

# Análise de Performance de Transporte (Dashboard Streamlit)

## Visão Geral do Projeto

Este projeto consiste em um dashboard interativo desenvolvido com Streamlit para analisar a performance de operações de transporte. Utilizando dados de entregas simuladas, o dashboard oferece uma visão clara e detalhada de Key Performance Indicators (KPIs) essenciais, como OTIF (On-Time In-Full), custo médio, distância média e percentual de atrasos. Além disso, fornece visualizações gráficas da evolução mensal do OTIF, atrasos por transportadora, distribuição de prazos de entrega e um mapa interativo das entregas.

## Recursos Principais:

- KPIs em Destaque: Visualização rápida dos principais indicadores de performance.
- Filtros Interativos: Capacidade de filtrar dados por Unidade Federativa (UF).
- Análises Gráficas:
  - \* Evolução Mensal do OTIF.
  - \* Atrasos por Transportadora.
  - \* Gráficos de dispersão e barras para entender custo, distância e status.
  - \* Distribuição de Prazos de Entrega.
  - \* Proporção de Entregas OK vs. Atrasadas.
  - \* Número de Pedidos por Cidade.
- Mapa Interativo de Entregas: Visualização das entregas no mapa, coloridas por status (OK/Atrasada), utilizando geocodificação.
- Insights Gerenciais: Recomendações baseadas nos dados para auxiliar na tomada de decisão.

## Tecnologias Utilizadas:

- Python 3.x
- Streamlit
- Pandas
- Plotly Express
- Matplotlib & Seaborn
- geopy

## Como Executar o Projeto Localmente:

### 1. Pré-requisitos:

# Análise de Performance de Transporte (Dashboard Streamlit)

- Python 3.x instalado.
- Arquivo entregas\_simuladas.csv na mesma pasta do script.

## 2. Instalação das dependências:

`pip install streamlit pandas plotly-express matplotlib seaborn geopandas geopy`

## 3. Executar o dashboard:

`cd C:\Projetos_data_science\Analise_logistica`

`streamlit run dashboard_analise.py`

## KPIs e Fórmulas:

### OTIF (On-Time In-Full):

Fórmula:  $OTIF = (\text{Número de Entregas OK} / \text{Total de Entregas}) \times 100$

Código: `otif = (df['Entrega_OK'] == True).mean() * 100`

### Percentual de Entregas com Atraso:

Fórmula:  $\text{Atraso Percentual} = (\text{Número de Entregas Atrasadas} / \text{Total de Entregas}) \times 100$

Código: `atraso_percentual = (df['Entrega_OK'] == False).mean() * 100`

### Custo Médio (R\$):

Fórmula:  $\text{Custo Médio} = \text{Custo de Cada Entrega} / \text{Total de Entregas}$

Código: `custo_medio = df['Custo_R$'].mean()`

### Distância Média (km):

Fórmula:  $\text{Distância Média} = \text{Distância de Cada Entrega} / \text{Total de Entregas}$

Código: `kms_medio = df['Distancia_km'].mean()`

### Prazo de Entrega (Dias):

Fórmula:  $\text{Data de Entrega} - \text{Data do Pedido}$

Código: `df['Prazo_Entrega_Dias'] = (df['Data_Entrega'] - df['Data_Pedido']).dt.days`

## **Análise de Performance de Transporte (Dashboard Streamlit)**

Custo por KM Individual:

Fórmula: Custo da Entrega / Distância da Entrega (km)

Código: `df_filt['Custo_por_KM_Individual'] = df_filt['Custo_R$'] / df_filt['Distancia_km']`

Estrutura do Repositório:

Analise\_logistica/

dashboard\_analise.py

entregas\_simuladas.csv

README.md