

# ED62A-COM2A

# ESTRUTURAS DE DADOS

Aula 00 - Plano da disciplina

Prof. Rafael G. Mantovani

13/08/2019

# Porque usar EDs?



# Porque usar EDs?



**Dados**

# Porque usar EDs?



**Dados**



**Algoritmo /  
Estrutura**



# Porque usar EDs?



**Dados**



**Algoritmo /  
Estrutura**



**Conhecimento**

# Porque usar EDs?



**Dados**



**Algoritmo /  
Estrutura**

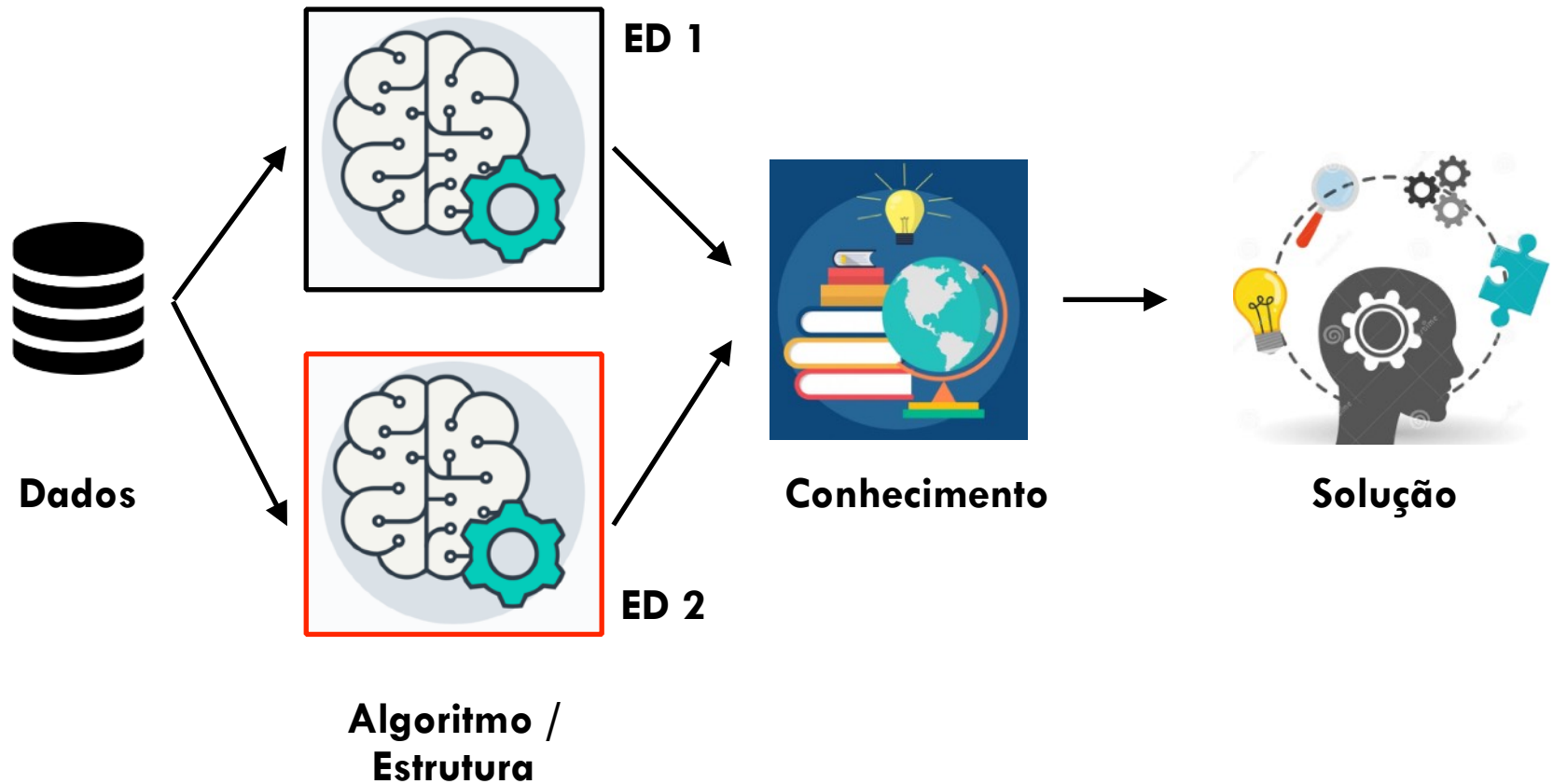


**Conhecimento**



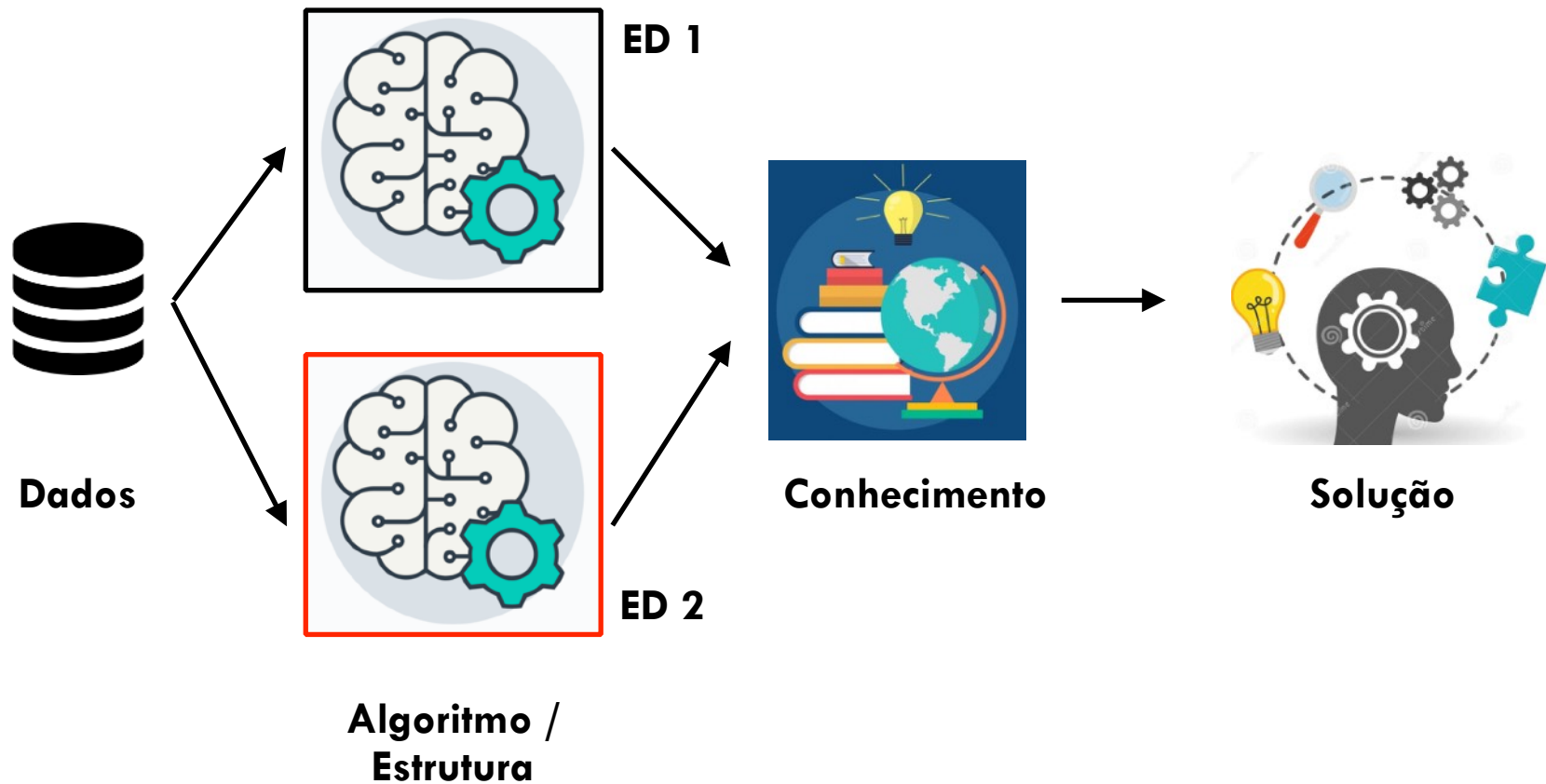
**Solução**

# Porque usar EDs?



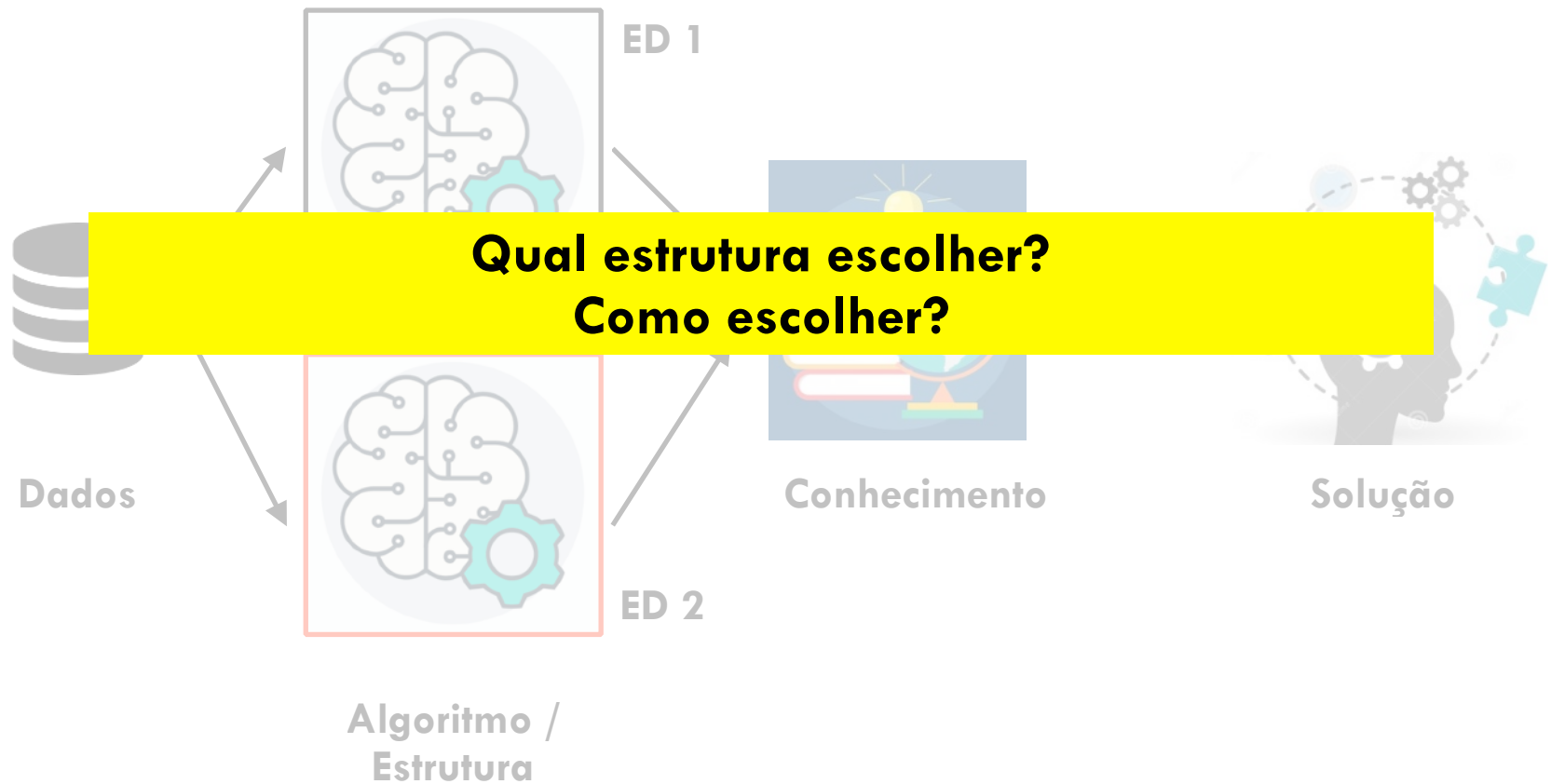
# Porque usar EDs?

**Diferentes formas de se resolver o mesmo problema !**





# Porque usar EDs?



# Porque usar EDs?

## Common Data Structure Operations

| Data Structure            | Time Complexity   |                   |                   |                   |                   |                   |                   |                   | Space Complexity    |
|---------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|---------------------|
|                           | Average           |                   |                   |                   | Worst             |                   |                   |                   | Worst               |
|                           | Access            | Search            | Insertion         | Deletion          | Access            | Search            | Insertion         | Deletion          |                     |
| <u>Array</u>              | $\theta(1)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$         |
| <u>Stack</u>              | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$         |
| <u>Queue</u>              | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$         |
| <u>Singly-Linked List</u> | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$         |
| <u>Doubly-Linked List</u> | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$       | $\theta(n)$       | $\theta(1)$       | $\theta(1)$       | $\theta(n)$         |
| <u>Skip List</u>          | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n \log(n))$ |
| <u>Hash Table</u>         | N/A               | $\theta(1)$       | $\theta(1)$       | $\theta(1)$       | N/A               | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$         |
| <u>Binary Search Tree</u> | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$         |
| <u>Cartesian Tree</u>     | N/A               | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | N/A               | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$         |
| <u>B-Tree</u>             | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(n)$         |
| <u>Red-Black Tree</u>     | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(n)$         |
| <u>Splay Tree</u>         | N/A               | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | N/A               | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(n)$         |
| <u>AVL Tree</u>           | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(n)$         |
| <u>KD Tree</u>            | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(\log(n))$ | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$       | $\theta(n)$         |

# Roteiro



- 1 Ementa**
- 2 Cronograma**
- 3 Avaliações**
- 4 Páginas com material da disciplina**
- 5 Referências**
- 6 Tarefas**

# Roteiro

- 1 Ementa**
- 2 Cronograma**
- 3 Avaliações**
- 4 Páginas com material da disciplina**
- 5 Referências**
- 6 Tarefas**

# Roteiro

- 1 Ementa**
- 2 Cronograma**
- 3 Avaliações**
- 4 Páginas com material da disciplina**
- 5 Referências**

# Ementa



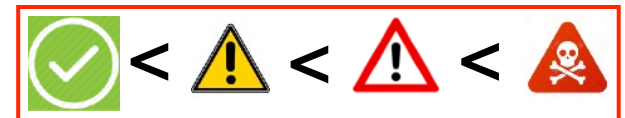
1. Tipos Abstratos de Dados
2. Listas Lineares
3. Filas
4. Pilhas
5. Árvores
6. Árvores AVL
7. Árvores B
8. Hash
9. Grafos

# Ementa

1. Tipos Abstratos de Dados
2. Listas Lineares
3. Filas
4. Pilhas
5. Árvores
6. Árvores AVL
7. Árvores B
8. Hash
9. Grafos



**Nível de  
Dificuldade**



# Roteiro

- 1 Ementa
- 2 Cronograma
- 3 Avaliações
- 4 Páginas com material da disciplina
- 5 Referências



# Cronograma

**Março**

TAD, Listas, Filas

**Abril**

Pilhas, Árvores,  
AVL, B-Tree

**Maio**

Heap, Hash  
Grafos

**Junho**

Algoritmos  
de Grafos

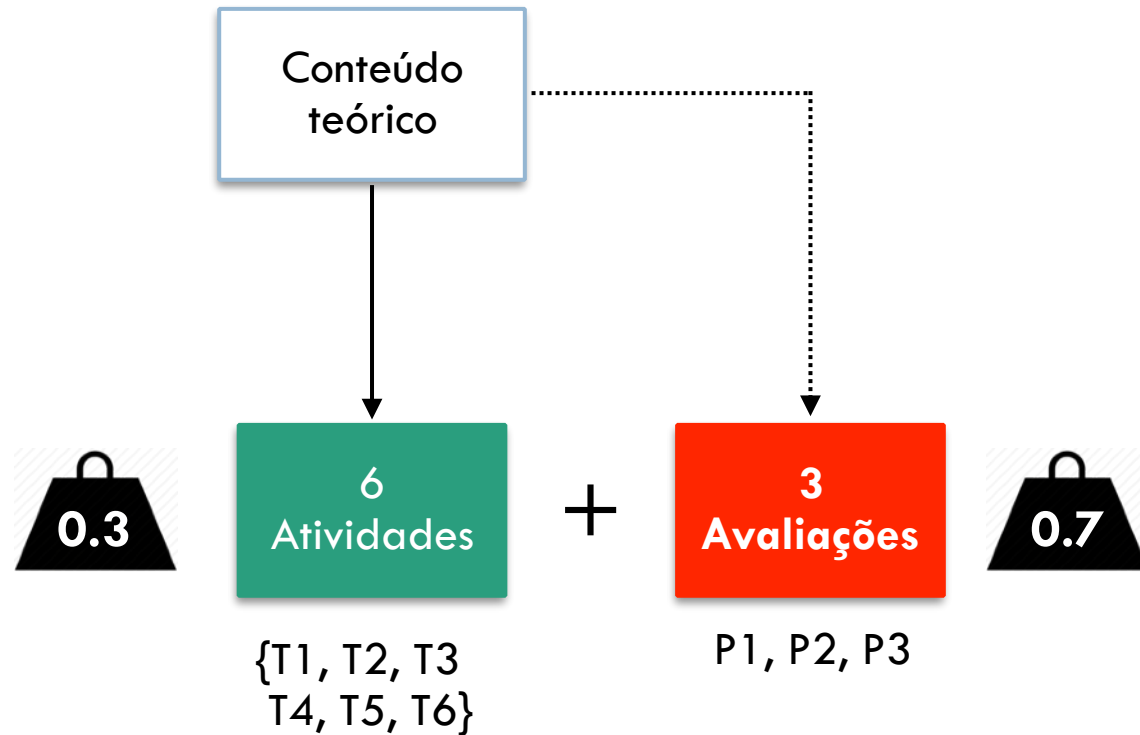
**Julho**

Exame

# Roteiro

- 1 Ementa
- 2 Cronograma
- 3 Avaliações
- 4 Páginas com material da disciplina
- 5 Referências

# Avaliações



# Ementa



1. Tipos Abstratos de Dados
2. Listas Lineares
3. Filas
4. Pilhas
5. Árvores [binárias]
6. Heaps
7. Árvores AVL
8. Árvores Red Black
9. Hash
10. Grafos

# Ementa

1. Tipos Abstratos de Dados
2. Listas Lineares
3. Filas
4. Pilhas
5. Árvores [binárias]
6. Heaps
7. Árvores AVL
8. Árvores Red Black
9. Hash
10. Grafos



**Nível de  
dificuldade  
(implementação)**



# Roteiro

- 1 Ementa
- 2 Cronograma
- 3 Avaliações
- 4 Páginas com material da disciplina
- 5 Referências
- 6 Tarefas

# Cronograma

**Agosto**

TAD, Listas,  
Filas, Pilhas

**Setembro**

Árvores, AVL,  
Red Black

**Outubro**

Heap, Hash

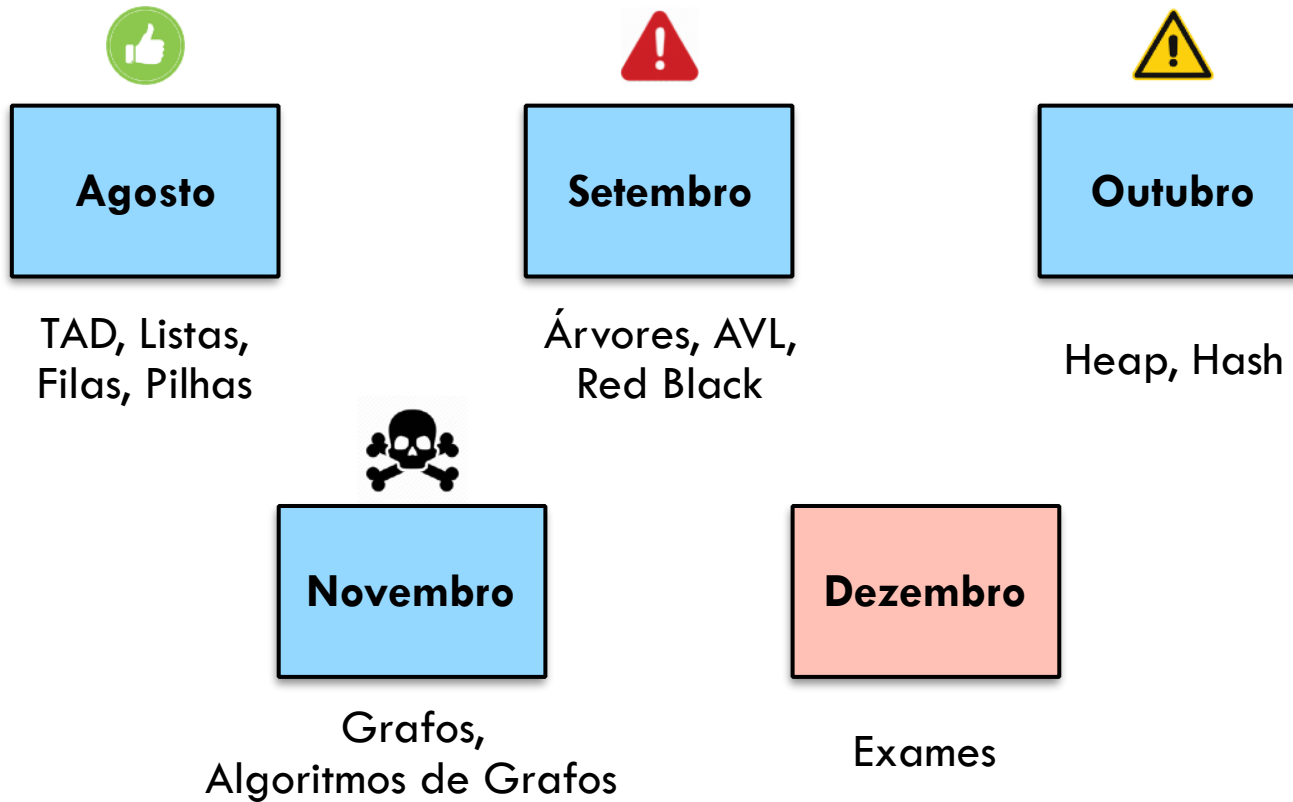
**Novembro**

Grafos,  
Algoritmos de Grafos

**Dezembro**

Exames

# Cronograma

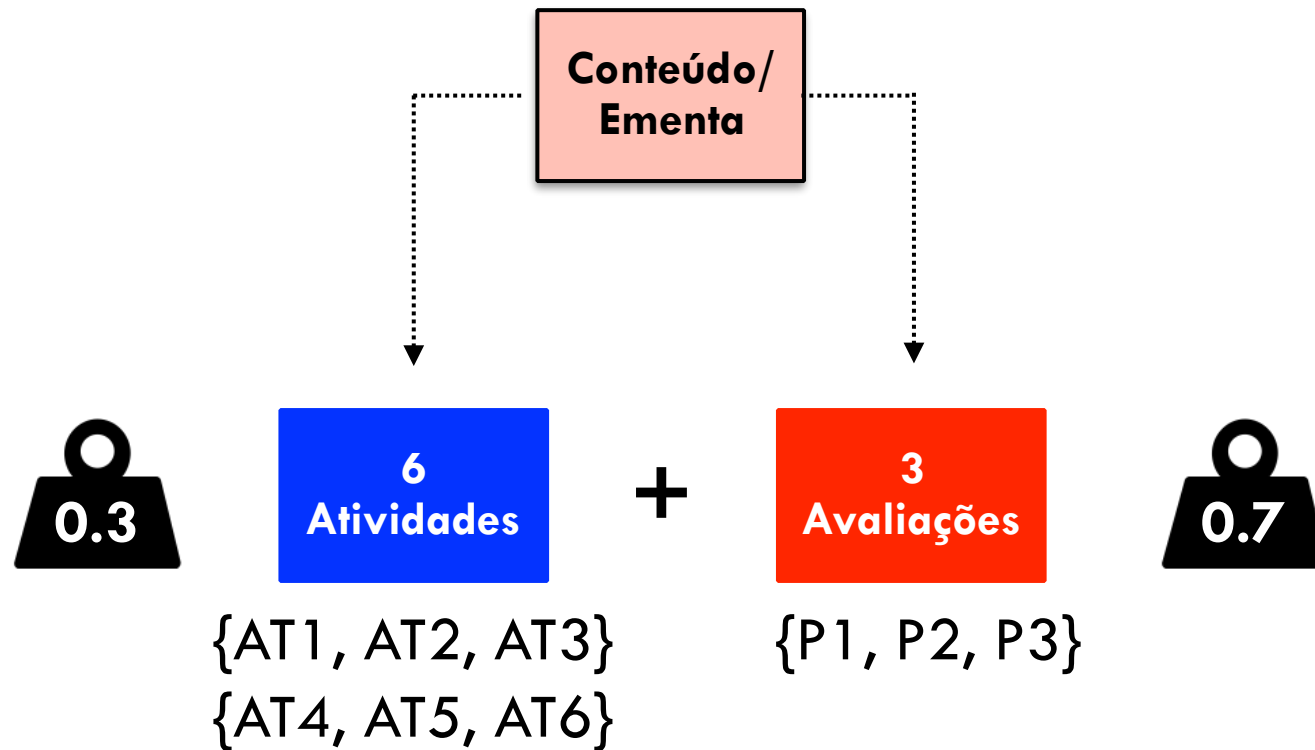




# Roteiro

- 1 Ementa
- 2 Cronograma
- 3 Avaliações
- 4 Páginas com material da disciplina
- 5 Referências
- 6 Tarefas

# Avaliações



# Provas

- Planejamento:
  - **P1**: 06/09 → {Listas, Pilhas, Filas}
  - **P2**: 29/10 → {Árvores, AVL, Hash, Heaps}
  - **P3**: **06/12** → {Grafos, Algoritmos de Grafos}
  - **Exame**: 13/12 → **TUDO**

# Provas

**P1**

Estruturas Elementares

10,00

sexta: 06/09

**P2**

Árvores, Hash, Heap

10,00

terça: 29/10

**P3**

Grafos

10,00

sexta: 06/12

**Exame: TUDO!**

10,00

sexta: 13/12

# Provas

**P1**  
Estruturas Elementares

10,00

sexta: 06/09

**P2**  
Árvores, Hash, Heap

10,00

terça: 29/10

**P3**  
Grafos

10,00

sexta: 06/12

Exame: **TUDO!**

10,00

sexta: 13/12

# Atividades práticas

- Uso de Estruturas de dados em problemas reais
  - **AT01**: Pilhas
  - **AT02**: Listas
  - **AT03**: Árvores
  - **AT04**: AVLs e Red-black
  - **AT05**: Grafos
  - **AT06**: Dijkstra

# Atividades práticas

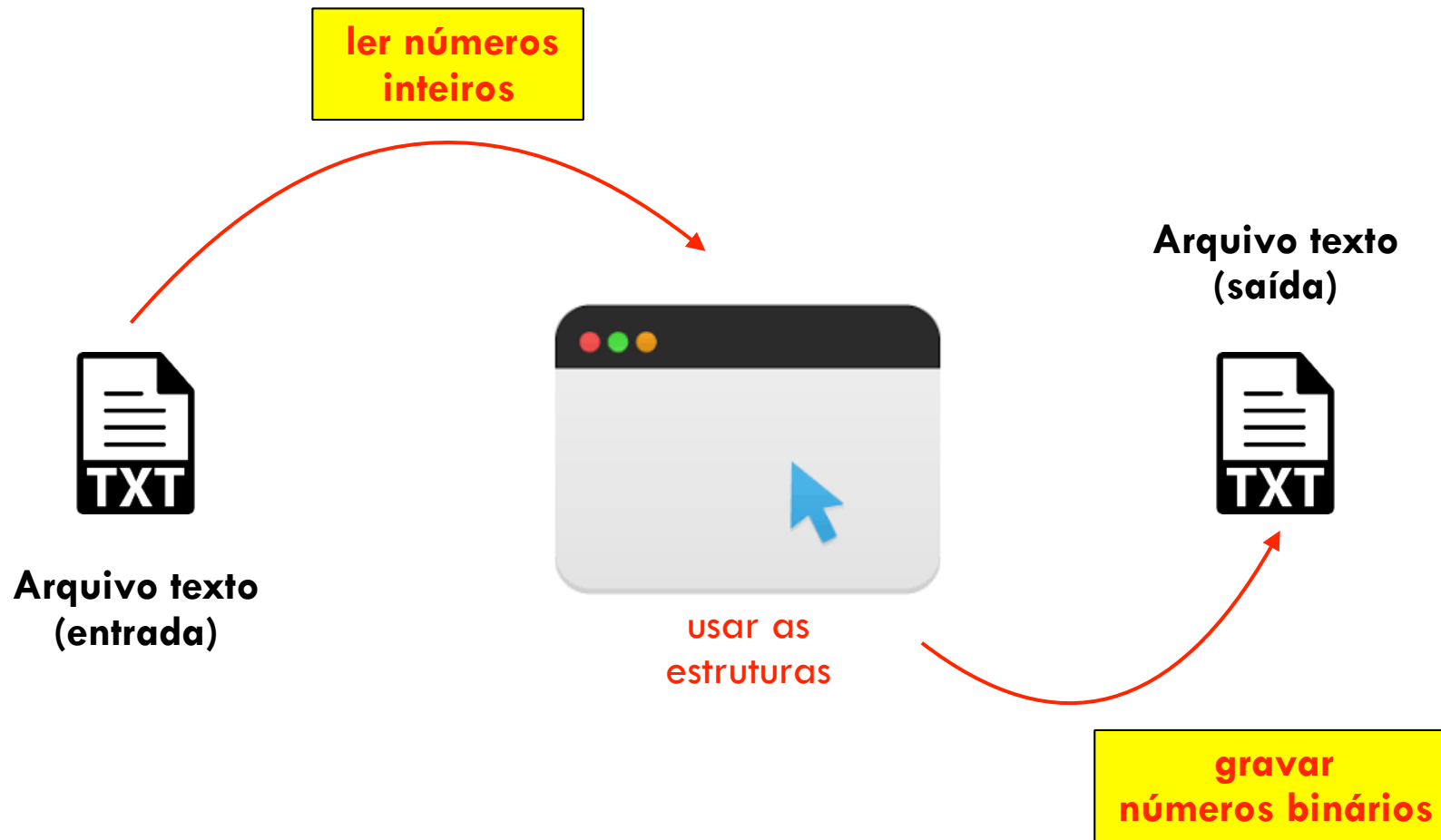
- Uso de Estruturas de dados em problemas reais
  - **AT01**: Pilhas
  - **AT02**: Listas
  - **AT03**: Árvores
  - **AT04**: AVLs e Red-black
  - **AT05**: Grafos
  - **AT06**: Dijkstra
  - **AT07**: Prim / Kruskal

# Atividades práticas





# Exemplo



# Atividades práticas

- **Prazos:** 1-2 semanas para desenvolvimento e entrega
- **Individuais** (evitar o plágio)
- Nota avaliada:

**Entregue no dia**

**100%**

**Atraso  $\leq$  4 dias**

**50%**

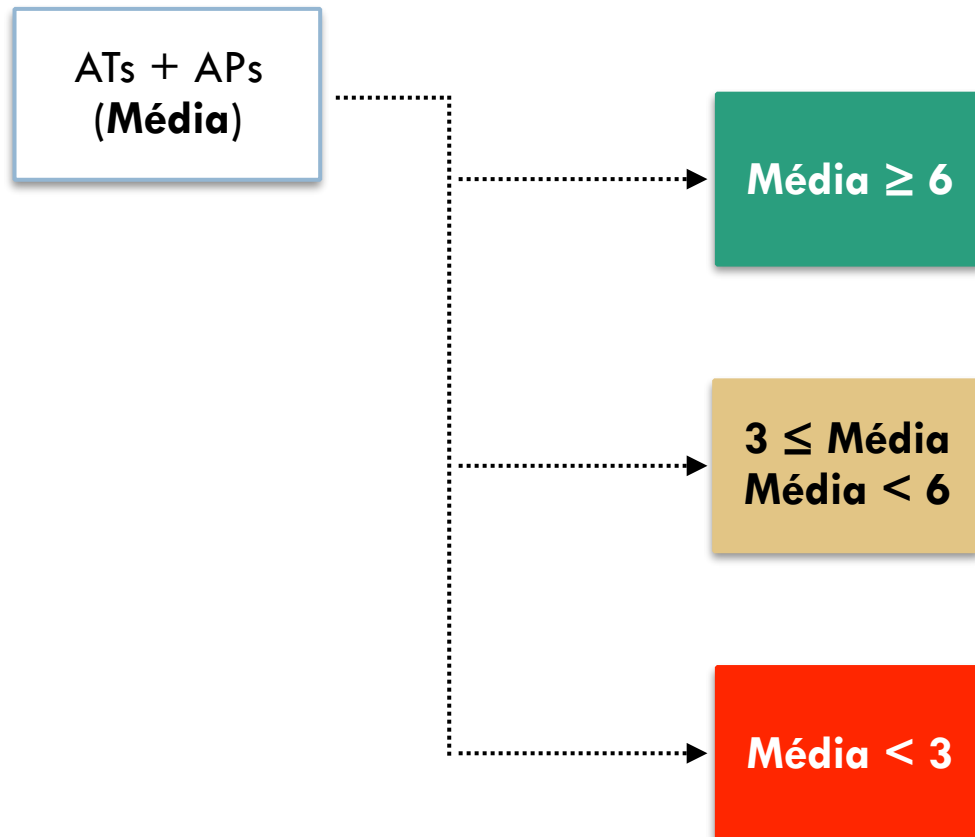
**> 4 dias**

**0%**

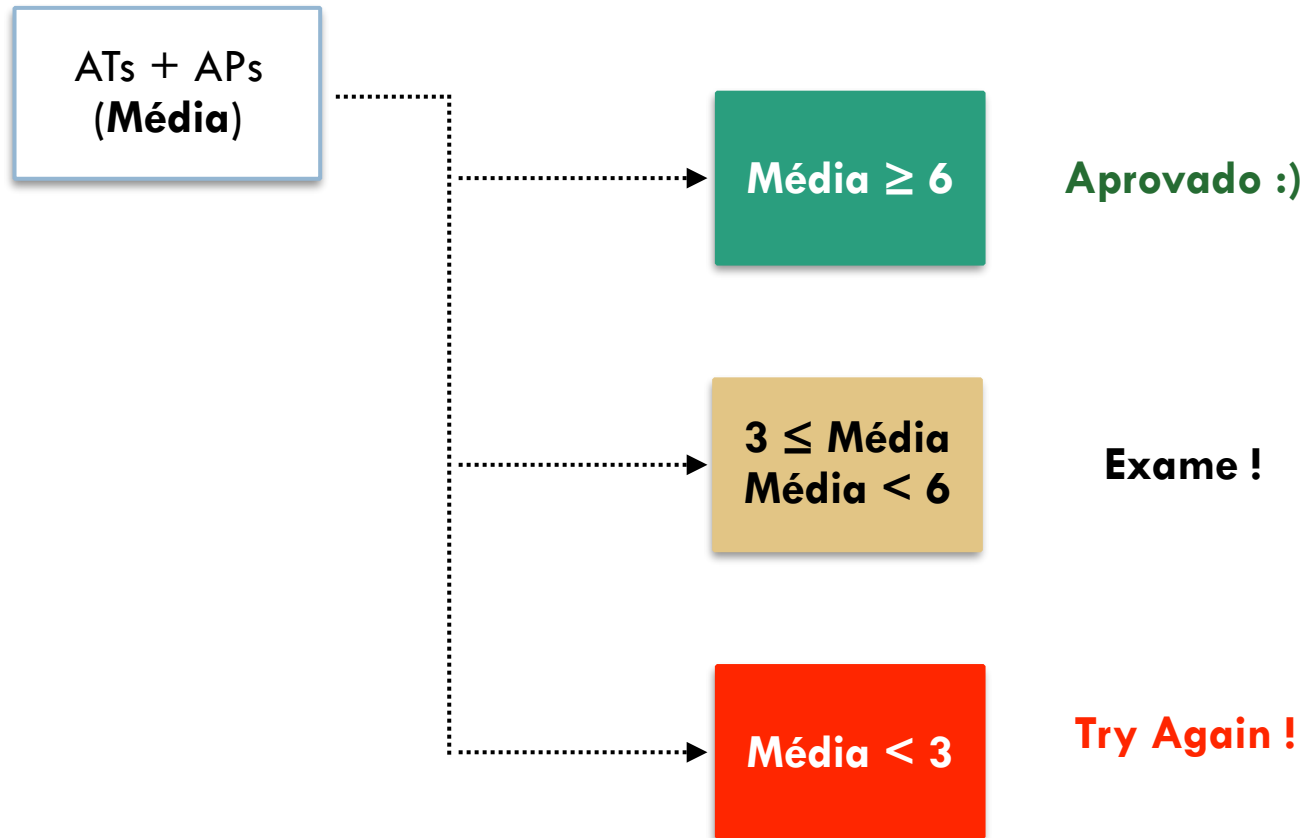
# Atividades práticas

- Implementação dos Algoritmos
  - C (não é permitido usar libs prontas)
  - IDE (livre escolha)
  
- Aplicação delas em domínios diferentes
  - manipulação de arquivos texto
  - entrada / saída
  
- Entregas:
  - **Moodle UTFPR**
  - Acompanhamento pelo GitHub Educational (links)

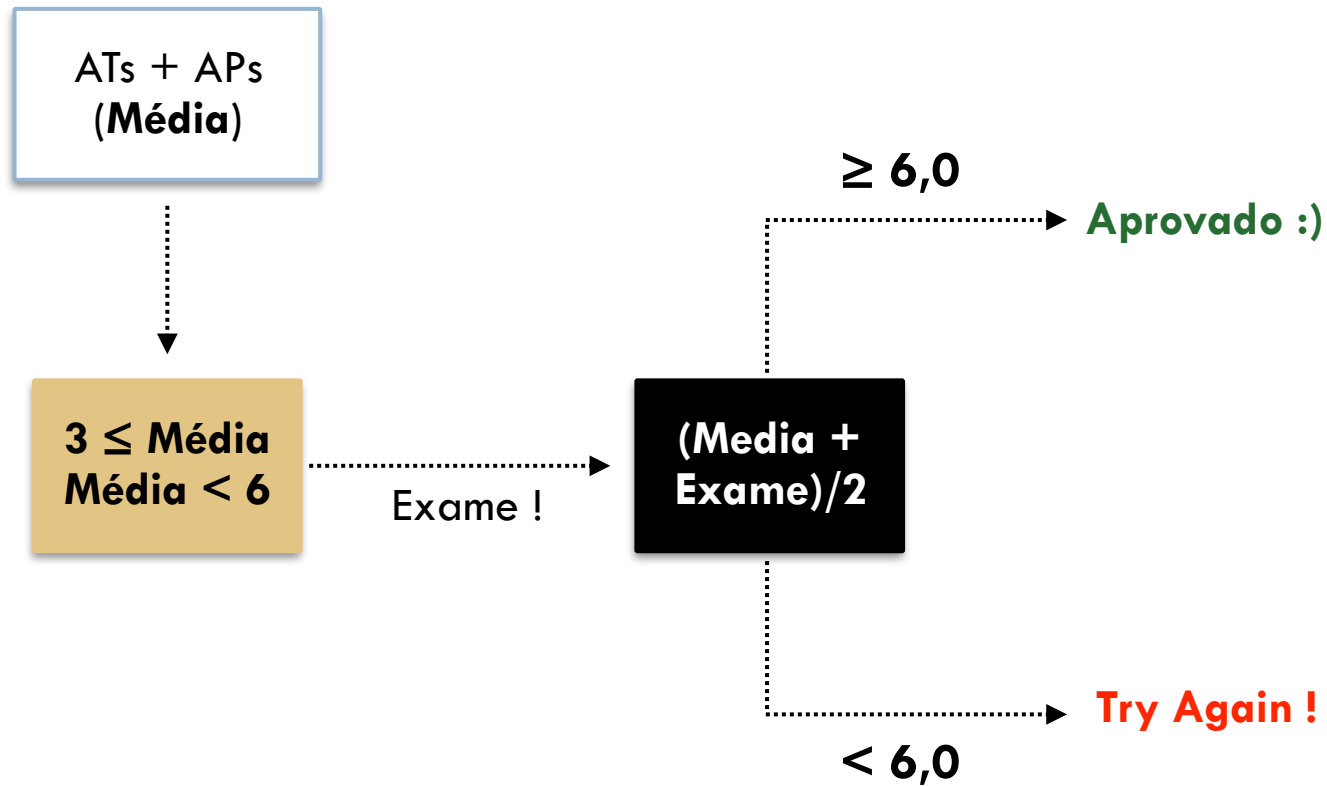
# Média Final



# Média Final



# Média Final && Exame



# Roteiro

- 1 Ementa
- 2 Cronograma
- 3 Avaliações
- 4 Páginas com material da disciplina
- 5 Referências
- 6 Tarefas

# Páginas com material

- <http://moodle.utfpr.edu.br>
  - Apucarana / Graduação / Engenharia de Computação / 2 período / Estrutura de Dados

Moodle UTFPR

Português - Brasil (pt\_br) ▾

## Estrutura de Dados

Página inicial ▶ Meus cursos ▶ ED62A-EC-AP

NAVEGAÇÃO

Página inicial

■ Painel

▶ Páginas do site

▼ Meus cursos

▶ ED62A-EC-AP

 Avisos

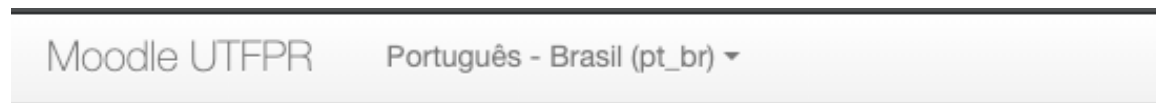
---

Tópico 1



# Páginas com material

- <http://moodle.utfpr.edu.br>
  - Apucarana / Graduação / Engenharia de Computação / 2 período / Estrutura de Dados



## Estrutura de Dados

**Senha: ed20192sem**

[Página inicial](#) ► [Meus cursos](#) ► [ED62A-EC-AP](#)

### NAVEGAÇÃO

[Página inicial](#)

■ [Painel](#)

► [Páginas do site](#)

▼ [Meus cursos](#)

► [ED62A-EC-AP](#)



Avisos

## Tópico 1

# Páginas com material (espelho)

□ <https://github.com/rgmantovani/dataStructures>

The screenshot shows the GitHub repository page for `rgmantovani / dataStructures`. The repository is described as "Data structures lectures from UTFPR - Apucarana". It has 1 commit, 1 branch, 0 releases, and 1 contributor. The main branch is `master`. The repository contains a `README.md` file, which is the only file shown in the file list. The commit history shows a single commit by `rgmantovani` with the message "first commit". The repository is currently on the `master` branch. The repository is public and has 1 star and 0 forks.

rgmantovani / dataStructures

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Data structures lectures from UTFPR - Apucarana

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

rgmantovani first commit Latest commit 2a8f9b3 a minute ago

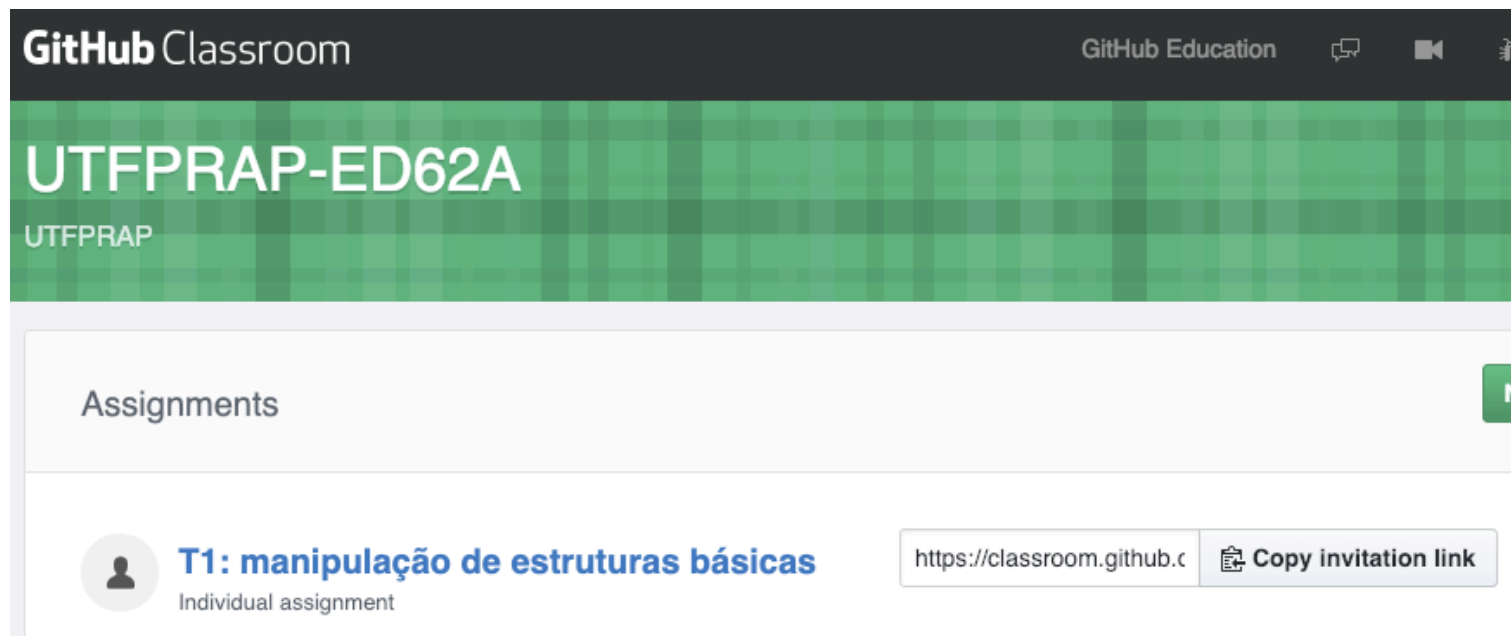
README.md first commit a minute ago

README.md

dataStructures

# Páginas com material (práticas)

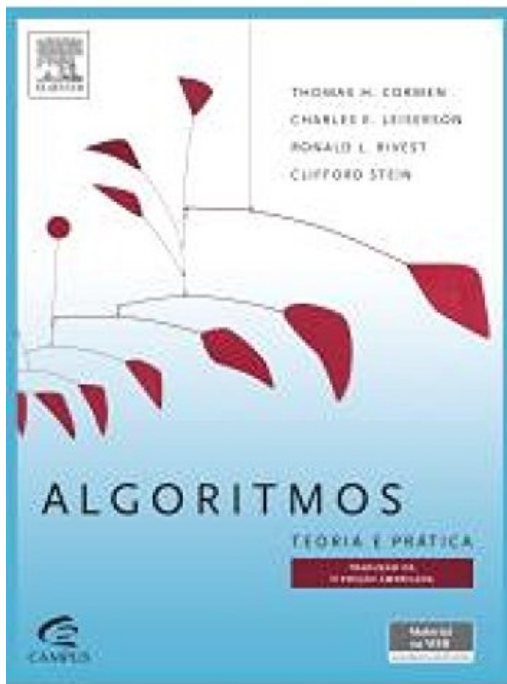
- <https://classroom.github.com/classrooms/37255181-utfprap-ed62a>



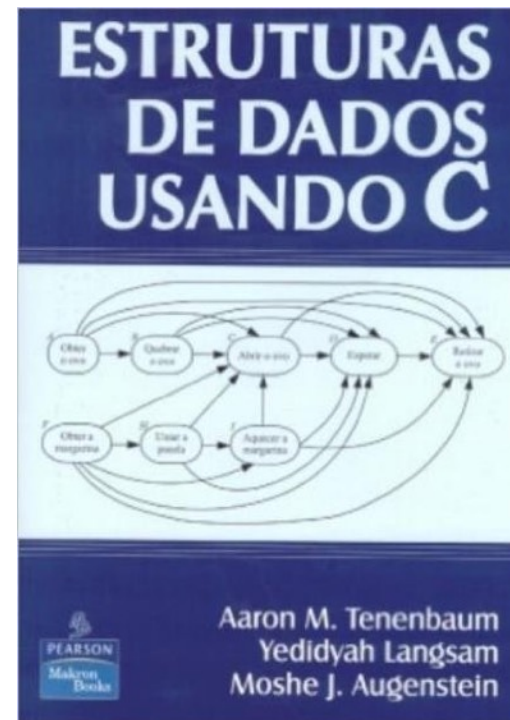
# Roteiro

- 1 Ementa
- 2 Cronograma
- 3 Avaliações
- 4 Página com material
- 5 Referências
- 6 Tarefas

# Referências sugeridas

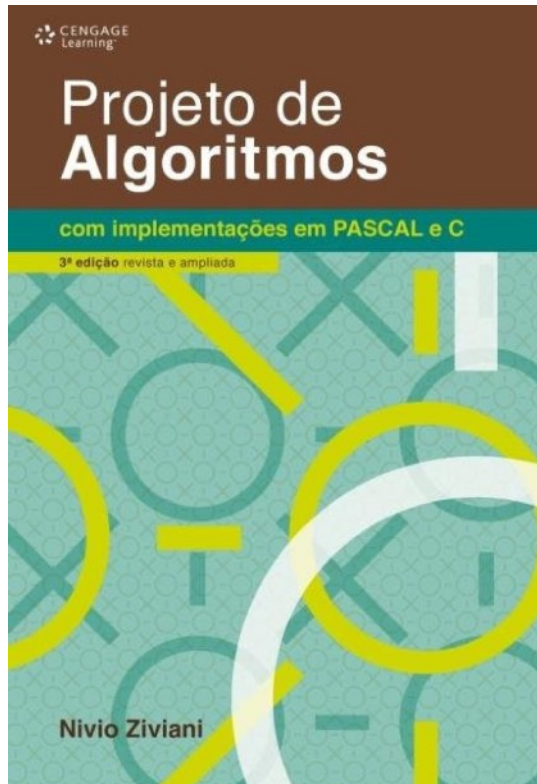


[Cormen et al, 2018]



[Tenenbaum et al, 1995]

# Referências sugeridas



[Ziviani, 2010]



[Drozdek, 2017]

# Perguntas?

Prof. Rafael G. **Mantovani**

[rafaelmantovani@utfpr.edu.br](mailto:rafaelmantovani@utfpr.edu.br)

# Tarefas

- 1 Criar conta no Github (usar e-mail institucional)**



# Tarefas

- 1 Criar conta no Github (usar e-mail institucional)
- 2 Ler tutorial de git (Moodle, GitHub da disciplina)

# Tarefas

- 1 Criar conta no Github (usar e-mail institucional)
- 2 Ler tutorial de git (Moodle, GitHub da disciplina)
- 3 Fazer um projeto teste, commitar e versionar arquivos

# Tarefas

- ❶ Criar conta no Github (usar e-mail institucional)
- ❷ Ler tutorial de git (Moodle, GitHub da disciplina)
- ❸ Fazer um projeto teste, commitar e versionar arquivos

**Baralho !!!**