

## PROJETO INTEGRADOR – SPRINT 3

Ana Júlia do Couto Brandão (Linkedin - <https://l1nk.dev/3PMzX>)

Hellen de Sousa Santos Carmo (Linkedin - <https://l1nk.dev/tlpKY>)

Marcos Vinicius Restani Avanzini (Linkedin - <https://acesse.one/pVKdh>)

Pedro Luis Cordeiro Dias Lourenço (Linkedin - <https://l1nk.dev/7NxUN>)

Steffany Caroline Vieira Santo (Linkedin - <https://l1nk.dev/0ycO6>)

Thayssa Andrade Santos (Linkedin - <https://l1nk.dev/BuKW9>)

Professor M2: Carlos Eduardo Bastos

Professor P2: Marcus Vinicius do Nascimento

### Resumo da Sprint 3:

Nessa sprint, foi utilizado o MySQL para formar o desenvolvimento do Power Bi, criando tela inicial com principais dados, como capacidade das fábricas, top cliente por frete, quantidade transportada e gráfico de evolução do transporte durante o ano.

Com o Python, utilizou-se a biblioteca PULP para fazer a otimização de custo, definindo então as variáveis, por exemplo os clientes, fabricas e custo entre eles, com isso concluiu-se as restrições de demanda e capacidade.

Realizou-se modificação nos códigos para otimização de custo e logo após executou-se o código para gerar um plano ideal.

Após realização acima, o próximo passo conectar código com o MySQL e inserir resultado da otimização no Power Bi

Palavras-Chave: Power Bi, Python e MySQL

## Abstract:

In this sprint, MySQL was used to develop Power Bi, creating an initial screen with the main data, such as factory capacity, top customer by freight, quantity transported and a graph showing the evolution of transportation over the year.

With Python, the PULP library was used to optimize the cost, defining the variables, such as customers, factories and the cost between them, thus completing the demand and capacity restrictions.

Modifications were made to the codes to optimize costs and immediately after executing the code to generate an ideal plan.

After completing the above, the next step was to connect the code to MySQL and insert the optimization results into Power Bi.

Keywords: Power Bi, Python e MySQL.

## 1. Contextualização da Sprint 3

Modelar as visões para o padrão previsto;

Utilização do modelo definido para executar a otimização;

Criar Visualização de otimização de rotas por veículo;

Montar visualizações iniciais;

Criar relatório da Sprint;

Tire dúvidas com o cliente;

Atualizar GitHub;

Criar apresentação para sprint.

## 2. Objetivos do projeto

Analisar a produtividade dos veículos, otimização de distribuição e custo de rotas de uma empresa embarcadora de carga, por meio de:

- i) Criação e modelagem do banco de dados em SQL;
- ii) Criação de um visualizador de indicadores em BI;
- iii) Aplicação do método de transportes para otimizar a distribuição.

### **3. Tecnologias utilizadas**

Estas foram as tecnologias solicitadas a serem utilizadas pelo cliente para a solução do projeto.

#### *3.1. Jira software*

Jira é um software comercial desenvolvido pela empresa Australiana Atlassian. É uma ferramenta que permite o monitoramento de tarefas e acompanhamento de projetos garantindo o gerenciamento de todas as suas atividades em único lugar.

#### *3.2. GitHub*

GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.

#### *3.3. Copilot*

A função do Copilot, é fornecer suporte em diversas áreas, desde responder a perguntas e fornecer informações até auxiliar na resolução de problemas e oferecer assistência em tarefas específicas.

#### *3.4. Python*

Python é uma linguagem versátil e poderosa usada em uma variedade de domínios, desde o desenvolvimento de software até análise de dados e automação

de tarefas. Sua popularidade crescente é atribuída à sua facilidade de uso e à vasta comunidade de desenvolvedores que apoiam.

### 3.5. *MySQL*

O MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) amplamente utilizado em muitas aplicações de software. Sua função principal é armazenar, organizar e gerenciar grandes volumes de dados de forma eficiente e segura.

### 3.6. *Power Bi*

O Power Bi tem a função de manipular e transformar os dados, além de oferecer visualizações usadas para personalizar a aparência e o comportamento dos gráficos e visualizações em seu projeto

### 3.7. *Canva*

Foi usado para criar a apresentação da nossa sprint, pois com o Canva, os usuários têm acesso a uma vasta biblioteca de modelos, imagens, ilustrações, ícones, fontes e outros elementos gráficos, facilitando a criação de designs profissionais com pouco esforço. Além disso, a plataforma oferece ferramentas de edição simples de usar, como arrastar e soltar, redimensionamento, ajuste de núcleos e adição de texto, tornando o processo de design rápido e eficiente.

## 4. **Coleta e descrição dos dados utilizados**

A base de dados foi fornecida pelo cliente, onde contém conhecimentos de rotas, fabricas e clientes.

- Rotas: Dt. Emissao; Dt. Entrega; Mes. Base; Ano. Exerc; CO. Fabrica; Cliente; Incoterm; Veículo; Qtd/pallet; Qtd. Transp; Moeda; Vlr. Frete; Dist.

- Fabricas: CO. Fabrica; NO\_MUN; NO\_MUN\_MIN; SG\_UF; LAT; LONG.
- Clientes: CO. Cliente; MUN; LAT; LONG.

## 5. Resultados

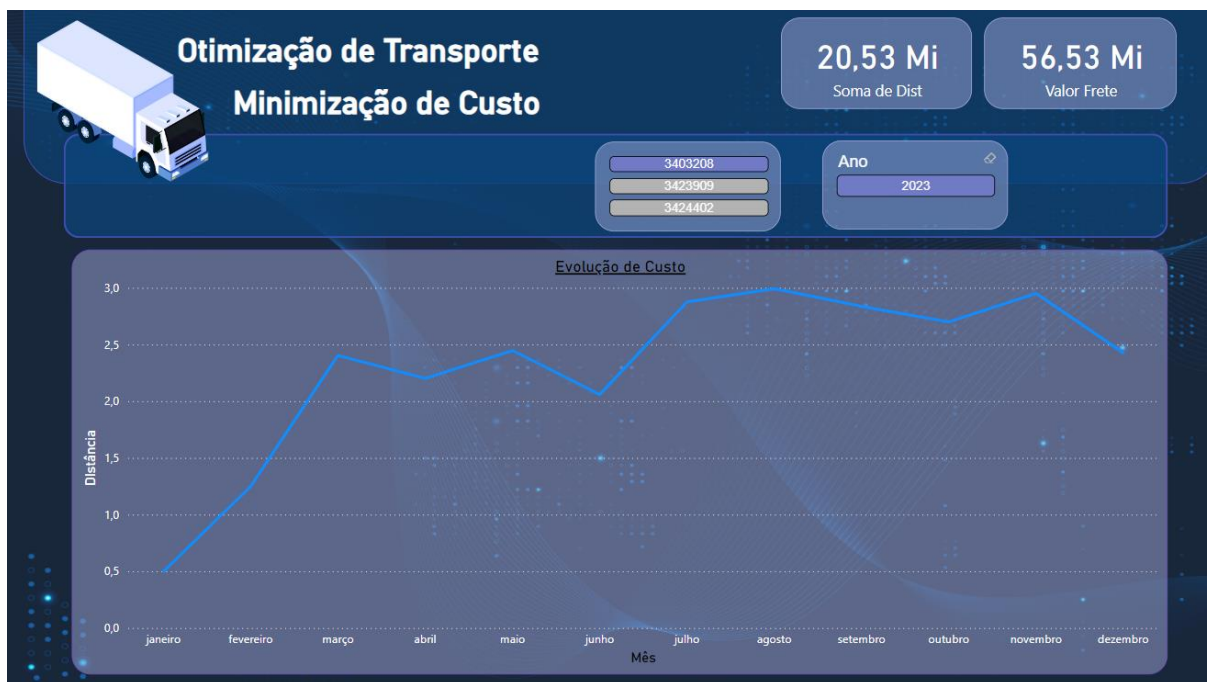
Para a criação do Dashboard no Power Bi, organizou-se o layout de modo a facilitar o acesso do cliente às informações, para isso, desenvolveu-se a parte estética, selecionou-se as informações que serão mostradas na plataforma de forma que possibilite uma fácil visualização e interpretação das informações ali presentes. Na parte superior do Dashboard, foram posicionados os botões de seleção, enquanto a parte inferior foi dedicada à exibição dos gráficos de KPIs.

Imagem 1 – Evolução Output e Input



Fonte: Própria autoria

Imagem 2 – Evolução dos custos por km.



Fonte: Própria autoria

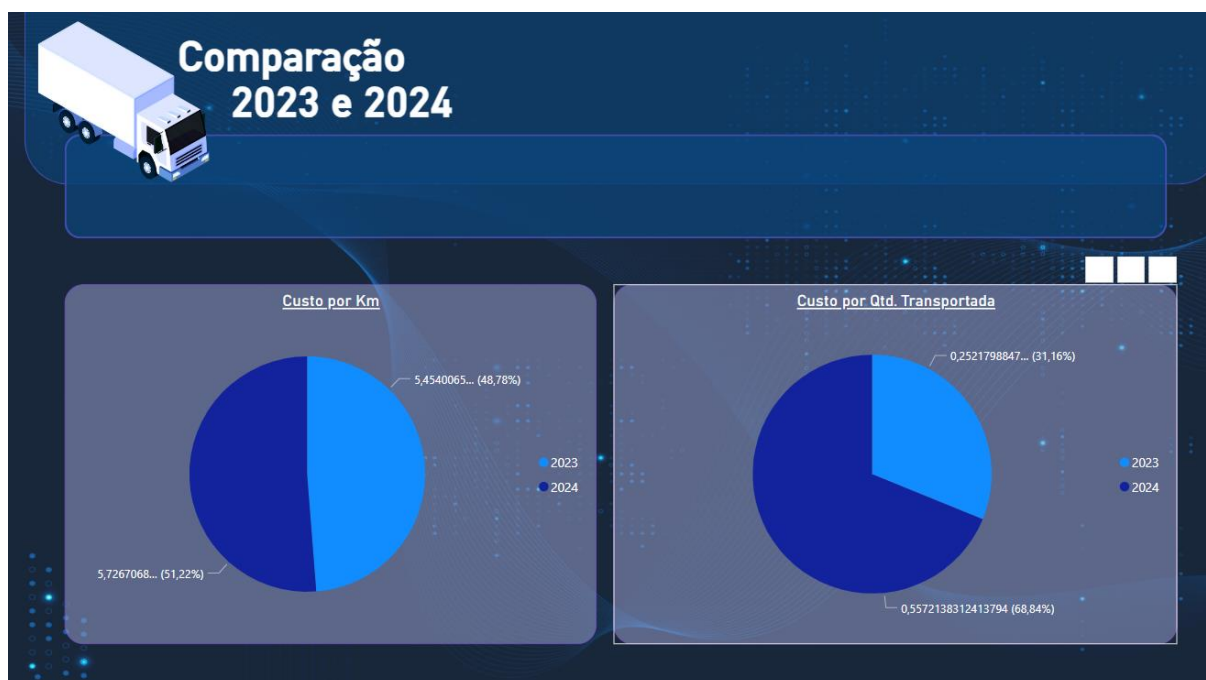
Imagem 3 – Evolução dos custos por quantidade transportada.



Fonte: Própria autoria



Imagem 4 – Comparação 2023 e 2024.



**Fonte:** Própria autoria

Os gráficos acima incluem: Produtividade mensal dos veículos; Evolução dos custos por km de cada rota, filtrando por fabricas; Evolução dos custos por unidade transportada, filtrando por fabricas.

Para otimização de minimização de custos, utilizamos programação em Python para execução do código. Inicialmente, é importada a biblioteca Pulp para definir o problema como um modelo de programação linear, especificando as variáveis de decisão, restrições e a função objetivo. As variáveis do código representam as fábricas (plants), os clientes (customers) e o custo de transporte entre eles (transportation\_costs), identificados por códigos.

Observação: Algumas fábricas foram identificadas como não realizando entregas para determinados clientes, resultando em um custo unitário de transporte igual a zero. Isso ocorre porque não há histórico de transporte entre essas fábricas e clientes específicos, e não há expectativa de que ocorram no futuro. Para evitar que o código de otimização considere erroneamente esses casos como possíveis rotas, foi implementado um mecanismo de filtragem dos custos de transporte iguais a zero. Isso cria um dicionário de custos filtrados, garantindo que esses custos sejam excluídos durante o processo de otimização.

**Imagem 5 – Variáveis Fábricas e Clientes**

```
# Define the problem data
plants = [3403208, 3423909, 3424402] # 3 plants
customers = [2301, 2302, 2303, 2304, 2305,
             2306, 2307, 2308, 2309, 2310,
             2311, 2312, 2313, 2314, 2315,
             2316, 2317, 2318, 2319, 2320,
             2321, 2322, 2323, 2324, 2325,
             2326, 2327, 2328, 2329, 2330,
             2331, 2332, 2333, 2334, 2335,
             2336, 2337, 2338, 2339, 2340,
             2341, 2342, 2343, 2344, 2345,
             2346, 2347, 2348, 2349, 2350,
             2351] # 51 customers
```

**Fonte:** Própria autoria

**Imagem 6 – Variável Custo de Transporte**

```
transportation_costs = {
    (3403208, 2301): 0.581689889,
    (3403208, 2302): 0.479859,
    (3403208, 2303): 0.440461389,
    (3403208, 2304): 0.456173529,
    (3403208, 2305): 0.492159853,
    (3403208, 2306): 0.544231152,
    (3403208, 2307): 0.531967,
    (3403208, 2308): 0.540897912,
    (3403208, 2309): 0.424910965,
    (3403208, 2310): 0.714332699,
    (3403208, 2311): 0.403713274,
    (3403208, 2312): 0.788882677,
    (3403208, 2313): 0.753914848,
    (3403208, 2314): 0.707153831,
    (3403208, 2315): 0.825211267,
    (3403208, 2316): 0.834051632,
```

**Fonte:** Própria autoria

Para calcular o custo unitário do Transporte foi utilizado um código no MySQL o qual calcula o custo unitário de transporte para o ano de 2024, considerando um aumento de 5% no valor do frete (porcentagem dada pelo cliente). Ele seleciona dados da tabela "rotas", onde o termo de comércio é 'CIF', agrupa os resultados por fábrica e cliente, e então calcula o custo unitário dividindo a soma dos valores do frete ajustados pela quantidade total transportada.



## Imagem 7 – Cálculo Custo de Transporte

```

1
2 • select
3
4     `CO.Fabrica`,
5
6     `CO.Cliente`,
7
8     SUM(`Vlr.Frete` * 1.05)/SUM(`Qtd.Transp`) as `Custo Unitário Transporte 2024`
9
10  from rotas
11
12  Where Incoterm = 'CIF'
13
14  group by
15
16     `CO.Fabrica`,
17
18     `CO.Cliente`
19

```

**Fonte:** Própria autoria

Já as restrições do código são definidas pela demanda dos clientes (customer\_demands) e a capacidade das fábricas (plant\_supplies), também identificadas por códigos.

## Imagem 8 – Restrição Demanda Cliente

```

customer_demands = {
    2301: 5973721,
    2302: 1778080,
    2303: 5958798,
    2304: 896173,
    2305: 3241494,
    2306: 3244827,
    2307: 12738726,
    2308: 6792503,
    2309: 7471374,
}

```

**Fonte:** Própria autoria

**Imagem 9 – Restrição Capacidade Fábricas**

```
plant_supplies = {
    3403208: 90000000,
    3423909: 90000000,
    3424402: 90000000
}
```

**Fonte:** Própria autoria

Após a definição das restrições e das variáveis, um código de otimização foi elaborado para minimizar o custo total de transporte. Isso é feito somando os custos de transporte para todas as combinações de plantas e clientes. Utiliza-se a função `LpMinimize` da biblioteca PuLP para realizar a minimização com base em programação linear.

**Imagem 10 – Função Objetivo**

```
# Create the linear programming problem
prob = LpProblem("Transportation_Cost_Minimization", LpMinimize)

# Define the decision variables
shipments = LpVariable.dicts("Shipments", (plants, customers), 0, None, cat='Integer')

# Define the objective function
try:
    prob += lpSum([transportation_costs[p, c] * shipments[p][c] for p in plants for c in customers])
except KeyError as e:
    print(f"Missing combination: {e}")

# Define the constraints
for p in plants:
    prob += lpSum([shipments[p][c] for c in customers]) <= plant_supplies[p]

for c in customers:
    prob += lpSum([shipments[p][c] for p in plants]) == customer_demands[c]
```

**Fonte:** Própria autoria

Depois de executar o código se obtêm o valor objetivo da minimização de custos e quais fabricas supriram as demandas.

**Imagem 11 – Resultado Total**

```
Objective value:          96704198.52407421
Enumerated nodes:         0
Total iterations:         0
Time (CPU seconds):       0.02
Time (Wallclock seconds): 0.02
```

**Fonte:** Própria autoria

## Imagem 12 – Resultado Fábrica por Cliente

```
Status: Optimal
Shipments_3403208_2301 = 0.0
Shipments_3403208_2302 = 1778080.0
Shipments_3403208_2303 = 5958798.0
Shipments_3403208_2304 = 896173.0
Shipments_3403208_2305 = 0.0
Shipments_3403208_2306 = 0.0
Shipments_3403208_2307 = 7344603.0
Shipments_3403208_2308 = 0.0
Shipments_3403208_2309 = 7471374.0
Shipments_3403208_2310 = 0.0
Shipments_3403208_2311 = 7295028.0
```

Fonte: Própria autoria

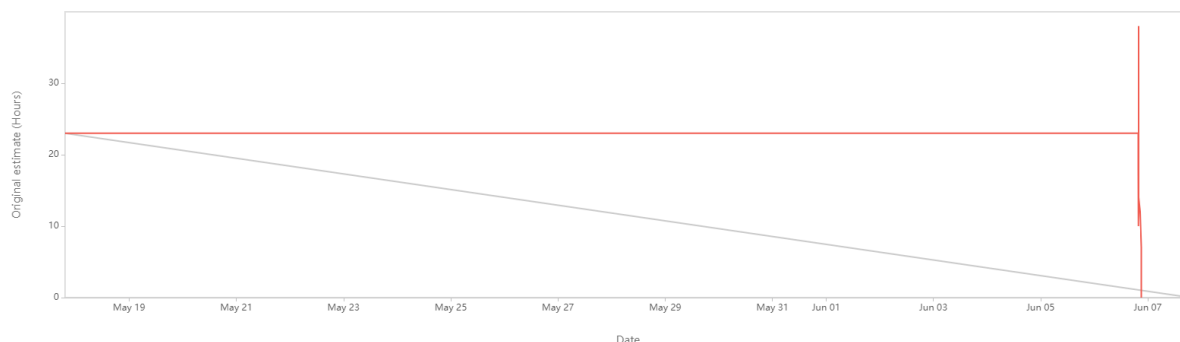
Para facilitar a visualização do progresso do projeto, foi desenvolvido um burndown da sprint diretamente no Jira Software. Isso permite acompanhar de perto o avanço das atividades e identificar áreas que podem ser melhoradas em relação ao gerenciamento do tempo durante o desenvolvimento. Além disso, o burndown também ajuda a identificar possíveis atrasos e tomar medidas corretivas de forma proativa para garantir que os objetivos da sprint sejam alcançados dentro do prazo estabelecido.

## Imagem 13 – Desenvolvimento do projeto

Data - 17 de maio de 2024 - 7 de junho de 2024

Trabalho restante  
Horas de trabalho restantes para concluir este sprint

Diretriz  
Taxa de queima ideal



Fonte: Própria autoria

Para fácil visualização do andamento do projeto foi desenvolvido no próprio Jira Software o burndown da sprint para assim notar o que podemos melhorar em questão ao tempo no desenvolvimento.