



Relatório do Projeto

Parte 2

Nome do Integrante	RA
Carlos Eduardo Rosendo Basseto	10449941
João Pedro Gianfaldoni	10409524
Luiz Henrique Ribeiro Pulga	10409246
Matheus Santiago de Brito	10408953

Relatório

Recicla Fluxo

Definição do Problema –

Esse projeto tem como objetivo facilitar a identificação dos pontos de coleta de reciclagem próximos à sua casa e informar quais itens cada um aceita. A motivação é incentivar o descarte consciente e tornar a reciclagem algo mais simples de encaixar na rotina de todo mundo. A justificativa está em como essa solução ajuda a reduzir o lixo em aterros e promove hábitos sustentáveis. Para isso, vamos usar um grafo não direcionado, que conecta os pontos de coleta e a residência do usuário. Cada ponto de coleta, junto com a casa do usuário, funciona como um vértice (nó) nesse grafo. As arestas (arcos) são as ligações que exibem a distância em quilômetros entre esses lugares. Além disso, só aparecem as arestas para os pontos que aceitam os itens escolhidos e que estejam ao alcance do usuário. Dessa forma, o problema central é indicar, de forma rápida, onde descartar cada tipo de material sem andar muito. Isso deixa a reciclagem mais acessível e encoraja todo mundo a participar. No fim, a ideia é que essas informações estejam sempre à mão, adaptadas ao dia a dia e sem complicações.



OBJETIVO ODS:

Cuidar do meio ambiente por meio da reciclagem e pontos de coleta



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



Execuções –

```
Menu de Opções:
a) Ler dados do arquivo grafo.txt
b) Gravar dados no arquivo grafo.txt
c) Inserir vértice
d) Inserir aresta
e) Remover vértice
f) Remover aresta
g) Mostrar conteúdo do arquivo
h) Mostrar grafo
i) Apresentar conexidade do grafo e o reduzido
j) Encerrar a aplicação
Escolha uma opção: a
Grafo carregado do arquivo com sucesso.
```

```
Menu de Opções:
a) Ler dados do arquivo grafo.txt
b) Gravar dados no arquivo grafo.txt
c) Inserir vértice
d) Inserir aresta
e) Remover vértice
f) Remover aresta
g) Mostrar conteúdo do arquivo
h) Mostrar grafo
i) Apresentar conexidade do grafo e o reduzido
j) Encerrar a aplicação
Escolha uma opção: c
Informe o rótulo do vértice: sapo
Informe o peso do vértice (ou deixe vazio):
Vértice inserido: id=61, label='sapo', peso=''
Arquivo atualizado com sucesso!
```

```
Menu de Opções:
a) Ler dados do arquivo grafo.txt
b) Gravar dados no arquivo grafo.txt
c) Inserir vértice
d) Inserir aresta
e) Remover vértice
f) Remover aresta
g) Mostrar conteúdo do arquivo
h) Mostrar grafo
i) Apresentar conexidade do grafo e o reduzido
j) Encerrar a aplicação
Escolha uma opção: c
Informe o rótulo do vértice: rato
Informe o peso do vértice (ou deixe vazio): 10
Vértice inserido: id=62, label='rato', peso='10'
Arquivo atualizado com sucesso!
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
50: label='50', peso=''
51: label='51', peso=''
52: label='52', peso=''
53: label='53', peso=''
54: label='54', peso=''
55: label='55', peso=''
56: label='56', peso=''
57: label='57', peso=''
58: label='58', peso=''
59: label='59', peso=''
60: label='60', peso=''
61: label='sapo', peso=''
62: label='rato', peso='10'
```

Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: d

Informe o vértice de origem: 61

Informe o vértice de destino: 62

Informe o peso da aresta: 777

Aresta inserida/atualizada entre 61 e 62 com peso 777.

Arquivo atualizado com sucesso!

```
54 <--> 50 com peso 2000
55 <--> 58 com peso 1000
56 <--> 58 com peso 1500
57 <--> 58 com peso 650
58 <--> 59 com peso 1800
58 <--> 60 com peso 2300
61 <--> 62 com peso 777
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: b

Arquivo atualizado com sucesso!

Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: f

Informe o vértice de origem da aresta a remover: 61

Informe o vértice de destino da aresta a remover: 62

Aresta entre 61 e 62 removida.

Arquivo atualizado com sucesso!

```
50 <--> 58 com peso 1500
57 <--> 58 com peso 650
58 <--> 59 com peso 1800
58 <--> 60 com peso 2300
-----
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: e

Informe o id do vértice a remover: 61

Vértice 61 e suas 0 arestas associadas foram removidos.

Arquivo atualizado com sucesso!

```
59: label='59', peso=''
60: label='60', peso=''
62: label='rato', peso='10'
```




UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: g

--- Conteúdo do Arquivo ---

2

61

1 "1" ""

2 "2" ""

3 "3" ""

4 "4" ""

5 "5" ""

6 "6" ""

7 "7" ""

8 "8" ""

9 "9" ""

10 "10" ""

11 "11" ""

12 "12" ""

13 "13" ""

14 "14" ""

15 "15" ""

16 "16" ""

17 "17" ""

18 "18" ""

19 "19" ""

20 "20" ""

21 "21" ""

22 "22" ""

23 "23" ""

24 "24" ""

25 "25" ""

26 "26" ""

27 "27" ""

28 "28" ""

29 "29" ""

30 "30" ""

31 "31" ""

32 "32" ""

33 "33" ""

34 "34" ""

35 "35" ""

36 "36" ""

37 "37" ""

38 "38" ""



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: i

O grafo não é conexo.

Componentes conexas encontradas:

Componente 1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]

Componente 2: [62]

Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: e

Informe o id do vértice a remover: 62

Vértice 62 e suas 0 arestas associadas foram removidos.

Arquivo atualizado com sucesso!

Menu de Opções:

- a) Ler dados do arquivo grafo.txt
- b) Gravar dados no arquivo grafo.txt
- c) Inserir vértice
- d) Inserir aresta
- e) Remover vértice
- f) Remover aresta
- g) Mostrar conteúdo do arquivo
- h) Mostrar grafo
- i) Apresentar conexidade do grafo e o reduzido
- j) Encerrar a aplicação

Escolha uma opção: i

O grafo é conexo.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Menu de Opções:

- a) Ler dados do arquivo grafo.txt
 - b) Gravar dados no arquivo grafo.txt
 - c) Inserir vértice
 - d) Inserir aresta
 - e) Remover vértice
 - f) Remover aresta
 - g) Mostrar conteúdo do arquivo
 - h) Mostrar grafo
 - i) Apresentar conexidade do grafo e o reduzido
 - j) Encerrar a aplicação
- Escolha uma opção: j
Encerrando a aplicação...
PS C:\Users\win\Desktop\ProjetoGrafos> █

Menu:

- a) Ler grafo
 - b) Gravar grafo
 - c) Inserir vértice
 - d) Inserir aresta
 - e) Remover vértice
 - f) Remover aresta
 - g) Mostrar arquivo
 - h) Mostrar grafo
 - i) Verificar conexidade
 - j) Sair
 - k) Caminho mínimo (Dijkstra)
 - l) Árvore Geradora Mínima (Prim)
 - m) Coloração de Vértices
- Opção: k
Informe o ID do vértice onde você está: 13
1 - 1 a 1400.0 de distância
2 - 2 a 1800.0 de distância
3 - 3 a 2100.0 de distância
4 - 4 a 2200.0 de distância
5 - 5 a 2300.0 de distância
6 - 6 a 2900.0 de distância
7 - 7 a 2500.0 de distância
8 - 8 a 3200.0 de distância
9 - 9 a 3500.0 de distância
10 - 10 a 4100.0 de distância
11 - 11 a 4000.0 de distância
12 - 12 a 3900.0 de distância
14 - 14 a 3800.0 de distância
15 - 15 a 3200.0 de distância
16 - 16 a 3600.0 de distância
17 - 17 a 4000.0 de distância
18 - 18 a 3600.0 de distância
19 - 19 a 4200.0 de distância
20 - 20 a 4600.0 de distância
21 - 21 a 5500.0 de distância
22 - 22 a 5500.0 de distância
23 - 23 a 5700.0 de distância
24 - 24 a 4300.0 de distância
25 - 25 a 5100.0 de distância
26 - 26 a 5150.0 de distância
27 - 27 a 5100.0 de distância
28 - 28 a 5000.0 de distância
29 - 29 a 4800.0 de distância
30 - 30 a 5000.0 de distância
31 - 31 a 4400.0 de distância
32 - 32 a 3700.0 de distância
33 - 33 a 4200.0 de distância
34 - 34 a 3200.0 de distância
35 - 35 a 6200.0 de distância
36 - 36 a 6200.0 de distância
37 - 37 a 5000.0 de distância
38 - 38 a 4800.0 de distância
39 - 39 a 5300.0 de distância
40 - 40 a 4400.0 de distância

Menu:

- a) Ler grafo
 - b) Gravar grafo
 - c) Inserir vértice
 - d) Inserir aresta
 - e) Remover vértice
 - f) Remover aresta
 - g) Mostrar arquivo
 - h) Mostrar grafo
 - i) Verificar conexidade
 - j) Sair
 - k) Caminho mínimo (Dijkstra)
 - l) Árvore Geradora Mínima (Prim)
 - m) Coloração de Vértices
- Opção: l
Árvore Geradora Mínima:
1 <--> 2 com custo 400.0
1 <--> 3 com custo 700.0
1 <--> 4 com custo 800.0
1 <--> 5 com custo 900.0
1 <--> 7 com custo 1100.0
1 <--> 13 com custo 1400.0
1 <--> 6 com custo 1500.0
1 <--> 8 com custo 1800.0
1 <--> 15 com custo 1800.0
1 <--> 34 com custo 1800.0
1 <--> 47 com custo 1900.0
47 <--> 24 com custo 1900.0
24 <--> 25 com custo 800.0
24 <--> 26 com custo 850.0
24 <--> 48 com custo 1100.0
24 <--> 9 com custo 1700.0
1 <--> 16 com custo 2200.0
1 <--> 18 com custo 2200.0
1 <--> 32 com custo 2300.0
32 <--> 58 com custo 2100.0
58 <--> 57 com custo 650.0
58 <--> 55 com custo 1000.0
58 <--> 29 com custo 1200.0
58 <--> 31 com custo 1400.0
58 <--> 30 com custo 1500.0
58 <--> 56 com custo 1500.0
58 <--> 59 com custo 1800.0
58 <--> 33 com custo 1900.0
58 <--> 36 com custo 2100.0
58 <--> 28 com custo 2200.0
58 <--> 60 com custo 2300.0
1 <--> 14 com custo 2400.0
1 <--> 12 com custo 2500.0
1 <--> 11 com custo 2600.0
1 <--> 17 com custo 2600.0
24 <--> 10 com custo 2600.0
1 <--> 19 com custo 2800.0
58 <--> 35 com custo 2800.0
58 <--> 53 com custo 2800.0



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
58 <--> 57 com custo 650.0
58 <--> 55 com custo 1000.0
58 <--> 29 com custo 1200.0
58 <--> 31 com custo 1400.0
58 <--> 30 com custo 1500.0
58 <--> 56 com custo 1500.0
58 <--> 59 com custo 1800.0
58 <--> 33 com custo 1900.0
58 <--> 36 com custo 2100.0
58 <--> 28 com custo 2200.0
58 <--> 60 com custo 2300.0
1 <--> 14 com custo 2400.0
1 <--> 12 com custo 2500.0
1 <--> 11 com custo 2600.0
1 <--> 17 com custo 2600.0
24 <--> 10 com custo 2600.0
1 <--> 19 com custo 2800.0
58 <--> 35 com custo 2800.0
58 <--> 53 com custo 2800.0
58 <--> 54 com custo 2800.0
58 <--> 27 com custo 2900.0
1 <--> 40 com custo 3000.0
1 <--> 41 com custo 3000.0
24 <--> 50 com custo 3000.0
1 <--> 20 com custo 3200.0
1 <--> 38 com custo 3400.0
1 <--> 42 com custo 3400.0
58 <--> 52 com custo 3500.0
1 <--> 37 com custo 3600.0
1 <--> 44 com custo 3600.0
24 <--> 51 com custo 3700.0
1 <--> 39 com custo 3900.0
1 <--> 46 com custo 4000.0
1 <--> 21 com custo 4100.0
1 <--> 22 com custo 4100.0
1 <--> 23 com custo 4300.0
1 <--> 45 com custo 4700.0
58 <--> 49 com custo 4700.0
1 <--> 43 com custo 4800.0
Custo total: 141600.0
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Menu:

- a) Ler grafo
 - b) Gravar grafo
 - c) Inserir vértice
 - d) Inserir aresta
 - e) Remover vértice
 - f) Remover aresta
 - g) Mostrar arquivo
 - h) Mostrar grafo
 - i) Verificar conexidade
 - j) Sair
 - k) Caminho mínimo (Dijkstra)
 - l) Árvore Geradora Mínima (Prim)
 - m) Coloração de Vértices
- Opção: m

Coloração de Vértices (número mínimo de cores):

Vértice 1 (1) -> Cor 1
Vértice 2 (2) -> Cor 2
Vértice 3 (3) -> Cor 2
Vértice 4 (4) -> Cor 2
Vértice 5 (5) -> Cor 2
Vértice 6 (6) -> Cor 2
Vértice 7 (7) -> Cor 2
Vértice 8 (8) -> Cor 2
Vértice 9 (9) -> Cor 2
Vértice 10 (10) -> Cor 2
Vértice 11 (11) -> Cor 2
Vértice 12 (12) -> Cor 2
Vértice 13 (13) -> Cor 2
Vértice 14 (14) -> Cor 2
Vértice 15 (15) -> Cor 2
Vértice 16 (16) -> Cor 2
Vértice 17 (17) -> Cor 2
Vértice 18 (18) -> Cor 2
Vértice 19 (19) -> Cor 2
Vértice 20 (20) -> Cor 2
Vértice 21 (21) -> Cor 2
Vértice 22 (22) -> Cor 2
Vértice 23 (23) -> Cor 2
Vértice 24 (24) -> Cor 3

Vértice 43 (43) -> Cor 2
Vértice 44 (44) -> Cor 2
Vértice 45 (45) -> Cor 2
Vértice 46 (46) -> Cor 2
Vértice 47 (47) -> Cor 2
Vértice 48 (48) -> Cor 2
Vértice 49 (49) -> Cor 2
Vértice 50 (50) -> Cor 2
Vértice 51 (51) -> Cor 2
Vértice 52 (52) -> Cor 2
Vértice 53 (53) -> Cor 2
Vértice 54 (54) -> Cor 2
Vértice 55 (55) -> Cor 2
Vértice 56 (56) -> Cor 2
Vértice 57 (57) -> Cor 2
Vértice 58 (58) -> Cor 4
Vértice 59 (59) -> Cor 2
Vértice 60 (60) -> Cor 2
Total de cores usadas: 4

Link do GitHub: <https://github.com/KaduRosendo/Grafos.git>

Link vídeo Youtube: https://youtu.be/OTieR-5FnPw?si=qSu4o94ulp3d_Cr1

Apresentação:



Recicla Fluxo: Otimizando a Reciclagem com Teoria dos Grafos

Bem-vindos à apresentação do nosso projeto **Recicla Fluxo**, desenvolvido pelos estudantes João Pedro Gianfaldoni, Matheus Santiago de Brito, Carlos Eduardo Rosendo Basseto e Luiz Henrique Ribeiro Pulga. Este projeto foi concebido como parte da nossa disciplina de Estruturas de Dados e Algoritmos, no curso de Ciência da Computação.

O Recicla Fluxo busca solucionar um problema prático enfrentado por muitos brasileiros: a dificuldade em localizar pontos adequados para descarte de materiais recicláveis. Utilizando conceitos avançados da teoria dos grafos, nosso aplicativo conecta pessoas a pontos de coleta próximos, tornando a reciclagem mais acessível e eficiente.

Nas próximas telas, exploraremos a motivação por trás do projeto e como implementamos algoritmos de grafos para criar soluções inovadoras de logística sustentável.

Problema e Solução Proposta

A reciclagem no Brasil enfrenta um obstáculo significativo: muitos cidadãos desconhecem onde descartar adequadamente seus materiais recicláveis. Esta lacuna de informação resulta em materiais potencialmente recicláveis sendo destinados a aterros sanitários, agravando problemas ambientais existentes.

Nossa solução aborda diretamente este desafio através de uma plataforma geolocalizada que, ao receber a posição do usuário e o tipo de material a ser reciclado, indica os pontos de coleta mais próximos e adequados para aquele material específico.



Com esta abordagem intuitiva, eliminamos a barreira informacional que impede muitos brasileiros de participarem ativamente em programas de reciclagem, contribuindo para os Objetivos de Desenvolvimento Sustentável da ONU.

O problema que buscamos resolver é algo cotidiano, mas que tem grandes impactos ambientais.

No Brasil, muitas pessoas ainda não sabem onde descartar corretamente seus resíduos recicláveis. Isso faz com que materiais que poderiam ser reutilizados acabem sendo jogados em aterros sanitários, contribuindo para a poluição e o desperdício de recursos.

A nossa proposta, representada nesse slide, é criar uma plataforma simples, intuitiva e eficiente que conecta o cidadão aos pontos de coleta mais próximos.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



O funcionamento é dividido em quatro etapas:

- **Primeiro**, o usuário compartilha sua **localização** atual com o aplicativo.
- **Depois**, ele informa o tipo de **material reciclável** que deseja descartar — por exemplo, papel, plástico, eletrônico etc.
- Em seguida, o aplicativo realiza uma **busca** inteligente usando algoritmos de grafos para processar esses dados.
- E, por fim, o sistema exibe o **resultado**: mostra os pontos de coleta mais próximos e traça a **rota ideal** para facilitar o deslocamento.

Essa abordagem é simples para o usuário, mas por trás dela existe uma lógica bem estruturada que nos permite transformar dados em ações práticas. Com isso, eliminamos a barreira da desinformação, e ajudamos o Brasil a se aproximar dos Objetivos de Desenvolvimento Sustentável da ONU.

Implementação com Teoria dos Grafos

A base técnica do Recicla Fluxo é fundamentada na Teoria dos Grafos. Modelamos o sistema representando pontos de coleta e a localização do usuário como vértices, enquanto as ruas e conexões entre estes pontos são representadas como arestas com pesos que indicam distância ou custo de deslocamento.

Nossa implementação utiliza algoritmos clássicos de grafos, adaptados para o contexto da reciclagem urbana, promovendo eficiência e sustentabilidade alinhadas aos ODS 3 (Saúde e Bem-Estar) e ODS 13 (Ação Contra a Mudança Global do Clima).



Algoritmo de Dijkstra

Utilizado para determinar o ponto de coleta mais próximo com base na localização do usuário, minimizando o tempo de deslocamento



Algoritmo de Prim

Gera árvores geradoras de custo mínimo para otimizar rotas de coleta para empresas e cooperativas de reciclagem



Coloração de Vértices

Organiza visualmente rotas e áreas de atendimento, minimizando conflitos em regiões próximas e facilitando a gestão logística

Esta abordagem computacional não apenas facilita o acesso dos cidadãos aos pontos de reciclagem, mas também oferece uma infraestrutura de dados que pode ser utilizada por gestores públicos para otimizar a distribuição de pontos de coleta nas cidades brasileiras.

Aqui nós explicamos como essa solução foi desenvolvida tecnicamente.

Toda a base do Recicla Fluxo foi construída usando a **Teoria dos Grafos**, onde cada ponto de coleta e cada usuário é representado como um **vértice**, e as conexões entre eles — ou seja, as ruas — são as **arestas** do grafo. Essas arestas possuem pesos que representam o custo de deslocamento, como a distância ou o tempo gasto.

Com essa estrutura, conseguimos aplicar **algoritmos clássicos** da computação de forma personalizada para a realidade urbana e ambiental.

O primeiro algoritmo que usamos é o **Dijkstra**, que calcula o caminho mais curto entre o usuário e o ponto de coleta, garantindo o deslocamento mais rápido e eficiente.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Já o **Algoritmo de Prim** foi aplicado para criar **árvores geradoras de custo mínimo** — isso é muito útil para as cooperativas de reciclagem e prefeituras que precisam planejar rotas de coleta em várias regiões da cidade.

E, por fim, utilizamos a **Coloração de Vértices**, que serve para dividir as zonas de coleta de forma organizada, evitando sobreposição entre rotas e melhorando a logística urbana.

Com isso, o aplicativo não só ajuda o cidadão comum, como também se transforma em uma ferramenta poderosa para **gestores públicos**, otimizando a distribuição dos pontos de coleta nas cidades brasileiras.

```
File Edit Selection View Go Run Terminal Help
PROJETO_1_GRAFOS
  grafo.txt
  GrafosReciclaFluxo.py
  ReciclaFluxoGR.graphml
  templateRelatorioProjetoTG_parte1.pdf
PS C:\Users\win\Desktop\Projeto_1_Grafos> python .\GrafosReciclaFluxo.py
=== Gerenciamento de Grafos ===

Menu:
a) Ler grafo
b) Gravar grafo
c) Inserir vértice
d) Inserir aresta
e) Remover vértice
f) Remover aresta
g) Mostrar arquivo
h) Mostrar grafo
i) Verificar conexidade
j) Sair
k) Caminho mínimo (Dijkstra)
l) Árvore Geradora Mínima (Prim)
m) Coloração de Vértices
Opção: a

Menu:
a) Ler grafo
b) Gravar grafo
c) Inserir vértice
d) Inserir aresta
e) Remover vértice
f) Remover aresta
g) Mostrar arquivo
h) Mostrar grafo
i) Verificar conexidade
j) Sair
k) Caminho mínimo (Dijkstra)
l) Árvore Geradora Mínima (Prim)
m) Coloração de Vértices
Opção: 
```

Leitura do grafo (opção 'a')

Inicialmente, carregamos os dados do grafo a partir de um arquivo `.txt`. Esse arquivo contém os vértices (com nome e tipo de material aceito) e as arestas com os pesos.

Visualização da estrutura (opção 'h')

Em seguida, mostramos os vértices e arestas. Isso confirma a integridade da nossa representação do grafo.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
File Edit Selection View Go Run Terminal Help
EXPLORER
PROJETO_1_GRAFOS
  grafo.txt
  GrafosRecidaFluxo.py
  RecidaFluxoGR.graphml
  templateRelatórioProjetoTG_parte1.pdf
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
53 <--> 58 com peso 2800
54 <--> 58 com peso 2800
55 <--> 58 com peso 1000
56 <--> 58 com peso 1500
57 <--> 58 com peso 650
58 <--> 59 com peso 1800
58 <--> 60 com peso 2300

Menu:
a) Ler grafo
b) Gravar grafo
c) Inserir vértice
d) Inserir aresta
e) Remover vértice
f) Remover aresta
g) Mostrar arquivo
h) Mostrar grafo
i) Verificar conexidade
j) Sair
k) Caminho mínimo (Dijkstra)
l) Árvore Geradora Mínima (Prim)
m) Coloração de Vértices
Opção: k
Informe o ID do vértice onde você está: 1
2 - 2 a 400.0 de distância
3 - 3 a 700.0 de distância
4 - 4 a 800.0 de distância
5 - 5 a 900.0 de distância
6 - 6 a 1500.0 de distância
7 - 7 a 1100.0 de distância
8 - 8 a 1800.0 de distância
9 - 9 a 2100.0 de distância
10 - 10 a 2700.0 de distância
11 - 11 a 2600.0 de distância
12 - 12 a 2500.0 de distância
13 - 13 a 1400.0 de distância
14 - 14 a 2400.0 de distância
15 - 15 a 1800.0 de distância
16 - 16 a 2200.0 de distância
17 - 17 a 2600.0 de distância
18 - 18 a 2200.0 de distância
19 - 19 a 2800.0 de distância
20 - 20 a 3200.0 de distância
21 - 21 a 4100.0 de distância
22 - 22 a 4100.0 de distância
23 - 23 a 4300.0 de distância
24 - 24 a 3100.0 de distância
25 - 25 a 3900.0 de distância
26 - 26 a 3950.0 de distância
27 - 27 a 3700.0 de distância
28 - 28 a 3600.0 de distância
29 - 29 a 3400.0 de distância
30 - 30 a 3600.0 de distância
31 - 31 a 3000.0 de distância
```

Cálculo do ponto de coleta mais próximo (opção 'k')

Aqui pedimos o ID do vértice correspondente à casa do usuário. O algoritmo de Dijkstra retorna todos os pontos alcançáveis ordenados pela distância. O mais próximo é exibido em primeiro lugar.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



```
File Edit Selection View Go Run Terminal Help
PROJETO_1_GRAFOS
  grafo.txt
  GrafosRecidaFluxo.py
  RecidaFluxoGR.graphml
  templateRelatorioProjetoTG_parte1.pdf

b) Gravar grafo
c) Inserir vértice
d) Inserir aresta
e) Remover vértice
f) Remover aresta
g) Mostrar arquivo
h) Mostrar grafo
i) Verificar conexidade
j) Sair
k) Caminho mínimo (Dijkstra)
l) Árvore Geradora Mínima (Prim)
m) Coloração de Vértices
Opção: l
Árvore Geradora Mínima:
1 <--> 2 com custo 400.0
1 <--> 3 com custo 700.0
1 <--> 4 com custo 800.0
1 <--> 5 com custo 900.0
1 <--> 7 com custo 1100.0
1 <--> 13 com custo 1400.0
1 <--> 6 com custo 1500.0
1 <--> 8 com custo 1800.0
1 <--> 15 com custo 1800.0
1 <--> 34 com custo 1800.0
1 <--> 47 com custo 1900.0
47 <--> 24 com custo 1900.0
24 <--> 25 com custo 800.0
24 <--> 26 com custo 850.0
24 <--> 48 com custo 1100.0
24 <--> 9 com custo 1700.0
1 <--> 16 com custo 2200.0
1 <--> 18 com custo 2200.0
1 <--> 32 com custo 2300.0
32 <--> 58 com custo 2100.0
58 <--> 57 com custo 650.0
58 <--> 55 com custo 1000.0
58 <--> 29 com custo 1200.0
58 <--> 31 com custo 1400.0
58 <--> 30 com custo 1500.0
58 <--> 56 com custo 1500.0
58 <--> 59 com custo 1800.0
58 <--> 33 com custo 1900.0
58 <--> 36 com custo 2100.0
58 <--> 28 com custo 2200.0
58 <--> 60 com custo 2300.0
1 <--> 14 com custo 2400.0
1 <--> 12 com custo 2500.0
1 <--> 11 com custo 2600.0
1 <--> 17 com custo 2600.0
24 <--> 10 com custo 2600.0
1 <--> 19 com custo 2800.0
58 <--> 35 com custo 2800.0
58 <--> 53 com custo 2800.0
58 <--> 54 com custo 2800.0
```

Árvore Geradora Mínima (opção 'l')

Executamos o algoritmo de Prim para gerar uma árvore mínima que conecta todos os pontos. Essa informação é essencial para planejamento logístico, pois garante menor custo para passar por todos os locais.”



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  PROJETO_1_GRAFOS
    grafo.txt
    GrafosReciclaFluxo.py
    ReciclaFluxoGR.graphml
    templateRelatorioProjetoTG_parte1.pdf
PROBLEMAS OUTPUT DEBUG CONSOLE TERMINAL
j) Sair
k) Caminho mínimo (Dijkstra)
l) Árvore Geradora Mínima (Prim)
m) Coloração de Vértices
Opção: m

Coloração de Vértices (número mínimo de cores):
Vértice 1 (1) -> Cor 1
Vértice 2 (2) -> Cor 2
Vértice 3 (3) -> Cor 2
Vértice 4 (4) -> Cor 2
Vértice 5 (5) -> Cor 2
Vértice 6 (6) -> Cor 2
Vértice 7 (7) -> Cor 2
Vértice 8 (8) -> Cor 2
Vértice 9 (9) -> Cor 2
Vértice 10 (10) -> Cor 2
Vértice 11 (11) -> Cor 2
Vértice 12 (12) -> Cor 2
Vértice 13 (13) -> Cor 2
Vértice 14 (14) -> Cor 2
Vértice 15 (15) -> Cor 2
Vértice 16 (16) -> Cor 2
Vértice 17 (17) -> Cor 2
Vértice 18 (18) -> Cor 2
Vértice 19 (19) -> Cor 2
Vértice 20 (20) -> Cor 2
Vértice 21 (21) -> Cor 2
Vértice 22 (22) -> Cor 2
Vértice 23 (23) -> Cor 2
Vértice 24 (24) -> Cor 3
Vértice 25 (25) -> Cor 2
Vértice 26 (26) -> Cor 2
Vértice 27 (27) -> Cor 2
Vértice 28 (28) -> Cor 2
Vértice 29 (29) -> Cor 2
Vértice 30 (30) -> Cor 2
Vértice 31 (31) -> Cor 2
Vértice 32 (32) -> Cor 2
Vértice 33 (33) -> Cor 2
Vértice 34 (34) -> Cor 2
Vértice 35 (35) -> Cor 2
Vértice 36 (36) -> Cor 2
Vértice 37 (37) -> Cor 2
Vértice 38 (38) -> Cor 2
Vértice 39 (39) -> Cor 2
Vértice 40 (40) -> Cor 2
Vértice 41 (41) -> Cor 2
Vértice 42 (42) -> Cor 2
Vértice 43 (43) -> Cor 2
Vértice 44 (44) -> Cor 2
Vértice 45 (45) -> Cor 2
Vértice 46 (46) -> Cor 2
```

Coloração de vértices (opção 'm')

Por fim, aplicamos a coloração. Essa etapa é fundamental para dividir áreas e evitar que a mesma rota seja usada por regiões vizinhas. Isso simula, por exemplo, a divisão entre zonas atendidas por diferentes caminhões.

Nosso repositório está disponível publicamente no GitHub. Ele contém:

- main.py: código principal com menu de execução e chamadas dos algoritmos;
- grafo.txt: grafo base, onde estão definidos os vértices (pontos e casas) e arestas (ruas com pesos);
- README.md: explicação do objetivo do projeto, instruções de uso e descrição das funcionalidades;
- Print screens da execução e testes.



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática
Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



KaduRosendo / Grafos

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file

Projeto_1

Projeto_2

Grafos2.md

GrafosReciclaFluxo.py

RelatorioProjetoTG_parte 2.pdf

grafo.txt

README.md

Grafos / Projeto_2 / Grafos2.md

matheus-sdb Create Grafos2.md

Preview Code Blame 22 lines (12 loc) · 528 bytes Code 55% faster with GitHub Copilot

Projeto Recicla Fluxo da Matéria Teoria dos Grafos

Integrantes -

João Pedro Gianfaldoni - 10409524

Matheus Santiago de Brito - 10408953

Carlos Eduardo Rosendo Basseto - 10409941

Luiz Henrique Ribeiro Pulga - 10409246

Conteúdo dos arquivos:

- Relatório atualizado do projeto 2
- Arquivo .py relacionado ao projeto Recicla Fluxo

Alterações:

- Última alteração feita pelo grupo:

22/05/2025 - João, Matheus, Carlos, Luiz - finalização da edição do vídeo e adição do link no relatório.