

# Tarefa 1-B

## Teoria dos Grafos 2025/2

André Vinícius Trentini, Nicole Martins da Costa

<sup>1</sup>Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC) – Joinville – SC – Brasil

### 1. Código

Esta tarefa foi implementada completamente na linguagem C, com o uso das bibliotecas *stdio.h*, *stdlib.h*, *string.h* e *math.h* no *Visual Studio Code*. O código-fonte está disponível em Bases Rotuladas.

A estrutura é composta de duas bibliotecas de funções e uma *main* que as executa. Os arquivos são:

1. **dataset.h**: contém todas as funções para a leitura do CSV, aplicação do limiar, contagem de componentes conexos e seus respectivos tamanhos e análise da separação por Tipos e todos os necessários para o cálculo da acurácia.
2. **dist\_euclidiana.h**: contém as funções para o cálculo inicial das distâncias normalizadas e criação do arquivo com as tais. Foi criado por motivo de organização do código.
3. **main.c**: possui a função *execute()*, que realiza a chamada de todas as funções necessárias para a análise das duas bases antes da chamada da função para a análise da tipagem do *cluster*.

### 2. Lógica

Após o cálculo das distâncias, o código aplica o limiar e calcula quantos componentes conexos se formam ao percorrer com DFS (*Depth First Search*), verificando durante cada iteração quais os tamanhos de cada componente para encontrar se existem pelo menos três com tamanho o suficiente. A decisão de se o tamanho é suficiente será explicada na Seção 2.1.

Para o caso de a separação por limiar não ser suficiente para a separação completa em exatos três *clusters*, os menores componentes e singulares são associados aos maiores. A lógica para tal será explicada na Seção 2.2.

Então, é dito por critério de maioria qual Tipo cada componente representa e, por fim, feita a matriz de confusão para o cálculo da acurácia desta afirmação.

#### 2.1. Componentes Encontrados

É considerado que a separação em três teve sucesso se forem encontrados pelo menos três componentes com diferença máxima igual 15 entre o menor e o maior, ou seja, a maioria dos vértices já foi agrupada entre eles.

#### 2.2. Separação em Exatos Três

No caso da separação pelo limiar não ser suficiente, é calculado o centróide de cada um dos três maiores componentes e os vértices ainda não atribuídos a um dos três é redistribuído com base na sua distância ser mais próxima de qual valor de centro.

### 3. Instruções

O código foi feito para ser executado em ambiente Linux e deve ser compilado com o comando abaixo:

```
gcc -o tarefa1b main.c -lm
```

### 4. Análise de Resultados

Abaixo serão mostrados os resultados finais encontrados em cada uma das bases, juntamente com o valor da acurácia da tipagem geral dos *clusters*.

#### 4.1. Base 1

Para a 1ª base, o código apontou **Limiar = 0.109** como suficiente e apresentou como resultado de cada componente os seguintes tamanhos:

1. 51 vértices
2. 50 vértices
3. 49 vértices

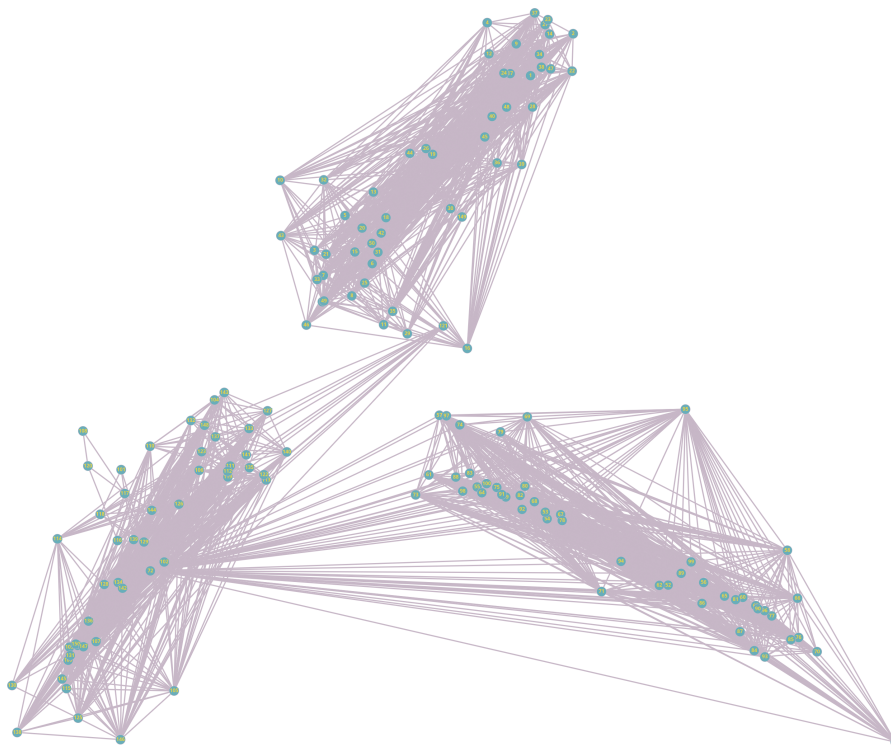


Figura 1. Grafo resultante da Base 1.

Real / Predito	P. Tipo 1	P. Tipo 2	P. Tipo 3
R. Tipo 1	50	0	0
R. Tipo 2	0	50	0
R. Tipo 3	1	0	49

Tabela 1. Matriz de confusão

Classe	TP	FP	FN	Precision	Recall	F1
Tipo 1	50	1	0	0.980	1.000	0.990
Tipo 2	50	0	0	1.000	1.000	1.000
Tipo 3	49	0	1	1.000	0.980	0.990

**Acurácia Global: 99.33%**

**Tabela 2. Métricas por classe com acurácia global.**

TP = True Positive FP = False Positive FN = False Negative

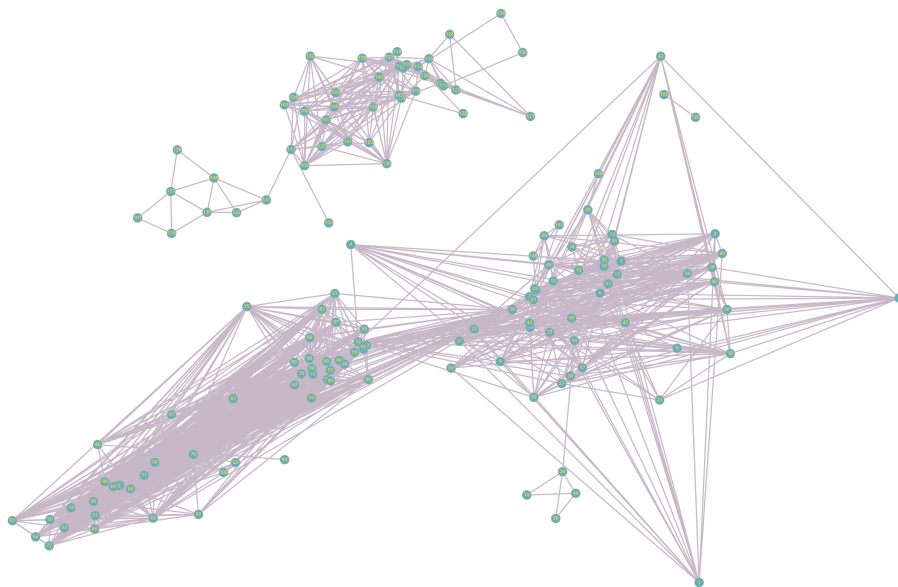
## 4.2. Base 2

Para a 2ª base, as distâncias euclidianas entre os vértices era pequena demais para ser separada apenas aplicando um limiar. O código encontrou **Limiar = 0.078** como resultado inicial para separação em três vértices suficientemente grandes para serem usados de base para o agrupamento dos demais componentes menores.

Os componentes e seus tamanhos foram como segue:

1. 53 vértices
2. 50 vértices
3. 44 vértices
4. 2 vértices
5. 1 vértice

Os três pontos restantes foram reatribuídos aos três maiores *clusters*, dois deles foram para o *cluster* 3 e o último para o *cluster* 1.



**Figura 2. Grafo resultante da Base 2.**

Real / Predito	P. Tipo 1	P. Tipo 2	P. Tipo 3
R. Tipo 1	50	0	0
R. Tipo 2	0	50	0
R. Tipo 3	4	0	46

**Tabela 3. Matriz de confusão.**

Classe	TP	FP	FN	Precision	Recall	F1
Tipo 1	50	4	0	0.926	1.000	0.962
Tipo 2	50	0	0	1.000	1.000	1.000
Tipo 3	46	0	4	1.000	0.920	0.958

**Acurácia Global: 97.33%**

**Tabela 4. Métricas por classe com acurácia global.**

TP = True Positive FP = False Positive FN = False Negative

## 5. Conclusão

Os valores de acurácia encontrados foram próximos de 100%, com a maior diferença sendo na 2ª base com cerca de 97% de resultados corretos de tipo dentro do *cluster*. Pode-se considerar como um treinamento suficientemente bom de separação de *Clusters* de Tipo.

## Referências

Costa, N. M. d. and Trentini, A. V. (2025). Bases\_rotuladas. [https://github.com/NicoleMCosta/Bases\\_Rotuladas](https://github.com/NicoleMCosta/Bases_Rotuladas). Acesso em: 05 de nov. 2025.