
PRG203402 - Lógica de Programação: Estruturas de decisão e repetição

João Cláudio Elsen Barcellos

Engenheiro Eletricista
Formado na Universidade Federal de Santa Catarina
campus Florianópolis
joaoclaudiobarcellos@gmail.com

6 de Maio de 2025

** Créditos ao Prof. Emerson Ribeiro de Mello, o qual criou e disponibilizou o template aqui usado, via ShareLaTeX*

*** Créditos ao Prof. Renan Augusto Starke, o qual forneceu parte do conteúdo usado nesta apresentação*



Na aula de hoje veremos...

- 1 Estrutura sequencial
- 2 Estrutura de decisão
- 3 Estrutura de repetição
- 4 Enquanto – while
- 5 Para – for



Estrutura sequencial



Estrutura sequencial

Conjunto de ações será executado em uma sequência linear de cima para baixo.

Entrada: Notas: $N1, N2, N3, N4$

Saída: Média: MA

// Variáveis

$N1, N2, N3, N4, MA \leftarrow 0$

// Entrada de dados

$N1 \leftarrow \text{leia}()$

$N2 \leftarrow \text{leia}()$

$N3 \leftarrow \text{leia}()$

$N4 \leftarrow \text{leia}()$

// Processamento

$MA \leftarrow (N1 + N2 + N3 + N4)/4$

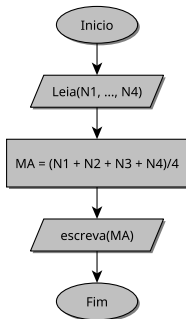
// Saída de dados

escreva(MA)

Algoritmo 1: Algoritmo sequencial: média aritmética



Estrutura sequencial



Entrada de dados

```
n1 = float(input('Digite nota 1: '))
n2 = float(input('Digite nota 2: '))
n3 = float(input('Digite nota 3: '))
n4 = float(input('Digite nota 4: '))
```

Processamento

```
ma = (n1 + n2 + n3 + n4)/4
```

Saída de dados

```
print('Média é: ' + str(ma))
```



Estrutura de decisão



Estrutura de decisão

Permite a escolha de um grupo de ações a ser executado quando determinadas condições são satisfeitas.

Simples:

if .. : Se .. :

Composto:

if .. : .. else: Se .. : senão .. :

Aninhado ou encadeado

if .. :

if .. :

if .. :

else



Estrutura de decisão – simples

Entrada: Notas: $N1, N2, N3, N4$

Saída: Situação do aluno

// Declaração de variáveis

$N1, N2, N3, N4, MA \leftarrow 0$

// Entrada de dados

$N1 \leftarrow \text{leia}()$

$N2 \leftarrow \text{leia}()$

$N3 \leftarrow \text{leia}()$

$N4 \leftarrow \text{leia}()$

// Processamento

$MA \leftarrow (N1 + N2 + N3 + N4)/4$

// Decisão

se $MA < 6$ **então**

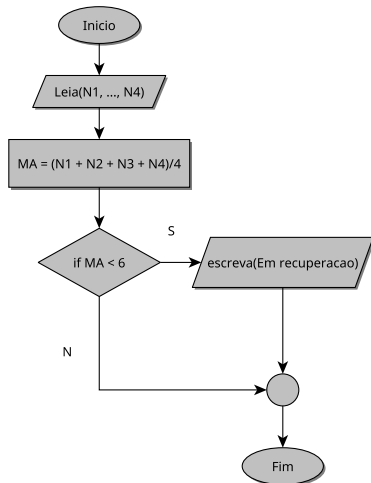
 escreva(Em recuperação)

fim

Algoritmo 2: Decisão simples: média aritmética



Estrutura de decisão – simples



Entrada de dados

```
n1 = float(input('Digite nota 1: '))
n2 = float(input('Digite nota 2: '))
n3 = float(input('Digite nota 3: '))
n4 = float(input('Digite nota 4: '))
```

Processamento

```
ma = (n1 + n2 + n3 + n4)/4
```

Saída de dados condicionada

```
if (ma < 6):
    print('Em Recuperação, média é: ' +
          str(ma))
```



Estrutura de decisão – composta

Entrada: Notas: $N1, N2, N3, N4$

Saída: Situação do aluno

// Declaração de variáveis

$N1, N2, N3, N4, MA \leftarrow 0$

// Entrada de dados

$N1 \leftarrow \text{leia}()$

$N2 \leftarrow \text{leia}()$

$N3 \leftarrow \text{leia}()$

$N4 \leftarrow \text{leia}()$

// Processamento

$MA \leftarrow (N1 + N2 + N3 + N4)/4$

// Decisão

se $MA < 6$ **então**

 escreva(Em recuperação)

senão

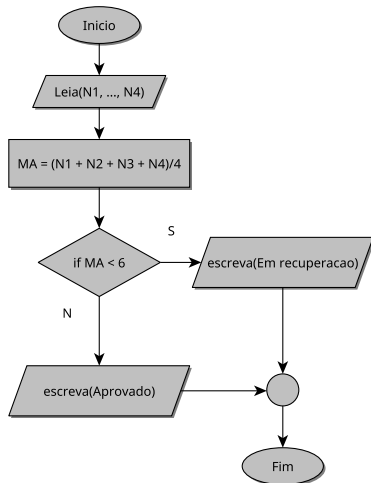
 escreva(Aprovado)

fim

Algoritmo 3: Decisão composta: média aritmética



Estrutura de decisão – composta



Entrada de dados

```
n1 = float(input('Digite nota 1: '))
n2 = float(input('Digite nota 2: '))
n3 = float(input('Digite nota 3: '))
n4 = float(input('Digite nota 4: '))
```

Processamento

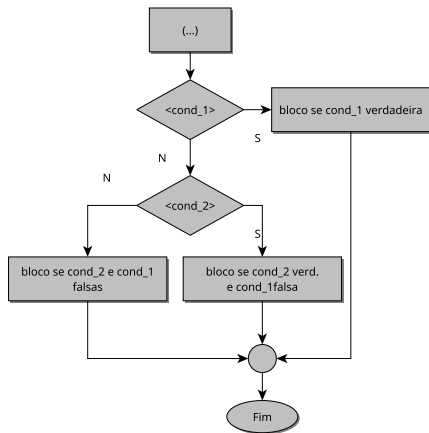
```
ma = (n1 + n2 + n3 + n4)/4
```

Saída de dados condicionada

```
if (ma < 6):
    print('Em Recuperação, média é: ' + str(ma))
else:
    print('Aprovado, média é: ' + str(ma))
```



Estrutura de decisão – aninhada



```
# Entrada de dados
```

```
n1 = float(input('Digite nota 1: '))
```

```
n2 = float(input('Digite nota 2: '))
```

```
n3 = float(input('Digite nota 3: '))
```

```
n4 = float(input('Digite nota 4: '))
```

```
# Processamento
```

```
ma = (n1 + n2 + n3 + n4)/4
```

```
# Saída de dados condicionada
```

```
if (ma < 6):
```

```
    print('Em Recuperação, média é: ' + str(ma))
```

```
else:
```

```
    print('Aprovado, média é: ' + str(ma))
```

```
    if (ma > 9):
```

```
        print('Sua media foi maior que 9.')
    )
```



Caso houver mais testes, pode-se utilizar a instrução **elif**:

```
temperatura = int(input("Qual a temperatura lá fora? "))

if temperatura > 40:
    print("Está quente lá fora")
elif temperatura < 15:          # Aqui há um novo teste!
    print("Está frio lá fora")
else:
    print("Está agradável")

print("Feito!")
```



Indentação importa. Cada linha do **if** que é indentada somente será executada se o teste for verdadeiro.

```
# Esse código não é válido
```

```
if a == 1:
    print("Indentado com 3 espaços.")
    print("Indentado com 4 três espaços. Aqui gera erro.")
    print("O computador vai querer que você decida.")
```

```
# Quando executado
```

```
File "<ipython-input-2-b72e5263e892>", line 4
    print("Indentado com 4 três espaços. Aqui gera erro.")
    ^
```

```
IndentationError: unexpected indent
```





Estrutura de repetição



Estrutura de repetição

Permite repetir um grupo de ações enquanto determinadas condições são satisfeitas.

Denominados laços ou *loops*

Tipos básicos:

enquanto: **while**

para: **for**

Todas estas estruturas podem ser aninhadas, ou seja, **while** contendo outro **while**, **while** contendo um ou vários **for**, ...



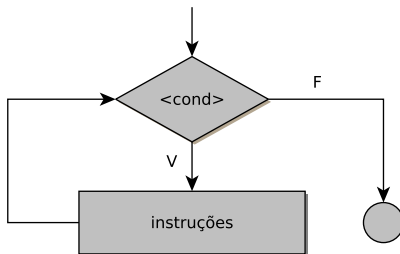
Enquanto – while



Enquanto – while

Teste lógico no **início do laço**

Executa repetição **enquanto** a condição for verdadeira



Média para 10 Alunos

Entrada: Notas de 10 alunos

Saída: Situação de cada aluno

// Declaração de variáveis

$N1, N2, N3, N4, MA \leftarrow 0$

$CONTADOR \leftarrow 0$

// Laço de repetição

enquanto $CONTADOR < 10$ **faça**

 // Entrada de dados

$N1 \leftarrow \text{leia}()$

$N2 \leftarrow \text{leia}()$

$N3 \leftarrow \text{leia}()$

$N4 \leftarrow \text{leia}()$

 // Processamento

$MA \leftarrow (N1 + N2 + N3 + N4)/4$

 // Decisão

se $MA < 6$ **então**

 escreva(Em recuperação)

senão

 escreva(Aprovado)

fim

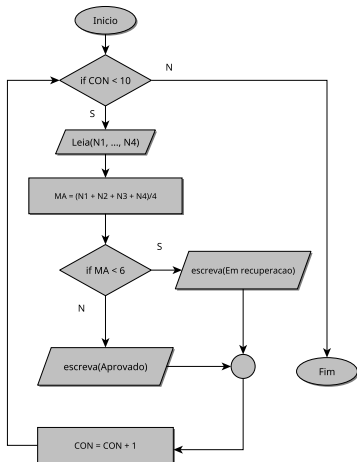
$CONTADOR \leftarrow CONTADOR + 1$

fim

Algoritmo 4: Cálculo de média para 10 alunos



Estrutura de repetição: while



```
# Variável de contagem (condição do while)
contador = 0
```

```
# Repetição
```

```
while (contador < 10):
    print('Aluno: ' + str(contador))
```

```
# Entrada de dados
```

```
n1 = float(input('Digite nota 1: '))
n2 = float(input('Digite nota 2: '))
n3 = float(input('Digite nota 3: '))
n4 = float(input('Digite nota 4: '))
```

```
# Processamento
```

```
ma = (n1 + n2 + n3 + n4) / 4
```

```
# Decisão
```

```
if ma < 6:
    print('Em recuperação: ' + str(ma))
else:
    print('Aprovado: ' + str(ma))
```

```
# Contador de alunos
```

```
contador = contador + 1
```

```
# Fim do while (indentado à direita)
print('Cálculos concluídos')
```



Estrutura de repetição: while

```
# Continuação (condição do while)
continuar = 's'

# Repetição
while (continuar == 's'):

    # Entrada de dados
    n1 = float(input('Digite nota 1: '))
    n2 = float(input('Digite nota 2: '))
    n3 = float(input('Digite nota 3: '))
    n4 = float(input('Digite nota 4: '))

    # Processamento
    ma = (n1 + n2 + n3 + n4) / 4

    # Decisão
    if ma < 6:
        print('Em recuperação: ' + str(ma))
    else:
        print('Aprovado: ' + str(ma))

    # Continuar ou não?
    continuar = input('Deseja continuar (s/n): ')

# Fim do while (indentado à direita)
print('Cálculos concluídos')
```



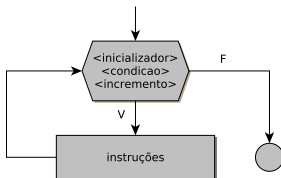
Para – for



Estrutura de repetição: for

Laços **for** são utilizados para iterar em uma sequência.

Em cada iteração, a variável definida no laço irá incrementar.



Estrutura de repetição: for

Strings são sequências de caracteres:

```
# Para cada letra em "banana"  
for letra in "banana":  
    print(letra)
```

Saída:

```
b  
a  
n  
a  
n  
a
```



Estrutura de repetição: for

```
# Para cada letra em "banana"
for letra in "banana":
    if letra == 'a':
        print('A')
    else:
        print(letra)
```

Saída:

```
b
A
n
A
n
A
```



Estrutura de repetição: sequências numéricas

```
# Para uma sequência de 6 números: [0,6[
for numero in range(6):
    print(numero)
```

Saída:

```
0
1
2
3
4
5
```



Sequências numéricas: função range

```
# Range começa em 0 por padrão:
for numero in range(6):      # Intervalo [0,5] ou [0,6[

# Range pode começar em 2 até 10, mas 10 não é incluso
for numero in range(2,10):   # Intervalo [2,10[

# Para mudar o intervalo de incremento
for numero in range(2,30,3): # Intervalo [2,30[
                             # com incremento de 3
```

range(início, fim, incremento)



Sequências numéricas: tabuada

```
# Pede para o usuário
tabuada = int(input('Qual tabuada? '))

print('Range(0,11)')
for numero in range(0,11):
    print(numero * tabuada)

# ou
print('Range mais esperto')
for numero in range(0, tabuada * 11, tabuada):
    print(numero)
```



Sequências numéricas: funções matemáticas

```
# Valores de uma parábola
```

```
for x in range(-10,11):  
    y = x**2 - 9*x + 2  
    print(y)
```

```
192      -16  
164      -18  
138      -18  
114      -16  
92       -12  
72       -6  
54        2  
38        12  
24  
12  
2  
-6  
-12
```



Indentação importa. Cada linha do **for/while** que é indentada somente será executada se o teste for verdadeiro.

```
# Esse código não é válido
```

```
while (a == 1):  
    print("Indentado com 3 espaços.")  
    print("Indentado com 4 três espaços. Aqui gera erro.")  
    print("O computador vai querer que você decida.")
```

```
# Quando executado
```

```
File "<ipython-input-2-b72e5263e892>", line 4  
    print("Indentado com 4 três espaços. Aqui gera erro.")  
    ^
```

```
IndentationError: unexpected indent
```



