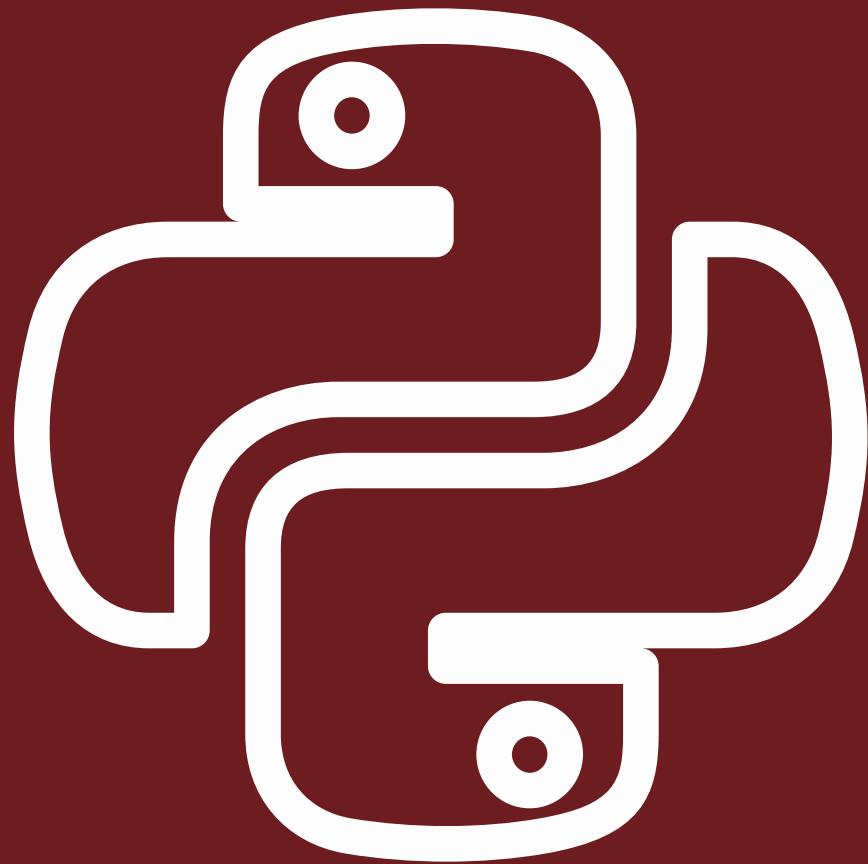


# ESTRUTURA DE DADOS



UNIVERSIDADE DE  
**vassouras**





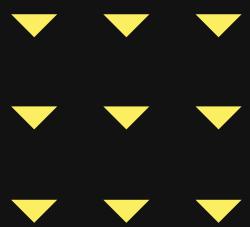
# EMENTA

Durante o semestre veremos os seguintes itens abaixo.

- Programação Básica em Python para nivelamento da turma.
- Notação **Big-O** (análise algoritmo)
- Vetores não ordenados e ordenados
- Pilhas, filas e deques
- Listas encadeadas (com Orientação a Objetos... trabalharemos com a ideia de ponteiros)



Durante o semestre veremos os seguintes itens abaixo. (continuação)



- Recursão (quando uma função chama ela mesma)
- Algoritmos de ordenação (Bubble sort, Selection sort, Insertion sort, Shell sort, Merge sort e Quick sort)
- Árvores
- Grafos (busca em largura e profundidade, Algoritmo de Dijkstra, busca gulosa e A\* (lê-se A estrela))

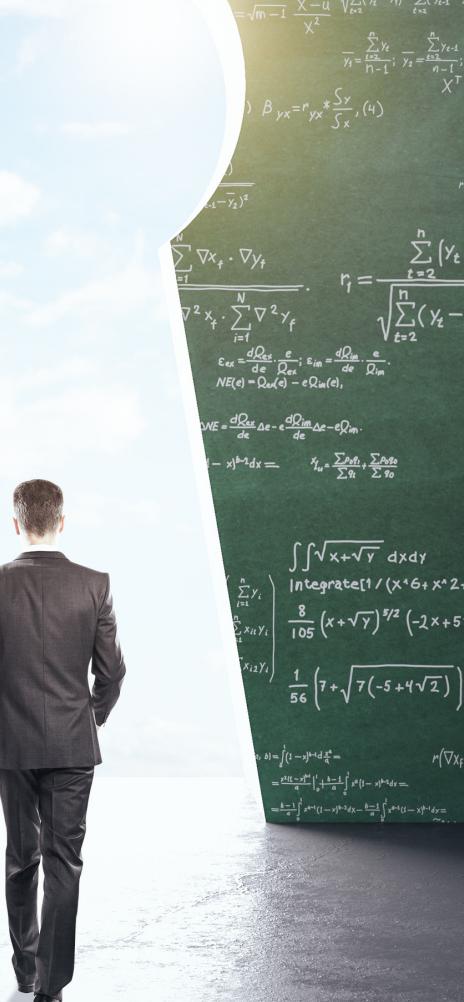
## Atenção



Os objetivos principais da Disciplina Estrutura de Dados **comigo** serão somente 2!

1. Entender a teoria e implementar/ testar as estruturas de dados!
2. Curtir o processo!

Pré-requisito **desejável** para melhor acompanhamento: lógica de programação

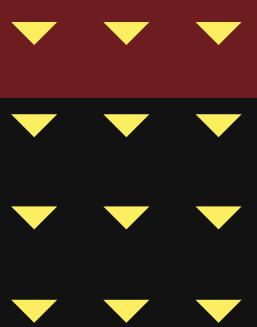


# **REFERÊNCIAS BIBLIOGRÁFICAS**

## Bibliografia básica



- EDELWEISS, NINA; RENATA, G. **Estruturas de Dados** - V18 - UFRGS. Grupo A, 2011. 9788577804504. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577804504/>.
- ESPÍNDOLA, C.T.; SANTOS, B.J.D.; OLIVEIRA, S.M.D.; ADRI, V. **Estrutura de Dados**. Grupo A, 2018. 9788595024328. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595024328/>.
- T., G.M.; ROBERTO, T. **Estruturas de Dados e Algoritmos em Java**. Grupo A, 2013. 9788582600191. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788582600191/>.



- PERKOVIC, Ljubomir. **Introdução à Computação Usando Python - Um Foco no Desenvolvimento de Aplicações.** Retirado de [https://integrada\[minhabiblioteca.com.br\]/#/books/9788521630937/](https://integrada[minhabiblioteca.com.br]/#/books/9788521630937/)
- KOFFMAN, B., E., WOLFGANG, T., P. A. **Objetos, Abstração, Estrutura de Dados e Projeto Usando C++.** Retirado de [https://integrada\[minhabiblioteca.com.br\]/#/books/978-85-216-2780-7/](https://integrada[minhabiblioteca.com.br]/#/books/978-85-216-2780-7/)
- Cury, Espíndola, T., Barreto, Santos, J. D., Saraiva, Oliveira, M. D., Vettorazzo, Adri. **Estrutura de Dados.** Retirado de [https://integrada\[minhabiblioteca.com.br\]/#/books/9788595024328](https://integrada[minhabiblioteca.com.br]/#/books/9788595024328)
- Drozdek, A. **Estrutura de Dados e Algoritmos em C++** - Tradução da 4<sup>a</sup> edição norte-americana. Retirado de [https://integrada\[minhabiblioteca.com.br\]/#/books/9788522126651/](https://integrada[minhabiblioteca.com.br]/#/books/9788522126651/)
- MARKENZON. **Estruturas de Dados e Seus Algoritmos.** Grupo GEN, 2010. 978-85-216-2995-5. Disponível em: [https://integrada\[minhabiblioteca.com.br\]/#/books/978-85-216-2995-5/](https://integrada[minhabiblioteca.com.br]/#/books/978-85-216-2995-5/).



# AVALIAÇÕES

## Avaliações



- P1:

Trabalho valendo 2,00. Prova valendo 8,00.

- P2:

Trabalho valendo 2,00. Prova valendo 8,00.

- 2º chamada:

Logo após perder uma prova, aconselho a providenciar o mais breve possível a documentação para a habilitação da 2º chamada.

Lembrando que:

1 - O aluno deve apresentar uma justificativa plausível.

2 - Independentemente da prova que o aluno perder, a matéria da 2º chamada será TODO o conteúdo e valendo 10,00.

Prova final: valendo 10,00 e será TODA a matéria.



**BORA COMEÇAR!**

**QUAL A DEFINIÇÃO DE  
ESTRUTURA DE DADOS ?**



## CONCEITO

Estrutura de Dados é uma área da Ciência da Computação que estuda as técnicas computacionais para a organização e manipulação eficiente de quaisquer quantidade de informações.



## CONCEITO

Em outras palavras, quando você desenvolver um sistema deve pensar em **quais estruturas de dados devem ser utilizadas!**

Sendo que a **escolha** de determinada estrutura de dados **influenciará** no desempenho do software.



## RESUMINDO

Entender sobre estrutura de dados é essencial para garantir que os algoritmos sejam **eficientes**, tanto em **tempo** de execução quanto em métricas de avaliação de **desempenho**.

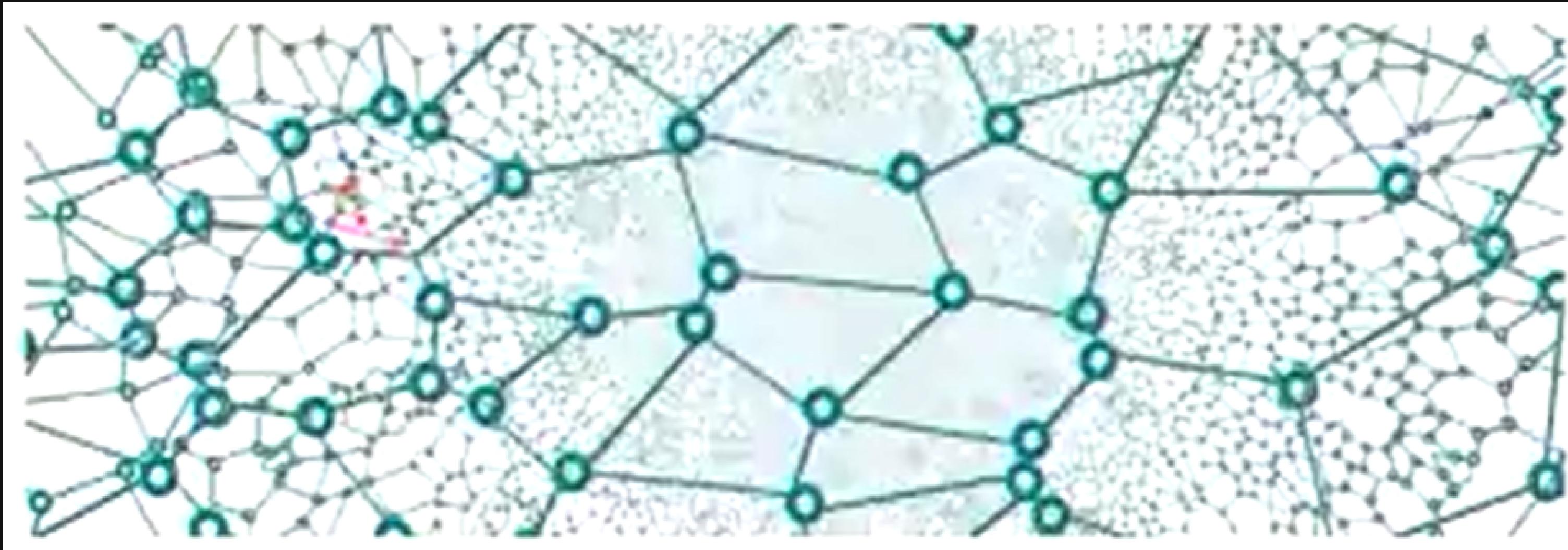


## UM EXEMPLO DE APLICAÇÃO

Temos uma rede de dispositivos eletrônicos. Tais dispositivos estão interligados por canais de comunicação. Cada canal tem um valor associado  $C$  (número real no intervalo  $0 \leq C \leq 1$ ) que representa sua confiabilidade. Interpreta-se como confiabilidade a probabilidade de que o canal não venha a falhar.

Forneça um algoritmo para encontrar o caminho mais confiável entre dois dispositivos dados

# Problema 1

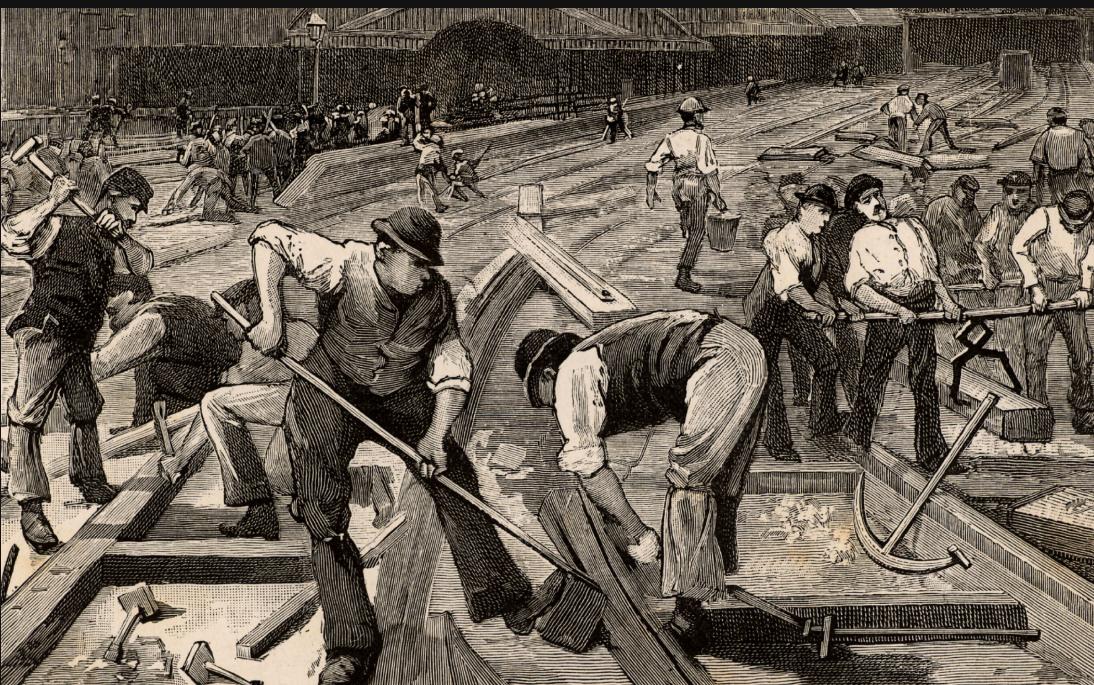


## Problema 2



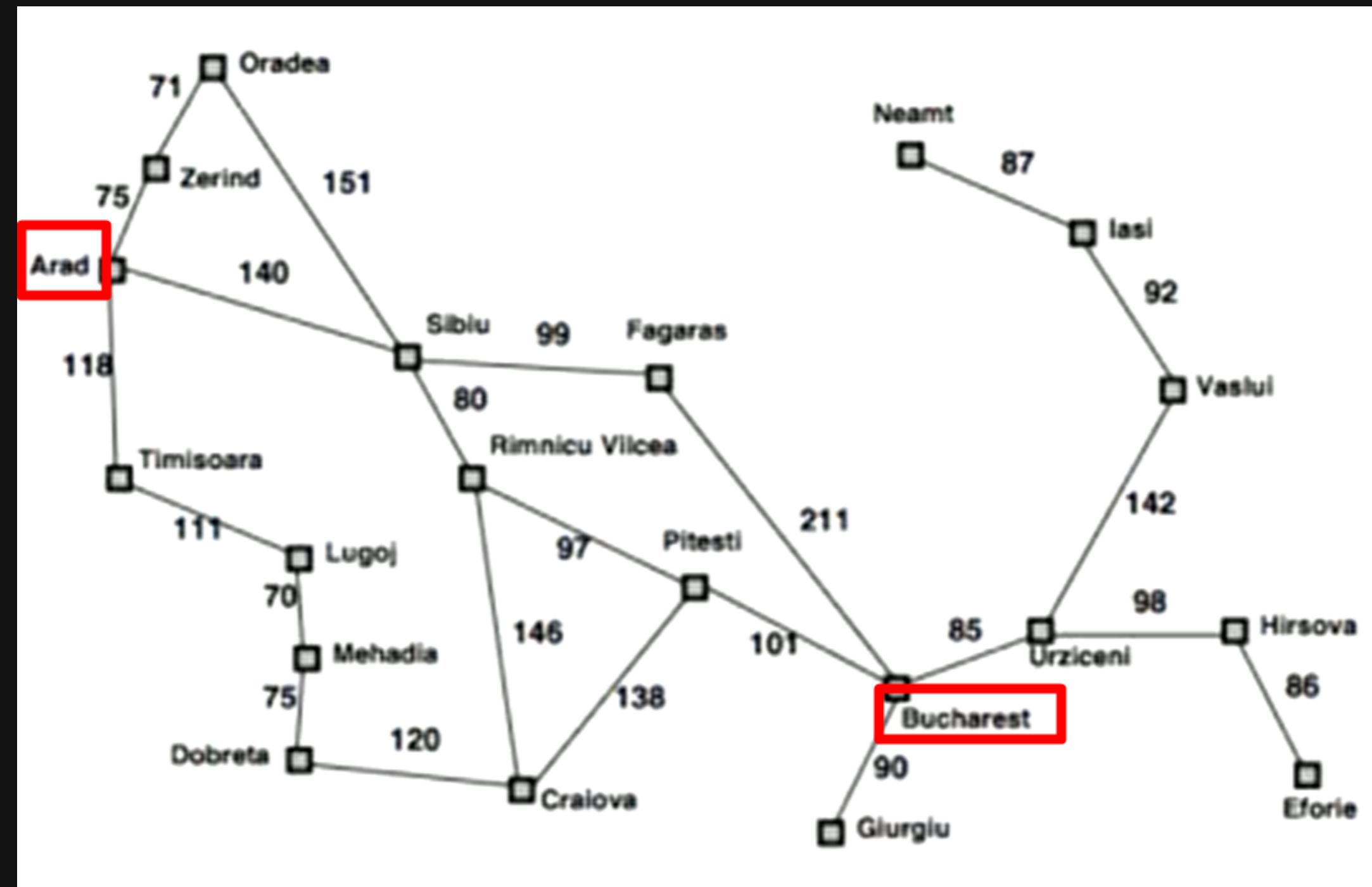
Considere uma rede de ferrovias conectando duas cidades através de um número de cidades intermediárias, onde cada ferrovia tem uma capacidade de transporte específica.

Como encontrar o fluxo máximo de passageiros entre as duas cidades?





Ir de Arad até Bucharest. Qual seria o menor caminho?



Cidades da Romênia



Como representar estes problemas em computadores?

Como construir os algoritmos necessários?

Que estrutura de dados utilizar?

De que forma os dados estarão organizados?

Que operações podem ser realizadas nestes dados?

**TODAS ESSAS PERGUNTAS SÃO RESPONDIDAS QUANDO SE ESTUDA A ÁREA DE ESTRUTURA DE DADOS**



Se organizem e formem 9 grupos de **5 pessoas**.

O grupo formado permanecerá até o final do semestre.



**5**

**OU**

VÁ P/ PRÓXIMO SLIDE

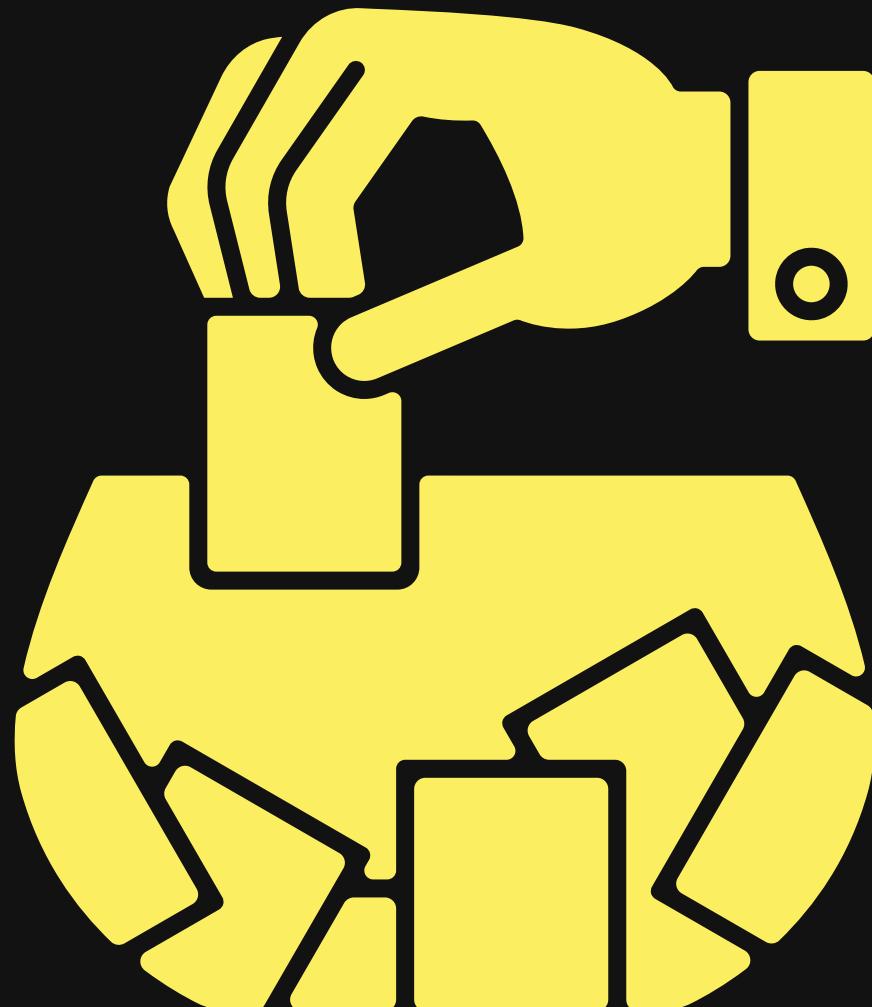
No momento vocês estão totalizando 51 pessoas, logo somente 1 grupo poderá ter 6 pessoas.

Resumindo: 9 grupos com 5 pessoas e 1 grupo com 6 pessoas.



Posso também sortear...

Talvez fique mais justo um sorteio para simularmos um ambiente empresarial onde não escolhemos nosso colega de equipe.





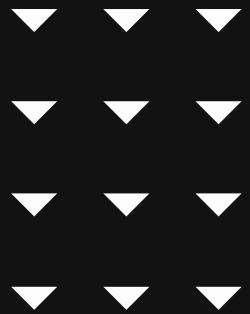
Estrutura de Dados  
Grupo do WhatsApp



# GRUPO DA DISCIPLINA

Foi criado um grupo no WhatsApp para passar informações, trabalhos e etc.

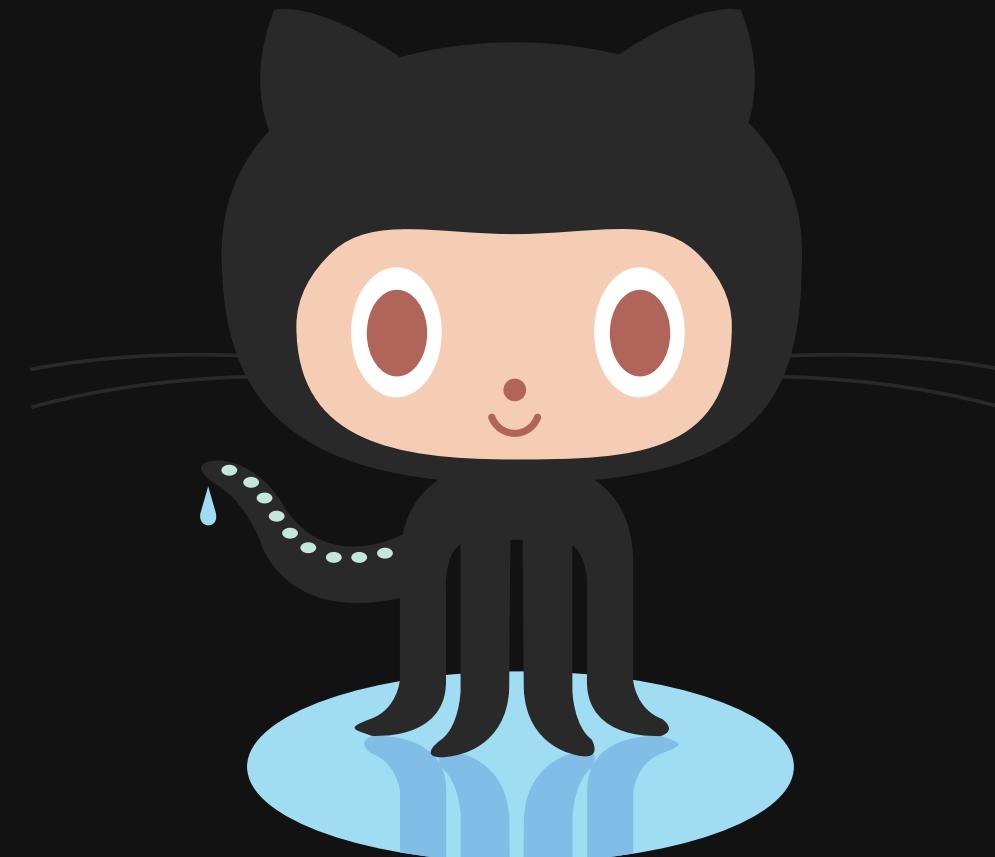
**Entrem.**



## Material Pedagógico

Utilizaremos o GitHub como nosso repositório de conteúdos.

Caso vocês ainda não tenham uma conta, criem uma o mais rápido possível!



<https://github.com/>

## Material Pedagógico

Utilizaremos o GitHub como nosso repositório de conteúdos.

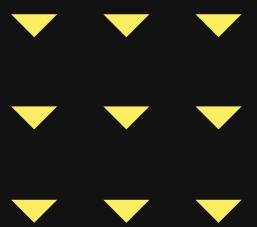
<https://github.com/cjannuzzi/ESW-estrutura-de-dados>

Nele vocês  
encontrarão todo o  
material da disciplina.

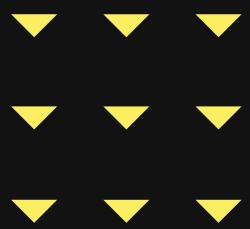


link do repo.

Mão no código

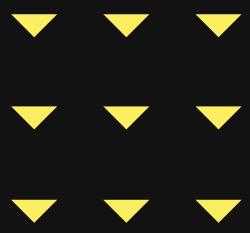


# Exercício



## 1. Coleta de Dados:

- criem um vetor para armazenar um conjunto de dados. Este pode ser, por exemplo, uma lista de idades ou qualquer outro conjunto de **dados numéricos simples**.
- O vetor deve ser preenchido com dados fornecidos pelo usuário, gerados aleatoriamente dentro de um intervalo definido, ou diretamente no código, conforme a preferência.

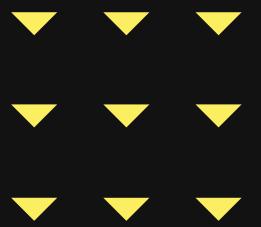


## 2. Processamento de Dados:

Implementar funções ou procedimentos para realizar as seguintes operações no vetor de dados:

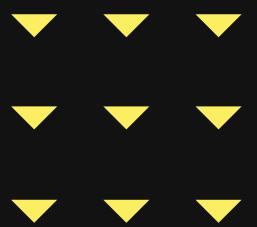
- Calcular a média: calcular a média dos valores armazenados no vetor.
- Encontrar o valor máximo e mínimo: identificar os valores máximo e mínimo dentro do vetor.
- Contar elementos dentro de um intervalo: contar quantos elementos estão dentro de um intervalo específico definido pelo usuário (por exemplo, quantas idades estão entre 18 e 25 anos).

# Mão no código - Análise de Dados com Vetores



## 3. Relatório dos Resultados:

Após realizar as operações acima, os alunos devem imprimir um relatório simples que mostre os resultados de cada operação (média, valor máximo e mínimo, e contagem de elementos dentro de um intervalo).



# Resolução

# Mão no código - Análise de Dados com Vetores

```
1 # 1. Criação do vetor com dados
2 dados = [20, 25, 22, 19, 30, 18, 24, 27, 21, 23, 29, 17]
3
4
5 # 2. Cálculo da média
6 def calcular_media(vetor):
7     return sum(vetor) / len(vetor)
8
9
10 media = calcular_media(dados)
11 print(f"Média: {media:.2f}")
12
13
14 # 3. Encontrar valor máximo e mínimo
15 def encontrar_maximo(vetor):
16     return max(vetor)
17
18
19 def encontrar_minimo(vetor):
20     return min(vetor)
21
22
23 maximo = encontrar_maximo(dados)
24 minimo = encontrar_minimo(dados)
25 print(f"Valor máximo: {maximo}")
26 print(f"Valor mínimo: {minimo}")
27
```

# Mão no código - Análise de Dados com Vetores

```
28
29 # 4. Contar elementos dentro de um intervalo
30 def contar_elementos_intervalo(vetor, inicio, fim):
31     contagem = 0
32     for valor in vetor:
33         if inicio <= valor <= fim:
34             contagem += 1
35     return contagem
36
37
38 contagem_intervalo = contar_elementos_intervalo(dados, 18, 25)
39 print(f"Elementos entre 18 e 25 anos: {contagem_intervalo}")
40
41
42 # Desafio Opcional: Ordenação do vetor (Bubble Sort)
43 def bubble_sort(vetor):
44     n = len(vetor)
45     for i in range(n):
46         for j in range(0, n - i - 1):
47             if vetor[j] > vetor[j + 1]:
48                 vetor[j], vetor[j + 1] = vetor[j + 1], vetor[j]
49     return vetor
50
51
52 dados_ordenados = bubble_sort(dados.copy()) # .copy() para não alterar o vetor original
53 print("Dados ordenados:", dados_ordenados)
54
```

*That's all Folks!*

ATÉ A PRÓXIMA!