

# Relatório de Análise de Desempenho de Entregas

Autores: João Paulo, João Pedro, Kiev Luiz

## 1. Tema

O objetivo deste projeto é analisar o desempenho dos entregadores de um serviço de delivery de comida. A análise foca em entender como diferentes fatores (como condições climáticas, trânsito, e características do entregador) impactam a eficiência, medida principalmente pelo tempo de entrega e velocidade média.

O produto final é um dashboard interativo (construído com Streamlit) que permite a visualização dessas métricas para tomada de decisão.

## 2. Descrição dos Dados Utilizados

### 2.1. Origem

Os dados brutos para esta análise foram extraídos do arquivo `food_delivery_dataset.csv`. A origem do conjunto dos dados é o Kaggle.

### 2.2. Estrutura dos Dados (Schema)

A tabela `deliveries` (definida em `table_deliveries.sql`) armazena os dados com a seguinte estrutura principal:

- **Identificadores:** `id` (do pedido), `delivery_person_id` (do entregador).
- **Informações do Entregador:** `delivery_person_age` (idade), `delivery_person_ratings` (classificação média).
- **Geolocalização:** `restaurant_latitude`, `restaurant_longitude`, `delivery_location_latitude`, `delivery_location_longitude`.
- **Tempos:** `order_date` (data), `time_orderd` (pedido feito), `time_order_picked` (pedido coletado), `preparing_time` (tempo de preparo do pedido em minutos), `delta_time` (tempo de entrega em minutos) e `time_taken` (tempo total gasto em minutos).
- **Condições da Entrega:** `weather_conditions` (clima), `road_traffic_density` (densidade do trânsito), `vehicle_condition` (condição do veículo), `festival` (se era um dia de festival).
- **Informações do Pedido:** `type_of_order` (tipo de comida), `type_of_vehicle` (tipo de veículo), `multiple_deliveries` (se carregava múltiplas entregas).

## 3. Explicação das Consultas e Views Criadas

Para alimentar o dashboard e permitir análises detalhadas, várias consultas foram realizadas em uma view SQL.

### 3.1. View Principal

A `view_master.sql` é o coração da análise. Ela é executada primeiro e serve como fonte de dados limpa e enriquecida para todas as outras consultas do dashboard.

- **CREATE VIEW `delivery_metrics_master` AS ...**
  - **Propósito:** Limpar os dados brutos da tabela `deliveries` e criar novas colunas (features) que são essenciais para a análise de faixas de referência.

- **Limpeza de Dados:** A view filtra agressivamente os dados, removendo quaisquer linhas onde time\_taken seja 0 ou nulo, ou onde ratings, age, geolocalização, clima ou tráfego sejam nulos ou 'NaN'. Isso garante que todas as métricas sejam calculadas apenas com registros completos.
- **Engenharia de Features (Colunas Criadas):**
  1. is\_festival: Converte a coluna festival (texto "Yes" ou "No") em um indicador numérico (1 ou 0).
  2. distance\_km: Calcula a distância em linha reta (usando a **Fórmula de Haversine**) entre o restaurante e o local de entrega.
  3. speed\_kmh: Calcula a velocidade real em **KM/H**, dividindo a distance\_km pelo tempo (convertido para horas).
  4. rating\_range: Categoriza os entregadores em faixas: 'high rating' (> 4.5), 'medium rating' (> 4.0), 'low rating' (> 3.5), e 'poor rating'.
  5. age\_group: Categoriza os entregadores em faixas etárias: 'Ages 18-24', 'Ages 25-34', e 'Ages 35+'.
  6. condition\_status: Cria uma categoria binária. É 'adverse\_condition' se o clima for ruim (Tempestade, Areia, Neblina, Vento) OU se o trânsito for 'High' (Alto) ou 'Jam' (Congestionado). Caso contrário, é 'normal\_condition'.
  7. proxy\_speed: Uma métrica de "pseudo-velocidade" (graus de distância ao quadrado por minuto) usada nos gráficos de eficiência.

### 3.2. Consultas do Dashboard (App)

As seguintes consultas são executadas pelo app.py e alimentam diretamente os gráficos e KPIs. Elas **sempre** consultam a delivery\_metrics\_master, e não a tabela deliveries original.

- **query\_time\_taken\_by\_weather\_conditions.sql**
  - **Propósito:** Alimenta os KPIs principais e o primeiro gráfico de barras.
  - **Lógica:** Agrupa os dados pelas duas categorias de condition\_status (normal vs. adverso) e calcula o tempo médio e o número total de entregas para cada uma.
- **query\_deliver\_efficiency\_by\_rating.sql**
  - **Propósito:** Alimenta o gráfico de "Eficiência por Classificação".
  - **Lógica:** Agrupa os dados pelas faixas de rating\_range (high, medium, etc.) e calcula a velocidade média (AVG(proxy\_speed)) para cada faixa.
- **query\_deliver\_efficiency\_by\_age.sql**
  - **Propósito:** Alimenta o gráfico "Eficiência por Faixa Etária".
  - **Lógica:** Agrupa os dados pelas faixas de age\_group (18-24, 25-34, 35+) e calcula a velocidade média (AVG(proxy\_speed)) para cada grupo.
- **query\_time\_by\_festival.sql**
  - **Propósito:** Alimenta a seção "Impacto de Festivais".
  - **Lógica:** Agrupa os dados pelo indicador is\_festival (1 ou 0) e calcula o tempo médio, velocidade média e total de entregas para cada status (Festival ou Sem Festival).
- **query\_top\_fastest\_delivery\_persons.sql e query\_top\_slowest\_delivery\_persons.sql**
  - **Propósito:** Identificar os 10 melhores e os 10 piores entregadores com base em sua velocidade média real.
  - **Lógica:** Agrupa por delivery\_person\_id e calcula a média de speed\_kmh (a velocidade real), filtrando velocidades irreais (> 200 km/h), e então ordena (DESC para os mais rápidos, ASC para os mais lentos).
- **query\_speed\_histogram.sql**
  - **Propósito:** Alimenta o histograma na seção "Visualizações adicionais".
  - **Lógica:** Simplesmente retorna todas as velocidades (speed\_kmh) calculadas, permitindo que o Plotly e o Streamlit façam a agregação e plotagem.

- **query\_speed\_vs\_rating.sql**
  - **Propósito:** Alimenta o gráfico de dispersão (scatter plot) para análise de correlação.
  - **Lógica:** Agrupa por delivery\_person\_id (para entregadores com 5+ entregas) e retorna a avg\_speed\_kmh e a avg\_rating daquele indivíduo.

## 4. Análise dos Dados (Resultados)

A análise dos dados, visualizada no dashboard, revela as seguintes tendências:

### 4.1. KPIs Principais

(Resultados de query\_time\_taken\_by\_weather\_conditions.sql)

- **Total de Entregas Analisadas:** 42.923
- **Tempo Médio (Condições Normais):** 13,79
- **Tempo Médio (Condições Adversas):** 17,43

### 4.2. Impacto das Condições no Tempo

A análise do gráfico "Impacto das Condições no Tempo de Entrega" .

- **Observação:** Em condições adversas o tempo médio de uma entrega é de 17,432 minutos, uma diferença de 3,65 min em comparação a condições normais.

### 4.3. Eficiência por Classificação (Rating)

O gráfico "Eficiência por Classificação" investiga se entregadores mais bem avaliados são mais eficientes (medido pela proxy\_speed).

- **Observação:** Entregadores mais rápidos são em geral mais bem avaliados.

### 4.4. Eficiência por Faixa Etária

O gráfico "Eficiência por Faixa Etária" mostra a relação entre idade e eficiência (medida pela proxy\_speed).

- **Observação:** Entregadores mais velhos tendem a ter uma velocidade máxima maior.

### 4.5. Impacto de Festivais no Tempo de Entrega

O gráfico " Impacto de Festivais no Tempo de Entrega" mostra a relação entre a existência de festivais e o tempo médio de entrega (medida pela proxy\_speed).

- **Observação:** Em dias de festival as entregas são quase duas vezes mais lentas.