



# Introdução ao Processamento de Dados Turma (2025.1)



## Estruturas de Decisão

**Tassio Sirqueira (IME/UERJ)**

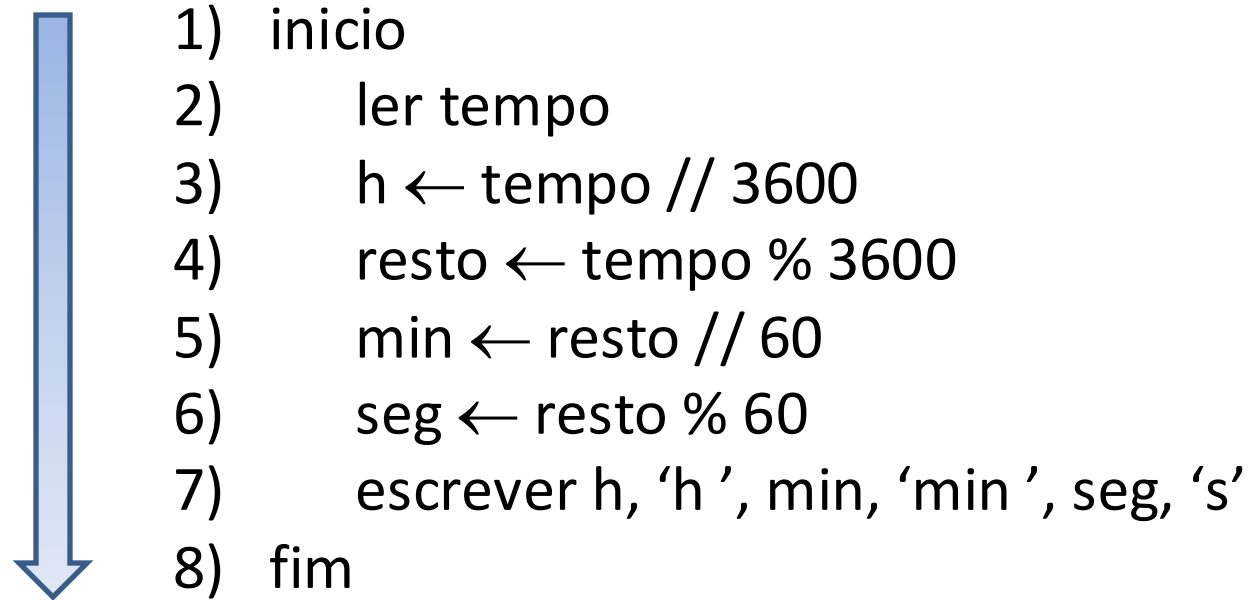
tassio.sirqueira@ime.uerj.br

# Decisão e Repetição

- Estruturas fundamentais para a programação.
- Praticamente todo o programa contém tais estruturas.
- Elas interferem no **fluxo de programação**: ordem em que as instruções (linhas) de programa são executadas.
- **Decisão**: executa um conjunto de instruções *se (if)* uma determinada condição é atendida.
- **Repetição**: executa repetidas vezes um conjunto de instruções *enquanto (while)* uma condição é atendida, ou por um número fixo de **iterações**.

# Fluxo de Processamento

Ordem em que as instruções (linhas) do algoritmo/programa são executados.

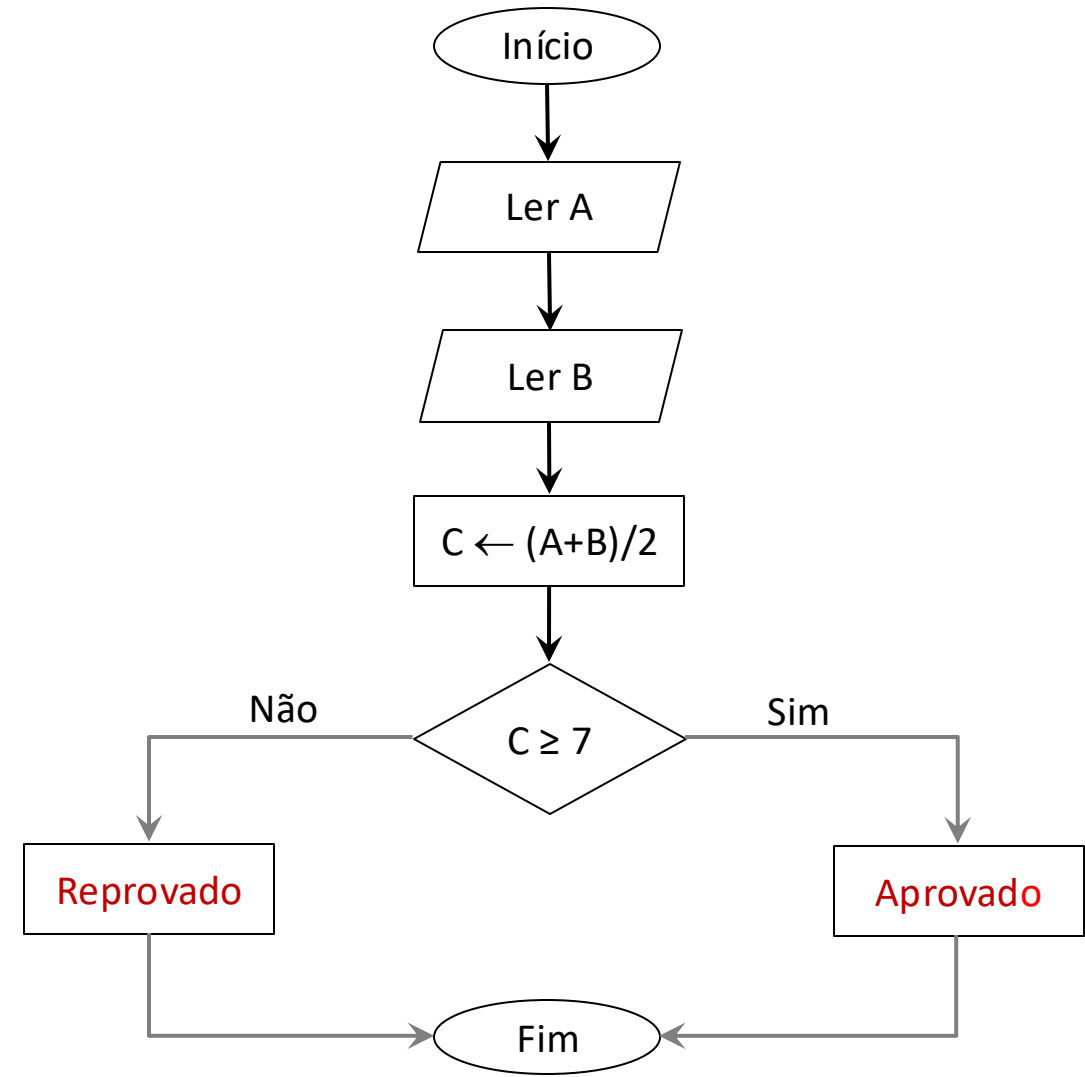


# Fluxo de Processamento

- Até agora vimos algoritmos que seguem um fluxo de processamento **sequencial**.
- Uma instrução é sempre executada depois da outra.
- A execução de certas instruções pode ser **condicionada**: só é executada se passar por um teste (**decisão**).
- **Estrutura de decisão**: como representar num programa a execução condicionada de instruções.

# Estrutura de Decisão

Como representar num programa a execução condicionada de instruções.



# Estrutura de Decisão

- A execução de certas instruções pode ser **condicionada**: só são executadas se passarem por um **teste** (decisão).
- A estrutura de decisão pode ser **simples** ou **composta**.

# Estrutura de Decisão

- Estrutura de decisão **simples**: quando a condição for atendida, executa uma ou mais instruções (um bloco de instruções).
- Se a condição não é atendida, o bloco de instruções condicionado não é executado e o **programa continua a partir do fim do bloco**.

```
...  
se condição então  
    instrução 1  
    instrução 2  
    ...  
    instrução n  
fim se  
...
```

# Estrutura de Decisão

- Estrutura de decisão **simples**: quando a condição for atendida, executa uma ou mais instruções (um bloco de instruções).
- A *condição* representa uma expressão que produz **um valor lógico**, ou uma **variável lógica**.

```
...  
se (A > B) então  
    instrução 1  
    instrução 2  
    ...  
    instrução n  
fim se  
...
```

```
...  
se aprovado então  
    instrução 1  
    instrução 2  
    ...  
    instrução n  
fim se  
...
```



# Estrutura de Decisão Simples

Exemplo: algoritmo que lê duas notas e escreve a média apenas se o aluno foi aprovado (média  $\geq 7$ ).

```
algoritmo aprovado_media
inicio
    ler P1
    ler P2
    media  $\leftarrow (P1+P2)/2$ 
    se (media  $\geq 7$ ) então
        escrever 'Aprovado com média = ', media
    fim se
fim
```

# Estrutura de Decisão Simples

Sintaxe em Python:

```
...  
if condicao:  
    instrucao_1  
    ...  
    instrucao_n  
...
```

- *condicao* é uma **expressão** com valor lógico, ou uma **variável lógica** (que teve seu valor atribuído anteriormente no programa).
- Repare os dois pontos (:) depois da *condicao*.
- Se não colocar o programa vai dar erro!

# Estrutura de Decisão Simples

Sintaxe em Python:

```
...  
if condicao:  
    instrucao_1  
    ...  
    instrucao_n  
...
```

- Repare que as instruções condicionadas (bloco) estão **indentados** (recuados) em relação a linha com o comando *if*.
- É assim que o Python entende quais instruções fazem parte do bloco!

# Estrutura de Decisão Simples

Exemplo: programa que lê duas notas e escreve a média apenas se o aluno foi aprovado (média  $\geq 7$ ).

```
P1 = float(input('Entre com a nota da P1:'))
P2 = float(input('Entre com a nota da P2:'))
media = (P1+P2)/2
if (media >=7):
    print('Aprovado com média:', media)
print('Fim do programa!')
```

# Estrutura de Decisão Simples

Exemplo: outra maneira de fazer a mesma coisa.

```
P1 = float(input('Entre com a nota da P1:'))
P2 = float(input('Entre com a nota da P2:'))
media = (P1+P2)/2
aprovado = media >= 7
if aprovado:
    print('Aprovado com média:', media)
print('Fim do programa!')
```

# Estrutura de Decisão Simples

Exemplo: outra maneira de fazer a mesma coisa.

```
P1 = float(input('Entre com a nota da P1:'))
P2 = float(input('Entre com a nota da P2:'))
media = (P1+P2)/2
aprovado = media >=7
if aprovado:
    print('Aprovado com média:', media)
print('Fim do programa!')
```

Esta linha é meio difícil de ler...

# Estrutura de Decisão Simples

Exemplo: outra maneira de fazer a mesma coisa.

```
P1 = float(input('Entre com a nota da P1:'))
P2 = float(input('Entre com a nota da P2:'))
media = (P1+P2)/2
aprovado = (media >=7)
if aprovado:
    print('Aprovado com média:', media)
print('Fim do programa!')
```

Melhor assim!

# Estrutura de Decisão Simples

Outro exemplo: ler um número inteiro e dizer se ele é impar.

```
numero = int(input('Entre com um número: '))  
if numero % 2 == 1:  
    print('O número é impar!')
```

E se o número for par?



# Estrutura de Decisão Simples

Outro exemplo: ler um número inteiro e dizer se ele é impar.

```
numero = int(input('Entre com um número: '))  
if numero % 2 == 1:  
    print('O número é impar!')  
print('Se o número não for impar, então ele é par!')
```

# Estrutura de Decisão Simples

Outro exemplo: ler um número inteiro e dizer se ele é impar.

```
numero = int(input('Entre com um número: '))  
if numero % 2 == 1:  
    print('O número é impar!')  
print('Se o número não for impar, então ele é par!')
```

Isso ficou meio estranho...

Seria bom que houvesse uma maneira de executar outras instruções se a condição falha.

# Estrutura de Decisão Composta

- Estrutura de decisão **composta**: se a condição for atendida, executa um bloco de instruções.
- Se a condição **não for atendida** (*se não*): executa um outro bloco de instruções.

```
...  
se condição então  
    instruções  
se não  
    outras instruções  
fim se  
...
```

# Estrutura de Decisão Composta

Sintaxe em Python:

```
...  
if condicao:  
    instrucao_1  
    ...  
    instrucao_n  
else:  
    instrucao_n_mais_1  
    ...  
    instrucao_m  
...
```

- O bloco de instruções depois do *else* só é executado **se a condição falha**.
- Neste caso, o bloco entre o *if* e o *else* não é executado.

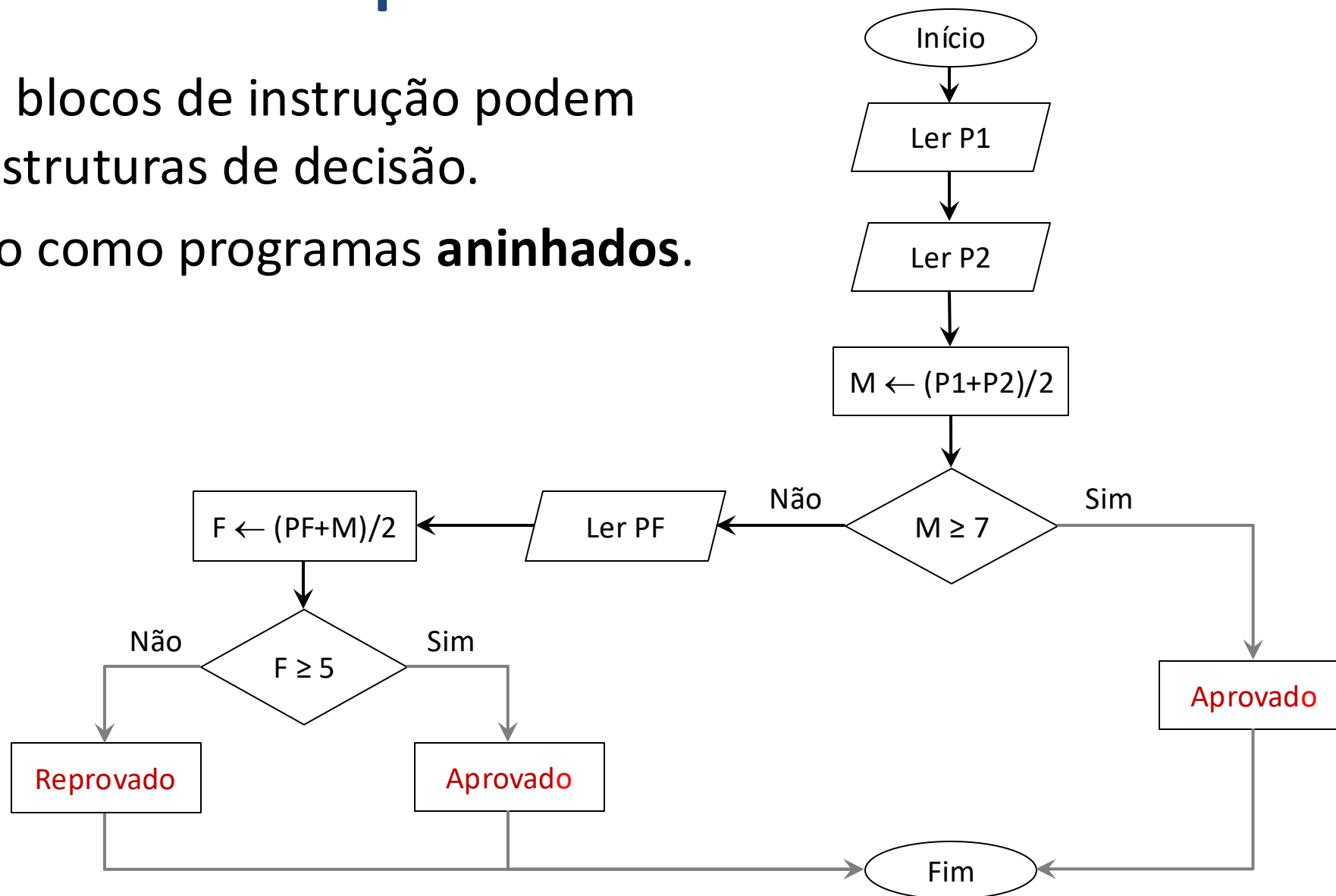
# Estrutura de Decisão Composta

Exemplo: ler um número inteiro e dizer se ele é **par ou impar**.

```
numero = int(input('Entre com um número: '))  
if numero % 2 == 1:  
    print('O número é impar!')  
else:  
    print('O número é par!')
```

# Estrutura de Decisão Composta

- Perceba que os blocos de instrução podem conter outras estruturas de decisão.
- Estes blocos são como programas **aninhados**.



# Estrutura de Decisão Composta

Exemplo:

- Programa que lê duas notas (P1 e P2) e diz se o aluno foi aprovado sem prova final.
- Se não, o programa lê a nota da prova final, e diz se o aluno foi aprovado ou não.

# Estrutura de Decisão Composta

```
P1 = float(input('Entre com a nota da P1:'))
P2 = float(input('Entre com a nota da P2:'))
media = (P1+P2)/2
if (media >=7):
    print('Aprovado com média:', media)
else:
    PF = float(input('Entre com a nota da PF:'))
    media = (media+PF)/2
    if (media >= 5):
        print('Aprovado com média:', media)
    else:
        print('Reprovado com média:', media)
```



# Estrutura de Decisão Composta

Exemplo: programa que lê dois números e diz qual dos dois é maior (se o primeiro ou o segundo número).

```
numero1 = int(input('Entre com o primeiro número: '))
numero2 = int(input('Entre com o segundo número: '))
if numero1 > numero2:
    print('O primeiro número é o maior!')
else:
    print('O segundo número é o maior!')
```

E se os dois números forem iguais?

# Estrutura de Decisão Composta

Exemplo: programa que lê dois números e diz qual dos dois é maior (se o primeiro ou o segundo número).

```
numero1 = int(input('Entre com o primeiro número: '))
numero2 = int(input('Entre com o segundo número: '))
if numero1 > numero2:
    print('O primeiro número é o maior!')
else:
    if numero1 < numero2:
        print('O segundo número é o maior!')
    else:
        print('Os dois números são iguais!')
```

# Sequência de Decisões

Se houver a necessidade de uma sequência de decisões como as do algoritmo anterior podemos utilizar a estrutura composta abaixo:

```
...  
if condicao_1:  
    bloco_de_instrucoes_1  
elif condicao_2:  
    bloco_de_instrucoes_2  
...  
elif condicao_n:  
    bloco_de_instrucoes_n  
else:  
    bloco_de_instrucoes_else  
...
```

# Sequência de Decisões

- Repare que apenas um bloco de instruções será executado.
- Se nenhuma condição é atendida, o bloco depois do *else* é executado.

```
...  
if condicao_1:  
    bloco_de_instrucoes_1  
elif condicao_2:  
    bloco_de_instrucoes_2  
...  
elif condicao_n:  
    bloco_de_instrucoes_n  
else:  
    bloco_de_instrucoes_else  
...
```

# Estrutura de Decisão Composta

Exemplo: programa que lê dois números e diz qual dos dois é maior (se o primeiro ou o segundo número).

```
numero1 = int(input('Entre com o primeiro número: '))
numero2 = int(input('Entre com o segundo número: '))
if numero1 > numero2:
    print('O primeiro número é o maior!')
else:
    if numero1 < numero2:
        print('O segundo número é o maior!')
    else:
        print('Os dois números são iguais!')
```

# Estrutura de Decisão Composta

Exemplo: programa que lê dois números e diz qual dos dois é maior (se o primeiro ou o segundo número).

```
numero1 = int(input('Entre com o primeiro número: '))
numero2 = int(input('Entre com o segundo número: '))
if numero1 > numero2:
    print('O primeiro número é o maior!')
elif numero1 < numero2:
    print('O segundo número é o maior!')
else:
    print('Os dois números são iguais!')
```

## Exercício (até a próxima aula)

- Escrever um programa que leia o peso e a altura de uma pessoa e calcule seu índice de massa corporal (IMC) de acordo com a fórmula:

$$\text{IMC} = \text{peso (em quilos)} \div \text{altura}^2 \text{ (em metros)}$$

- O programa deve classificar uma pessoa de acordo com o IMC, nas seguintes faixas:

Menor que 18,5 - Abaixo do peso

Entre 18,5 e 25,0 - Peso normal

Entre 25,0 e 30,0 - Pré-obesidade

Entre 30,0 e 35,0 - Obesidade Grau 1

Entre 35,0 e 40,0 - Obesidade Grau 2

Acima de 40 - Obesidade Grau 3

## Exercício (até a próxima aula)

- As faixas de IMC indicadas previamente são para pessoas (mulheres e homens) acima de 20 anos.
- Para adolescentes (entre 10 e 20 anos) as faixas são diferentes para homens e mulheres.
- Para **homens**:

Idade	Baixo Peso	Adequado	Sobrepeso
10	< 14,42	entre 14,42 e 19,60	>= 19,60
11	< 14,83	entre 14,83 e 20,35	>= 20,35
12	< 15,24	entre 15,24 e 21,12	>= 21,12
13	< 15,73	entre 15,73 e 21,93	>= 21,93
14	< 16,18	entre 16,18 e 22,77	>= 22,77
15	< 16,59	entre 16,59 e 23,63	>= 23,63
16	< 17,01	entre 17,01 e 24,45	>= 24,45
17	< 17,31	entre 17,31 e 25,28	>= 25,28
18	< 17,54	entre 17,54 e 25,95	>= 25,95
19	< 17,80	entre 17,80 e 26,36	>= 26,36



## Exercício (até a próxima aula)

- As faixas de IMC indicadas previamente são para pessoas (mulheres e homens) acima de 20 anos.
- Para adolescentes (entre 10 e 20 anos) as faixas são diferentes para homens e mulheres.
- Para **mulheres**:

Idade	Baixo Peso	Adequado	Sobrepeso
10	< 14,23	entre 14,23 e 20,19	>= 20,19
11	< 14,60	entre 14,60 e 21,18	>= 21,18
12	< 14,98	entre 14,98 e 22,17	>= 22,17
13	< 15,36	entre 15,36 e 23,08	>= 23,08
14	< 15,67	entre 15,67 e 23,88	>= 23,88
15	< 16,01	entre 16,01 e 24,29	>= 24,29
16	< 16,37	entre 16,37 e 24,74	>= 24,74
17	< 16,59	entre 16,59 e 25,23	>= 25,23
18	< 16,71	entre 16,71 e 25,56	>= 25,56
19	< 16,87	entre 16,87 e 25,85	>= 25,85

## Exercício (até a próxima aula)

Para fazer esta segunda parte (para pessoas com menos de 20 anos), vocês devem perguntar também o sexo ('M' ou 'F') e a idade da pessoa.



# Introdução ao Processamento de Dados Turma (2025.1)



## Estruturas de Decisão

**Tassio Sirqueira (IME/UERJ)**

tassio.sirqueira@ime.uerj.br