**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS**

**FATEC PROFESSOR JESSEN VIDAL**

**EVELYN VICTÓRIA PEREIRA DE MELO**

**FERNANDA DA SILVA ANDRADE**

**GABRIELA DE AVILA**

**LUCAS MATHEUS DO NASCIMENTO**

**LUCAS MONTEIRO DA COSTA SILVA**

**RAÍSSA NUNES CÔRREA**

**ANÁLISE DE PRODUTIVIDADE DE UMA EMPRESA EMBARCADORA DE CARGAS**

São José dos Campos

2024

**EVELYN VICTÓRIA PEREIRA DE MELO**

**FERNANDA DA SILVA ANDRADE**

**GABRIELA DE AVILA**

**LUCAS MATHEUS DO NASCIMENTO**

**LUCAS MONTEIRO DA COSTA SILVA**

**RAISSA NUNES CÔRREA**

**ANÁLISE DE PRODUTIVIDADE DE UMA EMPRESA EMBARCADORA DE CARGAS**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Logística.

**Orientador: Professor Me. Marcus Vinicius do Nascimento**

**Coorientador: Professor Me. Carlos Eduardo Bastos**

São José dos Campos

2024

Dedicatória

Dedicamos esta obra a Deus, aos familiares e aos amigos.

**AGRADECIMENTOS**

Agradecemos a Deus.

Agradecemos aos nossos familiares.

Agradecemos aos nossos amigos.

Epígrafe

“Nada é menos produtivo do que tornar eficiente algo que nem deveria ser feito.”

Peter F. Drucker

**RESUMO**

Nesta fase do projeto, com os dados tratados e consolidados, desenvolvemos um protótipo mais avançado no Power BI, visando aprimorar a visualização e a experiência do cliente. Com maior domínio das tecnologias aplicadas, a equipe está focada em realizar ajustes pontuais, de modo a alcançar o resultado final esperado com precisão. Assim, estamos alinhados ao objetivo proposto, assegurando uma entrega consistente e satisfatória ao cliente.

**ABSTRACT**

At this stage of the project, with the data processed and consolidated, we have developed a more advanced prototype in Power BI, aiming to enhance client visualization and experience. With a greater mastery of the applied technologies, the team is focused on making only necessary adjustments to achieve the expected final outcome with precision. In this way, we are aligned with the proposed objective, ensuring a consistent and satisfactory delivery for the client.

**SUMÁRIO**

[**1.**](#_heading=h.30j0zll) **Contextualização do projeto 9**

[**2.**](#_heading=h.1fob9te) **Objetivos do projeto 10**

[**3.**](#_heading=h.3znysh7) **Tecnologias utilizadas 11**

[3.1.](#_heading=h.2et92p0) Jira Software. 11

[3.2.](#_heading=h.tyjcwt) GitHub. 13

[3.3.](#_heading=h.3dy6vkm) Pacote Office. 14

[3.4.](#_heading=h.1t3h5sf) *Python*. 15

[3.5.](#_heading=h.4d34og8) MySQL 15

[3.6.](#_heading=h.2s8eyo1) PowerBI 20

[**4.**](#_heading=h.3rdcrjn) **RESULTADOS ESPERADOS 22**

[**5.**](#_heading=h.26in1rg) **REFERÊNCIAS 23**

# Contextualização do projeto

No setor logístico, onde a eficiência operacional e a redução de custos são fundamentais para manter a competitividade, a análise de produtividade de uma empresa embarcadora de cargas desempenha um papel crucial. Empresas desse segmento enfrentam o desafio constante de otimizar a gestão e movimentação das cargas, garantindo que suas operações sejam tanto econômicas quanto ágeis. Nesse contexto, a análise de produtividade surge como uma ferramenta vital para identificar pontos de melhoria e aprimorar os processos logísticos.

A análise de produtividade tem como objetivo central avaliar a eficiência das operações logísticas, utilizando indicadores-chave como o volume de cargas movimentadas, o tempo de ciclo das operações, a taxa de ocupação dos veículos, os custos operacionais e a taxa de entregas no prazo. Esses indicadores fornecem um panorama detalhado da performance operacional, permitindo a identificação de áreas que necessitam de ajustes. A metodologia adotada para essa análise inclui a coleta sistemática de dados e a realização de análises comparativas, com o propósito de identificar gargalos e pontos críticos no fluxo de trabalho.

Com base nos resultados da análise, a empresa pode implementar ações direcionadas para otimizar seus processos. Essas ações podem incluir a redução de custos operacionais, a melhoria da eficiência dos processos e o aprimoramento do atendimento ao cliente. A implementação das recomendações resultará em uma significativa redução dos gastos, um aumento na eficiência das operações e uma maior satisfação dos clientes. Dessa forma, a análise não apenas proporciona uma visão detalhada e estratégica, mas também contribui para o aprimoramento contínuo da empresa, tornando-a mais eficaz e competitiva no mercado logístico.

# Objetivos do projeto

A análise de produtividade é essencial para o sucesso de uma empresa embarcadora de cargas, permitindo avaliar o desempenho e identificar oportunidades de melhoria. Este estudo aborda a importância da análise no transporte de cargas, utilizando métodos e indicadores-chave de desempenho para otimizar operações. Inclui um estudo de caso de 3 fábricas, destacando desafios e benefícios práticos. A análise avalia a eficiência na gestão e movimentação de cargas por meio de indicadores como volume, tempo de ciclo, taxa de ocupação, custos e pontualidade. A metodologia cobre coleta de dados, análise comparativa e identificação de gargalos, resultando em recomendações para reduzir custos e melhorar a eficiência, impactando positivamente a satisfação dos clientes.

# Tecnologias utilizadas

## Jira Software.

Auxilia as equipes a criarem e classificarem, em nível de prioridade, as tarefas, atribuindo aos responsáveis, que, assim, podem acompanhar progressos e prazos. Além disso, a ferramenta fornece recursos avançados de relatórios e painéis para ajudar as equipes a acompanharem o desempenho do projeto.

Esta relevância também foi constatada por Oliveira. (2022 4.2 p. 31) na afirmação de que:

“No detalhamento é exibido a descrição de cada respectiva tarefa, qual a gerência solicitante, o responsável da gerência, data de início e conclusão da tarefa, planejador, número da ordem de serviço, evento e prioridade da demanda, *sprint* em que a tarefa foi priorizada, nome da empresa terceirizada na realização da tarefa, story point da tarefa e pôr fim a data em que a tarefa foi programada.”

A equipe iniciou o projeto noJira *Software*, criando *Sprints* e atribuindo as tarefas iniciais. Após alguns dias, o *backlog* completo do projeto foi criado e a ferramenta estava sendo utilizada sem problemas. Foi criada, também, uma planilha em Excel como um backup, caso o *backlog* seja perdido.

O *backlog* final foi dividido em *sprints* conforme abaixo:

*Backlog de Produto+GitHub+Jira* (30/08/24 - 27/08/24):

* Estruturar Jira;
* Estruturar *GitHub*;
* Estruturar relatório;
* Disponibilizar relatório no *GitHub*;
* Revisão de *Sprint*;
* Estruturar *backlog* das *sprint*s;
* Estruturar *backlog* do produto;
* Tratar base de dados CSV;
* Estudar SQL;
* Entender método para unir bases de dados através do *Python*;
* Workshop *Jira Software*.

*Sprint* 1 (27/09/24 - 18/10/24):

* Entender método para cálculo das distâncias
* Unir Planilhas – *Python*
* Construir função objetivo (Z)
* Levantamento de informações para análise (PO)
* Unir planilhas – SQL
* Comparar resultados
* Atualizar GitHub
* Atualizar relatório
* Disponibilizar relatório no GitHub
* Revisão da *sprint*
* MVP: Produtividade mensal dos veículos – Tela de BI

*Sprint* 3 (18/10/24 - 08/11/24):

* Atualizar GitHub
* Atualizar Relatório
* Disponibilizar Relatório no GitHub
* Revisão de Sprint
* Protótipo de Dashboard
* Realizar Testes de Indicadores no Dashboard
* Construir Função Objetivo (z)
* Verificar Eventuais Questões:Python + SQL
* Reunir Restrições para Aplicação do Método de Transportes
* Aplicação do Método de Transportes para Resolução do Problema
* Analisar Resultados
* Organizar Banco de Dados
* MVP: Custo por km/rota - Tela de BI

*Sprint* 4 (08/11/24 - 29/11/24):

* Em breve.

Feira de Soluções (29/11/24 - 12/12/24):

* Em breve.

## GitHub.

O GitHub é uma plataforma de hospedagem de código-fonte baseada na *web*, usada principalmente para controle de versão e colaboração em projetos de desenvolvimento de *software*. Os usuários podem colaborar em projetos públicos ou privados e contribuir com código, revisões, discussões e muito mais. A linguagem de programação utilizada para sintetizar os códigos é a Sintaxe de *Markdown*.

A Sintaxe de *Markdown* foi criada em 2004 por John Gruber, *Markdown* é uma ferramenta de formatação de ficheiros de texto. Os ficheiros de texto simples são mais vantajosos que outros formatos, sendo legíveis em praticamente qualquer dispositivo. São mais resistentes ao tempo do que outros tipos de ficheiro, como docx, pois estará familiarizado com as adversidades de compatibilidade envolvidos.

Utilizando essa ferramenta, pode-se produzir ficheiros simples legíveis como texto e podem ser estilizados em outras plataformas como o GitHu0b, que suporta e renderiza em HTML.

Para começar a programar com *Markdown*, existem 4 tipos de cabeçalho, que são renderizados pela # antes do texto de título. Sendo eles:

* # Primeiro Nível;
* ## Segundo Nível;
* ### Terceiro Nível;
* #### Quarto Nivel.

Os códigos acima, renderizados, ficam caracterizados conforme mostra a figura 1.

**Figura 1**

A screenshot of a computer

Description automatically generated

**Fonte**: Do autor.

Para acrescentar ênfase a uma palavra, o texto pode ser escrito em negrito, ou, itálico com os símbolos asterisco (\*\*) para negrito e *underscore* (\_\_) para itálico.

“A \*\*GeoNexus\*\* é uma \_startup\_ nova no mercado.”

Essa linha, renderizada se caracteriza como na figura 2, mostrada abaixo:

**Figura 2**

A black text on a white background

Description automatically generated

**Fonte:** Do autor.

Para elaboração de tabelas, originalmente, *Markdown* não as inclui. No entanto, para criar tabelas no *GitHub* se usam barras (|) para separar entre colunas e hífen (-) entre os cabeçalhos e o restante de itens da tabela. Por exemplo:

| Aluno | Função |

|-------- |-----------|

| Evelyn Victoria P. de Melo | Product Owner |

Que em *Markdown*, é renderizado conforme mostra figura abaixo.

**Figura 3**

A screenshot of a computer

Description automatically generated

Fonte: Do autor.

## Pacote Office.

O Pacote Office foi utilizado para atividades essenciais de gerenciamento de dados e documentação. Especificamente, o Microsoft Excel é uma ferramenta de planilhas eletrônicas amplamente utilizada para a manipulação, organização e análise de dados. Com recursos avançados, como fórmulas, gráficos, tabelas dinâmicas e filtros, o Excel facilita o tratamento de grandes volumes de dados. No contexto deste projeto, o Excel foi empregado para realizar o tratamento básico de bases de dados no formato CSV, permitindo a ordenação e filtragem dos dados, além de uma análise inicial para a tomada de decisões durante o desenvolvimento.

Já o Microsoft Word, é um processador de texto que oferece uma ampla gama de ferramentas para criação, edição e formatação de documentos. No presente projeto, o Word foi utilizado para a confecção do relatório, possibilitando a estruturação do conteúdo de forma clara e dentro das normas de formatação exigidas, como a ABNT, garantindo a apresentação profissional do documento final.

## *Python*.

**Cálculo de Distâncias entre Clientes e Fábricas utilizando *Python* e Pandas**

Aqui foi descrito o processo de criação de um *script* em *Python* que calcula a distância entre clientes e fábricas com base em coordenadas geográficas fornecidas em arquivos CSV. Utilizando bibliotecas como Pandas para manipulação de dados e Geopy para o cálculo das distâncias, o programa calcula e exporta as distâncias em formato CSV para análises posteriores.

Objetivo

Desenvolver um script em Python capaz de ler dados de localização (latitude e longitude) de clientes e fábricas, calcular as distâncias entre eles, e armazenar os resultados em um arquivo.

Metodologia

Para o desenvolvimento do código, foram seguidas as seguintes etapas:

1. Montagem do Google Drive: Utilizou-se a linha drive.mount('/content/drive') para carregar o diretório do Google Drive, onde se encontram os arquivos de entrada e saída.

2. Carregamento dos Arquivos CSV: Com o uso do Pandas, os arquivos CSV dos clientes e das fábricas foram lidos. Utilizou-se o delimitador ponto e vírgula (sep=';') e o encoding 'ISO-8859-1' para compatibilidade com o conteúdo dos arquivos.

3. Formatação dos Dados:

* Remoção de Espaços em Branco: clientes.columns.str.strip() e fabricas.columns.str.strip() para limpar os nomes das colunas.
* Conversão das Coordenadas: Como algumas coordenadas estavam em formato de string com vírgulas, criou-se a função converter\_coordenadas para convertê-las em valores numéricos.

4. Cálculo de Distâncias:

* Para cada par cliente-fábrica, a função calcular\_distancia calcula a distância em quilômetros, utilizando a função geodesic da biblioteca Geopy.
* Armazena-se cada cálculo em uma lista de dicionários, posteriormente convertida em DataFrame.

5. Armazenamento e Exportação dos Dados:

* O DataFrame resultante das distâncias calculadas é exportado para um arquivo CSV no Google Drive, com a precisão das distâncias arredondada para duas casas decimais.

**Resultados**

O código processa todos os pares cliente-fábrica e armazena as distâncias em um arquivo CSV. Os dados podem então ser usados para otimização logística ou análises de custo-benefício.

Para realizar a Pesquisa Operacional utilizando o método de transportes, informações como demanda por cliente e custo por unidade transportada foram necessárias. Os dados abaixo foram extraídos da base “Rotas” oferecida pelo cliente:

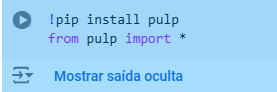
- Quantidade média adquirida por cliente (demanda média por cliente);

- Valor médio de frete por unidade transportada (custo médio por unidade transportada).

Obs.: Os dados foram calculados de forma automática através da criação de tabelas dinâmicas em Excel.

Após a aquisição dessas informações, o próximo passo foi seguir para o Python. A Pesquisa Operacional foi realizada de acordo com duas videoaulas do Professor Mestre Marcus Nascimento acompanhadas pelo grupo. Abaixo, há o passo a passo aplicado:

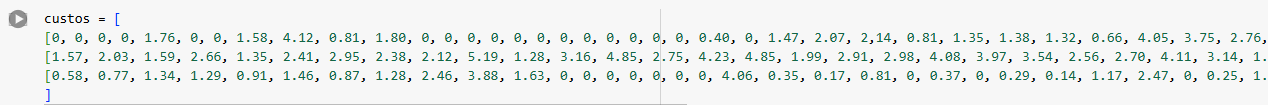
Acionar a biblioteca “Pulp”:

****

Adicionar os valores de capacidades e demandas para cada fábrica e cada cliente, respectivamente:



Adicionar custos por unidade de frete de cada fábrica para cada cliente:



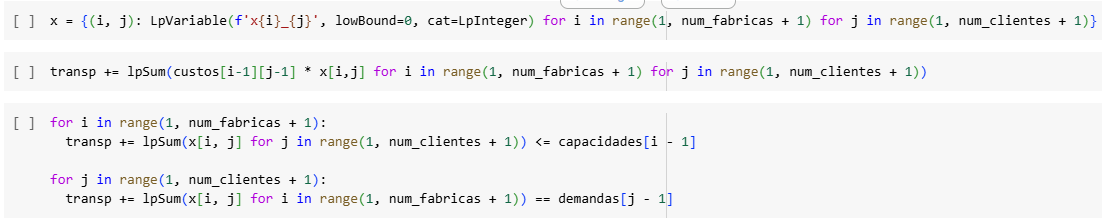
Informar a quantidade de fábricas e a quantidade de clientes:



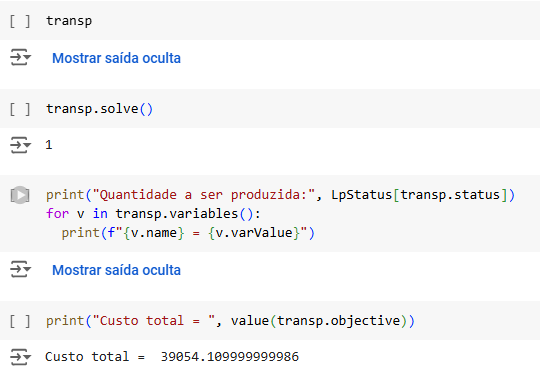
Informar a forma de otimização desejada – Método de Transportes, no caso:



Iniciar a aplicação do Método:



Finalizar o Método solicitando os resultados:



Dessa forma, o grupo adquiriu os resultados e, na próxima Sprint espera apresentar uma análise mais detalhada dos dados alcançados.

## MySQL

Este capítulo detalha os passos necessários para criar um banco de dados relacional utilizando o MySQL. O objetivo é transformar três arquivos CSV em tabelas de um banco de dados, além de criar consultas (*view*s) que facilitem a visualização dos dados. O processo inclui a criação das tabelas, importação dos dados, e a criação das *view*s.

A seguir, estão as explicações e os comandos para realizar cada etapa:

**Passo 1: Criação do Banco de Dados**

Um banco de dados é uma coleção de tabelas relacionadas, projetadas para armazenar dados de maneira eficiente e organizada. Para criar o banco de dados no MySQL, utilizamos o comando CREATE DATABASE, seguido por USE para selecionar o banco de dados ativo.

Exemplo de comando SQL:

CREATE DATABASE LogisticaDB;

USE LogisticaDB;

CREATE DATABASE: Cria um novo banco de dados no servidor MySQL.

USE: Define o banco de dados que será usado para as operações subsequentes.

**Passo 2: Criação das Tabelas**

No MySQL, uma tabela é uma estrutura composta por colunas (atributos) e linhas (tuplas) que armazenam os dados. Cada coluna deve ter um tipo de dado específico, como texto, números inteiros ou decimais.

**Tabela Clientes**

Esta tabela armazenará informações dos clientes, como código, município, latitude e longitude. Para garantir a integridade dos dados, a coluna CO\_Cliente será usada como chave primária, ou seja, um identificador único para cada cliente.

Exemplo de comando SQL:

CREATE TABLE Clientes (

CO\_Cliente INT PRIMARY KEY,

MUN VARCHAR(100),

LAT DECIMAL(10, 7),

LONG DECIMAL(10, 7)

);

INT: Tipo de dado numérico inteiro.

VARCHAR(100): Armazena textos de até 100 caracteres.

DECIMAL(10, 7): Tipo de dado para valores numéricos com casas decimais, útil para latitudes e longitudes.

PRIMARY KEY: Define a coluna CO\_Cliente como a chave primária da tabela, garantindo que os valores dessa coluna sejam únicos.

**Tabela Fábricas**

Essa tabela contém informações sobre as fábricas, como código da fábrica, município, estado, latitude e longitude. O código da fábrica (CO\_Fabrica) também é definido como chave primária.

Exemplo de comando SQL:

CREATE TABLE Fabricas (

CO\_Fabrica INT PRIMARY KEY,

NO\_MUN VARCHAR(100),

NO\_MUN\_MIN VARCHAR(50),

SG\_UF VARCHAR(2),

LAT DECIMAL(15, 8),

LONG DECIMAL(15, 8)

);

SG\_UF VARCHAR(2): Armazena a sigla do estado em dois caracteres (ex.: SP, RJ).

DECIMAL(15, 8): Precisa maior de precisão para latitudes e longitudes.

**Tabela Rotas**

A tabela Rotas relaciona as fábricas e os clientes, registrando pedidos, datas, transporte e valores. Essa tabela possui chaves estrangeiras (FOREIGN KEY) para assegurar a integridade dos dados, vinculando o código da fábrica e o código do cliente às tabelas correspondentes.

Exemplo de comando SQL:

CREATE TABLE Rotas (

Dt\_Pedido DATE,

Dt\_Emissao DATE,

Dt\_Entrega DATE,

Mes\_Base INT,

Ano\_Exec INT,

CO\_Fabrica INT,

CO\_Cliente INT,

Incoterm VARCHAR(20),

Veiculo VARCHAR(50),

Qtd\_pallets INT,

Qtd\_Transp DECIMAL(10, 2),

Moeda VARCHAR(3),

Vlr\_Frete DECIMAL(10, 2),

FOREIGN KEY (CO\_Fabrica) REFERENCES Fabricas(CO\_Fabrica),

FOREIGN KEY (CO\_Cliente) REFERENCES Clientes(CO\_Cliente)

);

FOREIGN KEY: Garante que os valores em CO\_Fabrica e CO\_Cliente existam nas tabelas Fabricas e Clientes, respectivamente. Isso evita inconsistências nos dados.

DATE: Tipo de dado para armazenar datas.

DECIMAL(10, 2): Valores decimais com 10 dígitos no total e 2 casas decimais, adequados para valores monetários ou quantidades transportadas.

**Passo 3: Importação dos Dados**

Após criar as tabelas, é necessário inserir os dados dos arquivos CSV. Isso pode ser feito utilizando o comando LOAD DATA INFILE, que permite carregar dados diretamente de arquivos para as tabelas do MySQL. Certifique-se de que os arquivos CSV estejam em um diretório acessível pelo MySQL.

**Importando dados para a tabela Clientes:**

LOAD DATA INFILE '/caminho\_para\_arquivo/Clientes.csv'

INTO TABLE Clientes

FIELDS TERMINATED BY ';'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(CO\_Cliente, MUN, LAT, LONG);

FIELDS TERMINATED BY: Define o delimitador de campos no arquivo CSV (neste caso, ponto e vírgula).

LINES TERMINATED BY: Define o delimitador de linha (neste caso, nova linha).

IGNORE 1 LINES: Ignora a primeira linha do arquivo CSV (os cabeçalhos das colunas).

**Importando dados para a tabela Fabricas:**

LOAD DATA INFILE '/caminho\_para\_arquivo/Fabricas.csv'

INTO TABLE Fabricas

FIELDS TERMINATED BY ';'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(CO\_Fabrica, NO\_MUN, NO\_MUN\_MIN, SG\_UF, LAT, LONG);

**Importando dados para a tabela Rotas:**

LOAD DATA INFILE '/caminho\_para\_arquivo/Rotas.csv'

INTO TABLE Rotas

FIELDS TERMINATED BY ';'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(Dt\_Pedido, Dt\_Emissao, Dt\_Entrega, Mes\_Base, Ano\_Exec, CO\_Fabrica, CO\_Cliente, Incoterm, Veiculo, Qtd\_pallets, Qtd\_Transp, Moeda, Vlr\_Frete);

**Passo 4: Criação das *View*s**

Uma *view* é uma consulta armazenada no banco de dados, que apresenta uma "visão" específica dos dados. As *view*s são úteis para simplificar consultas complexas, fornecer uma camada de segurança e abstrair os detalhes da estrutura de tabelas.

***View* para Clientes:**

CREATE VIEW View\_Clientes AS

SELECT CO\_Cliente, MUN, LAT, LONG

FROM Clientes;

Essa *view* facilita a consulta dos clientes e suas localizações sem a necessidade de interagir diretamente com a tabela.

***View* para Fabricas:**

CREATE VIEW View\_Fabricas AS

SELECT CO\_Fabrica, NO\_MUN, SG\_UF, LAT, LONG

FROM Fabricas;

Essa *view* exibe as fábricas e seus respectivos municípios e coordenadas geográficas.

***View* para Rotas:**

CREATE VIEW View\_Rotas AS

SELECT Dt\_Pedido, Dt\_Entrega, CO\_Fabrica, CO\_Cliente, Qtd\_pallets, Vlr\_Frete

FROM Rotas;

A view “View\_Rotas” facilita a visualização das rotas de entrega entre fábricas e clientes, incluindo a quantidade transportada e o valor do frete.

Anotações Gerais ao Longo do Desenvolvimento da Sprint: gráfico valor de frete por veículo - fazer ele por linhas utilizando a produtividade dos veículos quanto foi utilizado dividido pela capacidade. Valor do frete por município - alterar para um card com o frete total lead time (diferença entre tempo de pedido e data da entrega) fazer uma média e dividir. Imput - Capacidade de veículo. Output quanto foi utilizado, quantidade transportada dividida pela capacidade por porcentagem de produtividade mês a mês. Fábricas, cliente, veículos e incoterm.

## PowerBI

Power BI é uma ferramenta de business intelligence (BI) da Microsoft que permite aos usuários criarem relatórios e painéis de controle interativos para visualizar e analisar dados. Com o Power BI, é possível:

* Principais recursos:

1. Conexão a diversas fontes de dados (banco de dados, planilhas, serviços de nuvem)

2. Criação de relatórios e painéis personalizados

3. Visualizações de dados dinâmicas (gráficos, tabelas, mapas)

4. Análise de dados em tempo real

5. Compartilhamento de relatórios e painéis com outros usuários

6. Integração com outras ferramentas da Microsoft (Excel, Azure, Dynamics)

* Vantagens:

1. Facilidade de uso

2. Flexibilidade na criação de visualizações

3. Capacidade de análise de grandes conjuntos de dados

4. Integração com outras ferramentas da Microsoft

5. Custo competitivo

* Tipos de usuários:

1. Analistas de dados

2. Profissionais de BI

3. Gerentes e executivos

4. Usuários finais que precisam de insights de dados

* Versões:

1. Power BI Desktop (gratuito)

2. Power BI Pro (assinatura mensal)

3. Power BI Enterprise (assinatura anual)

4. Power BI Embedded (para desenvolvedores)

* Conhecimentos necessários:

1. Conhecimento básico de SQL

2. Familiaridade com ferramentas de BI

3. Conhecimento de dados e análise

4. Habilidades de visualização de dados

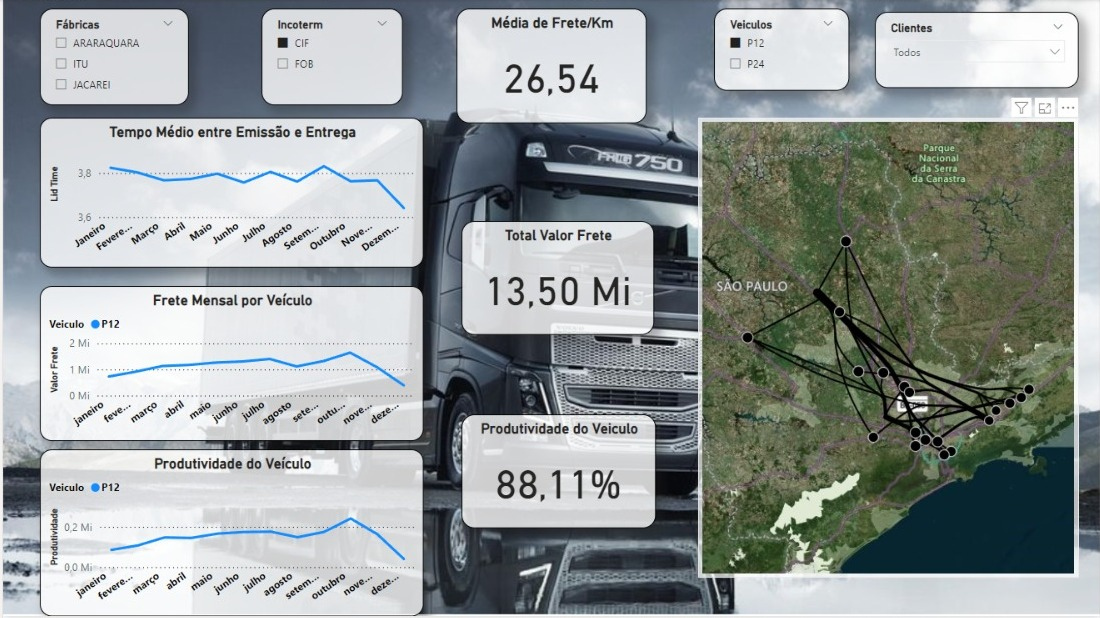
* Três gráficos de linha:
* Tempo médio entre emissão e entrega;
* Frete Mensal por veículo;
* Produtividade do veículo.
* Tr|ês cartões:
* Total valor Frete;
* Produtividade do veículo por porcentagem;
* Cálculo de Frete/Km

A partir dos resultados adquiridos no cálculo das distâncias entre fábricas e clientes através do Python, realizamos uma correlação desses dados com a base “Rotas” oferecida pelo cliente. Para extrair a informação de valor de frete por quilômetro, o procedimento abaixo foi realizado:

- A coluna “VLR Frete” foi dividida pela coluna “Distância” (adquirida a partir do cálculo em Python), resultando na coluna “Frete/km”:



**Figura 1: Protótipo de BI**



**Fonte:** Do autor.

# RESULTADOS ESPERADOS

Esperamos que, nesta *sprint*, a consolidação dos arquivos seja concluída com sucesso, permitindo uma integração eficiente dos dados. Além disso, buscamos apresentar ao cliente uma visualização aprimorada do projeto, por meio de um protótipo mais funcional em Power BI. Dessa forma, o foco para a próxima *sprint* será apenas na realização de ajustes finais, garantindo que o projeto esteja pronto para sua entrega.

# REFERÊNCIAS

*OpenAI*. (2024). *ChatGPT* (*Version* 2) [Software]. Disponível em: https://openai.com/chatgpt. Acesso em: 16 out. 2024.

W3Schools. SQL Tutorial. Disponível em: https://www.w3schools.com/sql/. Acesso em: 08 out. 2024.

SQLZoo. SQL Zoo Interactive Tutorial. Disponível em: https://sqlzoo.net/. Acesso em: 12 out. 2024.

Khan Academy. Intro to SQL: Querying and Managing Data. Disponível em: https://www.khanacademy.org/computing/computer-programming/sql. Acesso em: 12 out. 2024.

SQL-Easy. SQL Easy: Learn SQL Online. Disponível em: https://www.sql-easy.com/. Acesso em: 16 out. 2024.

Codecademy. Learn SQL. Disponível em: https://www.codecademy.com/learn/learn-sql. Acesso em: 16 out. 2024.

META PLATFORMS, Inc. Meta AI – WhatsApp. Versão 3.2. Califórnia: Meta, 2024. Programa de computador