

ED62A-COM2A

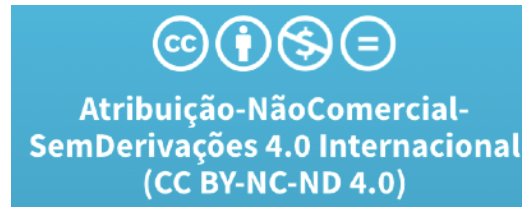
ESTRUTURAS DE DADOS

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

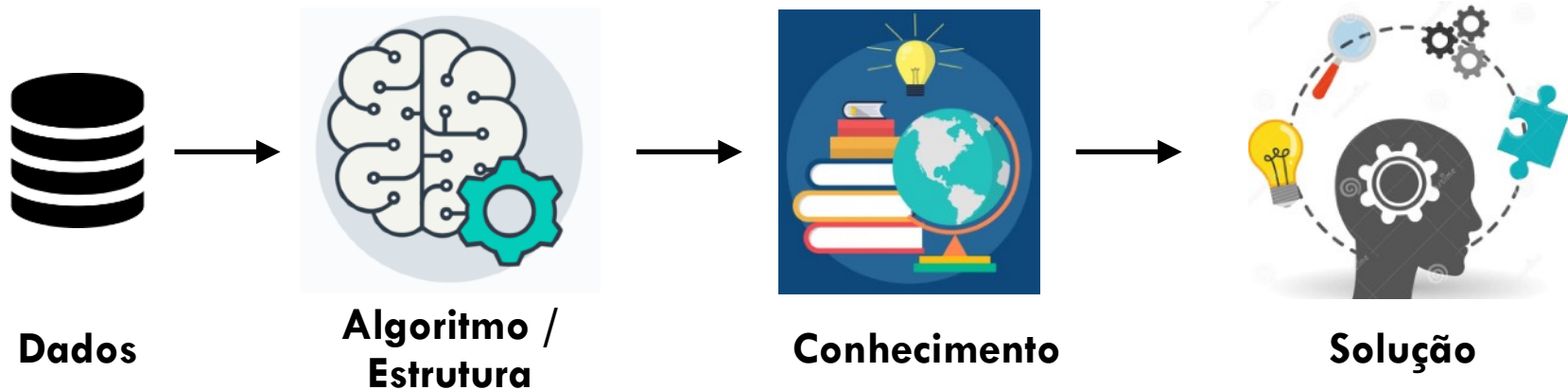


**Algoritmo /
Estrutura**



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?

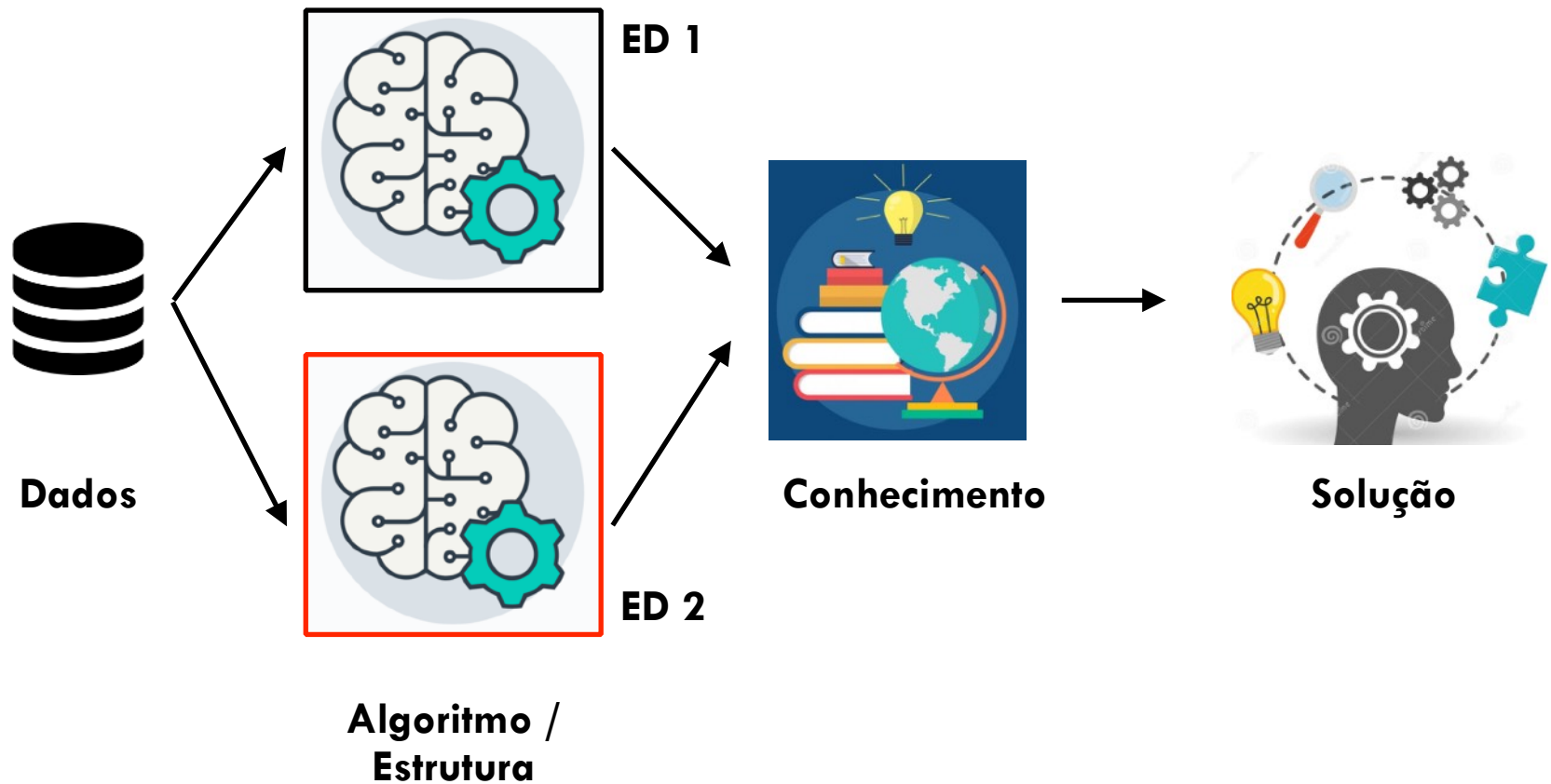


Mas ...

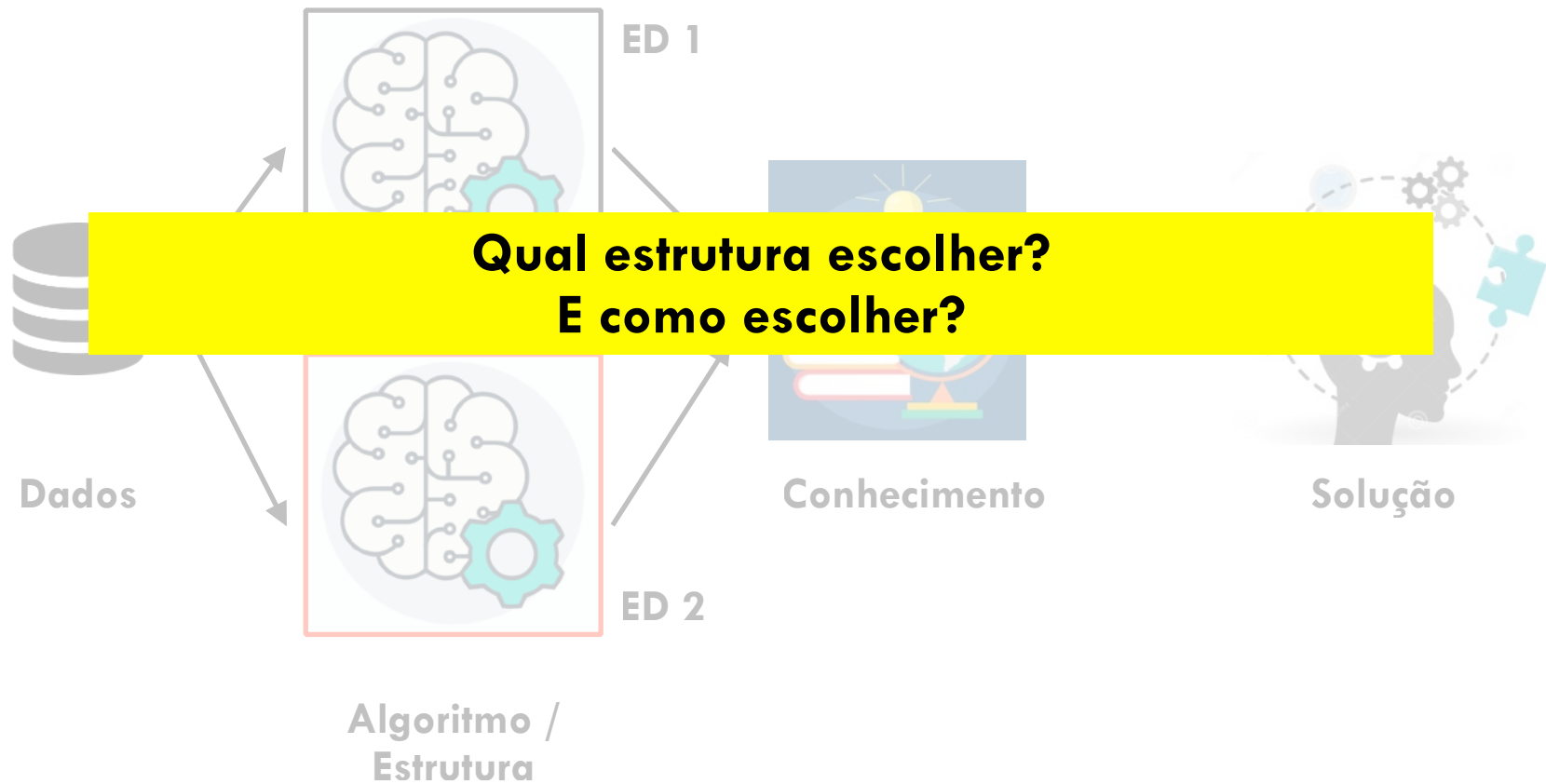


Porque usar EDs?

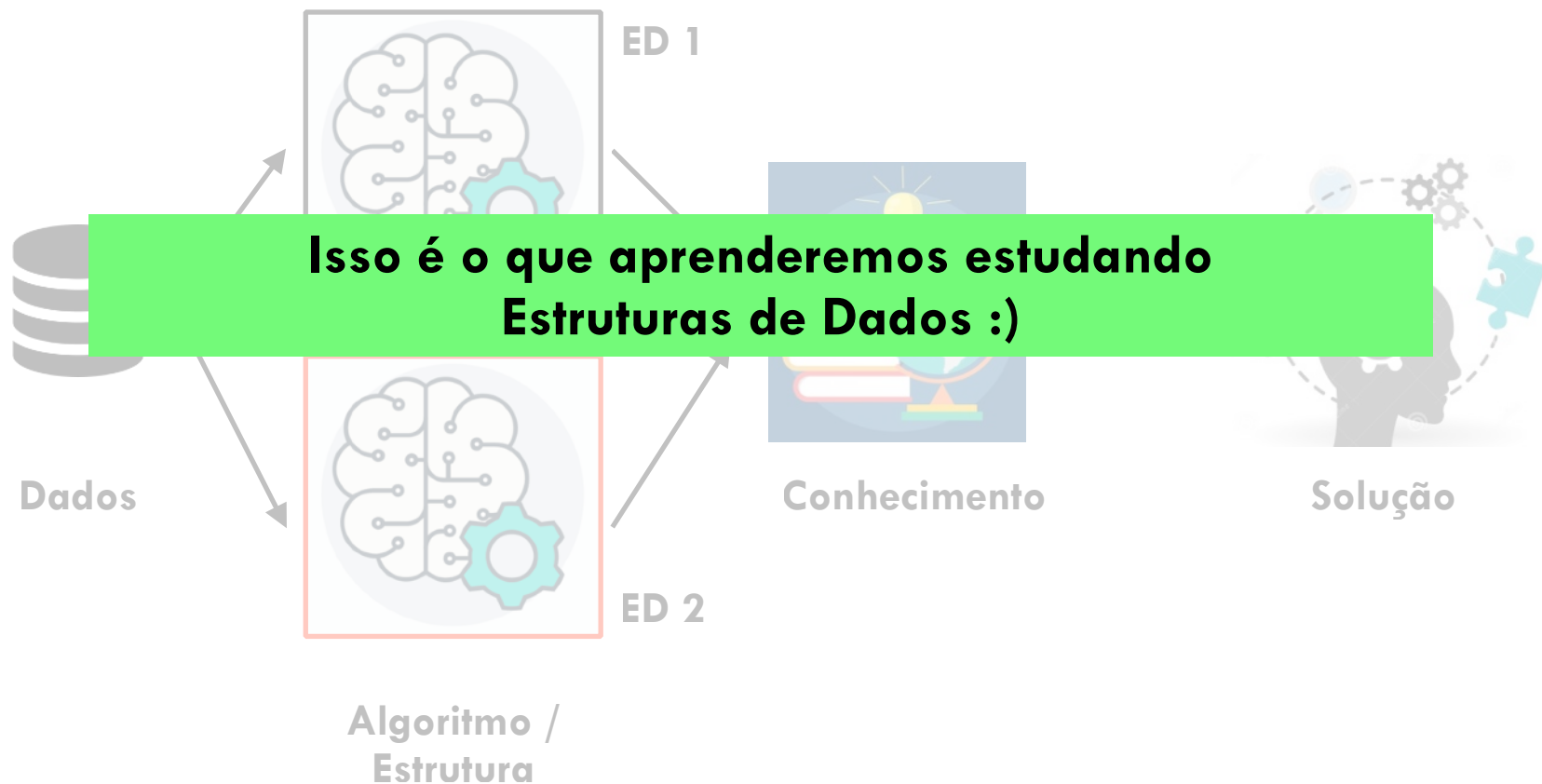
Existem diferentes formas de se resolver o mesmo problema !



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. Heaps
8. Hash
9. Grafos

Ementa

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



**Nível de
dificuldade
(implementação)**



Roteiro

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

Cronograma



Março

TAD, Listas,
Filas, Pilhas

Abril

Árvores, AVL,
Heap, Hash

Maio

Grafos,
Algoritmos de Grafos

Cronograma



Março

TAD, Listas,
Filas, Pilhas



Abril

Árvores, AVL,
Heap, Hash



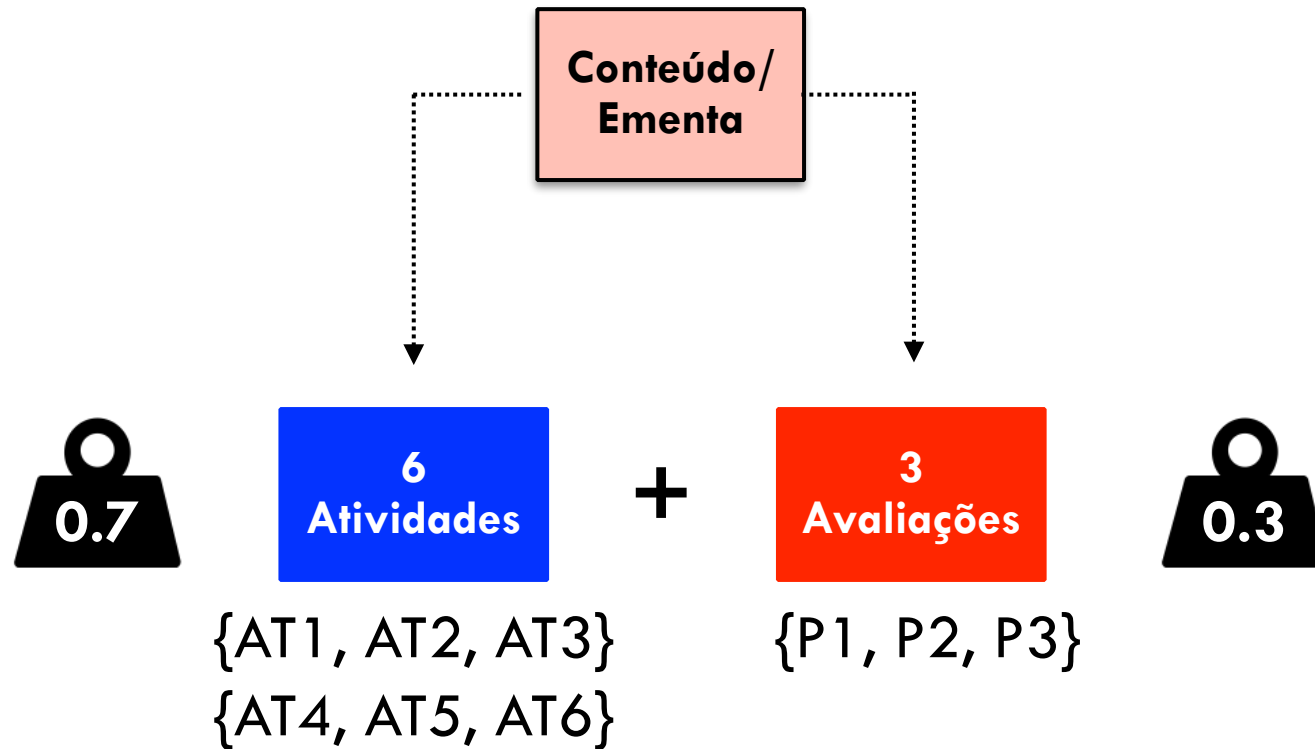
Maio

Grafos,
Algoritmos de Grafos

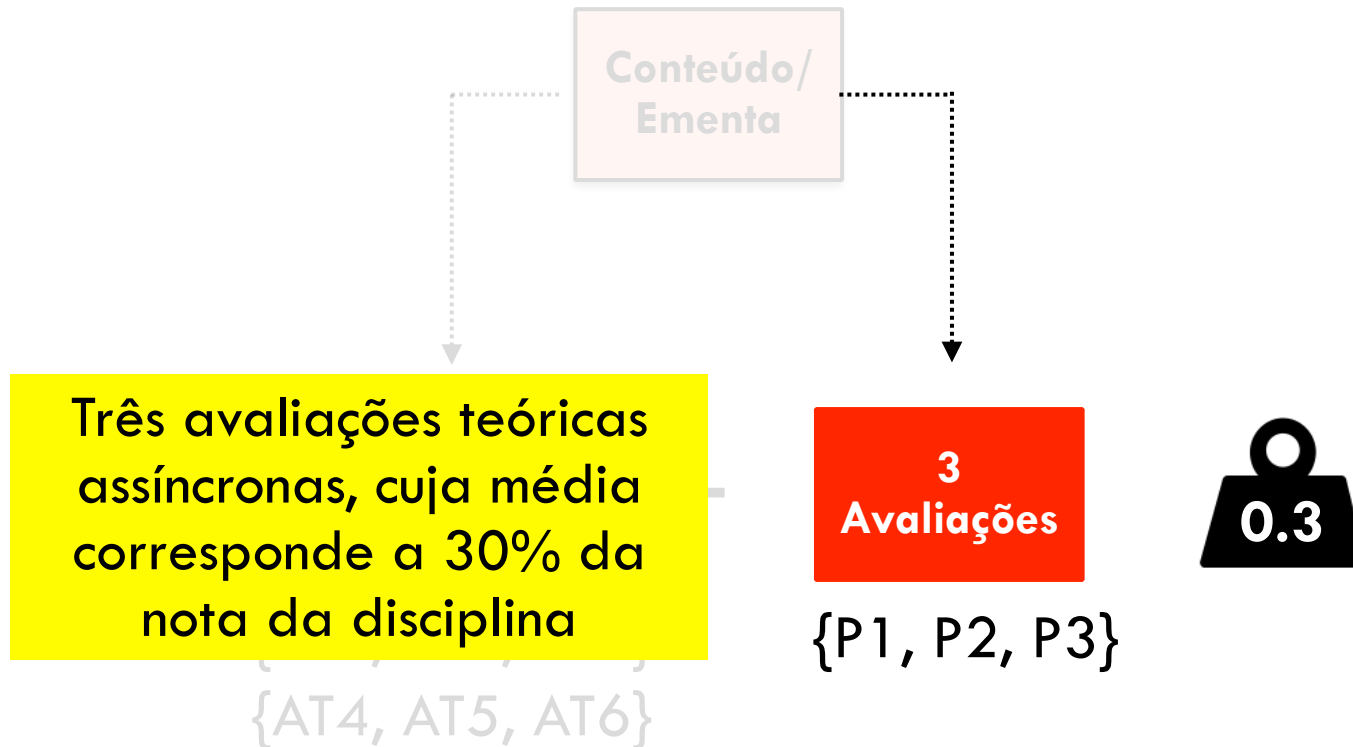
Roteiro

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

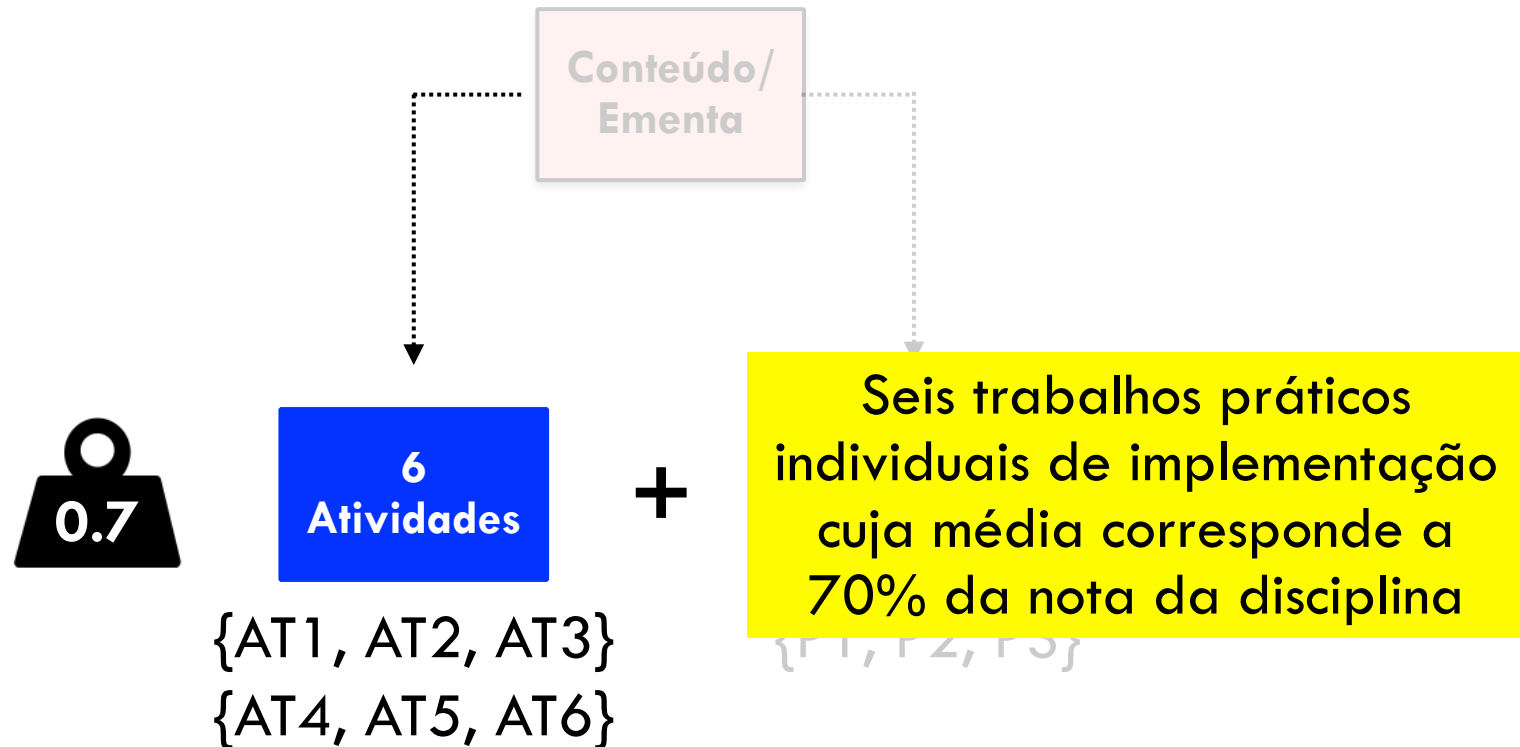
Avaliações



Avaliações



Avaliações



Avaliações

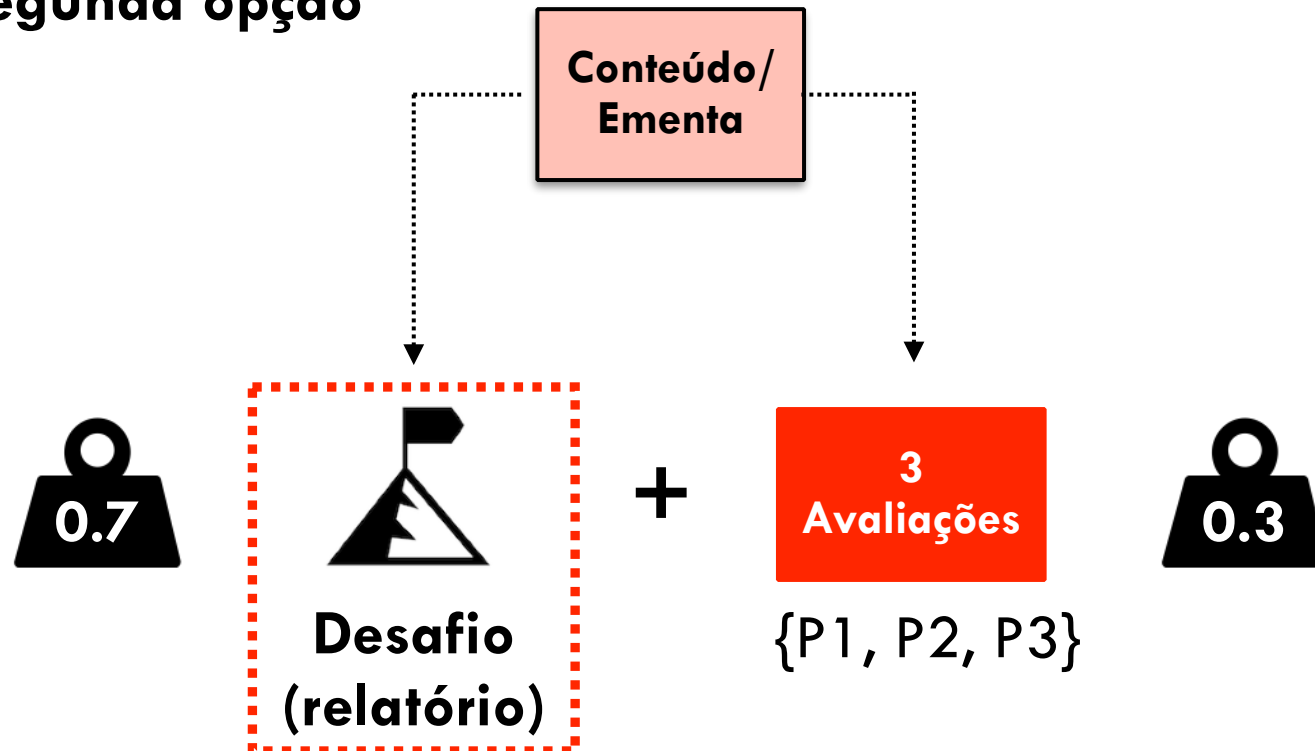


Mas ...



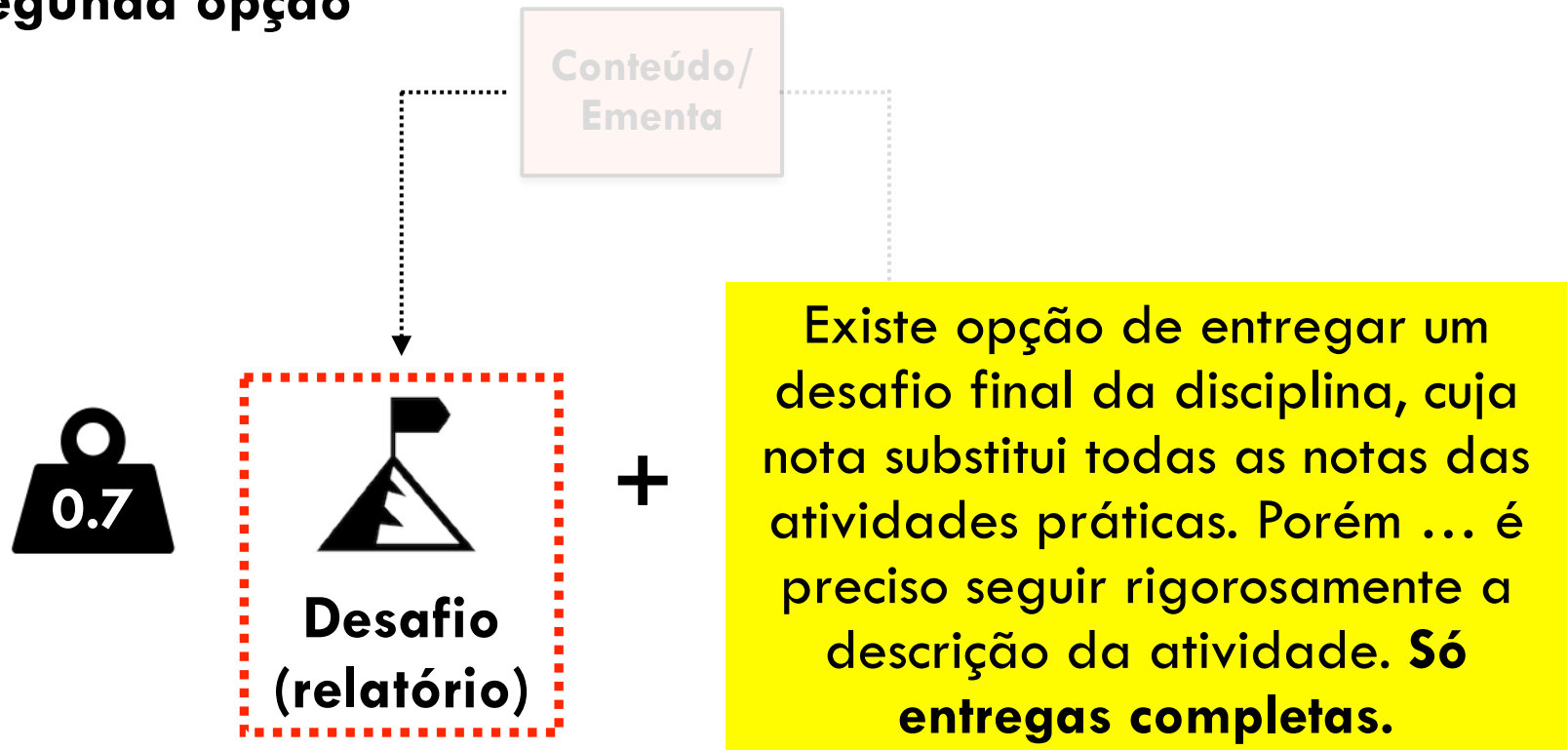
Avaliações

- Segunda opção



Avaliações

- Segunda opção



Provas

- Planejamento:
 - **P1**: 23/03 → {Listas, Pilhas, Filas}
 - **P2**: 23/04 → {Árvores, AVL, Hash, Heaps}
 - **P3**: 14/05 → {Grafos, Algoritmos de Grafos}
 - **Exame**: 18/05 → **Todo conteúdo da disciplina**

Provas

P1

Estruturas Elementares

10,00

sexta: 23/03

P2

Árvores, Hash, Heap

10,00

terça: 23/04

P3

Grafos

10,00

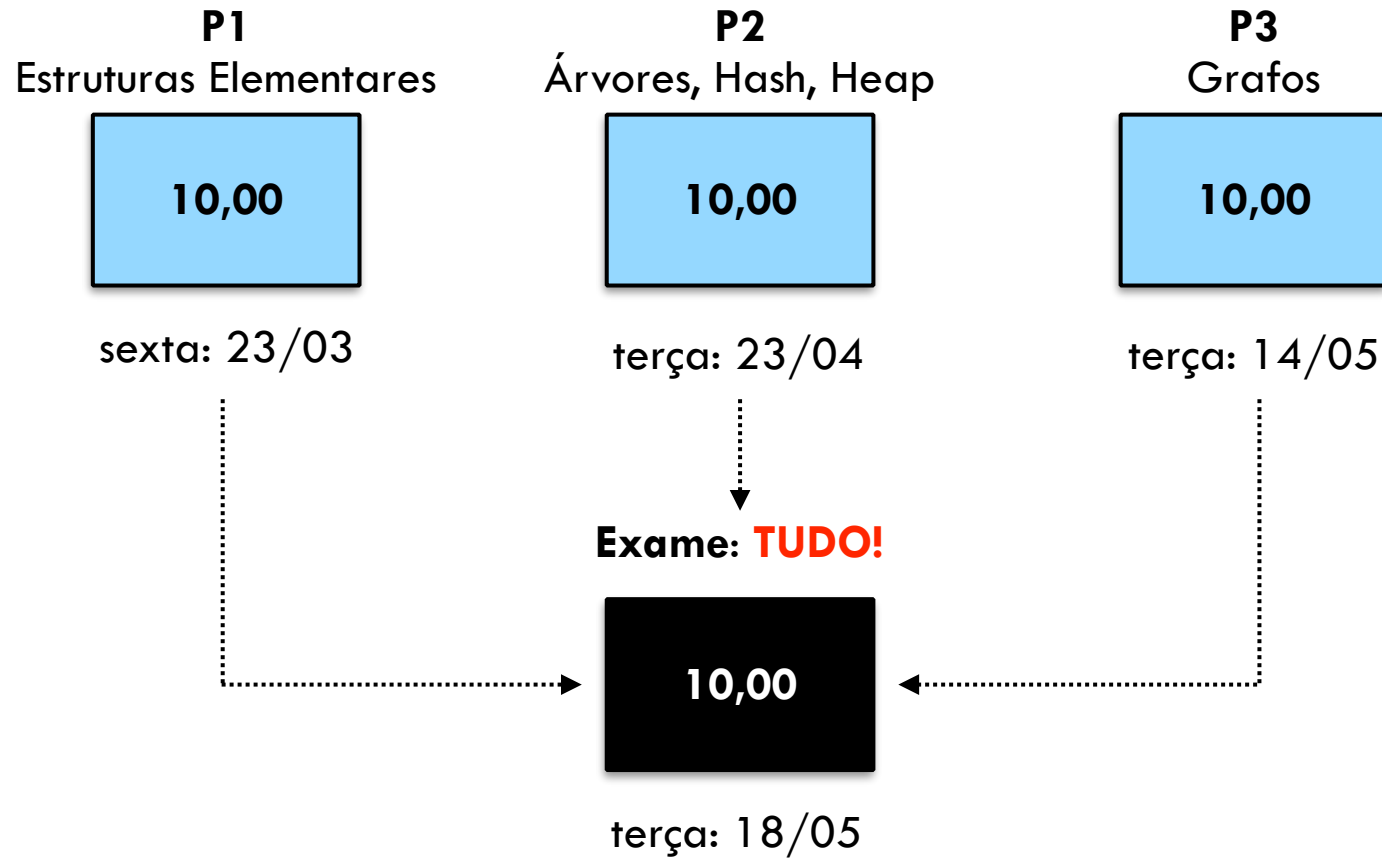
terça: 14/05

Exame: TUDO!

10,00

terça: 18/05

Provas



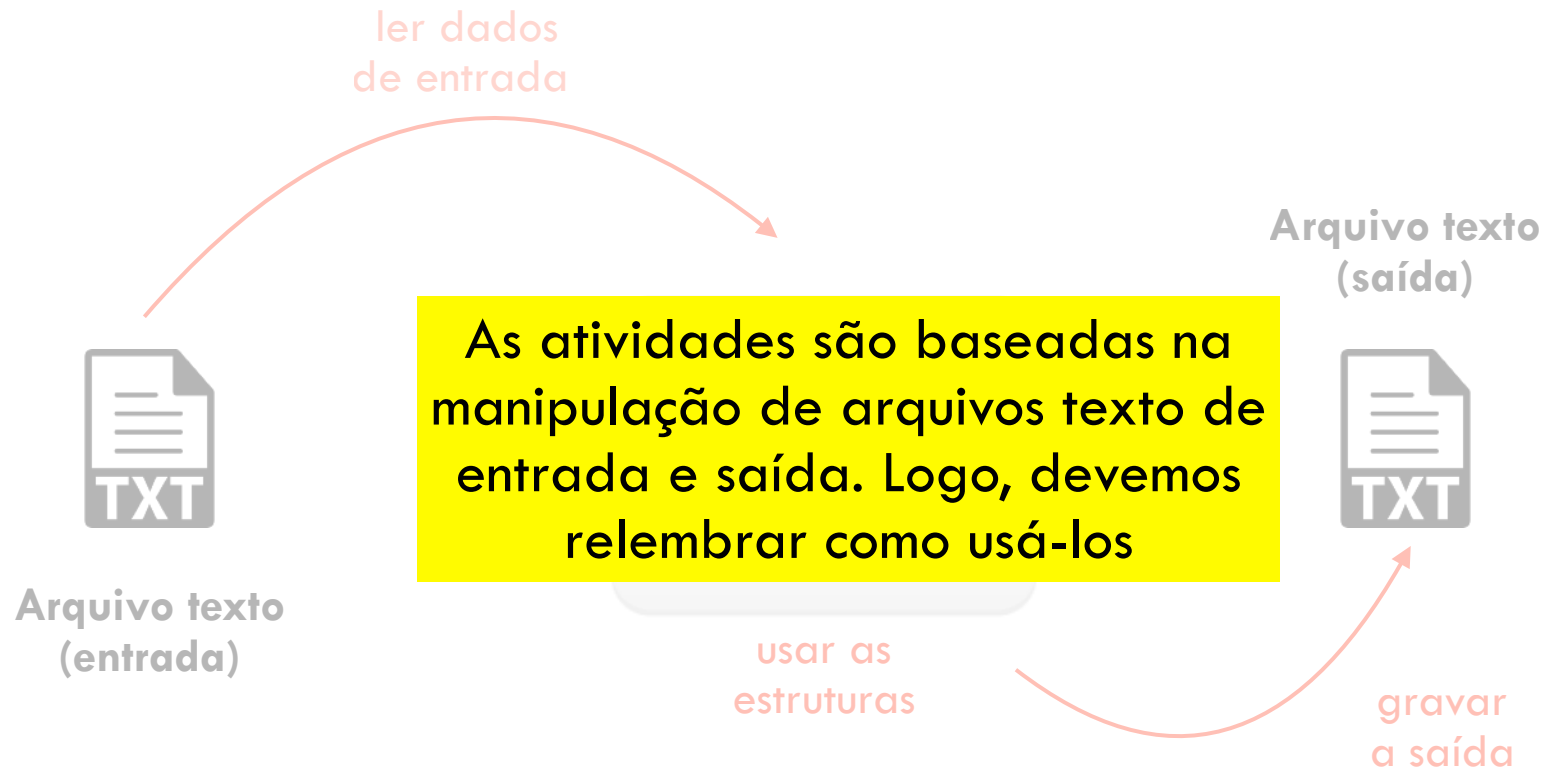
Atividades práticas

- Uso de Estruturas de dados em problemas reais
 - **AT01**: Pilhas
 - **AT02**: Listas
 - **AT03**: Árvores
 - **AT04**: AVLs
 - **AT05**: Grafos
 - **AT06**: Algoritmos de grafos

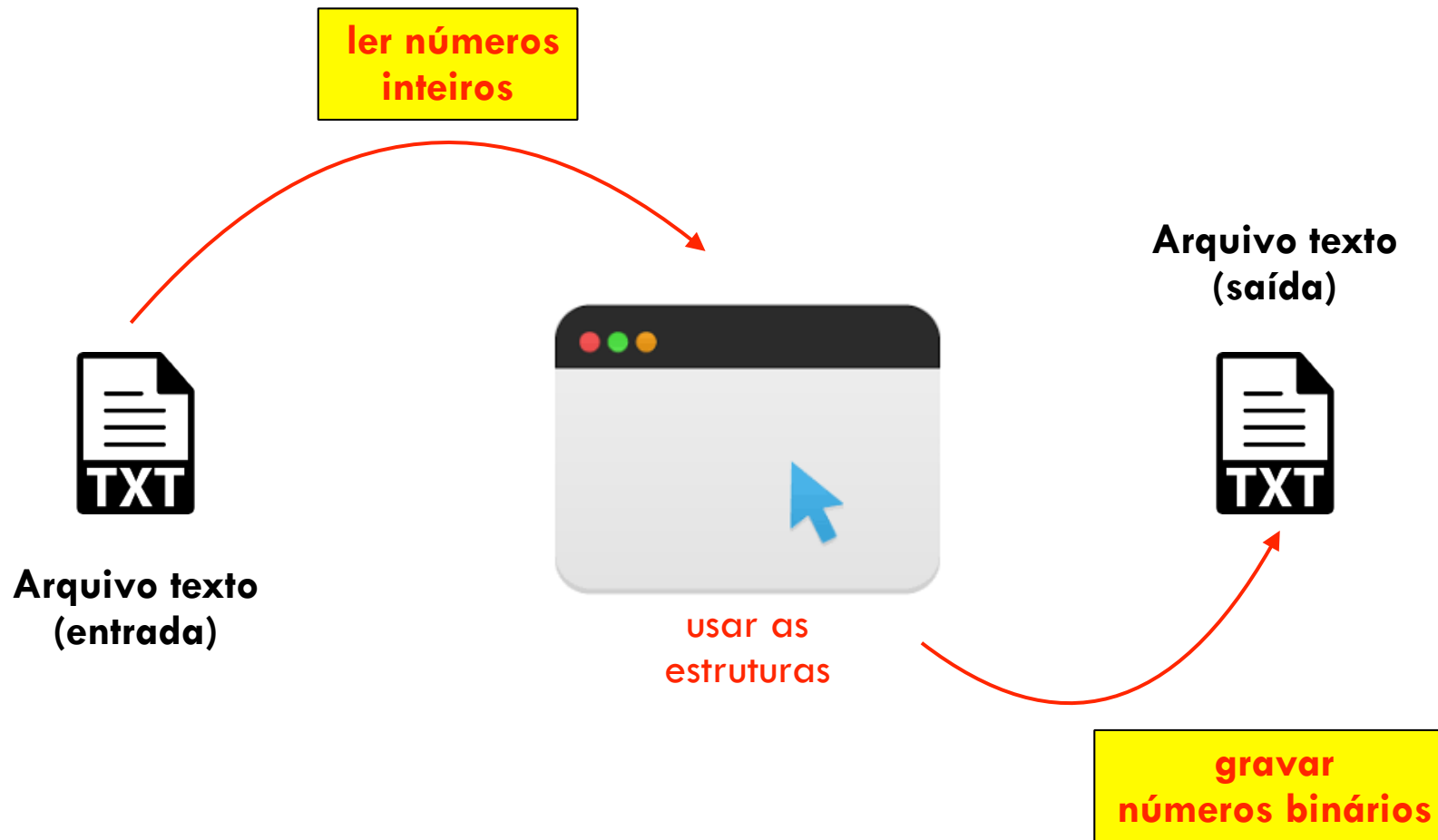
Atividades práticas



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 Educational (links)

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 :)

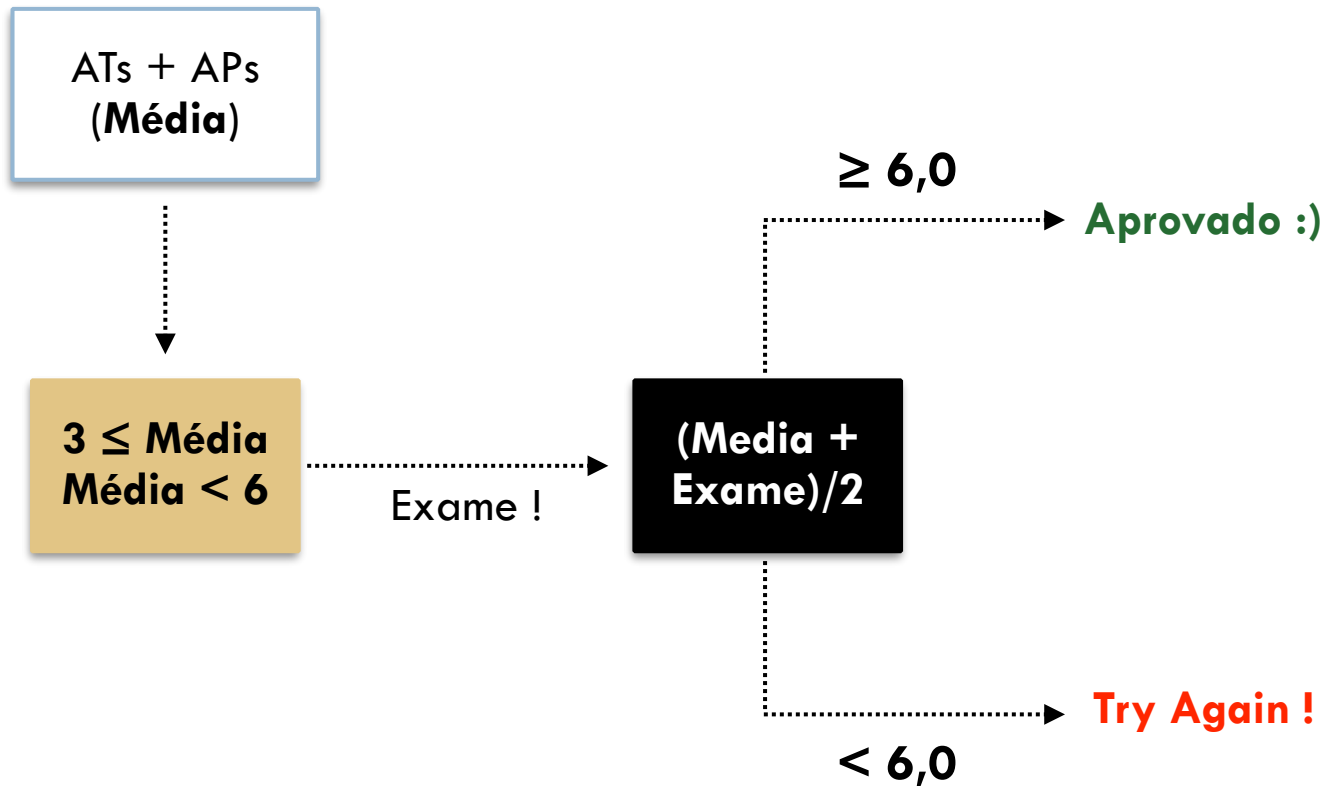
**$3 \leq \text{Média}$
Média < 6**

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 / 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

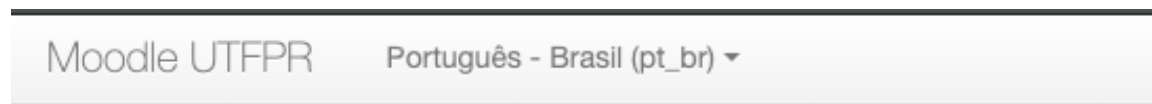
▶ ED62A-EC-AP

 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: ed2sem2020

[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.

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)

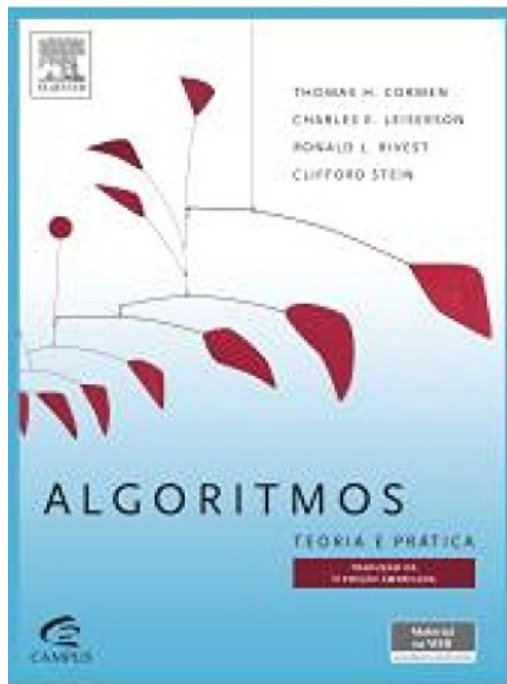
- <https://classroom.github.com/classrooms/37255181-utfprap-ed62a>

The screenshot shows the GitHub Classroom interface. At the top, there's a dark header with "GitHub Classroom" on the left and "GitHub Education" on the right. Below the header is a green banner with the course name "UTFPRAP-ED62A" and "UTFPRAP" underneath it. The main content area is titled "Assignments". Below this, there's a section for an assignment titled "T1: manipulação de estruturas básicas" with a sub-label "Individual assignment". To the right of the assignment title, there's a button that says "Copy invitation link" with a URL "https://classroom.github.c" partially visible.

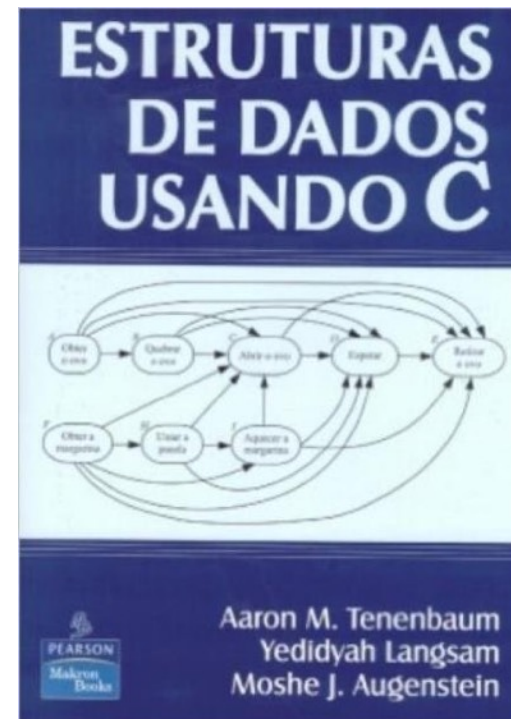
Roteiro

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

Referências sugeridas

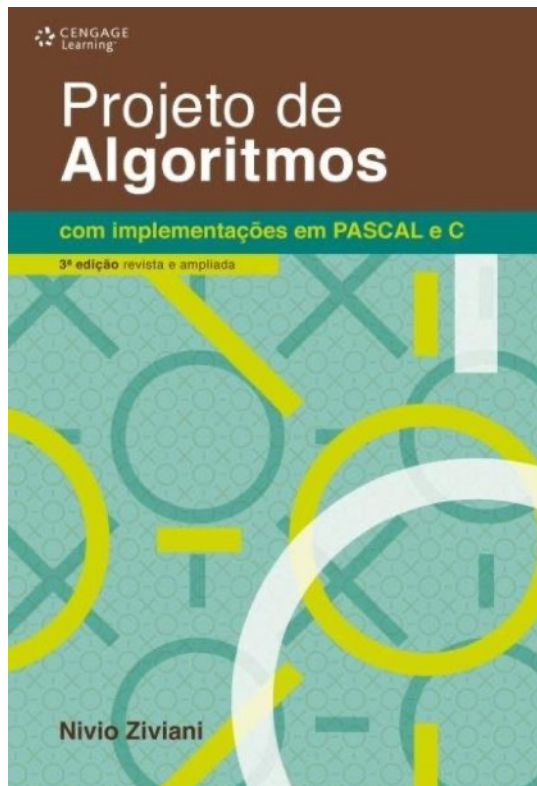


[Cormen et al, 2018]



[Tenenbaum et al, 1995]

Referências sugeridas



[Ziviani, 2010]



[Drozdek, 2017]

Informações Gerais



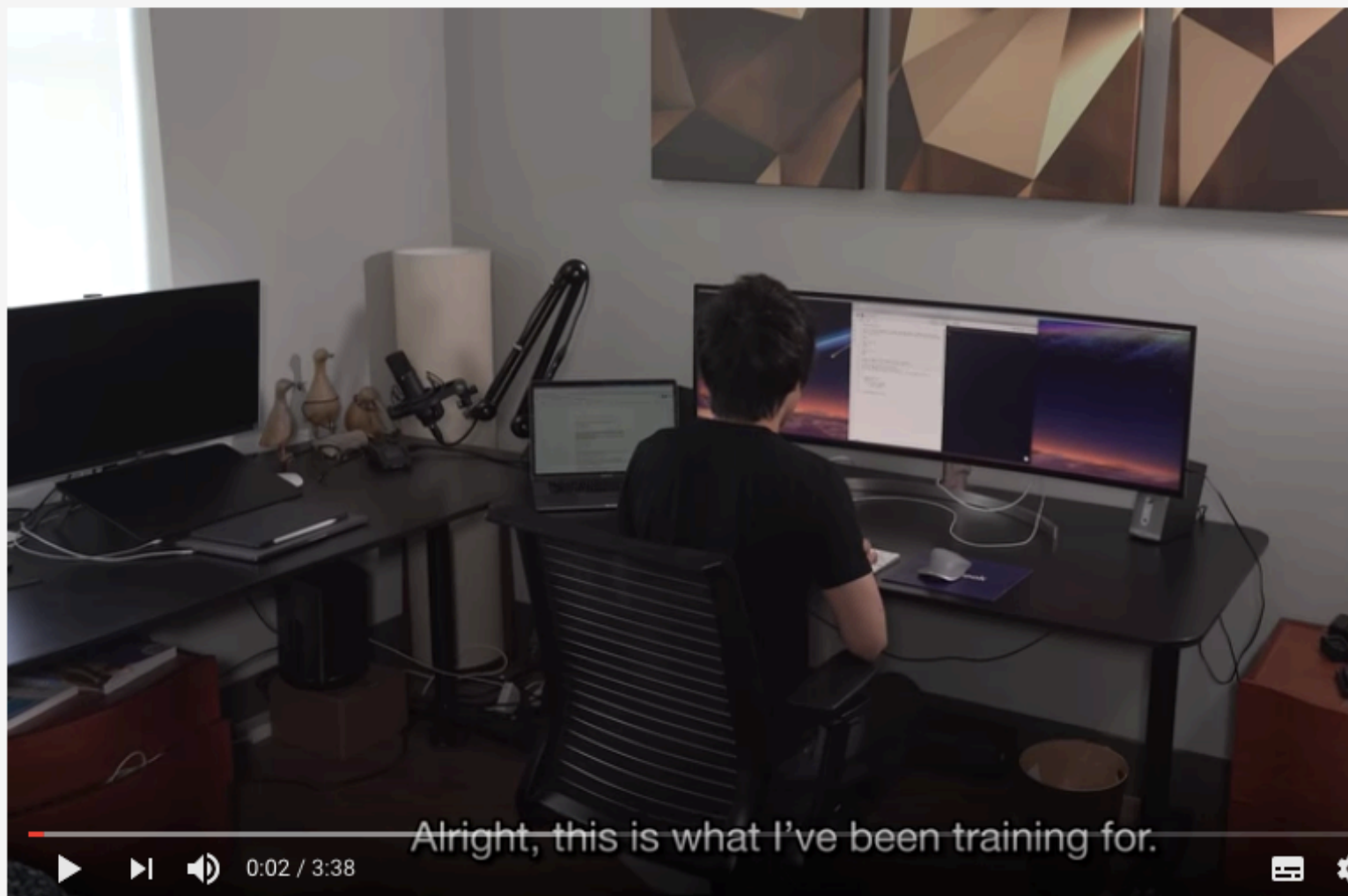
P-Aluno:

- Terças: 16:00 - 17:30
- Quartas: 14:40 - 16:30

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)**

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