

## PROJETO INTEGRADOR – SPRINT 2

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 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 2:

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.

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.

The code was then run to generate an optimal 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 2

Verificar possibilidade de views complementares;

Criar relatório da sprint;

Tirar dúvidas com o cliente;

Fazer Power Point para apresentação da sprint;

Corrigir outliers e erros da base;

Definir modelo de otimização;

Criar estrutura de dados Power Bi;

Atualizar GitHub;

Montar views iniciais para o projeto;

Criar padrão das views para Power Bi;

Criar código inicial para otimização;

Fazer conexão MySQL e Power Bi;

Criar os KPIs.

## **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 – Tela inicial no Power Bi



Fonte: Própria autoria

Os gráficos acima incluem: evolução do transporte, detalhado por frete e volume transportado; desempenho de capacidade das fábricas; ranking dos cinco

principais clientes por valor de frete e por volume transportado; análise da quantidade transportada por veículo; além de um mapa detalhado que mostra os locais de cada cliente com as respectivas quantidades entregues.

após a base de dados ter sido implementada no Power Bi, foi feito a análise da melhor forma de apresentá-los em um dashboard. Escolhendo o melhor gráfico para podermos mostrar os dados, tais como: top 5 clientes (por frete e quantidade), evolução do frete através de cada mês, identificação de veículos por quantidade transportada, etc.

após apresentado os erros de valores de frete e frete CIF ou FOB para nosso cliente, foi-nos passado que poderíamos excluí-los e desconsiderarmos eles para otimizarmos o transporte e minimizar o custo de frete.

Para a otimização de minimização de custos utilizou-se a biblioteca PuLP para definir o problema como um modelo de programação linear, especificando as variáveis de decisão, restrições e a função objetiva.

Primeiro foi feito a separação das variáveis como o código da fábrica, código de cliente e custo de transporte entre eles

## Imagem 2 – Definindo Variáveis

```
# 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

Depois definimos as restrições de demanda dos clientes e capacidade das fabricas

Obs.: Algumas fabricas não enviam para alguns clientes com isso para esse caminho não seja interpretado pelo código deixamos o preço de transporte como infinito para que ele nunca opte a utilizar esse caminho.

### Imagem 3 – Definindo Restrições

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

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

Fonte: Própria autoria

Depois fizemos a definição da função objetiva

### Imagem 4 – Definindo Função Objetiva

```
# 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:
```

Com esse código foi definida a LpMinimize que faz parte da biblioteca PuLP para que ele faça a minimização baseada em uma programação linear.

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

### Imagem 5 – Função Objetiva.

```
Result - Optimal solution found

Objective value:           4179880.94437757
Enumerated nodes:           0
Total iterations:           0
Time (CPU seconds):         0.01
Time (Wallclock seconds):   0.01
```

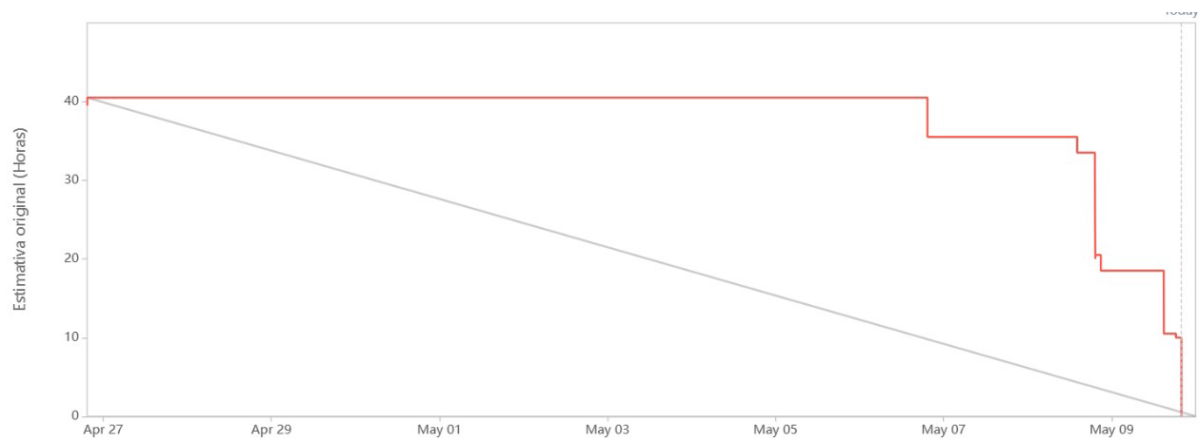
Fonte: Própria autoria

**Imagem 6 – Função Objetiva.**

```
Status: Optimal
Shipments_3403208_2301 = 0.0
Shipments_3403208_2302 = 0.0
Shipments_3403208_2303 = 0.0
Shipments_3403208_2304 = 896173.0
Shipments_3403208_2305 = 0.0
Shipments_3403208_2306 = 1525159.0
Shipments_3403208_2307 = 0.0
Shipments_3403208_2308 = 6792503.0
```

Fonte: Própria autoria

**Imagem 7 – Desenvolvimento do projeto**



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.