

EDCO3A

ESTRUTURAS DE DADOS 1

Aula 00 - Plano da disciplina

Prof. Rafael G. Mantovani

Licença

Este trabalho está licenciado com uma Licença CC BY-NC-ND 4.0:



maiores informações:

https://creativecommons.org/licenses/by-nc-nd/4.0/deed.pt_BR

Porque usar EDs?



Porque usar EDs?



Até agora aprendemos a manipular os comandos básicos durante a programação. Mas e se precisarmos de algo mais complexo cujos tipos básicos não são suficientes? O que fazer?

Porque usar EDs?



Dados

Porque usar EDs?

**Todo programa é alimentado
com dados (informações) do
mundo real**



Dados

Porque usar EDs?



Dados



**Algoritmo /
Estrutura**



Porque usar EDs?

Nós então criamos soluções por meio de algoritmos e programas complexos, para processar essa informação.



Dados



**Algoritmo /
Estrutura**



Porque usar EDs?



Dados



**Algoritmo /
Estrutura**



Conhecimento

Porque usar EDs?

Esses algoritmos/programas geram conhecimento sobre o domínio/problema em questão.



Dados

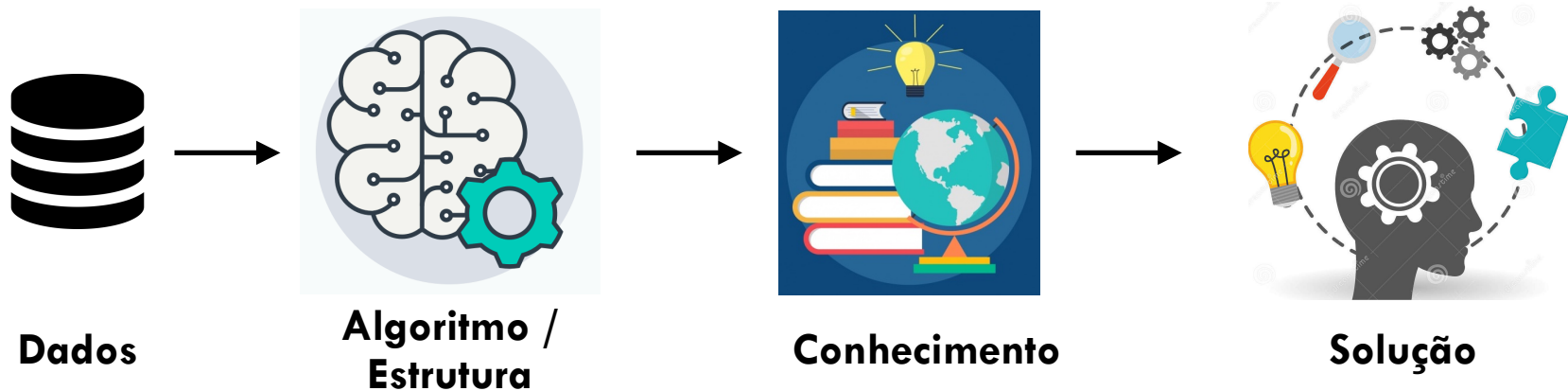


**Algoritmo /
Estrutura**



Conhecimento

Porque usar EDs?



Porque usar EDs?

**E por fim, geramos
soluções reais baseadas
nesse conhecimento
adquirido :)**



Dados



**Algoritmo /
Estrutura**



Conhecimento



Solução

Porque usar EDs?

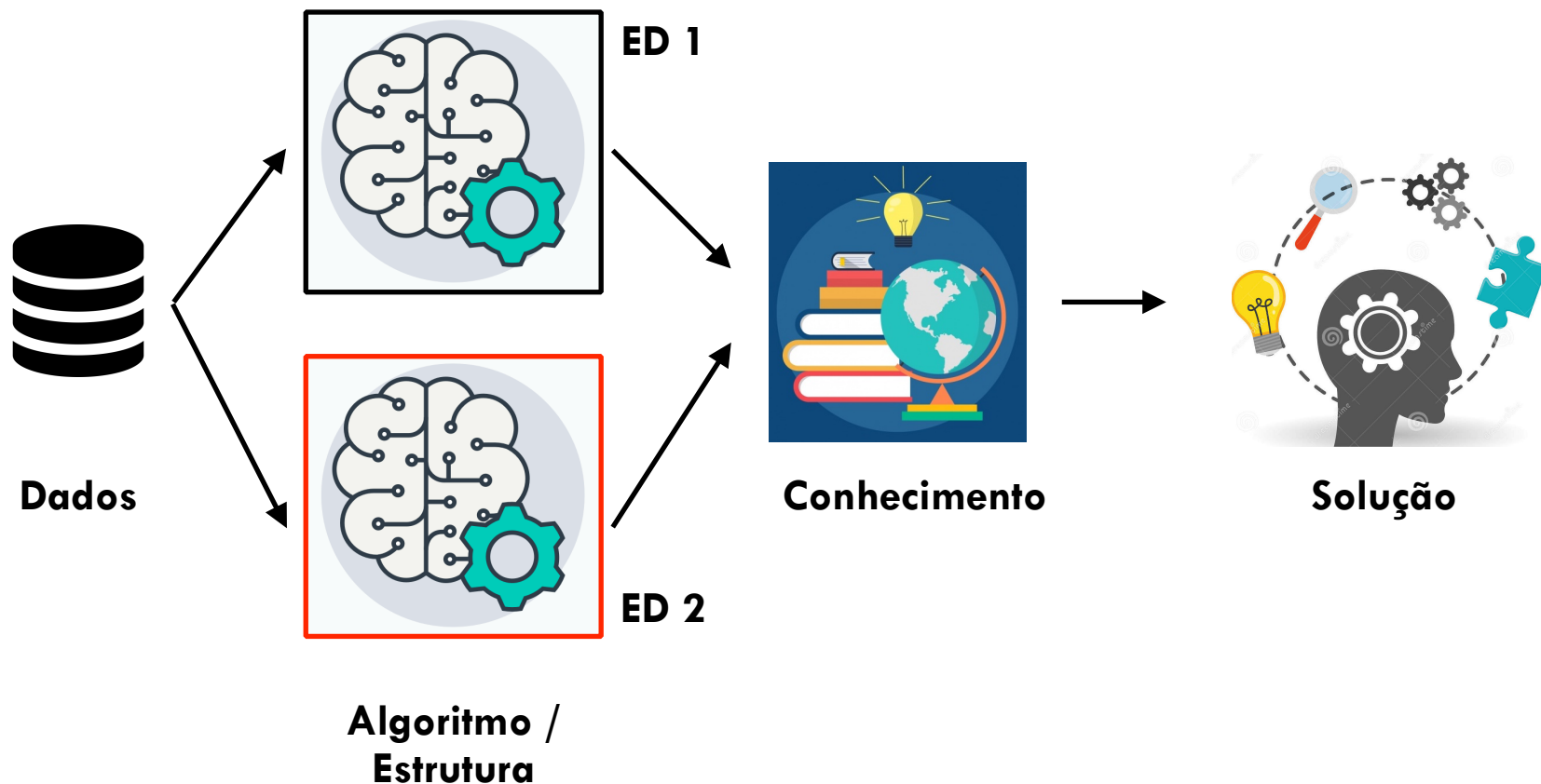


Mas ...

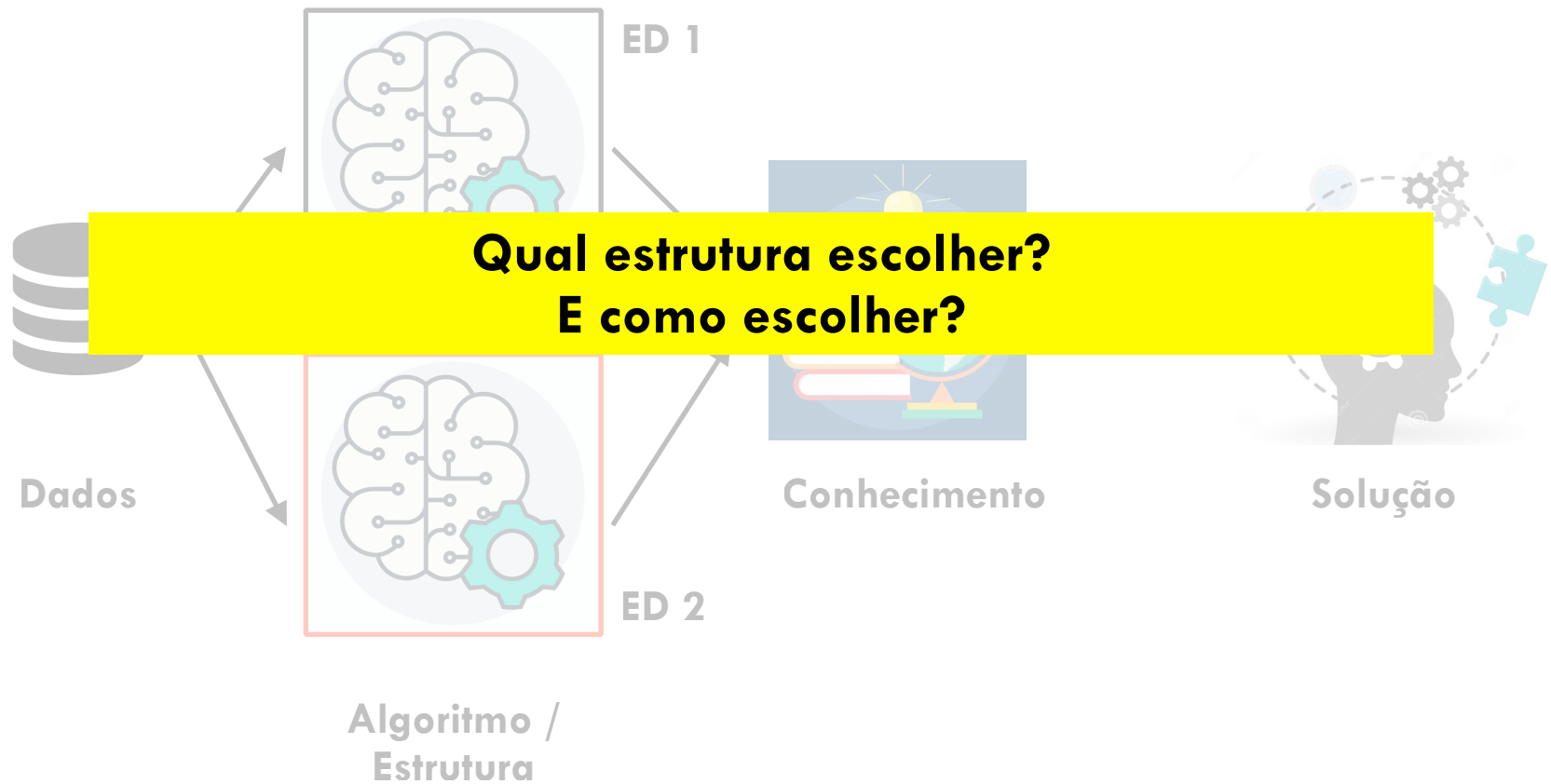


Porque usar EDs?

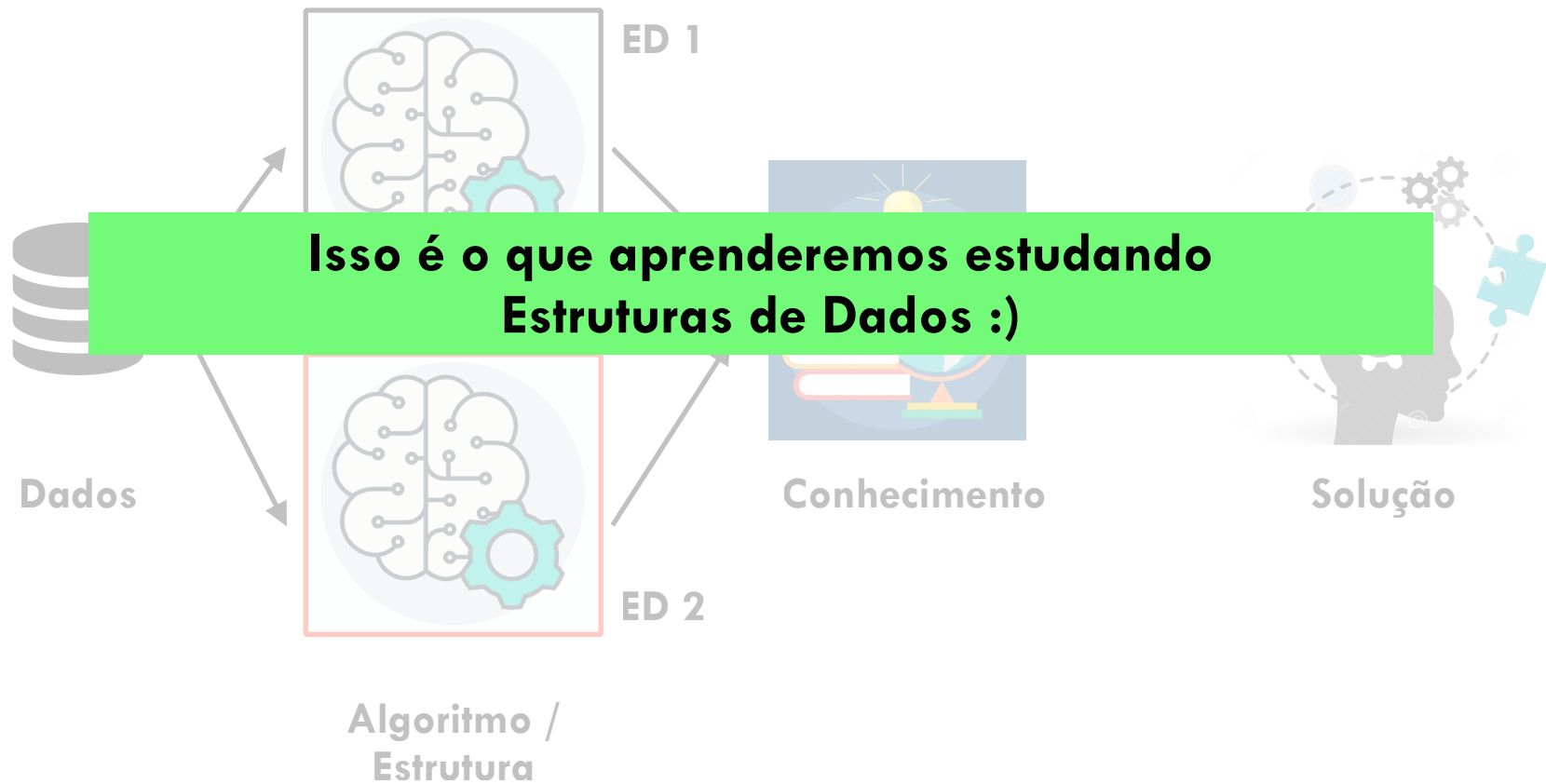
Existem diferentes formas de se resolver o mesmo problema !



Porque usar EDs?



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>	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(\log(n))$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$	$\theta(n)$

Porque usar EDs?

Cada estrutura tem um custo associado a suas operações, e até o fim da nossa disciplina, vamos entender essa tabela :)

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**

Ementa



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

Ementa

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



**Nível de
dificuldade
(implementação)**



Roteiro

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

Cronograma



Junho

TAD, Listas,
Filas, Pilhas

Julho

Árvores, AVL,
Heap, Hash

Agosto

Grafos,
Algoritmos de Grafos
Ordenação

Cronograma



Junho

TAD, Listas,
Filas, Pilhas



Julho

Árvores, AVL,
Heap, Hash



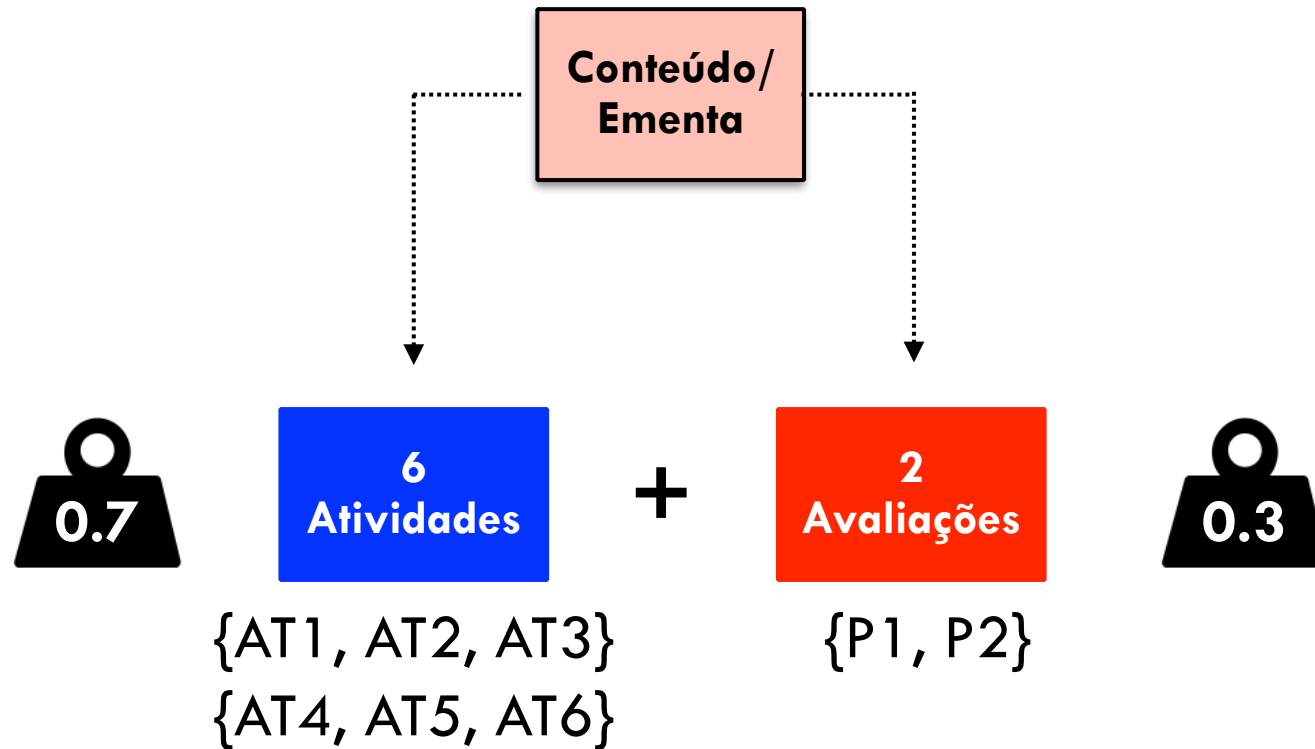
Agosto

Grafos,
Algoritmos de Grafos
Ordenação

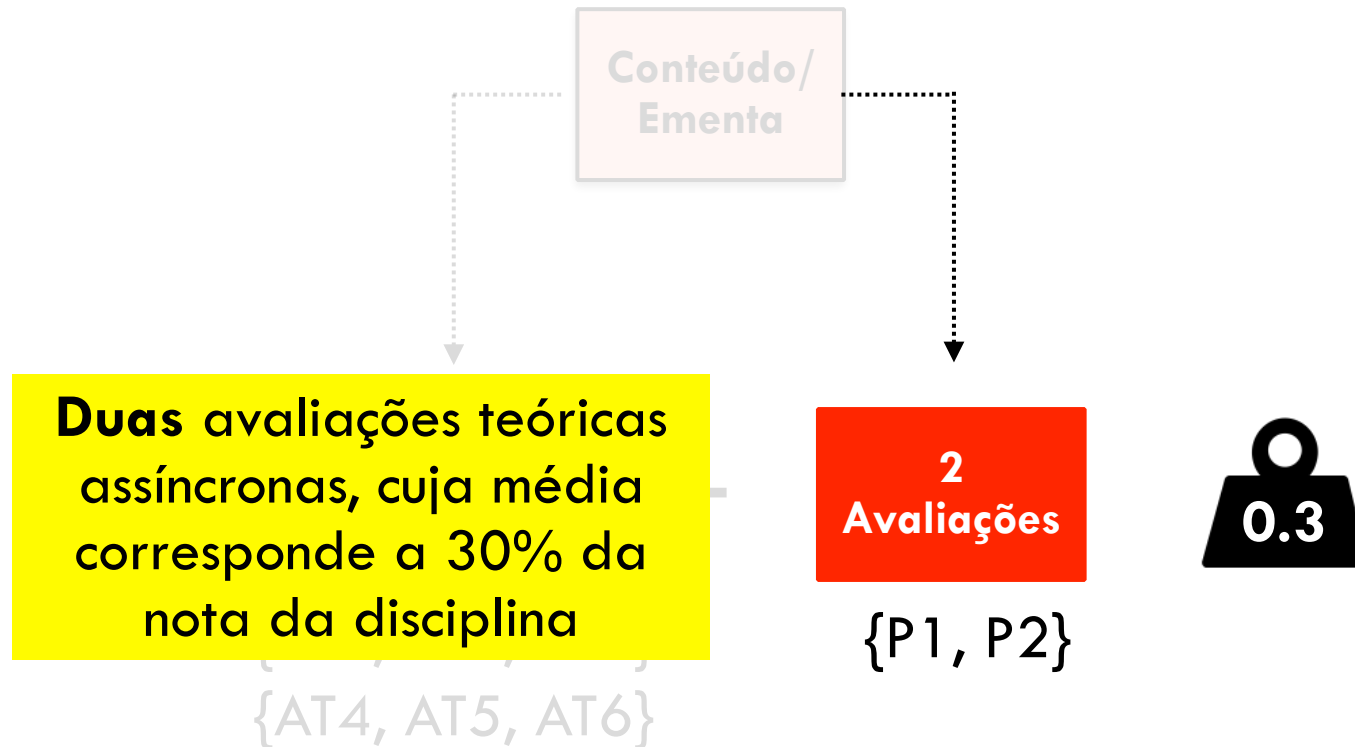
Roteiro

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

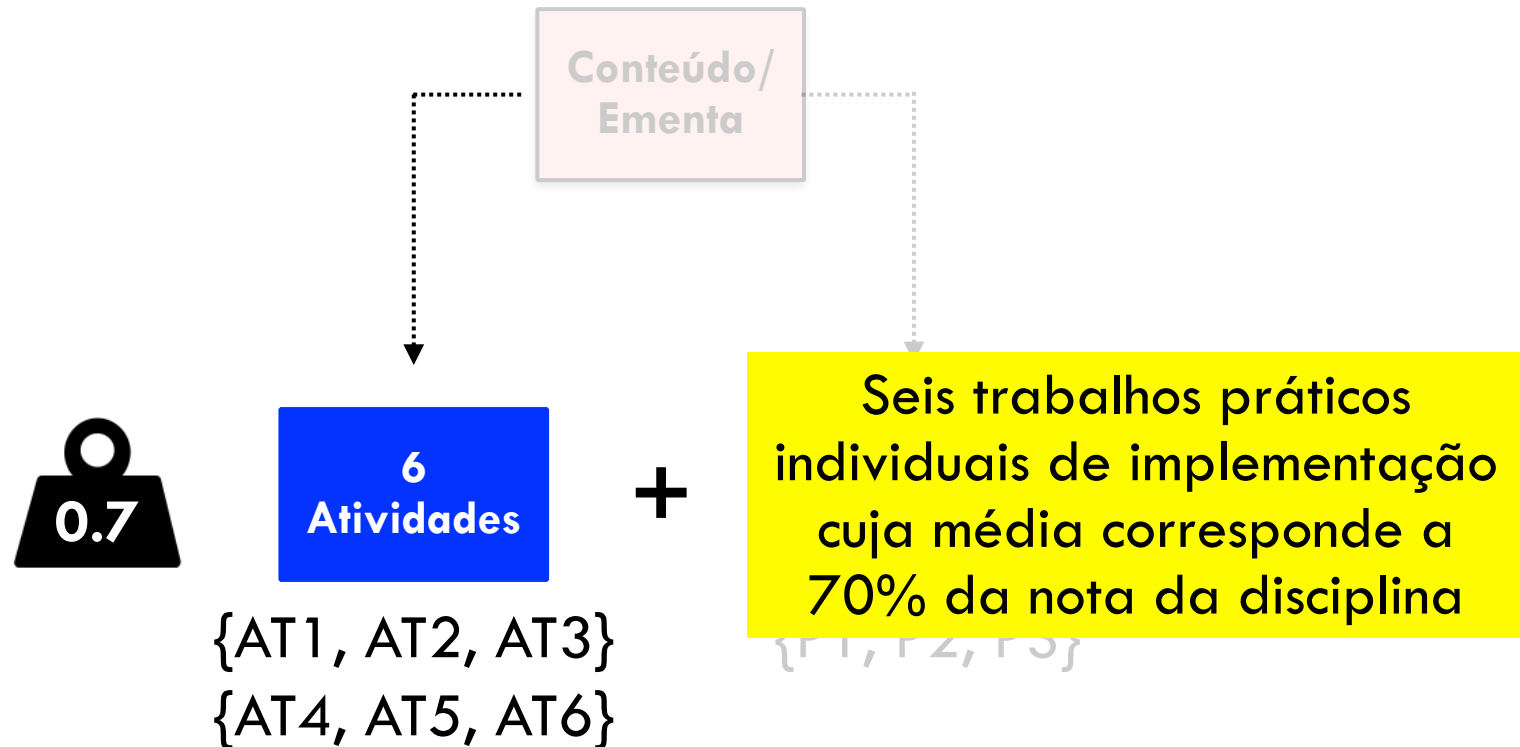
Avaliações



Avaliações



Avaliações



Avaliações

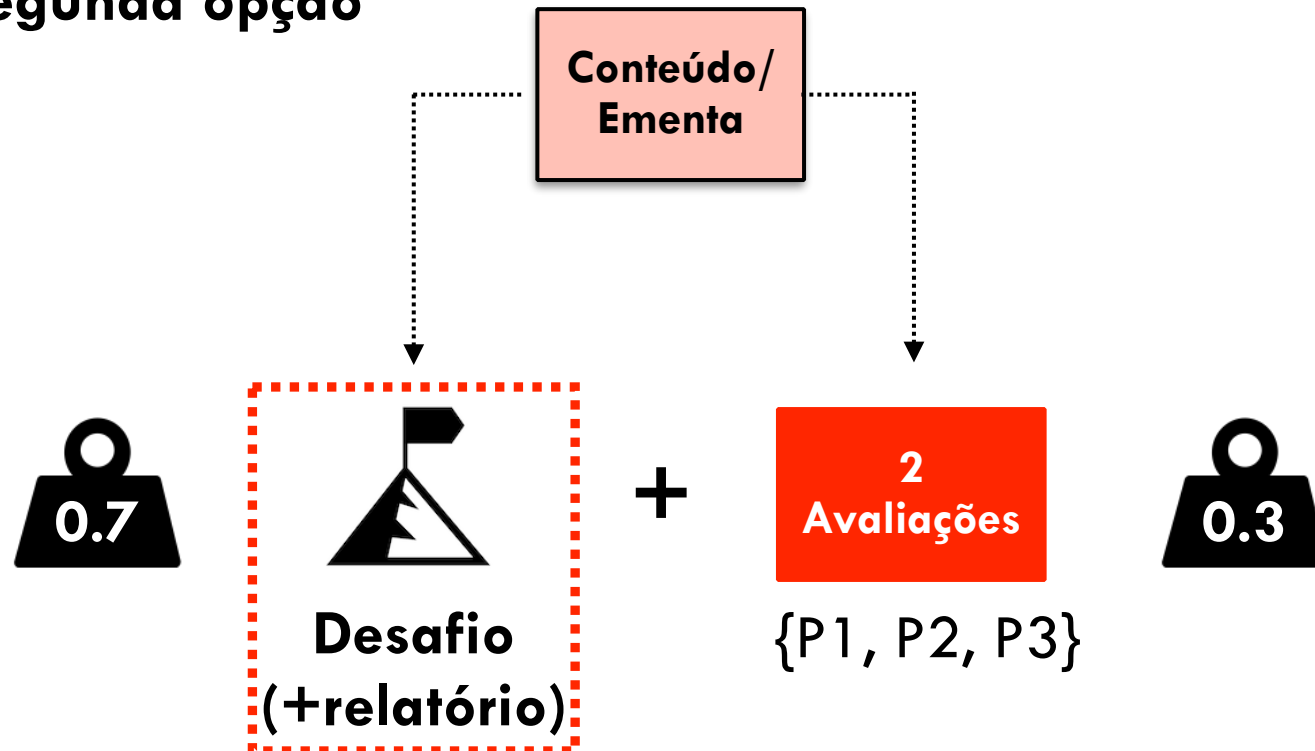


Mas ...



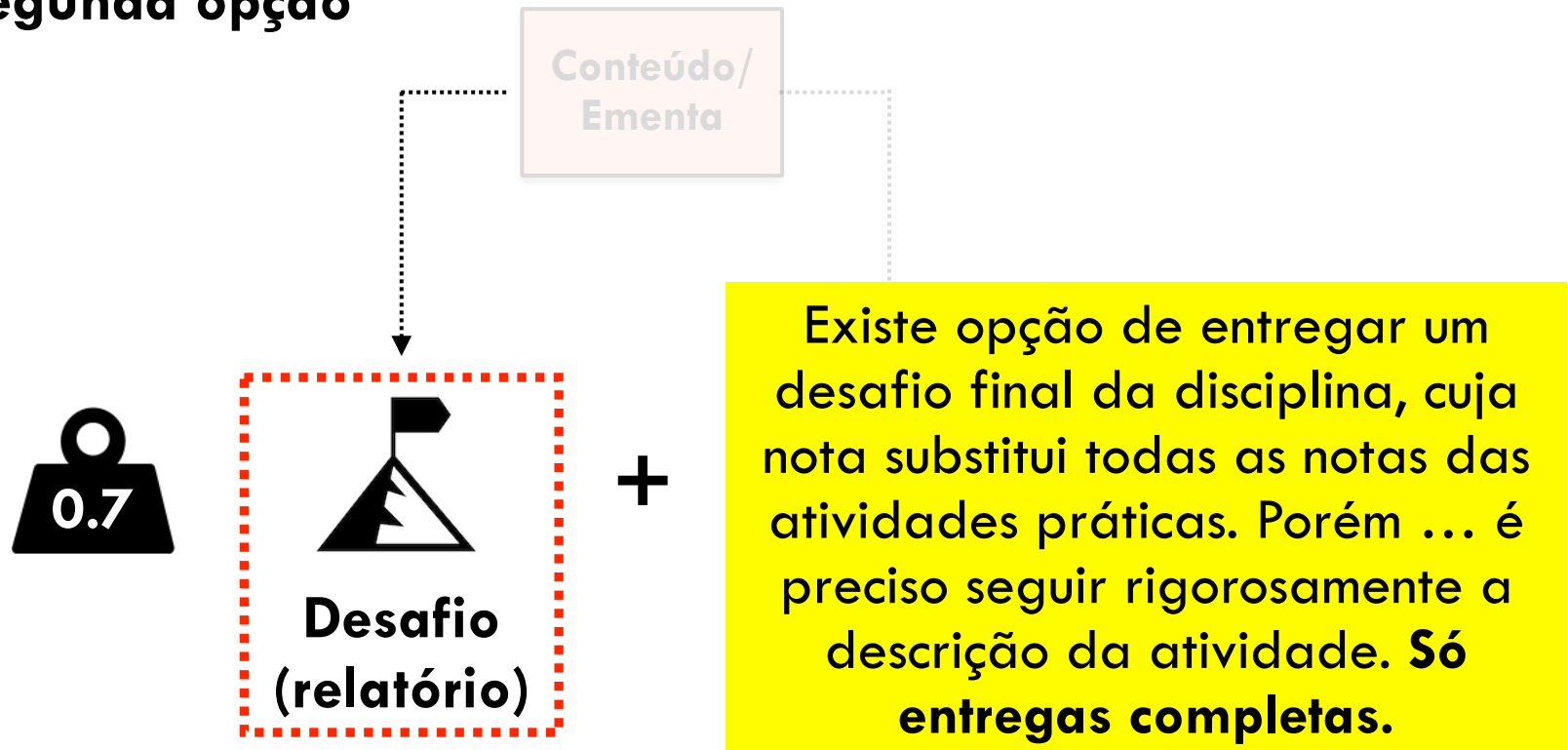
Avaliações

- Segunda opção



Avaliações

- Segunda opção



Provas (turma A)

- Planejamento (**turma A**)

- **P1**: 21/07 → {Listas, Pilhas, Filas, Árvores}

- **P2**: 25/08 → {AVL, Hash, Heaps, Grafos, Ordenação}

- **Exame**: 30/08 → **Todo conteúdo da disciplina**

Provas (turma A)

P1

Estruturas Elementares
+ Árvores

10,00

dia: 21/07

P2

AVLs, Hash, Grafos
Ordenação

10,00

dia: 25/08

10,00

dia: 30/08

Provas (turma A)

P1
Estruturas Elementares
+ Árvores

10,00

dia: 21/07

P2
AVLs, Hash, Grafos
Ordenação

10,00

dia: 25/08

Exame: **TUDO!**

10,00

dia: 30/08

Provas (turma C)

- Planejamento (**turma C**)
 - **P1**: 20/07 → {Listas, Pilhas, Filas, Árvores}
 - **P2**: 24/08 → {AVL, Hash, Heaps, Grafos, Ordenação}
 - **Exame**: 31/08 → **Todo conteúdo da disciplina**

Provas (turma C)

P1

Estruturas Elementares
+ Árvores

10,00

dia: 20/07

P2

AVLs, Hash, Grafos
Ordenação

10,00

dia: 24/08

10,00

dia: 31/08

Provas (turma C)

P1
Estruturas Elementares
+ Árvores

10,00

dia: 20/07

P2
AVLs, Hash, Grafos
Ordenação

10,00

dia: 24/08

Exame: **TUDO!**

10,00

dia: 31/08

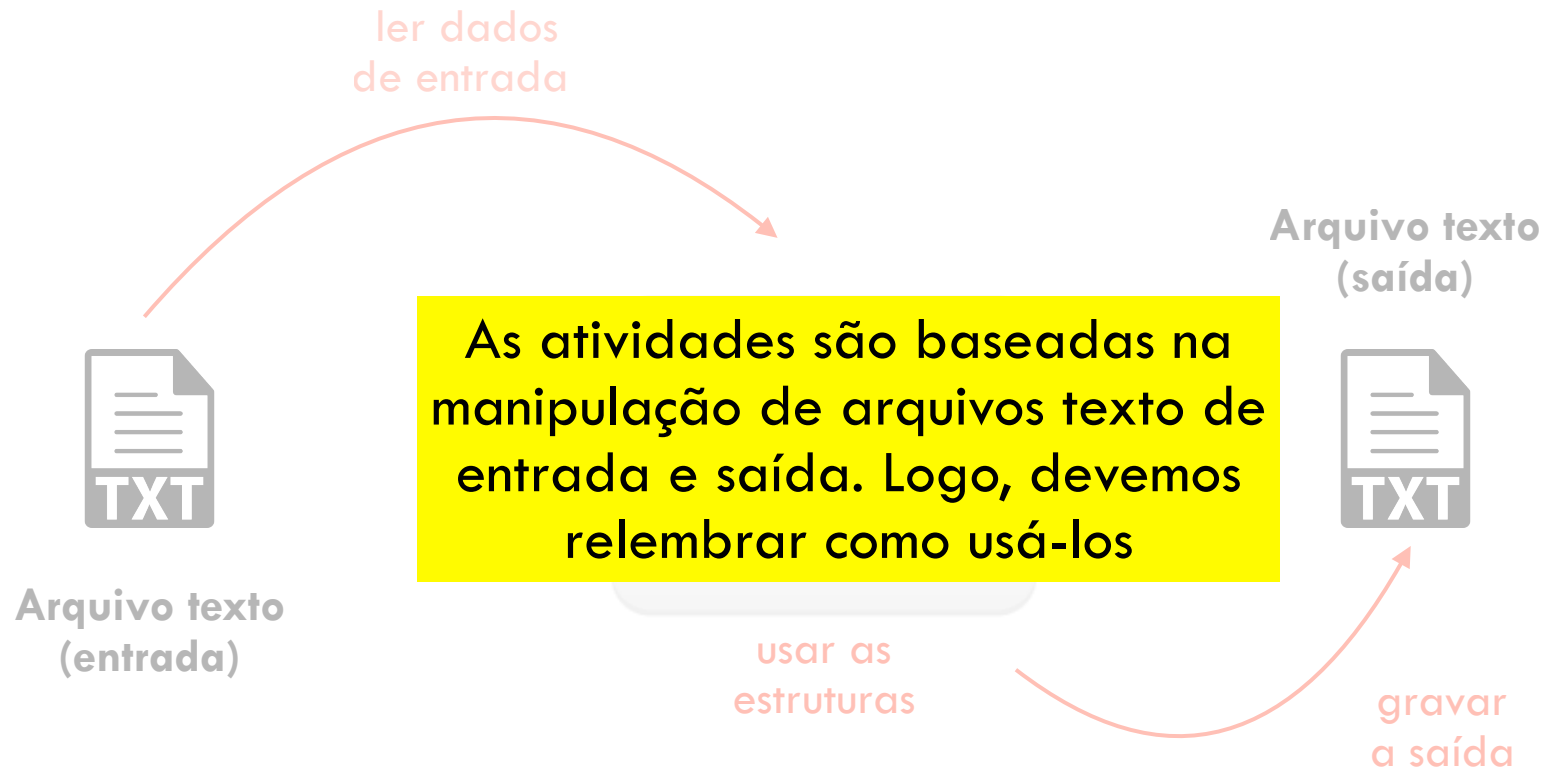
Atividades práticas

- Uso de Estruturas de dados em problemas reais
 - **AT01**: Pilhas
 - **AT02**: Listas
 - **AT03**: Árvores
 - **AT04**: AVLs
 - **AT05**: Grafos - BFS e DFS
 - **AT06**: Dijkstra - Caminhos Mínimos

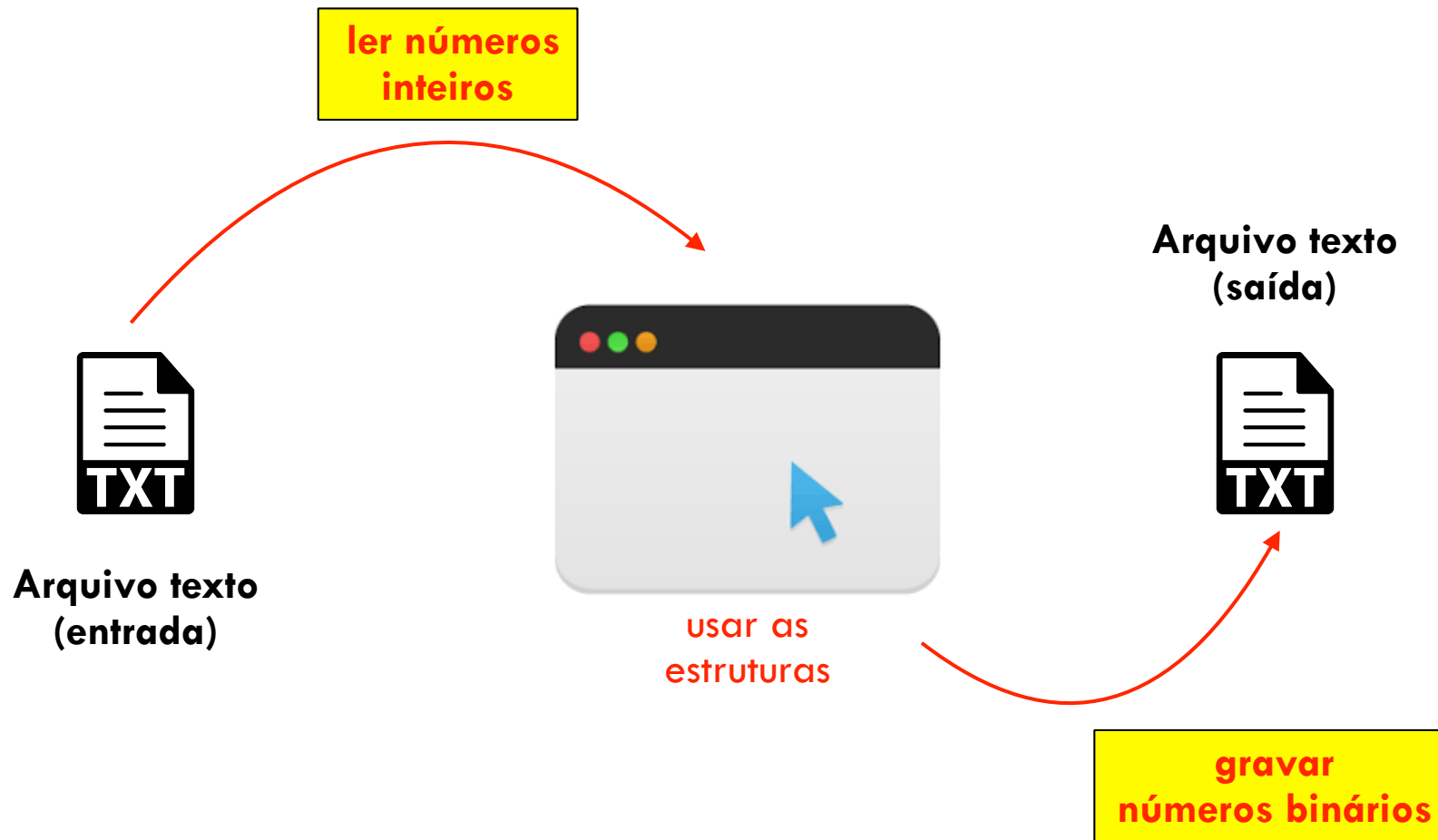
Atividades práticas



Atividades práticas



Exemplo:



Atividades práticas

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

Entregue no dia

100%

Atraso ≤ 4 dias

50%

> 4 dias

0%

Atividades práticas

- Implementação dos Algoritmos
 - C (não é permitido usar libs prontas)
 - IDE (livre escolha)

- Aplicação das EDs em domínios diferentes
 - manipulação de arquivos texto
 - entrada / saída

- Entregas:
 - **Moodle UTFPR**
 - Acompanhamento pelo GitHub

Média Final

$$0.3 * ATs + 0.7 * APs$$

(Média)

Média ≥ 6

**$3 \leq \text{Média}$
Média < 6**

Média < 3

Média Final

$$0.3 * ATs + 0.7 * APs$$

(Média)

Média ≥ 6

Aprovado :)

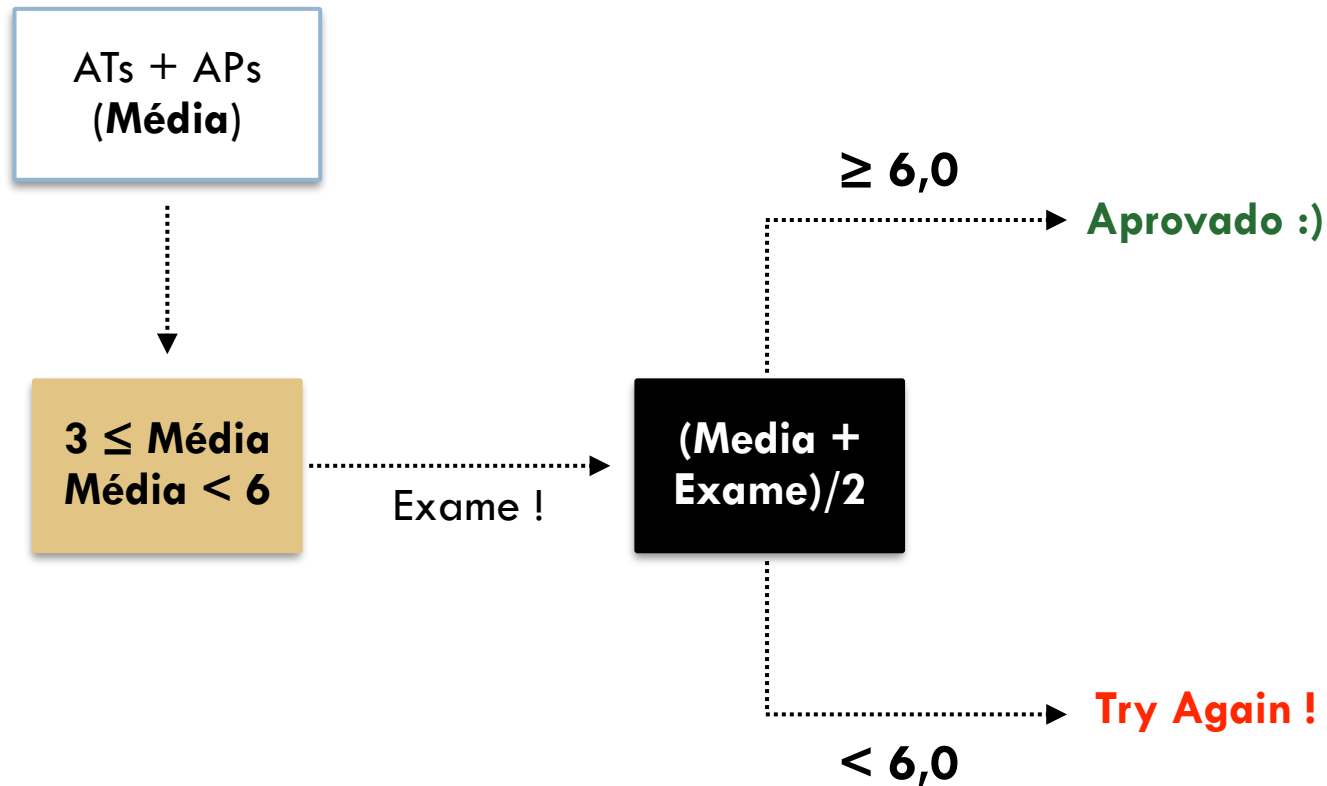
**$3 \leq \text{Média}$
Média < 6**

Exame !

Média < 3

Try Again !

Média Final && Exame

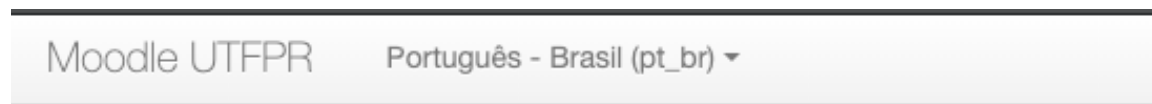


Roteiro

- 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 / 3 período / Estrutura de Dados



Estrutura de Dados

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

NAVEGAÇÃO



[Página inicial](#)

■ [Painel](#)

► [Páginas do site](#)

▼ [Meus cursos](#)

► [ED61A-EC-AP](#)

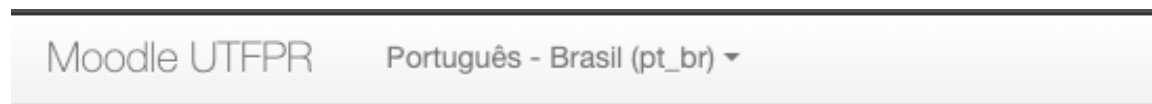


[Avisos](#)

Tópico 1

Páginas com material

- <http://moodle.utfpr.edu.br>
 - Apucarana / Graduação / Engenharia de Computação / 3 período / Estrutura de Dados



Estrutura de Dados

Senha: ed2021[α|c]

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

NAVEGAÇÃO

[Página inicial](#)

■ [Painel](#)

► [Páginas do site](#)

▼ [Meus cursos](#)

► [ED62A-EC-AP](#)

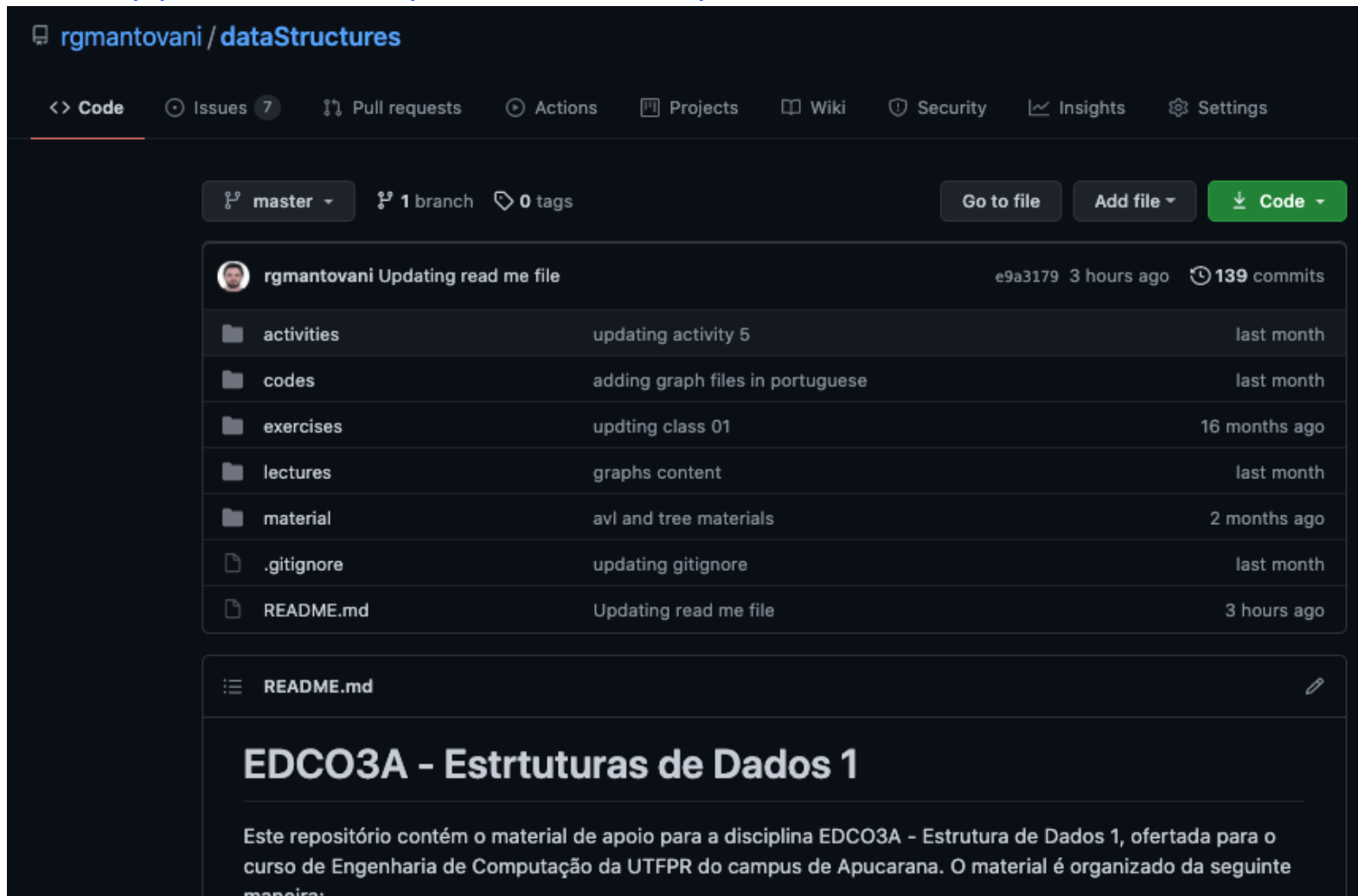


Avisos

Tópico 1

Páginas com material (espelho)

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



rgmantovani / dataStructures

<> Code Issues 7 Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

rgmantovani Updating read me file e9a3179 3 hours ago 139 commits

activities	updating activity 5	last month
codes	adding graph files in portuguese	last month
exercises	updtng class 01	16 months ago
lectures	graphs content	last month
material	avl and tree materials	2 months ago
.gitignore	updating gitignore	last month
README.md	Updating read me file	3 hours ago

README.md

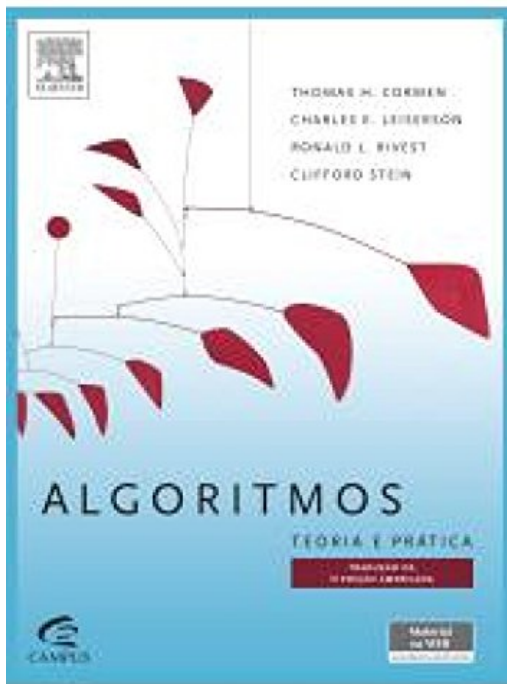
EDCO3A - Estruturas de Dados 1

Este repositório contém o material de apoio para a disciplina EDCO3A - Estrutura de Dados 1, ofertada para o curso de Engenharia de Computação da UTFPR do campus de Apucarana. O material é organizado da seguinte maneira:

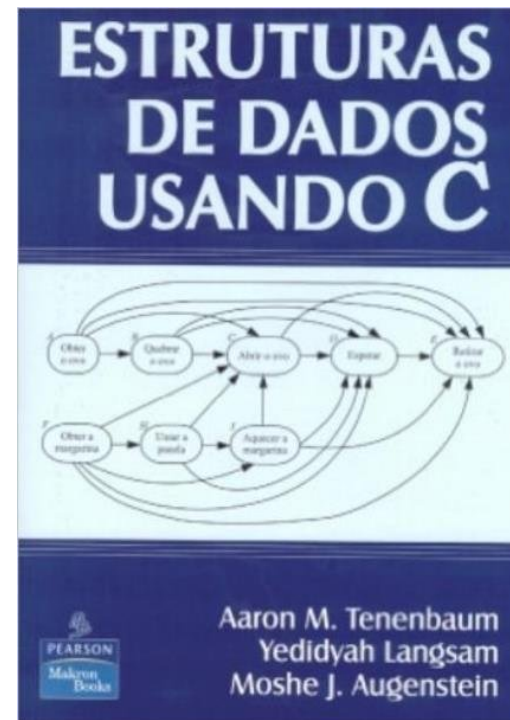
Roteiro

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

Referências sugeridas

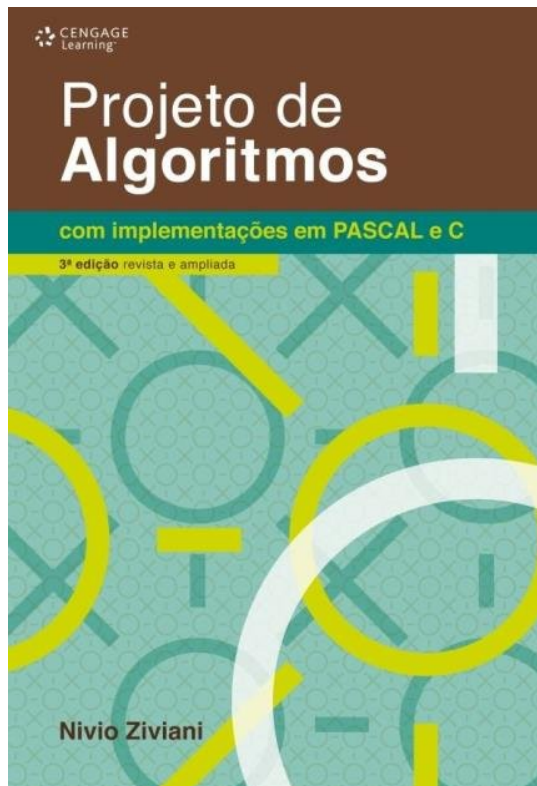


[Cormen et al, 2018]



[Tenenbaum et al, 1995]

Referências sugeridas



[Ziviani, 2010]



[Drozdek, 2017]

Informações Gerais



P-Aluno:

- Terças: 14:30 - 16:30
- Quartas: 16:40 - 18:20

Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br



If Programming Was An Anime

1.257.294 visualizações • 24 de fev. de 2020



115 MIL



1,5 MIL



COMPARTILHAR

Tarefas

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

Tarefas

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

Tarefas

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