do ETL
108 def main():
109     print("Iniciando ETL completo com extração, transformação e carga de dados:")
110
111     # extração e transformação de cada conjunto de dados
112     channels = transformar_channels(padronizar_colunas(ler_csv('channels.csv')))
113     deliveries = transformar_deliveries(padronizar_colunas(ler_csv('deliveries.csv')))
114     drivers = transformar_drivers(padronizar_colunas(ler_csv('drivers.csv')))
115     hubs = transformar_hubs(padronizar_colunas(ler_csv('hubs.csv')))
116     orders = transformar_orders(padronizar_colunas(ler_csv('orders.csv')))
117     payments = transformar_payments(padronizar_colunas(ler_csv('payments.csv')))
118     stores = transformar_stores(padronizar_colunas(ler_csv('stores.csv')))
119
120     # conexão com banco e carga final das tabelas
121     engine = conectar_postgres()
122     salvar_tabela(channels, 'dim_channels', engine)
123     salvar_tabela(deliveries, 'dim_deliveries', engine)
124     salvar_tabela(drivers, 'dim_drivers', engine)
125     salvar_tabela(hubs, 'dim_hubs', engine)
126     salvar_tabela(orders, 'fato_orders', engine)
127     salvar_tabela(payments, 'dim_payments', engine)
128     salvar_tabela(stores, 'dim_stores', engine)
129
130     print("Processo ETL finalizado com sucesso e dados carregados no PostgreSQL.")
131
132 # executa o ETL apenas quando este arquivo for executado diretamente
133 if __name__ == '__main__':
134     main()

```

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

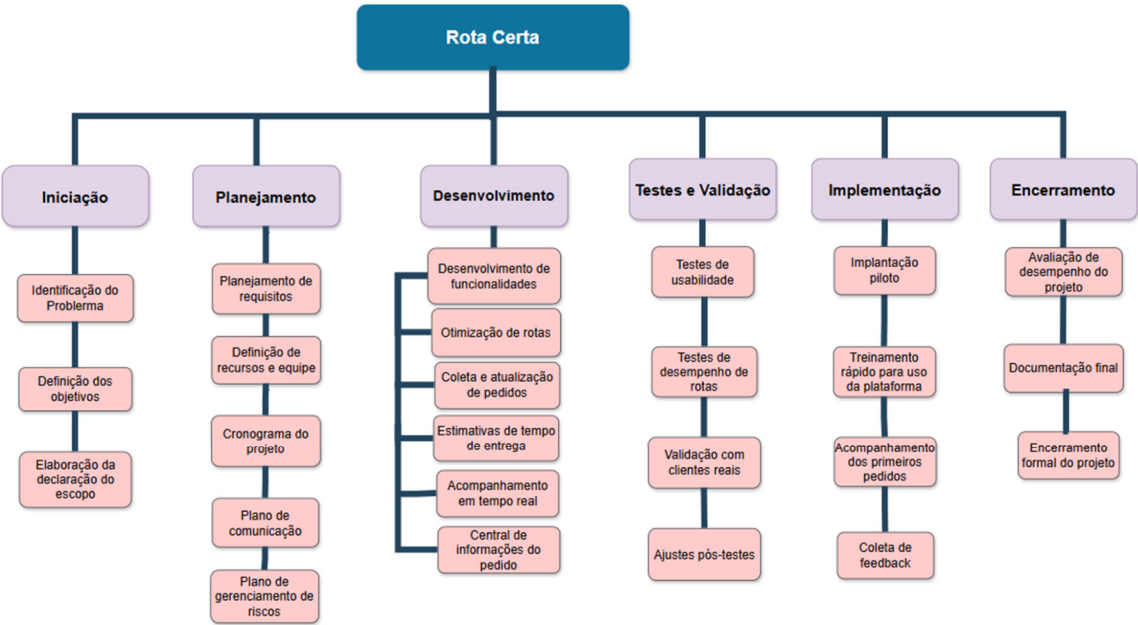
4. Gestão de Projetos

4.1 Estrutura Analítica do Projeto (EAP)

A Estrutura Analítica do Projeto (EAP) é uma ferramenta essencial na gestão de projetos que tem como objetivo principal dividir o escopo total do projeto em partes menores e mais gerenciáveis. Essa decomposição hierárquica permite organizar e visualizar melhor todas as entregas e tarefas necessárias para a conclusão do projeto, facilitando o planejamento, a execução e o controle das atividades.

A EAP serve como base para a definição do cronograma, da estimativa de custos, da alocação de recursos e do acompanhamento do progresso, uma vez que permite identificar claramente todas as etapas e sub-etapas envolvidas. Ao estruturar o projeto dessa forma, torna-se possível distribuir responsabilidades de forma mais eficiente, prever riscos com maior precisão e monitorar o andamento das atividades com mais clareza. Dessa maneira, a EAP contribui significativamente para que o projeto atinja seus objetivos dentro dos prazos e custos previstos, promovendo maior organização, controle e eficiência ao longo de toda a sua execução.

Figura 9– Modelo EAP



Fonte: (Própria, 2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

5. Análise de Dados e Funções Estatísticas

Questionamento 1:

Qual filial tem o maior número de pedidos? Identificar a filial com mais pedidos ajuda a entender quais locais têm maior demanda, auxiliando no planejamento de expansão, investimentos ou ações de marketing específicas.

Código 5- Código em Python para descobrir qual filial tem o maior número de pedidos

```

pip install kaggle
import os
import json

kaggle_token = {
    "username": "vitoriacoutinho15",
    "key": "03271912fa073b00413398859138186e"
}
os.makedirs(os.path.expanduser("~/kaggle"), exist_ok=True)
with open(os.path.expanduser("~/kaggle/kaggle.json"), "w") as f:
    json.dump(kaggle_token, f)

os.chmod(os.path.expanduser("~/kaggle/kaggle.json"), 0o600)
!kaggle datasets download -d nosbielcs/brazilian-delivery-center

import os
os.listdir()
import zipfile

with zipfile.ZipFile("brazilian-delivery-center.zip", "r") as zip_ref:
    zip_ref.extractall("dados_delivery")

import pandas as pd
stores = pd.read_csv("dados_delivery/stores.csv", encoding='latin1')
stores.head(100)
contagem = orders["store_id"].value_counts()
mais_frequente = contagem.idxmax()
frequencia = contagem.max()
print(f"Store_id mais frequente: {mais_frequente} (aparece {frequencia} vezes)")

```

Fonte: (Própria, 2025)

Figura 10- Saída do código em Python mostrando qual filial tem o maior número de pedidos

```
Store_id mais frequente: 53 (aparece 14004 vezes)
```

Fonte: (Própria, 2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

A partir do código usando a função *max*, foi possível identificar qual loja tem o maior número de pedidos no Data Warehouse, sendo essa a loja com o ID de número 53, aparecendo 14004 vezes.

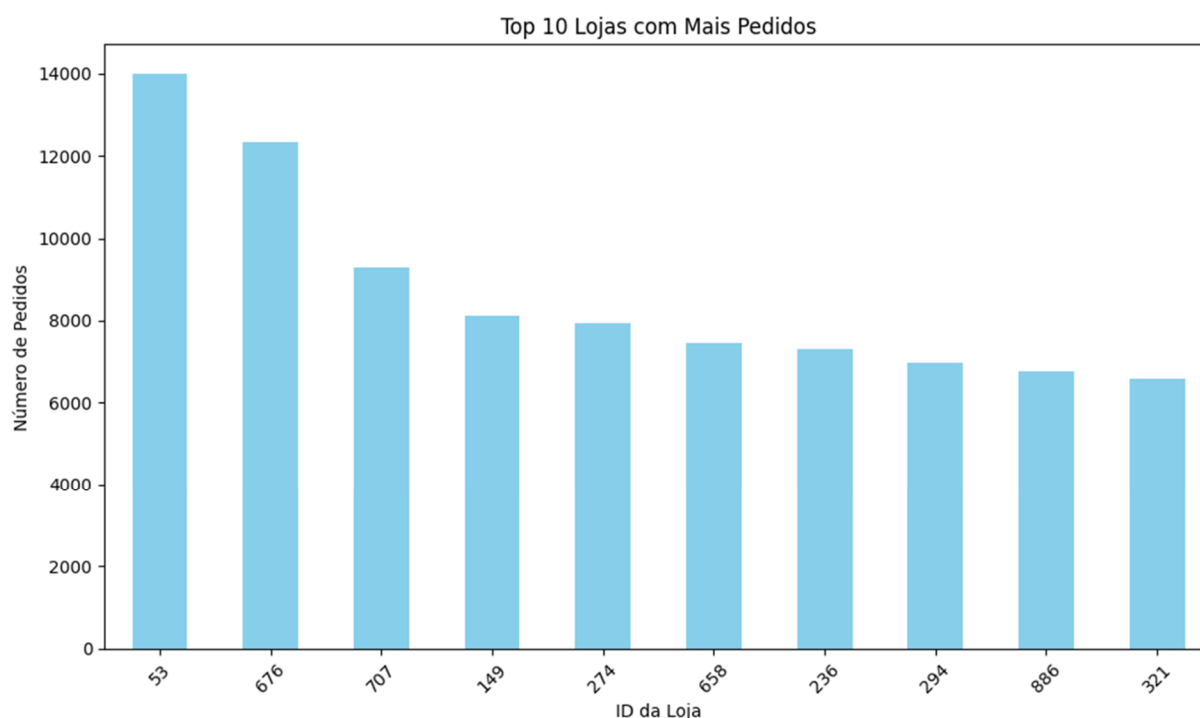
Código 6- Código em Python para identificar as 10 lojas com mais pedidos

```
import matplotlib.pyplot as plt

contagem = orders["store_id"].value_counts().head(10)
plt.figure(figsize=(10, 6))
contagem.plot(kind="bar", color="skyblue")
plt.title("Top 10 Lojas com Mais Pedidos")
plt.xlabel("ID da Loja")
plt.ylabel("Número de Pedidos")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Fonte: (Própria,2025)

Figura 11– Gráfico de top 10 lojas com mais pedidos



Fonte: (Própria, 2025)

A partir do gráfico, foi possível visualizar, além da com o maior número de pedidos, outras 9 lojas com mais pedidos.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Questionamento 2:

Qual a duração média do ciclo de entrega? Identificar a duração média do ciclo de entrega pode ajudar a tomar decisões baseadas em dados para melhorar rotas, fornecedores ou métodos de entrega.

Código 7– Código em Python para identificar a duração média do ciclo de entrega

```

pip install kaggle
import os
import json

kaggle_token = {
    "username": "vitoriacoutinho15",
    "key": "03271912fa073b00413398859138186e"
}
os.makedirs(os.path.expanduser("~/kaggle"), exist_ok=True)
with open(os.path.expanduser("~/kaggle/kaggle.json"), "w") as f:
    json.dump(kaggle_token, f)

os.chmod(os.path.expanduser("~/kaggle/kaggle.json"), 0o600)

!kaggle datasets download -d nosbielcs/brazilian-delivery-center

import os
os.listdir()
import zipfile

with zipfile.ZipFile("brazilian-delivery-center.zip", "r") as zip_ref:
    zip_ref.extractall("dados_delivery")

import pandas as pd

orders = pd.read_csv("dados_delivery/orders.csv")
orders.head(100)
media = orders['order_metric_transit_time'].mean()
print(f"Tempo médio de entrega de ciclo de entrega: {media} minutos.")

```

Fonte: (Própria,2025)

Figura 12– Saída do código em Python mostrando a duração média do ciclo de entrega

```

Tempo médio de entrega de ciclo de entrega: 46.84963694330628 minutos.

```

Fonte: (Própria, 2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

A partir do código, foi identificado a partir da função mean (média) da biblioteca Pandas que a duração média do ciclo de entrega é de aproximadamente 45 e 47 minutos.

Código 8- Código do gráfico

```
import matplotlib.pyplot as plt
import pandas as pd

media_geral = orders['order_metric_transit_time'].mean()

media_por_loja = orders.groupby('store_id')['order_metric_transit_time'].mean()

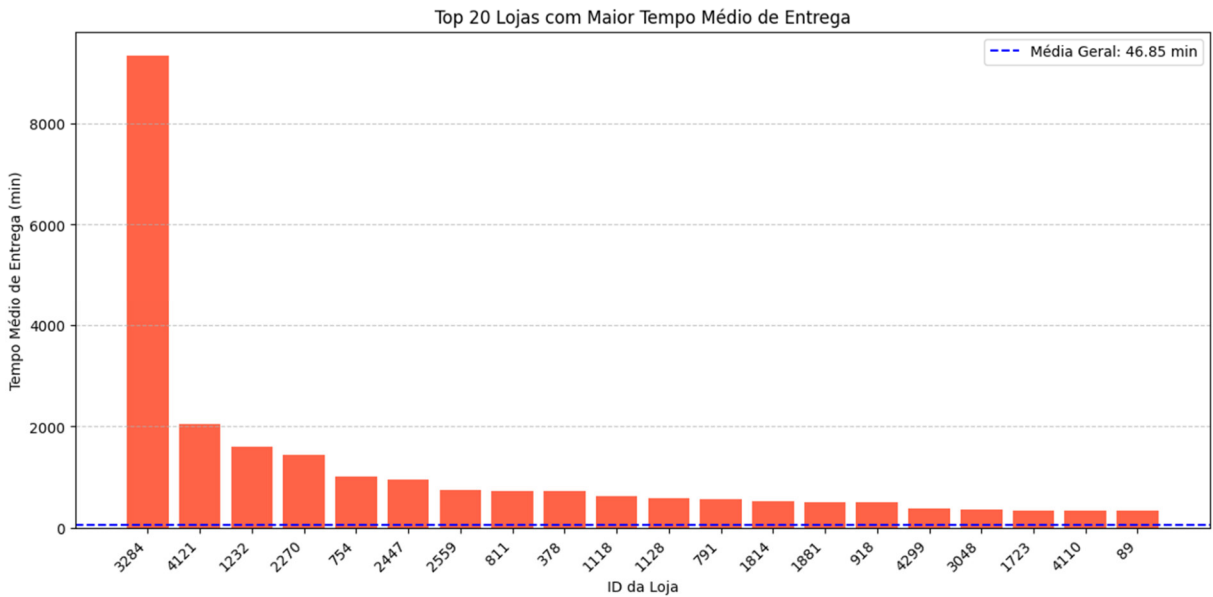
top_20_lojas = media_por_loja.sort_values(ascending=False).head(20)

plt.figure(figsize=(12, 6))
bars = plt.bar(top_20_lojas.index.astype(str), top_20_lojas.values,
color='tomato')
plt.axhline(media_geral, color='blue', linestyle='--', label=f'Média Geral:
{media_geral:.2f} min')

plt.xticks(rotation=45, ha='right')
plt.xlabel('ID da Loja')
plt.ylabel('Tempo Médio de Entrega (min)')
plt.title('Top 20 Lojas com Maior Tempo Médio de Entrega')
plt.legend()
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Fonte: (Própria,2025)

Figura 13– Gráfico de top 20 lojas com maior tempo médio de entrega



Fonte: (Própria, 2025)

A partir do gráfico, é possível visualizar as lojas que precisam de mais atenção quando se trata do tempo de entrega, concluindo que a com o maior tempo é a loja de ID 3284.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Questionamento 3:

Os pedidos são mais cancelados ou menos cancelados quando a loja tem planos de assinatura? Saber se os pedidos cancelados são mais frequentes em mercados sem plano de assinatura pode ser útil pois se houver uma correlação, a empresa pode concluir que lojas com planos de assinatura oferecem uma experiência mais confiável (ou não).

Código 9- Código em Python para identificar porcentagem de pedidos não cancelados e cancelados quando a loja tem planos de assinatura

```
from google.colab import files
uploaded = files.upload()
import pandas as pd

df = pd.read_csv('orders_stores.csv')
df.head()

com_plano_nao_cancelados = df[(df['order_status'] != 'CANCELED') &
(df['store_plan_price'] > 0)]
i = com_plano_nao_cancelados.shape[0]
com_plano_cancelados = df[(df['order_status'] == 'CANCELED') &
(df['store_plan_price'] > 0)]
j = com_plano_cancelados.shape[0]
print(f"Lojas com planos de fidelidade - Pedidos não cancelados: {i}. Pedidos cancelados: {j}.")

total_planos = i + j
porcentagem_nao_cancelados = (100 * i) / total_planos
porcentagem_cancelados = 100 - porcentagem_nao_cancelados
print(f'Valor total de pedidos com planos: {total_planos}. Porcentagem nao cancelada: {porcentagem_nao_cancelados} / Porcentagem cancelada: {porcentagem_cancelados}.')
```

Fonte: (Própria,2025)

Figura 14– Saída do código em Python mostrando a porcentagem de pedidos cancelados e não cancelados de lojas com planos de fidelidade

```
Lojas com planos de fidelidade - Pedidos não cancelados: 179652. Pedidos cancelados: 12805.
Valor total de pedidos com planos: 192457. Porcentagem nao cancelada: 93.34656572637005 / Porcentagem cancelada: 6.653434273629955.
```

Fonte: (Própria,2025)

A partir do código, foi identificado que as lojas com planos de fidelidade têm 179652 pedidos não cancelados, resultando em aproximadamente 93% e 12805 pedidos cancelados, resultando em aproximadamente 6%.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Código 10– Código em Python para identificar porcentagem de pedidos não cancelados e cancelados quando a loja sem planos de assinatura

```
from google.colab import files
uploaded = files.upload()
import pandas as pd

df = pd.read_csv('orders_stores.csv')
df.head()

sem_plano_nao_cancelados = df[(df['order_status'] != 'CANCELED') &
(df['store_plan_price'] == 0)]
x = sem_plano_nao_cancelados.shape[0]
sem_plano_cancelados = df[(df['order_status'] == 'CANCELED') &
(df['store_plan_price'] == 0)]
y = sem_plano_cancelados.shape[0]
print(f"Lojas sem planos de fidelidade - Pedidos não cancelados: {x}. Pedidos cancelados: {y}.")
```

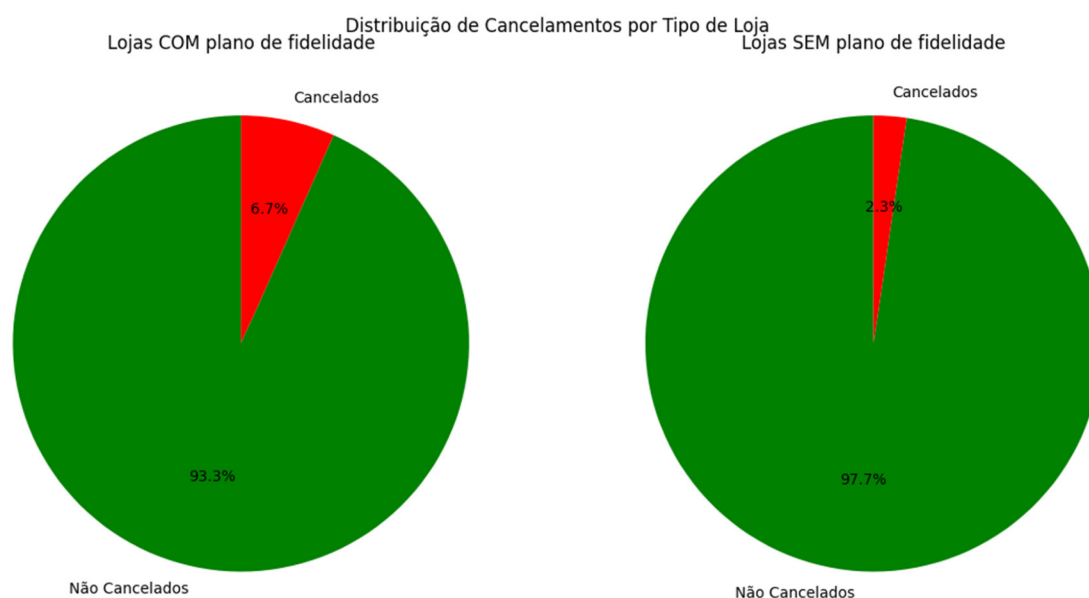
Fonte: (Própria,2025)

Figura 15– Saída do código em Python mostrando a porcentagem de pedidos cancelados e não cancelados de lojas sem planos de assinatura

```
Lojas sem planos de fidelidade - Pedidos não cancelados: 89370. Pedidos cancelados: 2136.
Valor total de pedidos com planos: 91506. Porcentagem nao cancelada: 97.66572683758442 / Porcentagem cancelada: 2.3342731624155846.
```

Fonte: (Própria,2025)

Figura 16– Gráfico de distribuição de cancelamentos por tipo de loja



Fonte: (Própria, 2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

A partir do gráfico, foi possível visualizar que lojas sem plano de fidelidade tem pedidos menos cancelados do que os feitos com plano de fidelidade.

Questionamento 4:

Quando a loja tem assinatura, existem mais pedidos? Identificar se quando a loja tem assinatura existem mais pedidos ajuda a entender se o plano pode não estar atraente para os clientes, seja pelo preço, benefícios fracos ou falta de divulgação.

Código 11– Código em Python para identificar se existem mais pedidos feitos em lojas com plano de assinatura ou em lojas sem plano de assinatura

```
from google.colab import files
uploaded = files.upload()
import pandas as pd

df = pd.read_csv('orders_stores.csv')
df.head()

total_sem_planos = x + y
porcentagem_nao_cancelados = (100 * x) / total_sem_planos
porcentagem_cancelados = 100 - porcentagem_nao_cancelados
print(f'Valor total de pedidos com planos: {total_sem_planos}. Porcentagem nao cancelada: {porcentagem_nao_cancelados} / Porcentagem cancelada: {porcentagem_cancelados}.')
```

Fonte: (Própria,2025)

Figura 17- Saída do código em Python mostrando a porcentagem de pedidos em lojas com planos de assinatura e lojas sem planos de assinatura

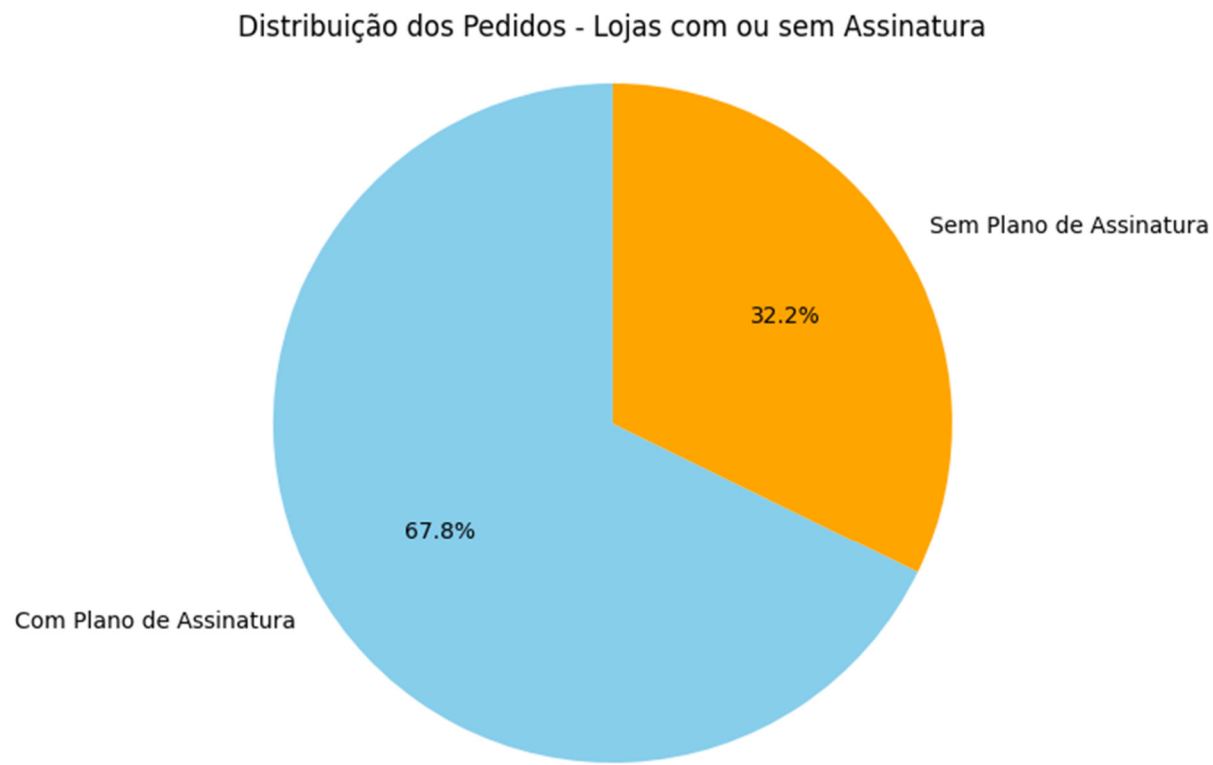
```
Porcentagem de pedidos - Com planos de assinatura: 67.77537918672503 / Sem planos de assinatura: 32.22462081327497.
```

Fonte: (Própria, 2025)

Sendo assim, foi identificado que os pedidos feitos sem plano de assinatura são cancelados com menos frequência do que os com assinatura.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Figura 18– Gráfico de distribuição de pedidos de lojas com ou sem assinatura



Fonte: (Própria, 2025)

Questionamento 5: (estatística)

Para avaliar o impacto dos canais de venda — próprios ou marketplace — sobre o valor dos pedidos e o tempo total de entrega, formulamos a seguinte hipótese de negócio: pedidos realizados por canais próprios tendem a ter maior valor e a serem entregues mais rapidamente. Para isso, utilizamos as variáveis *channel_type* (classificadas em *Own Channel* e *Marketplace*), *order_amount* (valor do pedido) e *order_metric_cycle_time* (tempo total do ciclo do pedido). A análise foi conduzida por meio de métodos estatísticos, entre eles a análise com boxplot, que permite uma comparação visual das distribuições, e com o cálculo do intervalo de confiança das médias para cada tipo de canal, fornecendo uma estimativa mais robusta da diferença entre os grupos.

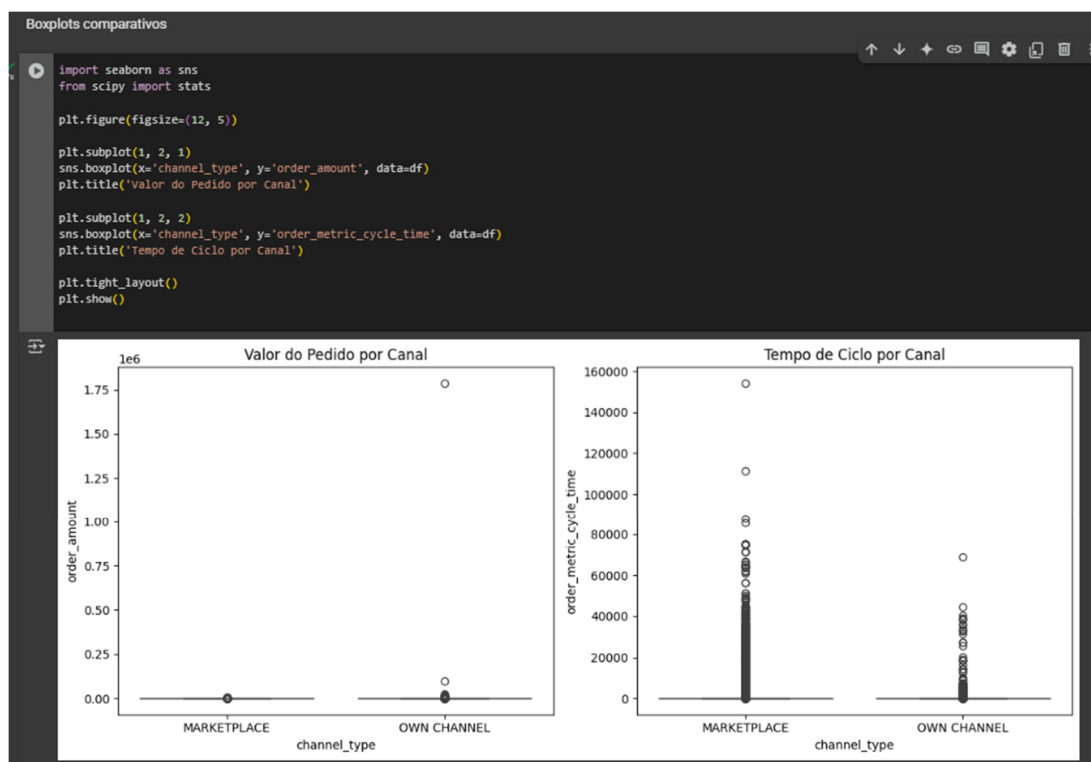
Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Código 12– Código e saída no python da análise estatística



Fonte: (Própria, 2025)

Código 13- Código e saída Boxplot



Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Questionamento 6:

Existe diferença significativa na taxa de sucesso das entregas entre os diferentes hubs? Entender essa diferença é importante para identificar possíveis variações no desempenho operacional, o que pode indicar a necessidade de melhorias específicas em determinados hubs.

Assim, foi realizado um teste estatístico Qui-quadrado para comparar as taxas de entregas bem-sucedidas e não realizadas entre os hubs. O resultado desse teste indica se as diferenças observadas entre os hubs são estatisticamente relevantes ou podem ser atribuídas ao acaso.

Código 14– Código em Python para analisar a diferença estatística na taxa de entregas bem-sucedidas entre hubs usando o teste Qui-quadrado

```
import pandas as pd
from scipy.stats import chi2_contingency

channels = pd.read_csv('channels.csv', encoding='latin1')
deliveries = pd.read_csv('deliveries.csv', encoding='latin1')
drivers = pd.read_csv('drivers.csv', encoding='latin1')
hubs = pd.read_csv('hubs.csv', encoding='latin1')
orders = pd.read_csv('orders.csv', encoding='latin1')
stores = pd.read_csv('stores.csv', encoding='latin1')
orders_reduced = orders[['delivery_order_id', 'store_id']]
stores_reduced = stores[['store_id', 'hub_id']]
hubs_reduced = hubs[['hub_id', 'hub_name']]
orders_stores = orders_reduced.merge(stores_reduced, on='store_id', how='left')
orders_stores_hubs = orders_stores.merge(hubs_reduced, on='hub_id', how='left')
deliveries_orders = deliveries[['delivery_order_id', 'delivery_status']].merge(
orders_stores_hubs, on='delivery_order_id', how='left')
delivery_status_by_hub =
deliveries_orders.groupby('hub_name')['delivery_status'].value_counts().unstack().fillna(0)
delivery_status_by_hub['CANCELED'] = delivery_status_by_hub.sum(axis=1) -
delivery_status_by_hub.get('DELIVERED', 0)
contingency_table = delivery_status_by_hub[['DELIVERED', 'CANCELED']]
print("Tabela de contingência:\n")
print(contingency_table)
chi2, p, dof, expected = chi2_contingency(contingency_table)
print("\nResultado do Teste Qui-quadrado:")
print(f"Estatística Qui-quadrado: {chi2:.4f}")
print(f"Valor-p: {p:.4f}")
print(f"Grau de liberdade: {dof}")
print("\nFrequências esperadas:\n", pd.DataFrame(expected, index=contingency_table.index,
columns=contingency_table.columns))
if p < 0.05:
print("\nExiste diferença estatística significativa na taxa de sucesso entre os hubs (Rejeita H0).")
else:
print("\nNão existe diferença estatística significativa na taxa de sucesso entre os hubs (Não rejeita H0).")
```

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Figura 19– Saída do código em Python mostrando a tabela de contingência e resultados do teste Qui-quadrado para a taxa de entregas bem-sucedidas por hub

Tabela de contingência:

delivery_status	DELIVERED	CANCELED
hub_name		
AVENUE SHOPPING	13875.0	115.0
BEACH SHOPPING	17236.0	290.0
BLACK SHOPPING	13049.0	194.0
COFFEE SHOPPING	21487.0	198.0
COLOR SHOPPING	5399.0	77.0
ELIXIR SHOPPING	7308.0	1391.0
FORTRAN SHOPPING	9774.0	163.0
FUNK SHOPPING	0.0	28.0
GAROA SHOPPING	0.0	1.0
GOLDEN SHOPPING	44324.0	1197.0
GREEN SHOPPING	21213.0	380.0
HIP HOP SHOPPING	21077.0	303.0
HOTMILK SHOPPING	7505.0	87.0
HUBLESS SHOPPING	2.0	3.0
PAGODE SHOPPING	23235.0	184.0
PEOPLE SHOPPING	19896.0	153.0
PHP SHOPPING	4976.0	129.0
PURPLE SHOPPING	17500.0	364.0
PYTHON SHOPPING	7641.0	227.0
R SHOPPING	232.0	3.0
RAP SHOPPING	18718.0	248.0
REACT SHOPPING	3339.0	28.0
RED SHOPPING	0.0	2.0
RIVER SHOPPING	6939.0	118.0
RUBY SHOPPING	9710.0	219.0
SAMPA SHOPPING	10767.0	170.0
SMALL SHOPPING	13828.0	238.0
SQL SHOPPING	10959.0	139.0
STAR SHOPPING	5612.0	228.0
SUBWAY SHOPPING	23386.0	213.0
WOLF SHOPPING	12380.0	386.0

Resultado do Teste Qui-quadrado:
Estatística Qui-quadrado: 11701.7460
Valor-p: 0.0000
Grau de liberdade: 30

Fonte: (Própria,2025)

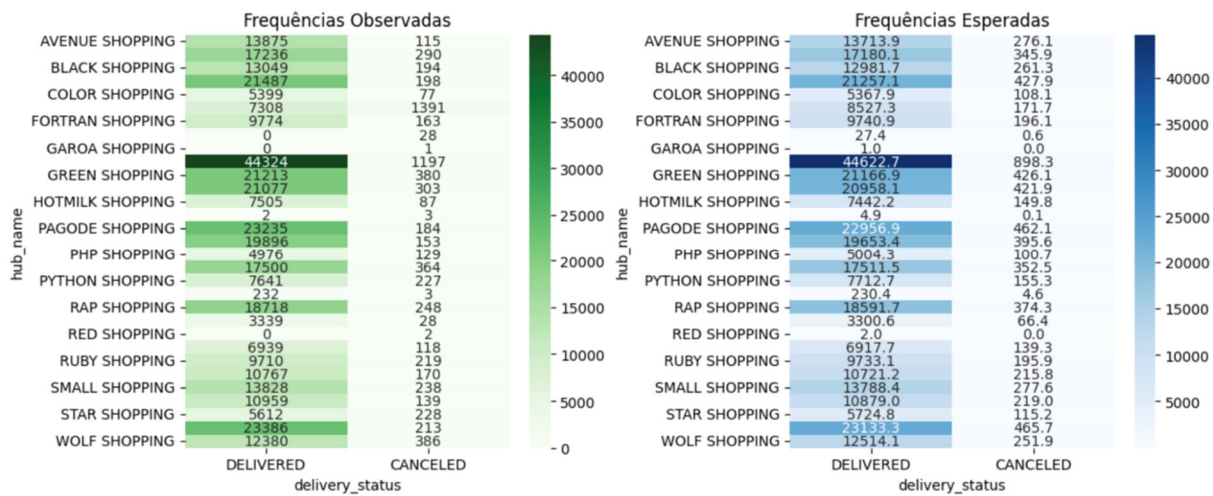
Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Figura 20– Saída do código em Python mostrando resultados da frequência para entregas entregues e canceladas por hub

Frequências esperadas:		
delivery_status	DELIVERED	CANCELED
hub_name		
AVENUE SHOPPING	13713.924581	276.075419
BEACH SHOPPING	17180.145976	345.854024
BLACK SHOPPING	12981.665706	261.334294
COFFEE SHOPPING	21257.073234	427.926766
COLOR SHOPPING	5367.937885	108.062115
ELIXIR SHOPPING	8527.335949	171.664051
FORTAN SHOPPING	9740.905544	196.094456
FUNK SHOPPING	27.447454	0.552546
GAROA SHOPPING	0.980266	0.019734
GOLDEN SHOPPING	44622.699131	898.300869
GREEN SHOPPING	21166.888740	426.111260
HIP HOP SHOPPING	20958.092033	421.907967
HOTMILK SHOPPING	7442.181231	149.818769
HUBLESS SHOPPING	4.901331	0.098669
PAGODE SHOPPING	22956.854879	462.145121
PEOPLE SHOPPING	19653.357679	395.642321
PHP SHOPPING	5004.259113	100.740887
PURPLE SHOPPING	17511.475962	352.524038
PYTHON SHOPPING	7712.734711	155.265289
R SHOPPING	230.362564	4.637436
RAP SHOPPING	18591.729350	374.270650
REACT SHOPPING	3300.556402	66.443598
RED SHOPPING	1.960532	0.039468
RIVER SHOPPING	6917.738797	139.261203
RUBY SHOPPING	9733.063414	195.936586
SAMPA SHOPPING	10721.171776	215.828224
SMALL SHOPPING	13788.424815	277.575185
SQL SHOPPING	10878.994639	219.005361
STAR SHOPPING	5724.754793	115.245207
SUBWAY SHOPPING	23133.302801	465.697199
WOLF SHOPPING	12514.078713	251.921287
Existe diferença estatística significativa na taxa de sucesso entre os hubs (Rejeita H0). Fonte: (Própria,2025)		

A partir do código, foi identificado que existem diferenças na quantidade de entregas bem-sucedidas e não realizadas entre os hubs e diferenças estatísticas significativas a taxa de sucesso entre os hubs

Figura 21– Gráfico de distribuição de entregas bem-sucedidas e não realizadas por hub



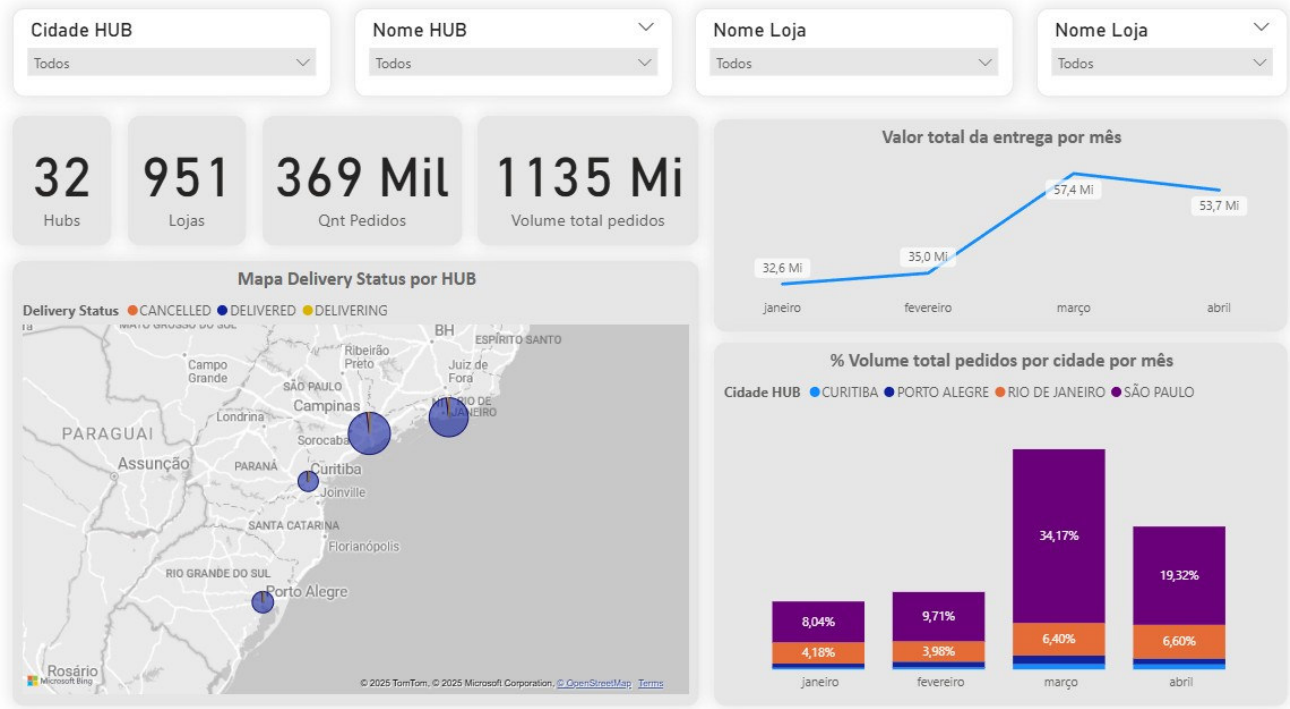
Fonte: (Própria, 2025)

A partir do gráfico, foi possível visualizar que alguns hubs apresentam uma proporção maior de entregas bem-sucedidas em comparação a outros, evidenciando variações no desempenho entre os centros de distribuição.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

Dashboard:

Figura 22– Dashboard



Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

6. Programação de Banco de Dados

Como parte das exigências da disciplina de Programação de Banco de Dados, foram desenvolvidas rotinas utilizando a linguagem PL/pgSQL no ambiente do Data Warehouse criado para o projeto. Os requisitos envolvem o uso de stored procedures com cursores não vinculados e a criação de triggers com funções associadas, com o objetivo de automatizar processos de análise e consistência dos dados.

A seguir, apresentam-se as implementações realizadas.

6.1 Stored Procedure com Cursor Não Vinculado de Query Dinâmica

Foi implementado um bloco anônimo com um cursor dinâmico que percorre registros da tabela `tb_order`, com a finalidade de calcular a soma de valores de pedidos com status 'delivered', agrupados por mês. A consulta SQL é construída de forma dinâmica utilizando a função `format()`, e os resultados são exibidos por meio de instruções `RAISE NOTICE`.

Código 15– Código em PL pg/SQL para analisar a soma das entregas por mês

```
DO $$
DECLARE
    cur_valor_entregas_mes REFCURSOR;
    variavel_status VARCHAR(200) := 'delivered';
    variavel_valor_entregas_mes INT;
    variavel_mes TEXT;
    variavel_nome_tabela VARCHAR(200) := 'tb_order';
BEGIN
    OPEN cur_valor_entregas_mes FOR EXECUTE
        format('
            SELECT SUM(order_amount) AS valor_entrega,
                   EXTRACT(MONTH FROM order_moment_created)::TEXT AS mes
            FROM %I
            WHERE order_status = $1
            GROUP BY mes
            ORDER BY mes
            ', variavel_nome_tabela)
    USING variavel_status;

    LOOP
        FETCH cur_valor_entregas_mes INTO variavel_valor_entregas_mes, variavel_mes;
        EXIT WHEN NOT FOUND;
        RAISE NOTICE 'Mês: %, Valor total entregue: %', variavel_mes, variavel_valor_entregas_mes;
    END LOOP;

    CLOSE cur_valor_entregas_mes;
END;
$$;
Fonte: (Própria,2025)
```

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

6.2 Trigger com Função Associada

Para garantir a consistência dos dados na tabela `tb_store`, foi criada uma função que avalia o campo `store_plan_price`. Sempre que ocorrer uma atualização, a trigger associada define automaticamente o valor do campo `have_plan_price` como `TRUE`, caso o plano seja maior que zero, ou `FALSE`, caso contrário. Essa ação ocorre antes da execução do comando `UPDATE`, assegurando integridade no registro da presença ou ausência de plano ativo.

Código 16– Código em PL pg/QGL para garantir a atualização na tabela stores

```
CREATE OR REPLACE FUNCTION fn_have_plan_price()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.store_plan_price IS NULL OR NEW.store_plan_price = 0 THEN
        NEW.have_plan_price := FALSE;
    ELSE
        NEW.have_plan_price := TRUE;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER tg_have_plan_price
BEFORE UPDATE ON tb_store
FOR EACH ROW
EXECUTE FUNCTION fn_have_plan_price();
```

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

7. Conclusão

Ao longo deste projeto, foi possível construir um Data Warehouse robusto e aplicar um processo completo de ETL, integrando e organizando os dados de maneira eficiente, o que possibilitou análises estratégicas voltadas ao contexto do Empório Quel. A centralização das informações e a padronização dos dados viabilizaram uma estrutura sólida para a exploração analítica, respeitando os princípios de qualidade, consistência e integridade.

Dessa forma, foi possível empregar técnicas de Mineração de Dados, aliadas a recursos de Inteligência Artificial e métodos estatísticos, revelando padrões ocultos, tendências operacionais e correlações significativas entre variáveis logísticas e comportamentais. A partir das análises realizadas, foram gerados insights estratégicos com foco na melhoria da operação de delivery, como a identificação de hubs mais eficientes, avaliação do tempo médio de entrega, análise do impacto dos planos de assinatura sobre os pedidos e a relação entre os canais de venda e o desempenho operacional.

Além disso, a implementação de dashboards analíticos possibilitou uma visualização clara e acessível dos principais indicadores de desempenho, permitindo o monitoramento contínuo da operação. A utilização de ferramentas como Python e PostgreSQL, combinadas com metodologias de análise estatística, fortaleceu o processo de extração de conhecimento, permitindo à equipe compreender melhor os fatores que impactam a eficiência logística e o comportamento do consumidor.

O projeto também demonstrou como o uso de dados externos — neste caso, uma base pública do Delivery Center — pode ser adaptado à realidade de uma empresa local, como o Empório Quel, servindo como modelo para a aplicação de tecnologias de Big Data em micro e pequenos empreendimentos. A construção de um ambiente estruturado de dados e a aplicação de técnicas de mineração viabilizam o acesso de empresas de menor porte a recursos que, tradicionalmente, são utilizados por grandes corporações, promovendo a democratização da inteligência de mercado.

Essas informações se mostraram valiosas para a tomada de decisões mais informadas e proativas, contribuindo para o aprimoramento da gestão do Empório Quel. Assim, os objetivos iniciais do projeto foram plenamente atingidos, demonstrando a importância da integração entre dados, tecnologia e inteligência analítica na transformação da informação em vantagem competitiva.

Ademais, todas as perguntas de pesquisa propostas ao longo do desenvolvimento do projeto foram devidamente investigadas e respondidas com base em análises estatísticas, testes de hipóteses e visualizações gráficas, comprovando a efetividade da metodologia adotada. Questões como a identificação dos hubs com melhor desempenho, a relação entre os planos de assinatura e a taxa de cancelamento, o impacto dos canais de venda sobre o valor dos pedidos e o tempo de entrega, entre outras, foram exploradas com profundidade e resultaram em descobertas relevantes para a tomada de decisão estratégica. A análise de dados logísticos, por meio de testes como o t de Student e o Qui-quadrado, permitiu compreender o comportamento da operação com respaldo estatístico, conferindo maior confiabilidade aos resultados obtidos.

Conclui-se, portanto, que os objetivos propostos no início do projeto foram plenamente alcançados. A construção do Data Warehouse, a execução do processo de ETL, a aplicação de técnicas de Mineração de Dados e a geração de insights estratégicos se integraram de forma coerente e eficaz. O projeto contribuiu significativamente para demonstrar como a organização e o tratamento inteligente de dados podem transformar informações dispersas em conhecimento útil e acionável, mesmo em contextos de empresas de pequeno porte. A experiência adquirida na modelagem de dados, na programação com Python e PL/pgSQL, e na condução de análises exploratórias e estatísticas consolidou o aprendizado dos participantes e deixou um legado de aplicabilidade prática para o Empório Quel. O trabalho reforça o papel da ciência de dados como aliada essencial à inovação e ao crescimento sustentável dos negócios.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

8. Cronograma

Tabela 1- Cronograma-

Etapa	Atividade	Data de Início	Data de Término	Observações
Planejamento Inicial	Reunião de Kick-off	04/03/2025	04/03/2025	Definição dos objetivos, escopo e responsabilidades.
Planejamento Inicial	Elaboração do Plano de Projeto	05/03/2025	09/03/2025	Documento com cronograma, dados da empresa, aplicações a serem usadas.
Planejamento Inicial	Definição das Metodologias de Pesquisa e Ferramentas	10/03/2025	17/03/2025	Escolha de metodologias e ferramentas.
Análise competitiva	Identificação de estratégias para o negócio	18/03/2025	21/03/2025	Discussão das premissas do projeto
Definição do projeto e objetivos	Definição do sistema a ser desenvolvido	21/03/2025	25/03/2025	Sistema resultante da análise competitiva
Desenvolvimento do Sistema	Reunião de Planejamento Técnico	26/03/2025	30/03/2025	Definição requisitos, tecnologias e arquitetura.
Desenvolvimento do Sistema	Desenho da Arquitetura do Sistema	01/04/2025	08/04/2025	Criação dos diagramas de arquitetura e UI.
Desenvolvimento do Sistema	Desenvolvimento do Backend	09/04/2025	23/04/2025	Codificação do backend. MVP
Desenvolvimento do Sistema	Desenvolvimento do Frontend	24/04/2025	02/05/2025	Desenvolvimento da interface do usuário. MVP
Desenvolvimento do Sistema	Integração do Sistema	03/05/2025	08/05/2025	Integração e testes iniciais.
Testes e Validação	Testes Funcionais	08/05/2025	13/05/2025	Testes de funcionalidades do sistema.
Testes e Validação	Correção de Bugs	14/05/2025	30/05/2025	Correção dos bugs identificados.
Documentação e Apresentação	Elaboração apresentação	14/05/2025	30/05/2025	Slides de apresentação
Data Analytics e documentos técnicos	Preparação de documentos técnicos e análise de dados	30/05/2025	13/06/2025	Dashboard e resultados da análise de dados

Fonte: (Própria,2025)

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

9. Referências Bibliográficas

COUTINHO, Vitória. ProjetoIntegrador3.ipynb. 2025. Disponível em: <https://colab.research.google.com/drive/1-xTsueLRYvC86Skbybq6AnfeYnU2sfO2?usp=sharing>. Acesso em: 13 maio 2025.

_____. ProjetoIntegrador3PlanPrice.ipynb. 2025. Disponível em: <https://colab.research.google.com/drive/1Fd-dKw2h7-iFR8vF0pssXiki7bb5o394?usp=sharing>. Acesso em: 13 maio 2025.

EBIT/NIELSEN. Estudo de mercado de e-commerce. 2020. Disponível em: <https://www.ecommercebrasil.com.br/artigos/o-varejo-no-brasil-e-na-america-latina-quem-sai-ganhando>. Acesso em: agosto-2024.

GOOMER. Crescimento do delivery no Brasil. Disponível em: <https://goomer.com.br/blog/crescimento-delivery-brasil>. Acesso em: setembro-2024.

INNOVA MARKET INSIGHTS apud INGREDIENTS NETWORK. Taxa de crescimento do setor de alimentos artesanais. 2023. Disponível em: <https://www.innovamarketinsights.com/trends-hub/>. Acesso em: agosto-2024.

LEADSTER. Estatísticas de marketing digital. Disponível em: https://leadster.com.br/blog/estatisticas-de-marketingdigital#Estatisticas_que_mostram_que_o_marketing_digital_ja_superou_o_tradicional. Acesso em: setembro-2024.

SEMRUSH. Tendências e estatísticas de pequenas empresas no Brasil. Disponível em: <https://pt.semrush.com/blog/tendencias-e-estatisticas-pequenas-empresas-brasil/#evoc%C3%AA.-est%C3%A1-pronto-para-come%C3%A7ar-um-novo-neg%C3%B3cio>. Acesso em: setembro-2024.

KAGGLE. Delivery Center: Food & Goods orders in Brazil. Disponível em: <https://www.kaggle.com/datasets/nosbielcs/brazilian-delivery-center>. Acesso em: abril-2025.

Inteligência de Mercado para Empório Quel	Versão:<8.0>
Documento Projeto Integrador III	Data: 26/mai/25

10. APÊNDICE A – Gestão de Projetos

Declaração do Problema

Tabela 2- Declaração do Problema

O problema de:	Demora nas entregas por delivery devido à localização desfavorável.
Afeta:	Clientes que dependem do serviço de entrega.
O impacto é o seguinte:	Insatisfação dos clientes, perda das vendas e danos á reputação da loja.
Uma solução bem-sucedida seria:	Otimizar a logística de entregas e firmar parcerias com serviços de entrega mais rápidos.

Fonte: (Própria,2025)

Declaração do Produto

Tabela 3- Declaração do Produto

aPara:	Lojas de produtos naturais com foco em vendas por delivery.
Que:	Enfrentam dificuldades com localização desfavorável e atrasos nas entregas, o que prejudica a experiência do cliente.
O Rota Certa:	É uma plataforma de gestão de entregas e comunicação com o cliente, voltada para otimizar rotas e prazos de entrega.
Oferece:	Coleta e atualização de dados dos pedidos, cálculo inteligente de rotas, estimativas precisas de entrega, acompanhamento em tempo real.

Fonte: (Própria,2025)

Resumo dos Recursos

Tabela 4 - Declaração dos Recursos

Benefício do Cliente	Recursos do Rota Certa
Entregas mais rápidas e eficientes.	Otimização de rotas com base na localização do cliente.
Redução da frustração com atrasos.	Painel de acompanhamento da entrega para o cliente.
Maior confiança no serviço da loja.	Canal de atendimento automatizado com respostas rápidas.
Facilidade para resolver problemas com pedidos.	Aplicativo com interface simples para acompanhar status do pedido
Mais comodidade e conveniência.	Centralização de informações do pedido.

Fonte: (Própria,2025)