API III – Análise de produtividade de veículos, de otimização de distribuição e de custo de rotas de uma empresa embarcadora de carga.

Elaine Cristina Fernandes Gonçalves (linkedin - <https://abrir.link/2t69P>)

Jessica Tinoco Bernardo (linkedin - <https://shre.ink/8GYh>)

João Pedro Cardoso de Oliveira (linkedin - <https://abrir.link/edY91>)

Joyce Prudêncio (linkedin - <https://abrir.link/vnIKm>)

Rebeca Fonseca de Abreu (linkedin - <https://abrir.link/1an95>)

Vitor Ávila (linkedin - <https://abrir.link/PdhkP>)

Vitor Hugo Caetano das Merces (linkedin - <https://abrir.link/aPPru>)

Professor M2: Carlos Eduardo Bastos

Professor P2: Marcus Vinícius do Nascimento

Resumo do projeto:

Um parceiro interno apesentou um projeto onde, uma empresa embarcadora de carga busca a otimização de distribuição e de custo de rotas. Este projeto tem como objetivo criar e modelar um banco de dados em SQL com visualizador de indicadores em Power BI, aplicando método de transportes para otimização da distribuição.

Palavras- Chave: Otimização; Carga; Rotas; Método de Transportes.

Abstract:

A internal partner presented a project where a cargo shipping company seeks to optimize distribution and route cost. This project aims to create and model a SQL database with a Power BI indicator viewer, applying transportation method for distribution optimization.

Keywords: Optimization; Cargo; Routes; Transportation Method.

# Contextualização do projeto

Uma empresa embarcadora de carga está buscando melhorar sua eficiência operacional e reduzir custos através de uma análise abrangente da produtividade de seus veículos, da otimização de distribuição e da gestão de custos das rotas. Este projeto visa implementar soluções baseadas em dados para otimizar o transporte de carga, reduzir tempos de entrega e minimizar despesas logísticas

.

# Objetivos do projeto

Análise de produtividade de veículos, de otimização de distribuição e de custo de rotas de uma empresa embarcadora de carga com: ​

i) criação de um banco de dados em SQL​ e modelagem em Python​

ii) criação de um visualizador de indicadores em BI​

iii) aplicação do método de transportes para otimização da distribuição​

# Fundamentação dos métodos analíticos e das tecnologias utilizados

## Métodos analíticos utilizados

Neste projeto o método utilizado para análise é o “problema de transportes” estando dentro da área de estudo de Pesquisa Operacional, visando maior lucratividade através da compreensão das restrições do problema.

## Tecnologias da Informação

JiraSoftware – Gerenciamento de tempo e atividades

GitHub – Documentação do projeto

Power BI – Criação do Dashboard

MySQL – Criação do Banco de dados

Python – Modelagem das bases fornecidas

# Coleta e descrição dos dados utilizados

Os dados utilizados foram entregues pelo cliente, as bases de dados fornecidas de clientes (contém a localização e código dos clientes, município e latitude e longitude), fábricas (contém a localização e código das fábricas, município e latitude e longitude) e rotas (contém a data de emissão e entrega, código da fábrica e cliente, o tipo de operação FOB ou CIF, tipo de veículo e quantidade de pallets, quantidade transportada, moeda utilizada, valor de frete e distância).

Esses dados serão analisados e manipulados utilizando a programação em Python e inseridos em um banco relacional SQL.

Vale ressaltar que a base passará por diversas modelagens antes da entrega final ao cliente, conforme as remodelagens ocorrem, as atualizações do que foi feito ocorrerão por meio de documentos como esse.

# Resultados

Para a primeira sprint o objetivo era estruturar o GitHub, Jira Software e o backlog do produto. Após isso, a atenção foi voltada para a análise das bases.

O primeiro passo para o desenvolvimento do projeto foi a validação dos dados entregues pelo cliente, então prosseguimos da seguinte maneira:

Validação da base de clientes e de fábricas, que contém suas respectivas localizações. As localizações foram dispostas em um mapa, dessa forma foi possível autenticar a coerência dessas informações.

**Figura 1** – Códigos, criação do mapa.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: De autoria própria

Os marcadores verdes correspondem aos clientes enquanto os vermelhos correspondem às fábricas:

**Figura 02** – Mapa com distribuição de clientes e fábricas com marcadores.

Mapa

Descrição gerada automaticamente

Fonte: De autoria própria

Na sequência verificou-se a possível existência de inconsistências na base de rotas, nos campos de data de emissão e entrega.

**Figura 03** – Código, diferença entre data de emissão e entrega.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: De autoria própria

Com essa validação foram encontradas cinco datas inconsistentes, sendo elas:

**Figura 04** – Datas inconsistentes, Data de emissão maior que a entrega

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Fonte: De autoria própria

Prosseguindo, na mesma base, foram buscados valores de Free On Board (FOB) maiores que zero, pois nesse caso o valor deve ser zero, tendo em vista que são os clientes quem pagam por essa operação.

**Figura 05** – Código

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: De autoria própria

Assim foram encontradas 8767 inconsistências, segue uma amostra dos valores encontrados nas mesmas:

**Figura 06** – Demonstração de inconsistências

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: De autoria própria

Por fim as bases foram unidas, tendo então as latitudes e longitudes de origem (fábricas) e destino (clientes) na mesma base. Dessa forma foi realizada uma consulta no Google Maps (utilizando a latitude e longitude disponibilizadas) para calcular as distâncias entre fábricas e clientes e comparar com as distâncias fornecidas.

**Figura 07** – Código, aplicação da Application Programming Interface do Google Maps

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: De autoria própria

**Figura 08** – Código, consulta linha a linha das rotas

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: De autoria própria

Com isso calculamos a distância e tempo para cada rota fabrica-cliente, pela API (Application Programming Interface) do Google.

Podemos observar os resultados abaixo, nos campos; **Km\_api** e **tempo\_min\_api**:

**Figura 09** – Demostração dos valores encontrados.



Fonte: De autoria própria

Após isso confrontamos a distância encontrada via API (calculado) com a distância disponibilizada pelo cliente em sua base de rotas (real), utilizando métricas estatísticas observamos uma diferença entre as informações, essa diferença pode ser observada no campo **ERRO** que nos mostra a diferença da distância fornecida com a distância calculada.

**Figura 10** – Diferença entre real e calculado.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Fonte: De autoria própria

Definimos, para análise, uma distância de 5 quilômetros para limite de diferença entre real e calculado, sendo apontado todos os valores que ultrapassem este limite.

**Figura 11** – Diferença entre real e calculado maiores que 5 quilômetros.

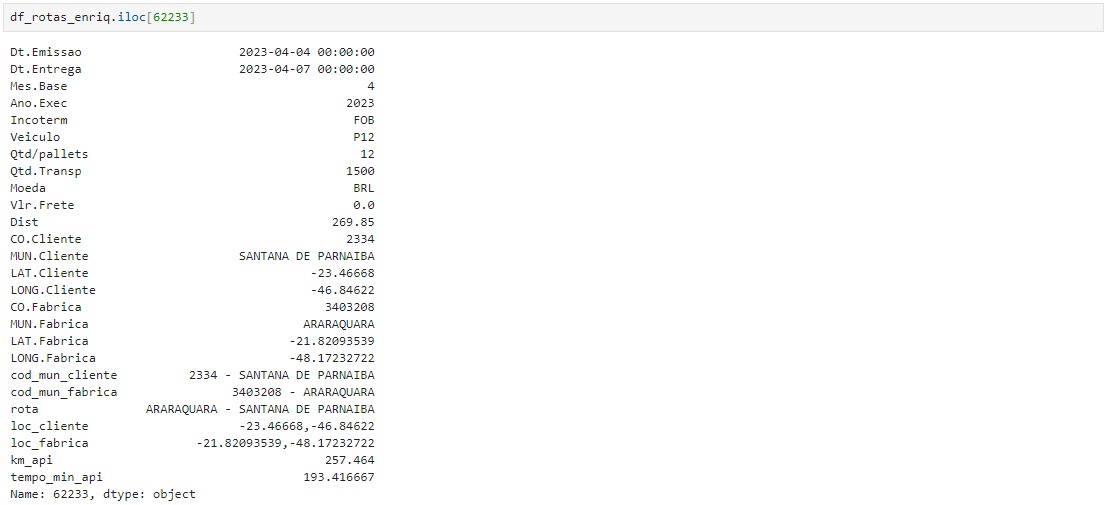
Interface gráfica do usuário, Aplicativo, Tabela

Descrição gerada automaticamente

Fonte: De autoria própria

Por fim, segue um exemplo de uma rota selecionada e as informações sobre ela.

**Figura 12** – Informações sobre a rota selecionada.



Fonte: De autoria própria