

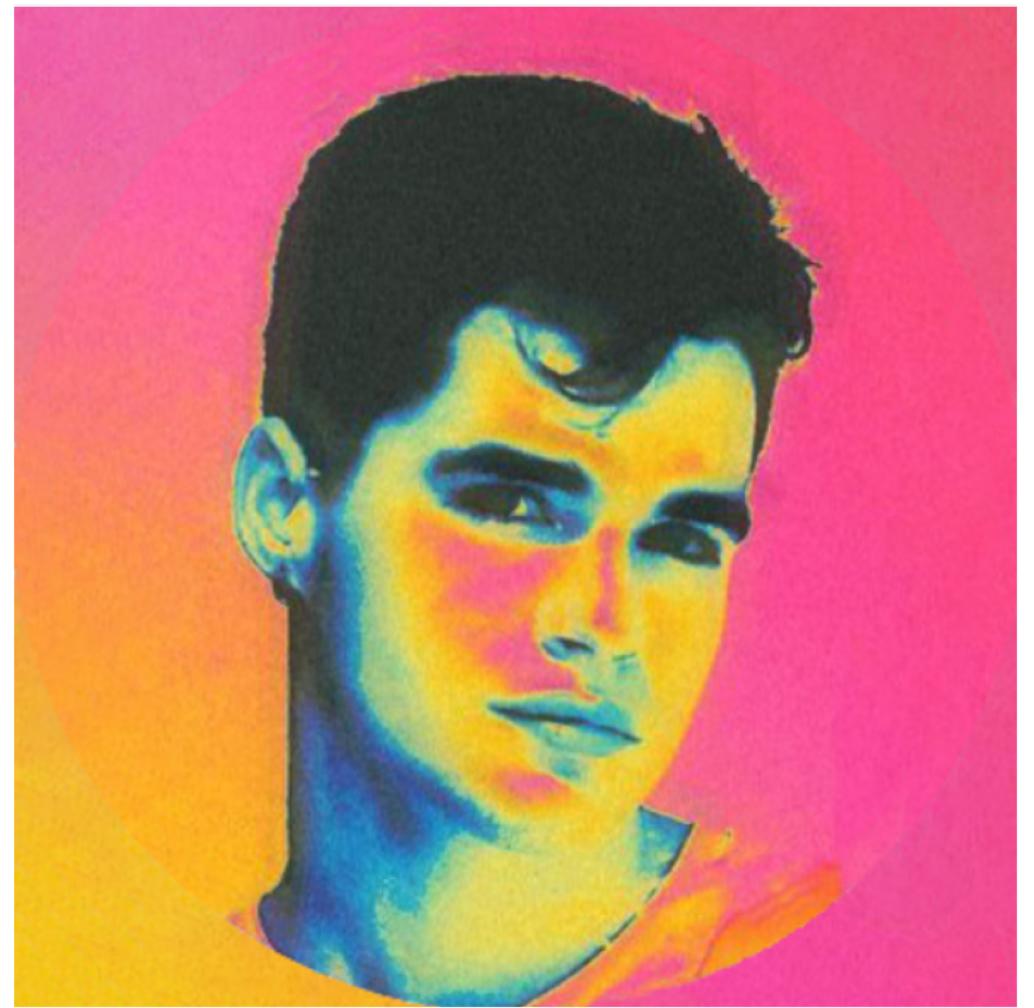


SPRINT 2

Otimização para rotas de leitura de hidrômetros de consumo mensal



Time



Antônio Moraes



Heloisa Oliveira



Patrick Savoia



Enzo Bressane



Lucas Nunes



Ever Felliphe

Sprints

Sprint 1

Entendimento do contexto do problema
Entendimento de Negócio
Entendimento da Experiência do Usuário

Sprint 2

Entendimento do contexto do problema
Resolução de uma versão simplificada do problema.
Protótipo do front-end
Artigo - versão inicial

Sprint 3

Planejador e back-end da aplicação
Front-end da aplicação
Artigo - 2a versão

Sprint 4

Complexidade e corretude do algoritmo
Aplicação
Artigo

Sprint 5

Preparação para a entrega
Artigo completo
Testes de usabilidade
Apresentação Final

Sumário

- Introdução ao Problema
- Transição de Modelos Matemáticos
- Tratamento de Dados
- Parâmetros de Entrada
- Saída dos Dados
- Clusterização
- Implementação Inicial do Algoritmo
- Wireframe
- Documentação e Artigo

DESLOCAMENTO

ATRASO TEMPO

ACIDENTES RISCO

HORAS EXTRAS

CUSTOS TRANSPORT

Recapitulando a Sprint 1

Restrições Do Projeto

$$\frac{\sum_{i=1}^n \sum_{j \neq i, j=1}^n c_{ij} \cdot x_{ij}}{velocidade} + \frac{2 \cdot n}{60min} \leq 22dias \cdot 6h/dia = 132h$$

Tratamento de Dados

- Otimização dos dados para o algoritmo
- Endereços repetidos
- Inconsistencias dos dados (formatações erradas e valores nulos)

Dados nulos

```
In [ ]: # Checagem por dados nulos
null_mask = total_sample.isnull().any(axis=1)
null_rows = total_sample[null_mask].count().max()

print(f"Há {null_rows} linhas com dados nulos na amostra total")
```

```
In [ ]: # Visualização dos dados nulos
total_sample[null_mask].head()
```

Ao que tudo indica, os dados nulos se referem a logradouros sem número registrado. Essa hipótese é verificada abaixo.

```
In [ ]: # Visualização de registros nulos em cada coluna
for col, null_count in total_sample[null_mask].count().items():
    if null_rows == null_count:
        print(f'A coluna "{col}" não contém registros nulos.')
    else:
        print(f'A coluna "{col}" possui {null_rows - null_count} registros nulos.')
```

Como a coluna "NUMERO" é a única com dados nulos, ou seja, mesmo sem número ainda se tem todos os outros dados relativos aquela rota e aquele logradouro, por ora os valores nulos serão substituídos por 0. Afinal, essa ação é necessária também para a conversão de tipos mencionada anteriormente.

```
n [ ]: # Trocando valores nulos por 0
total_sample.fillna(0, inplace=True)

# Reavaliando a máscara para filtrar o dataframe por dados nulos
null_mask = total_sample.isnull().any(axis=1)

if int(total_sample[null_mask].count().max()) > 0:
    print("Mesmo após substituição, o dataset ainda contém dados nulos.")
else:
    print("Todos os valores nulos foram substituídos com sucesso.")
```

Conversão de tipos

```
n [ ]: # Checagem do tipo da coluna "NUMERO" da amostra, pois não aparecia ser float na visualização inicial ()
print(total_sample["NUMERO"].dtype)
```

Visto que o tipo da coluna "NUMERO" é object (string), provavelmente há registros fora do padrão que contêm letras. Esses registros são tratados abaixo.

Alguns números foram registrados com complementos, então será criada uma coluna de complemento para armazenar esses dados.

```
In [ ]: # Ambas as funções retornam strings pois os dados da coluna na qual serão aplicadas estão neste formato
def get_adress_number_and_complement(s : pd.Series) -> list[str, str]:
    """
    Recebe um registro do dataset das colunas "NUMERO" e "COMPLEMENTO" e retorna o número
    devidamente processado (ou seja, com letras e caracteres não-numéricos retirados) e o
    complemento do endereço caso esteja contido na coluna "NUMERO". Caso contrário, o
    complemento é retornado como "N" (de "Não").

    Parâmetros:
    - s: pd.Series / registro das colunas "NUMERO" e "COMPLEMENTO".

    Retorno:
    - [número sem caracteres não-numéricos, complemento].
    """
    try:
        return [str(int(s["NUMERO"])), "N"]
    except:
        return get_number_and_complement_within_str(s["NUMERO"].strip())
```

SPRINT 2

Parâmetros de Entrada

Quantidade de Pontos

Quantidade de Dias

Saída de Dados

Rotas

Tempo estimado da rota

Distancia a ser percorrida da rota

Clusterização

K Means

Relativamente simples de implementar.

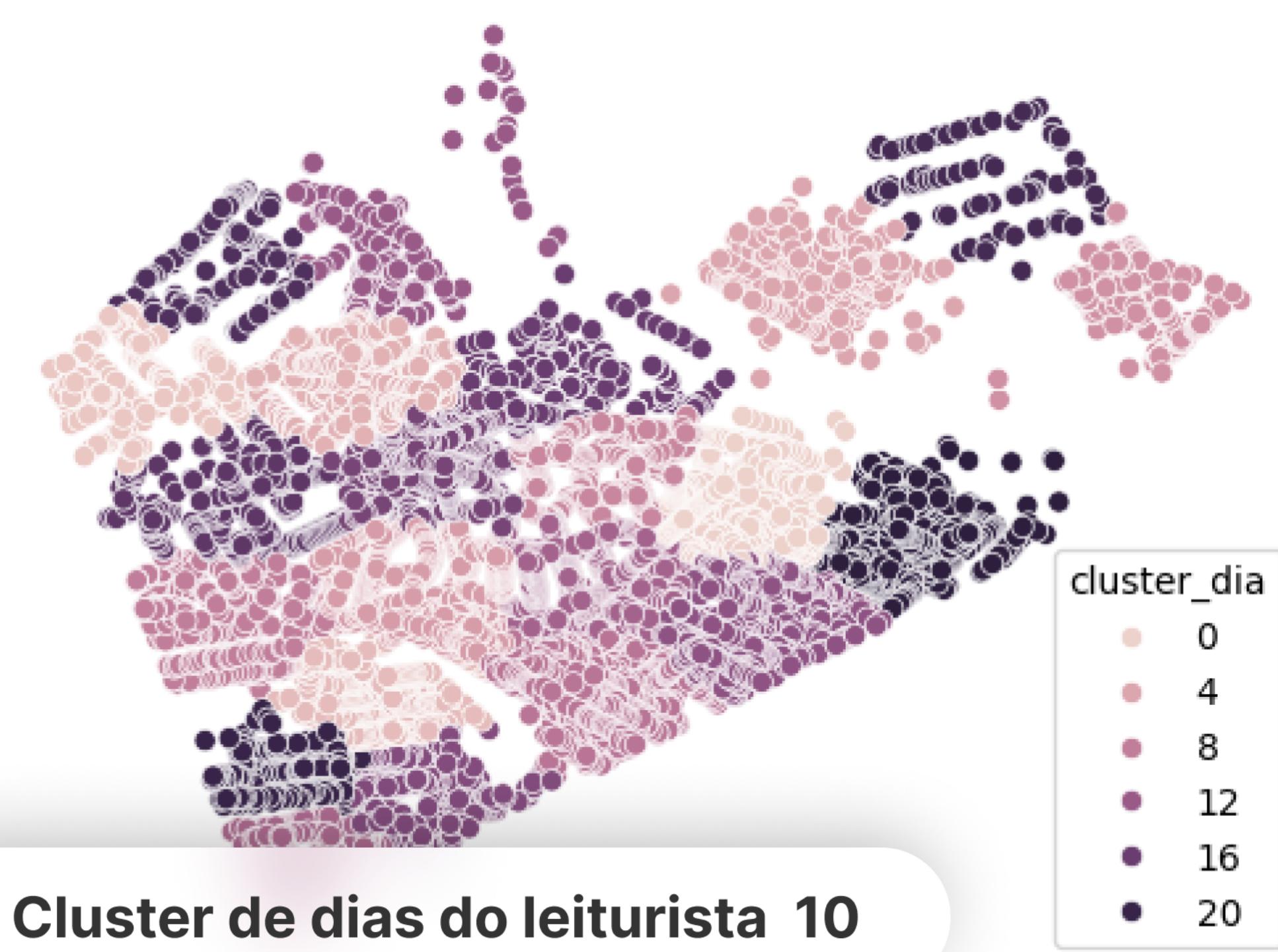
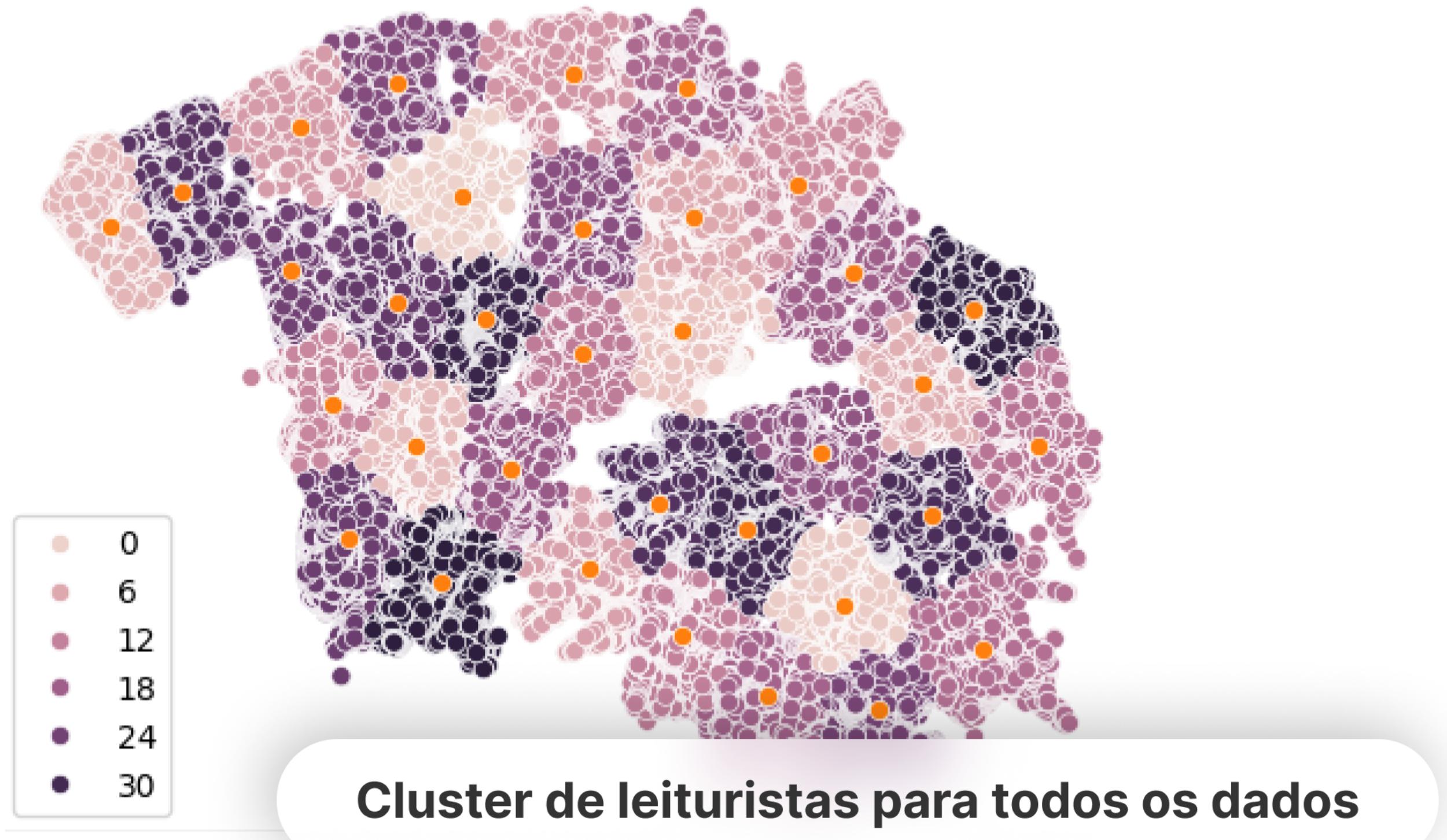
Escalona para grandes conjuntos de dados.

Garante a convergência.

Faça uma inicialização a quente das posições de centroides.

Adapta-se facilmente a novos exemplos.

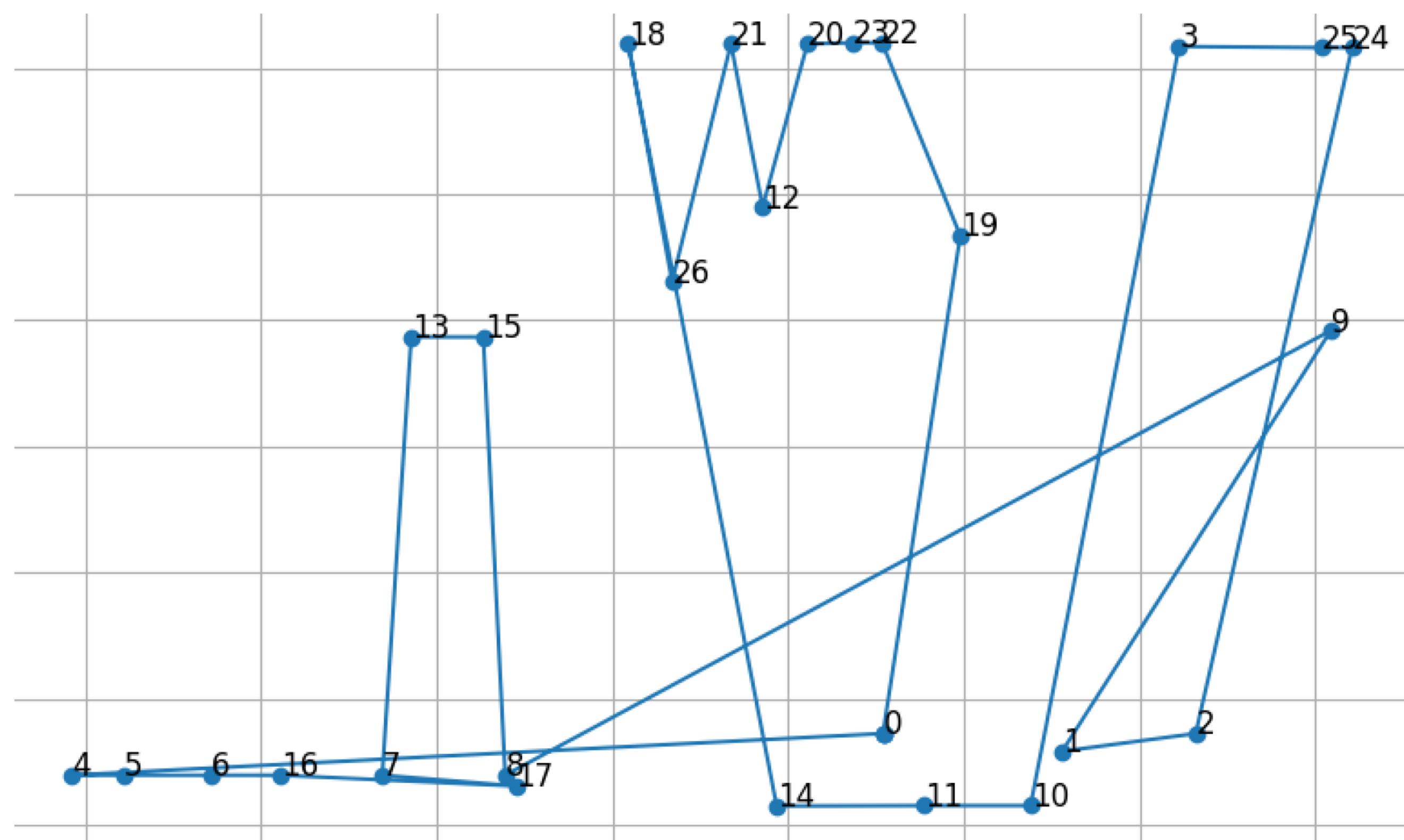
Generaliza a clusters de diferentes formas e tamanhos, como clusters elípticos.



Implementação Inicial de Algoritmo

Guloso

Seleciona o ponto mais próximo ou a melhor opção local a cada etapa, buscando uma solução global.



Wireframe

Wireframe

Wireframe

Wireframe

Wireframe

Wireframe

Wireframe

Inserir dados

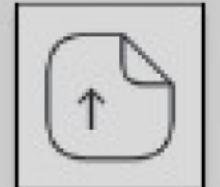
- 1 - Clique no ícone  para adicionar os dados.
- 2 - Adicione os dados de localização para que o algoritmo consiga rodar adequadamente.
- 3 - Após isso, atualize a página para que você consiga inserir as variáveis de entrada, utilizar o algoritmo de otimização e visualizar suas rotas.



Problemas com dados

- 1 - Clique no ícone  para adicionar os dados.
- 2 - Adicione os dados de localização para que o algoritmo consiga rodar adequadamente.
- 3 - Após isso, atualize a página para que você consiga inserir as variáveis de entrada, utilizar o algoritmo de otimização e visualizar suas rotas.

Infelizmente, tivemos um problema ao tentar fazer upload do arquivo. Tente novamente.



WIRE FRAME

Página Visão Única

Variáveis de entrada

Output

Trocar dados

Visão Única Visão Geral

Rotas

Leituristas: 20 Tempo: 20 Casas visitadas: 20

MAPA

Insira o total de dias de leitura

Insira as horas de trabalhos

Insira a quantidade de leituristas

Insira velocidade dos leituristas

Insira tempo de leitura

Otimizar

WIRE FRAME

Página Visão Única

Variáveis de entrada

Output

← Trocar dados

Visão Única Visão Geral

Leituristas: 20 Tempo: 20 Casas visitadas: 20

Rota 2	Rota 3	Rota 4
Leituristas: 20	Leituristas: 20	Leituristas: 20
Tempo: 20	Tempo: 20	Tempo: 20
Casas visitadas: 20	Casas visitadas: 20	Casas visitadas: 20
Ver mais >	Ver mais >	Ver mais >

Rota 5	Rota 6	Rota 8
Leituristas: 20	Leituristas: 20	Leituristas: 20
Tempo: 20	Tempo: 20	Tempo: 20

Insira o total de dias de leitura

Insira as horas de trabalhos

Insira a quantidade de leituristas

Insira velocidade dos leituristas

Insira tempo de leitura

Otimizar

Documentação e Artigo



Sprints

Planejador e back-end da aplicação

Front-end da aplicação

Artigo - 2a versão

Obrigado

Obrigado

Obrigado

Obrigado

Obrigado

Obrigado

Obrigado