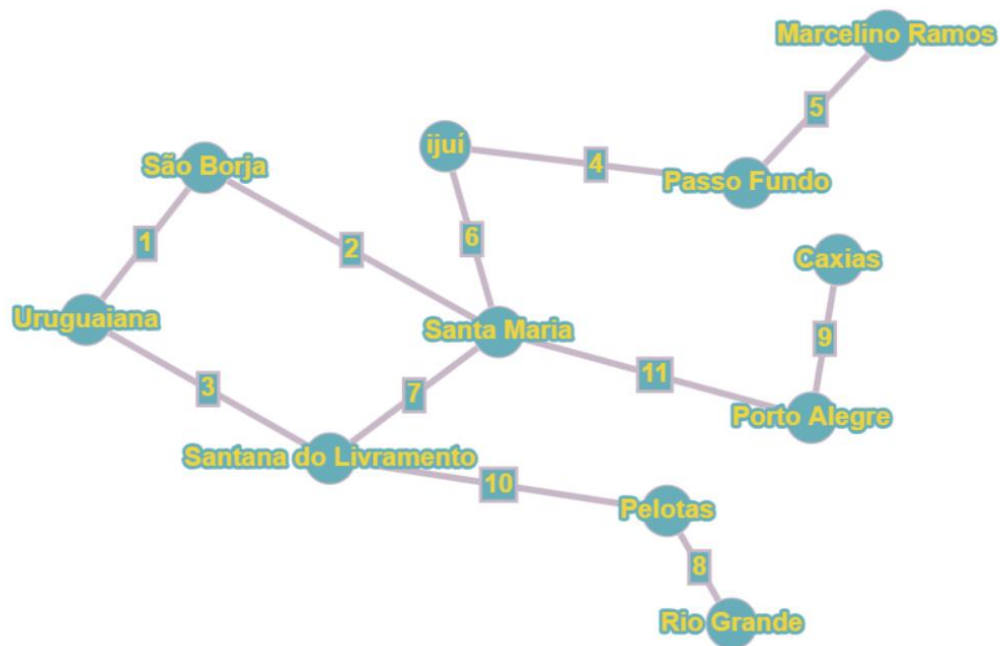


## Grafo de rota ferroviária do Rio Grande do Sul

Este relatório destaca a aplicação dos grafos na representação de rotas ferroviárias no Rio Grande do Sul. Em um estado com uma extensa malha ferroviária, foi utilizado um grafo representando a conexão ferroviária entre as principais cidades da Região, ao todo foram 11 cidades representadas.



A disposição dos vértices foi representada com base no mapa da malha ferroviária do estado do Rio Grande do Sul de 1925.



As cidades representadas no grafo foram:

- Marcelino Ramos
- Ijuí
- Passo Fundo
- São Borja
- Caxias
- Uruguaiana
- Santa Maria
- Santana do Livramento
- Pelotas
- Rio Grande
- Porto Alegre

Código em python

Definindo os nós

```
1  import networkx as nx
2  import matplotlib.pyplot as plt
3
4  F = nx.Graph()
5
6  #definindo as cidades (vértices)
7  F.add_node("Marcelino Ramos")
8  F.add_node("Passo Fundo")
9  F.add_node("Ijuí")
10 F.add_node("Santa Maria")
11 F.add_node("Porto Alegre")
12 F.add_node("Caxias")
13 F.add_node("São Borja")
14 F.add_node("Uruguaiana")
15 F.add_node("Santana do Livramento")
16 F.add_node("Pelotas")
17 F.add_node("Rio Grande")
18
```

Definindo as conexões

```

#definindo as rotas (arestas)
F.add_edge("Pelotas", "Rio Grande", weight=8)
F.add_edge("Rio Grande", "Pelotas", weight=8)
F.add_edge("Pelotas", "Santana do Livramento", weight=10)
F.add_edge("Santana do Livramento", "Santa Maria", weight=3)
F.add_edge("Uruguaiana", "São Borja", weight=1)
F.add_edge("São Borja", "Santa Maria", weight=2)
F.add_edge("Santana do Livramento", "Santa Maria", weight=7)
F.add_edge("Santa Maria", "Porto Alegre", weight=11)
F.add_edge("Porto Alegre", "Caxias", weight=9)
F.add_edge("Santa Maria", "Ijuí", weight=6)
F.add_edge("Ijuí", "Passo Fundo", weight=4)
F.add_edge("Passo Fundo", "Marcelino Ramos", weight=5)
F.add_edge("Marcelino Ramos", "Passo Fundo", weight=5)
F.add_edge("Passo Fundo", "Ijuí", weight=4)
F.add_edge("Ijuí", "Santa Maria", weight=6)
F.add_edge("Porto Alegre", "Santa Maria", weight=11)
F.add_edge("Santa Maria", "Santana do Livramento", weight=7)
F.add_edge("São Borja", "Uruguaiana", weight=1)
F.add_edge("Santa Maria", "São Borja", weight=2)
F.add_edge("Santana do Livramento", "Pelotas", weight=10)
F.add_edge("Caxias", "Porto Alegre", weight=9)
F.add_edge("Santana do Livramento", "Uruguaiana", weight=4)

```

Calculando e mostrando o grau de cada nó

```

#mostrar o grau de cada cidade
degree = F.degree("Pelotas")
print(f"O grau do nó {'Pelotas'} é {degree}.")
degree = F.degree("Marcelino Ramos")
print(f"O grau do nó {'Marcelino Ramos'} é {degree}.")
degree = F.degree("Passo Fundo")
print(f"O grau do nó {'Passo Fundo'} é {degree}.")
degree = F.degree("Ijuí")
print(f"O grau do nó {'Ijuí'} é {degree}.")
degree = F.degree("Santa Maria")
print(f"O grau do nó {'Santa Maria'} é {degree}.")
degree = F.degree("Porto Alegre")
print(f"O grau do nó {'Porto Alegre'} é {degree}.")
degree = F.degree("Caxias")
print(f"O grau do nó {'Caxias'} é {degree}.")
degree = F.degree("São Borja")
print(f"O grau do nó {'São Borja'} é {degree}.")
degree = F.degree("Uruguaiana")
print(f"O grau do nó {'Uruguaiana'} é {degree}.")
degree = F.degree("Santana do Livramento")
print(f"O grau do nó {'Santana do Livramento'} é {degree}.")
degree = F.degree("Rio Grande")
print(f"O grau do nó {'Rio Grande'} é {degree}.")

```

Escolhendo por onde começar a busca em largura

```

#Busca em largura
print("Busca Em Largura")
print("Começar a rota de por qual cidade? ")
print("1 - Marcelino Ramos")
print("2 - Santa Maria")
print("3 - Rio Grande")
print("4 - Porto Alegre")

while True:
    caminho = int(input("Digite a opção: "))
    match caminho:
        case 1:
            bfs_path = list(nx.bfs_tree(F, source='Marcelino Ramos'))
            comeca = 'Marcelino Ramos'
            break
        case 2:
            bfs_path = list(nx.bfs_tree(F, source='Santa Maria'))
            comeca = 'Santa Maria'
            break
        case 3:
            bfs_path = list(nx.bfs_tree(F, source='Rio Grande'))
            comeca = 'Rio Grande'
            break
        case 4:
            bfs_path = list(nx.bfs_tree(F, source='Porto Alegre'))
            comeca = 'Porto Alegre'
            break
        case _:
            print("Opção inválida, Digite Novamente.")
    print(f"Caminho da busca em largura começando por {comeca}:", bfs_path)

```

## Escolhendo por onde começar a busca em profundidade

```
#Busca em profundidade
print("Busca Em Profundidade")
print("Começar a rota de por qual cidade? ")
print("1 - Marcelino Ramos")
print("2 - Santa Maria")
print("3 - Rio Grande")
print("4 - Porto Alegre")

while True:
    caminho = int(input("Digite a opção: "))
    match caminho:
        case 1:
            dfs_path = list(nx.dfs_preorder_nodes(F, source="Marcelino Ramos"))
            começa = 'Marcelino Ramos'
            break
        case 2:
            bfs_path = list(nx.dfs_preorder_nodes(F, source='Santa Maria'))
            começa = 'Santa Maria'
            break
        case 3:
            bfs_path = list(nx.dfs_preorder_nodes(F, source='Rio Grande'))
            começa = 'Rio Grande'
            break
        case 4:
            bfs_path = list(nx.dfs_preorder_nodes(F, source='Porto Alegre'))
            começa = 'Porto Alegre'
            break
        case _:
            print("Opção inválida, Digite Novamente.")
    print(f"Caminho da busca em profundidade começando por {começa}:", bfs_path)
```

Utilizando a biblioteca `matplotlib` para mostrar uma representação gráfica na tela

```
plt.show()
```