



# **25070 - Construção de Algoritmos e Programação**

## **Aula 5 - 01/04/2019**

**Joice Otsuka**

*joice@ufscar.br*

**Algoritmos - Estruturas de repetição**

---

# Controle de fluxo: Estruturas de repetição

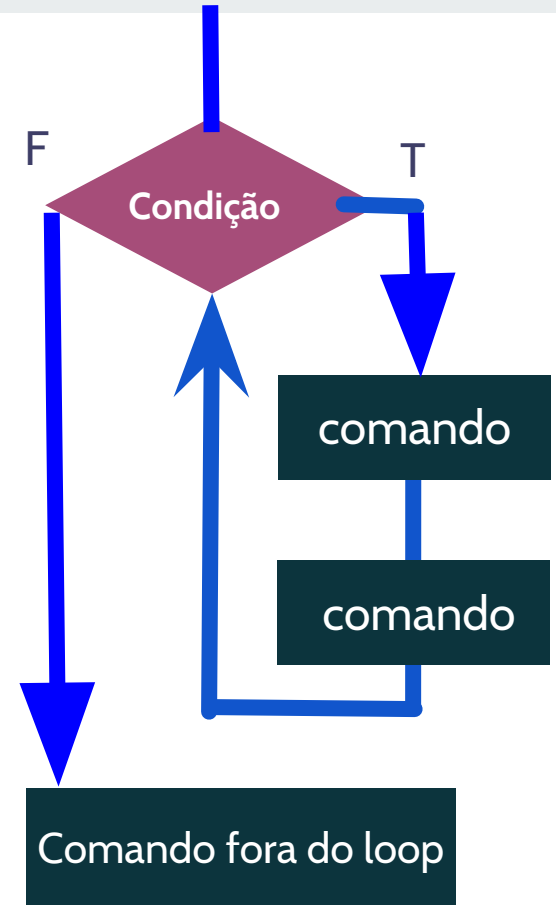


## Estruturas de repetição (iterativas)

- Permitem a execução de um bloco de comandos várias vezes
  - Geralmente dependente de uma condição
- Permitem controlar o número de repetições de um bloco de comandos
  - Repita enquanto / repita até condição verdadeira
    - *while* (enquanto.. faça)
    - *repeat until* (repita até)
  - Repita para cada elemento de um conjunto especificado
    - *for*

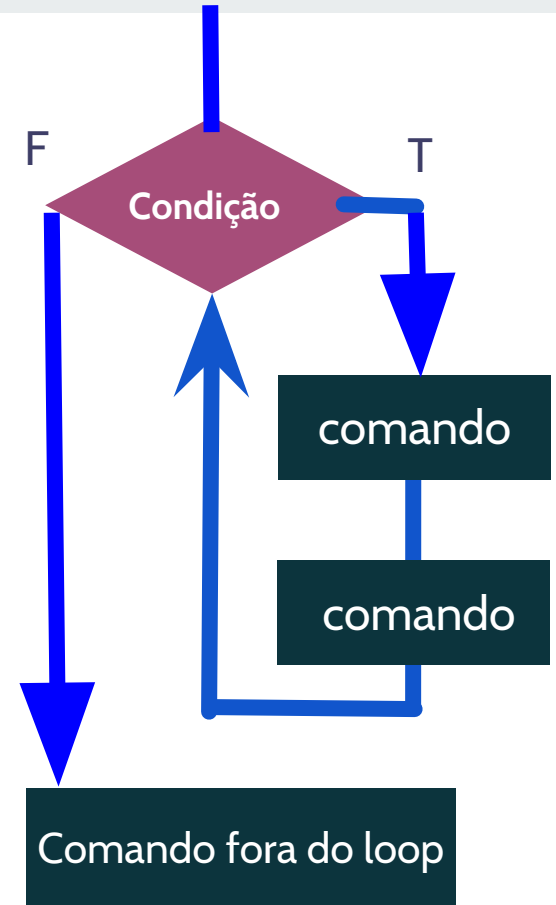
## Comando while (repita enquanto)

- Condição de repetição é verificada antes de entrar no bloco de repetição.
- A repetição ocorre enquanto a condição for verdadeira. Pára quando for falsa
- Comandos do bloco de repetição são executados a cada iteração (repetição)
- Repetição pode ocorrer 0 ou + vezes



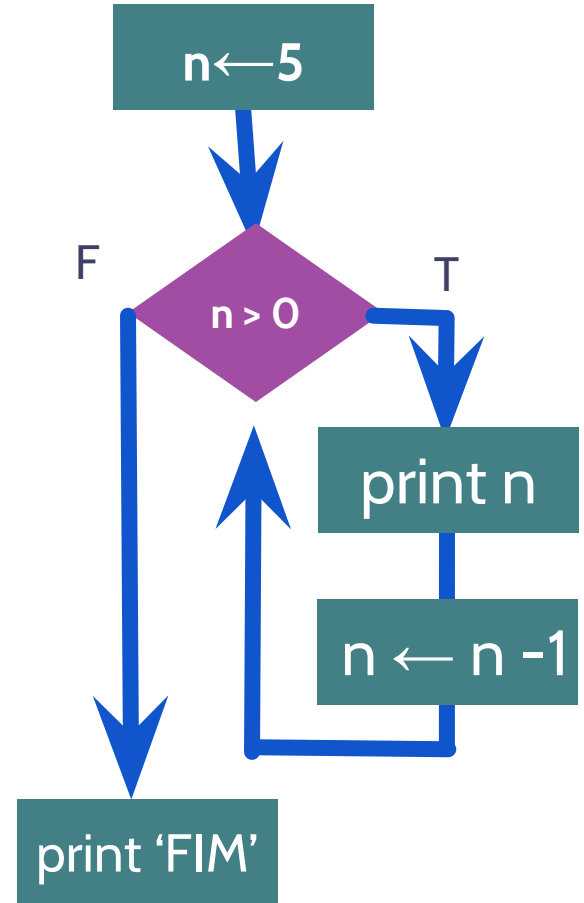
## Comando while (repita enquanto)

```
while (condicao) do  
    Bloco de comandos repetido  
enquanto a condição é  
    verdadeira  
end while
```



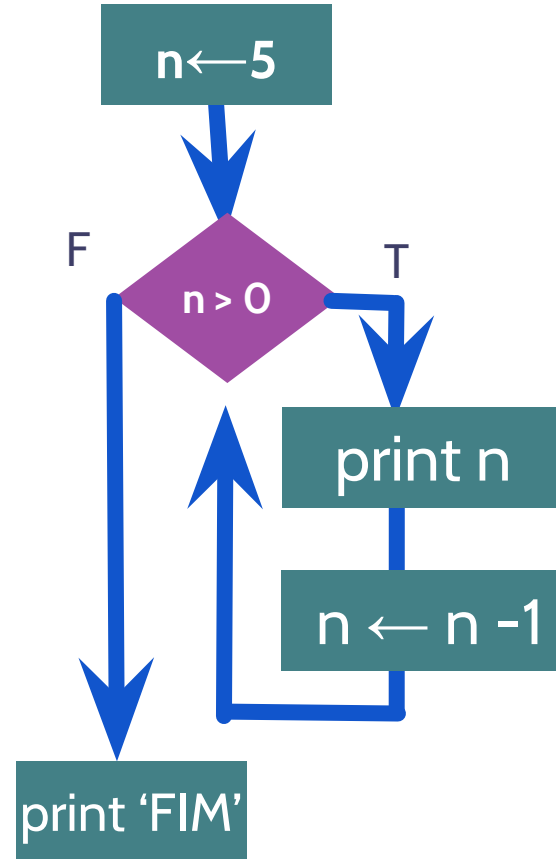
## Exemplo

```
n ← 5
while (n > 0) do
  print(n)
  n ← n - 1
end while
print('FIM')
```



## Comando while (repita enquanto)

- Cuidados:
  - A condição tem que se tornar falsa em algum momento, caso contrário, o loop nunca se encerra e teremos um **loop infinito!**
  - Logo, deve haver algum comando que atualize a condição dentro do loop, de forma que em algum momento a condição seja falsa

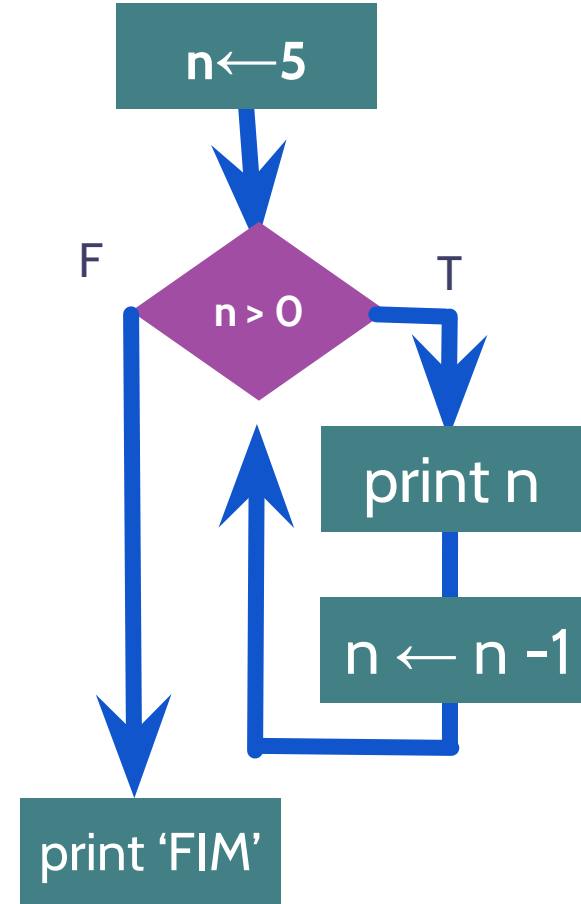


## Exemplo

```
n ← 5
while (n > 0) do
  print(n)
  n ← n - 1
end while
print('FIM')
```

Condição de parada

Atualização

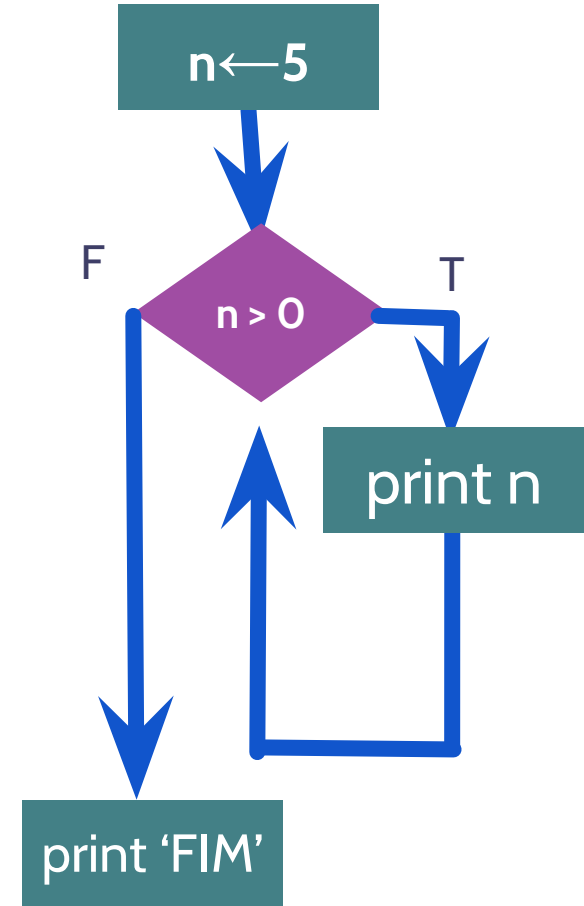




## Exemplo

```
n ← 5
while (n > 0) do
  print(n)
end while
print('FIM')
```

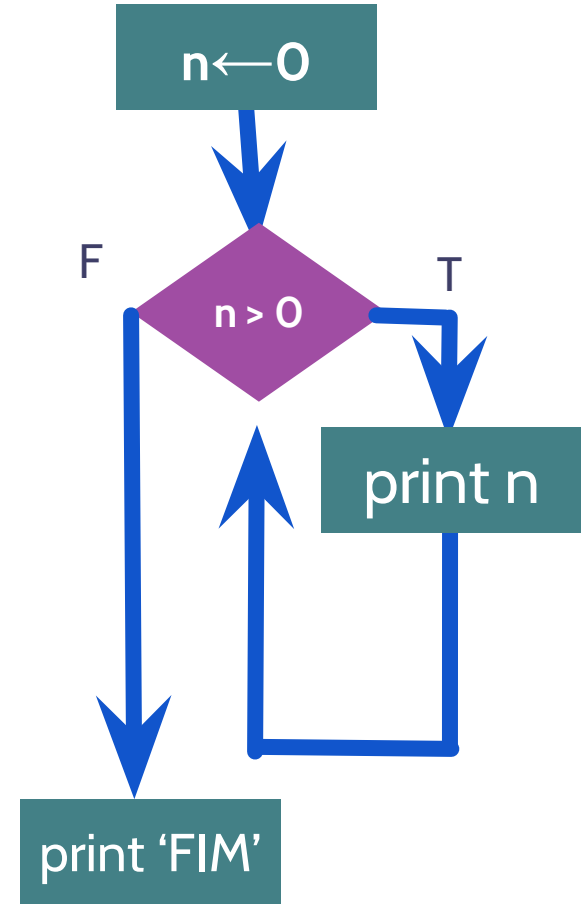
**Temos um loop infinito!!!**

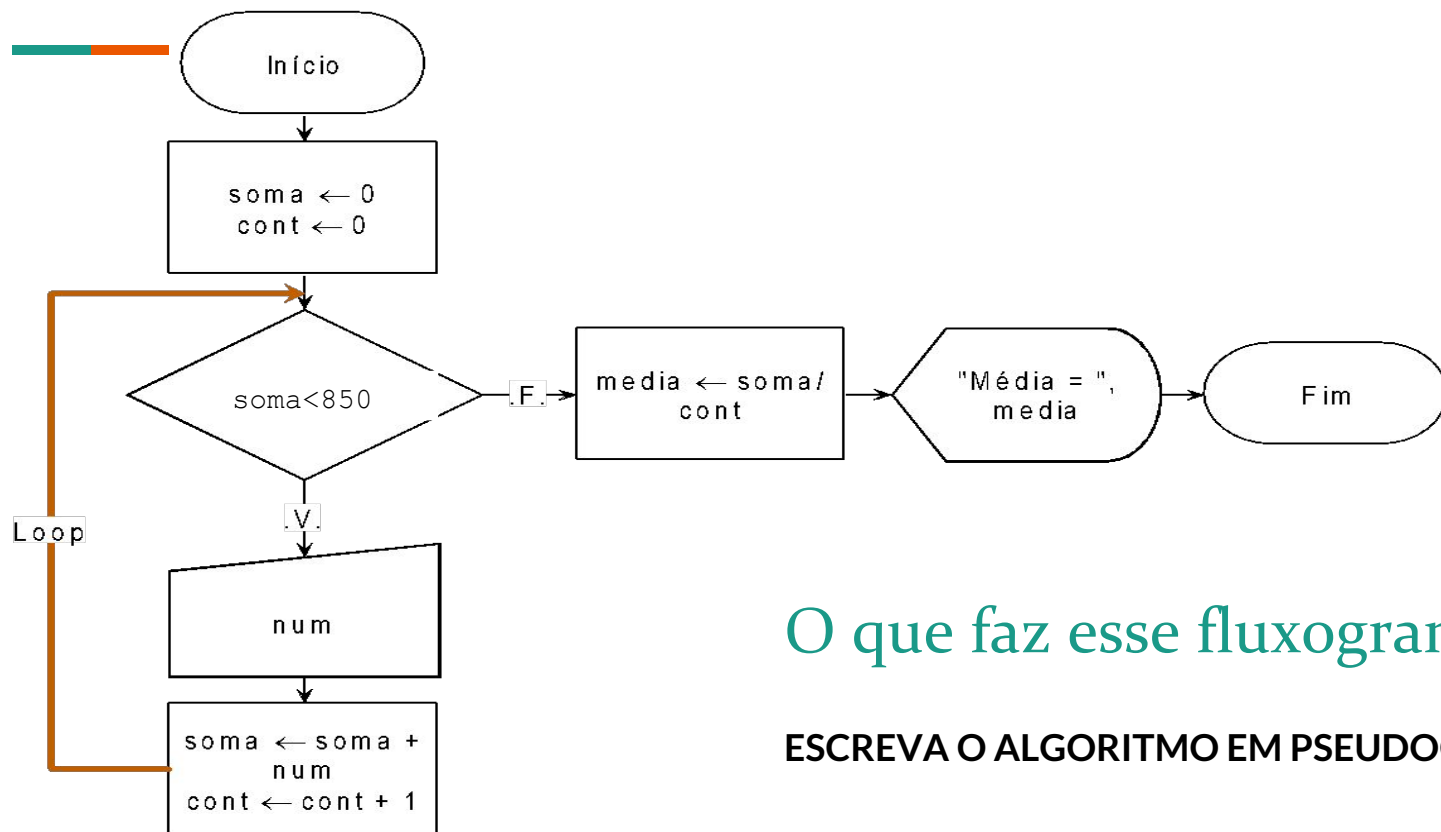


## Exemplo

```
n ← 0
while (n > 0) do
  print(n)
end while
print('FIM')
```

Não entra no loop!!!



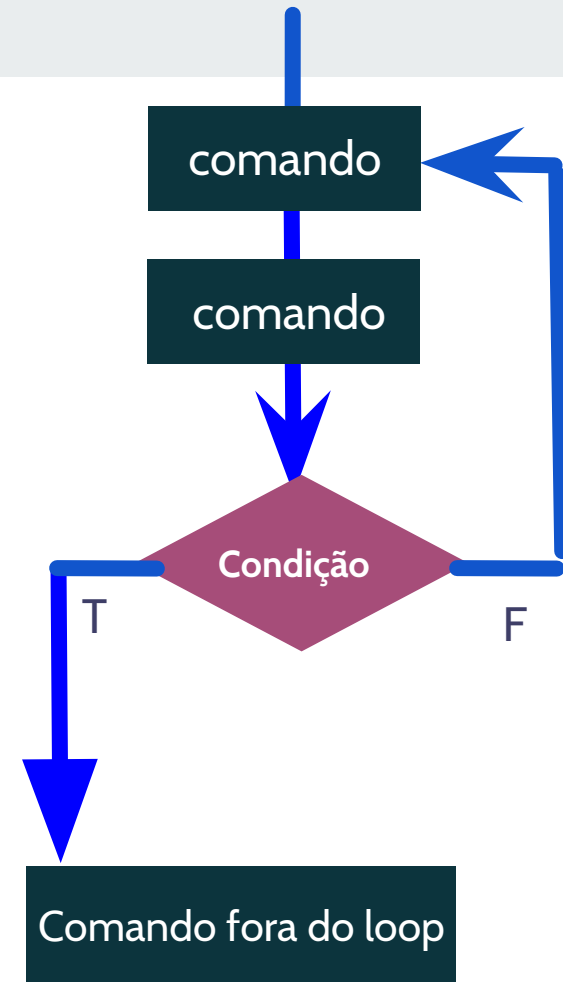


O que faz esse fluxograma?

ESCREVA O ALGORITMO EM PSEUDOCÓDIGO

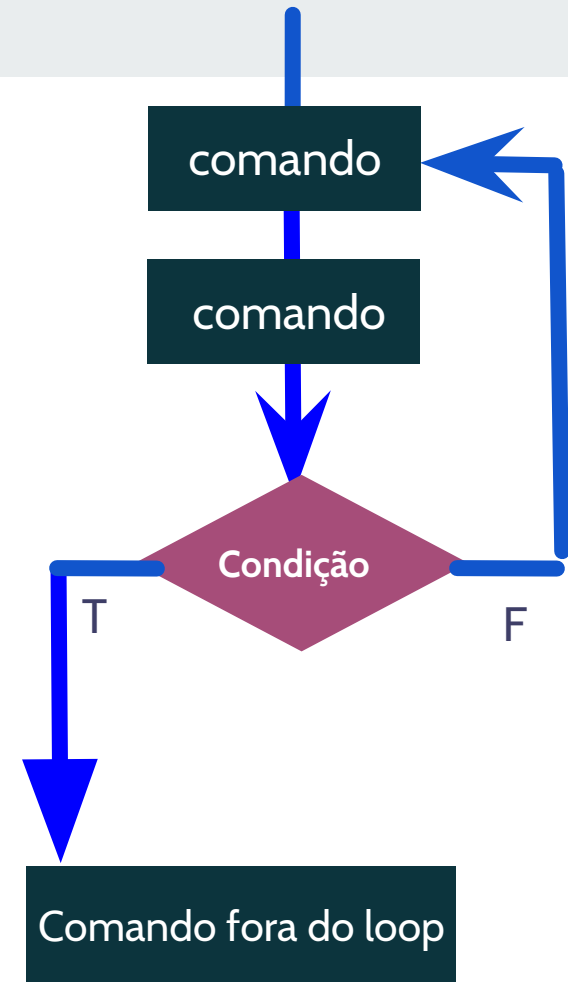
## Comando repeat until (repita até)

- Condição de repetição é verificada depois de entrar no bloco de repetição.
- A repetição ocorre até a condição se tornar verdadeira (condição de parada). Continua enquanto for falsa.
- Comandos do bloco de repetição são executados a cada iteração (repetição)
- Repetição pode ocorrer 1 ou + vezes



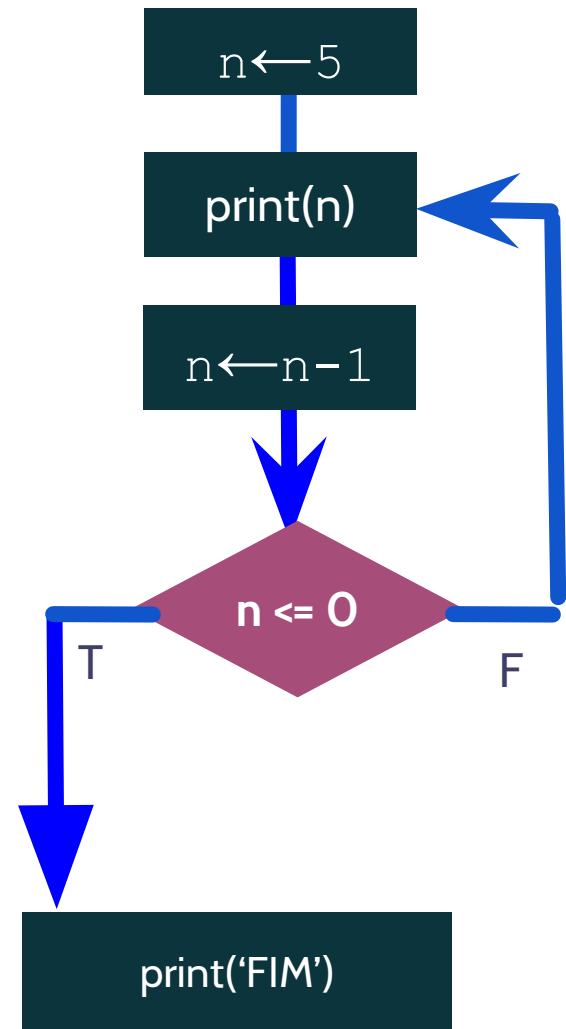
## Comando repeat until (repita até)

```
repeat
  Bloco de comandos repetido
  até a condição se tornar
  verdadeira
until (condicao)
```



## Exemplo

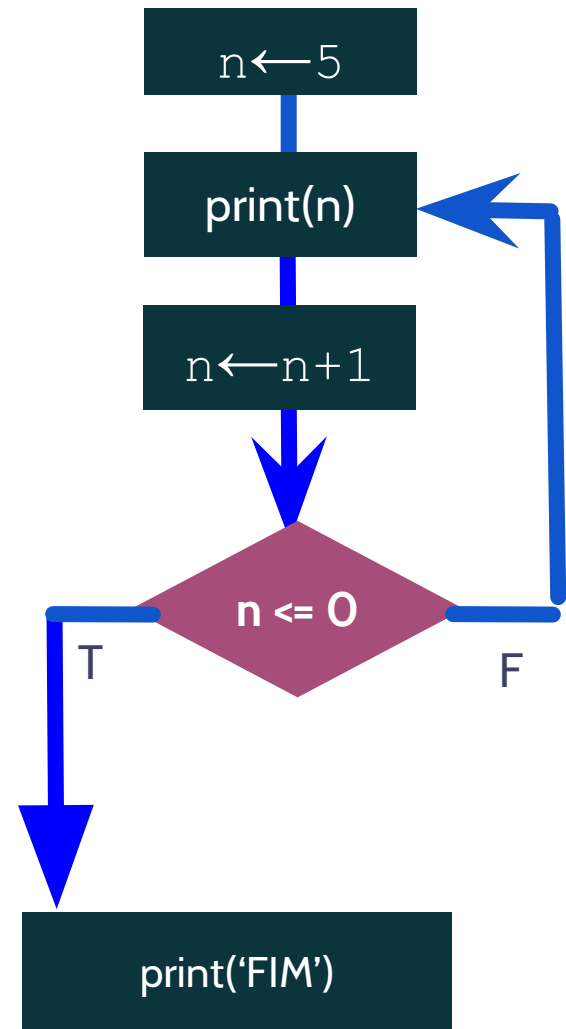
```
n ← 5
repeat
  print(n)
  n ← n - 1
until (n ≤ 0)
print('FIM')
```



## Exemplo

```
n ← 5
repeat
  print(n)
  n ← n + 1
until (n ≤ 0)
print('FIM')
```

**Temos um loop infinito!!!**





## Exercício

Refaça o exercício do slide 11 utilizando a estrutura repita até.





## Loop indefinido

- Os comandos **while** e **repeat** permitem a criação de loops indefinidos
  - While - repete enquanto a condição for verdadeira
  - Repeat - repete até que a condição de parada se torne verdadeira
- Nem sempre é possível prever quando isso ocorrerá



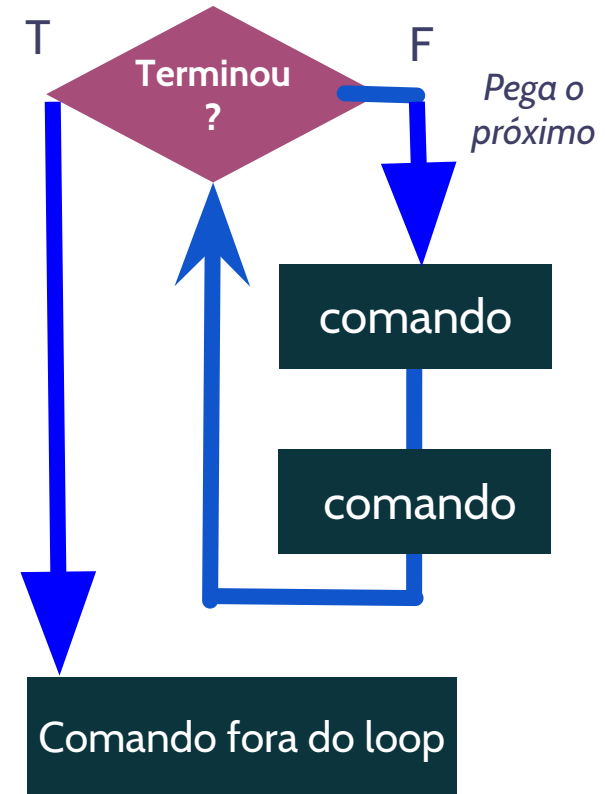
## Exercício:

- Escreva um algoritmo para controlar uma pequena máquina registradora. Você deve solicitar ao usuário que digite o código do produto e a quantidade a ser comprada, até que digite 0 para o código. Ao final, exiba o total da compra.
- Utilize a tabela ao lado para obter o preço de cada produto. Qualquer outro código deve gerar a mensagem de erro “Código inválido”.

Código	Preço
1	0.50
2	1.00
3	4.00
5	7.00
9	8.00

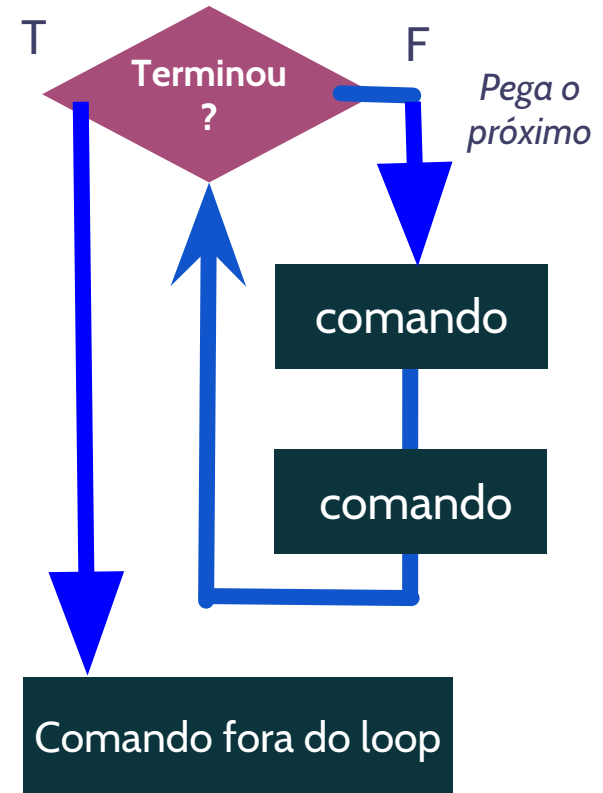
## Comando for (Repita para)

- Loop definido, sempre sabemos quantas repetições ocorrerão
- Repete o bloco de comandos para cada elemento de um conjunto definido
- A variável de controle assume o valor de um elemento do conjunto a cada iteração (pega automaticamente o próximo a cada repetição)



## Comando for (Repita para)

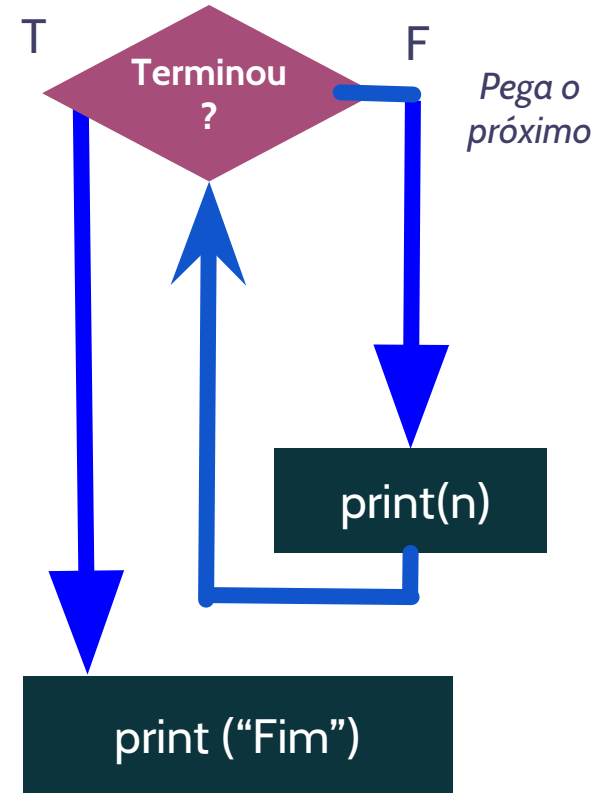
```
for (especificação do conjunto)do  
    Bloco de comandos repetido a  
    cada elemento do conjunto  
end for
```



## Comando for (Repita para)

```
for n ← 1 to 5 do  
    print (n)  
end for  
print ("Fim")
```

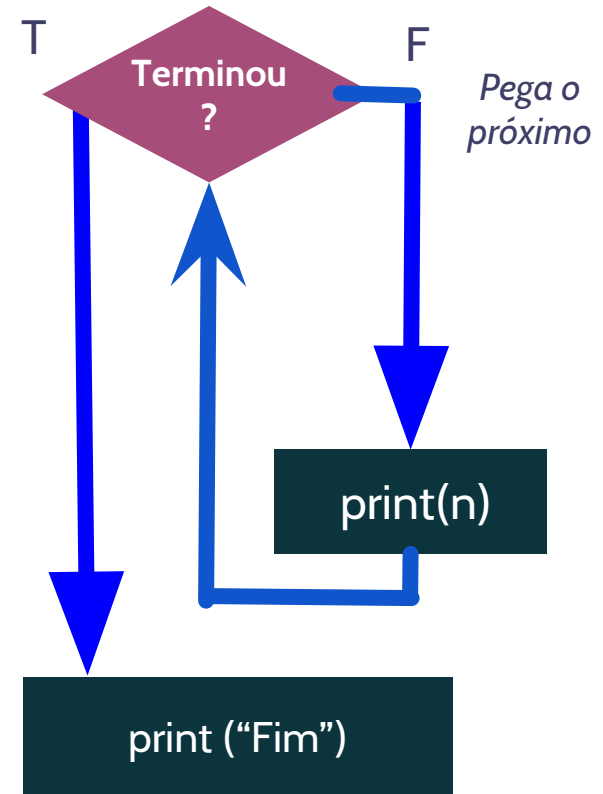
```
para n ← 1 até 5 faça  
    imprima (n)  
fim para  
imprima ("Fim")
```



## Comando for (Repita para)

```
for n ← 5 to 0 step -1 do  
    print (n)  
end for  
print ("Fim")
```

```
for n ← 1 to 5 step 0.1 do  
    print (n)  
end for  
print ("Fim")
```





## Exercícios

Faça um algoritmo que imprima os números pares de 1 a 100, inclusive.

Faça um algoritmo que acumule a soma dos números ímpares de 1 a 100.

Faça um algoritmo que calcule a média dos valores de um conjunto de  $n$  inteiros.

Faça um algoritmo que verifique o maior valor de um conjunto de  $n$  inteiros.



## for x repeat until/while

- Por que os exercícios do slide anterior puderam ser solucionados com o comando for?
- Poderia ser solucionado com repeat until/while?
- Os exercícios dos slides 11 e 18 poderiam ser solucionados com o comando for??





## Estrutura de repetição

- Repita enquanto (*while*) / repita até (*repeat*)
  - repete enquanto/até condição verdadeira
- Repita para (*for*)
  - Loop definido, executa número pré-definido de vezes (para cada elemento de um conjunto especificado)



# Repetições aninhadas

- Podemos combinar vários comandos de repetição criando uma iteração dentro de outra.
- Exemplo: impressão de tabuadas de 1 a 10

Tabuada de 1

1 x 1 = 1

2 x 1 = 2

3 x 1 = 3 ...

Tabuada de 2

1 x 2 = 2

2 x 2 = 4 ...



## Exercício

- Solicite que o usuário informe uma operação dentre (+, -, \* ou /) ou 0 para encerrar. Qualquer outra entrada deverá ser informada como “Entrada inválida!”
- Para cada operação válida informada, imprima a tabuada de 1 a 10 para a operação informada.
- Utilize o comando **while** apenas para loops indefinidos e utilize o comando **for** para loops definidos.



Lista de exercícios CAP (Prof. Jander): <https://goo.gl/mm8fut>

- Seção 3.6