



1001350 - Construção de Algoritmos e Programação

Aulas 5 e 6 - 14 e 16/04/2025

Joice Otsuka

Estruturas de repetição

Controle de fluxo: Estruturas de repetição

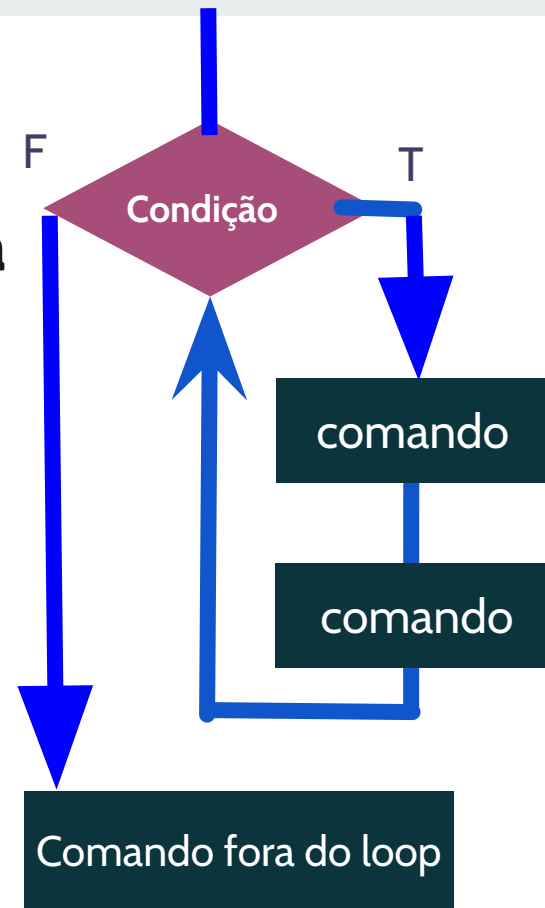


Estruturas de repetição (iterativas)

- 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
 - enquanto (condição verdadeira) faça
 - repita ... até (condição verdadeira)
 - Repita para cada elemento de um conjunto especificado
 - para ... faça

Comando: enquanto (condição) faça

- 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



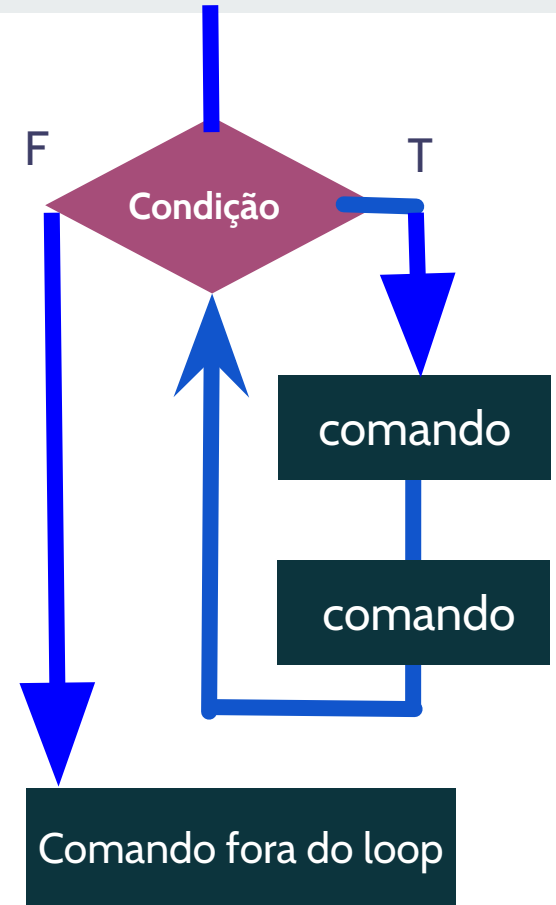
Comando: enquanto (condição) faça

enquanto (condição) faça

Bloco de comandos é repetido

enquanto a condição é
verdadeira

fimenquanto





Exemplo

$n \leftarrow 5$

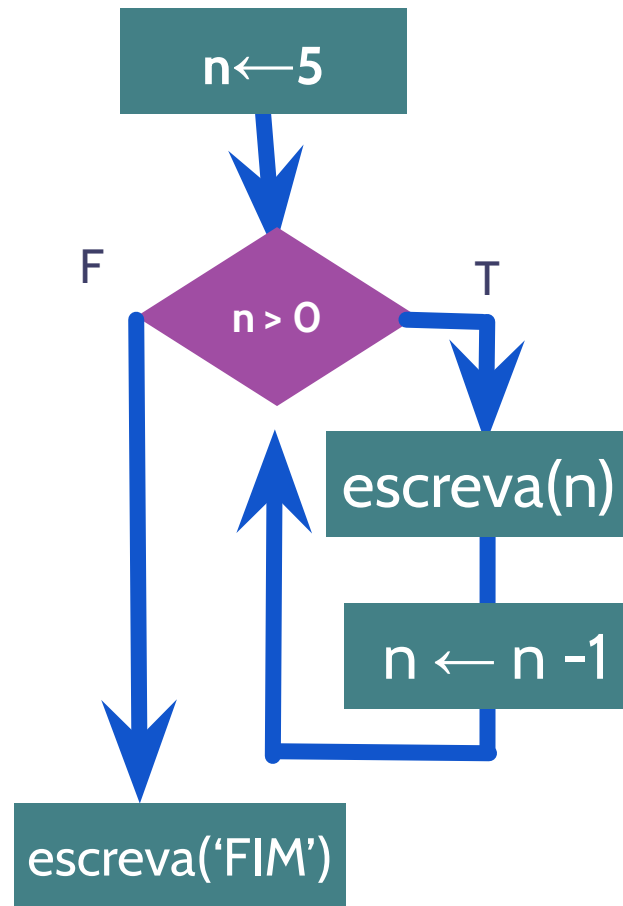
enquanto ($n > 0$) **faça**

$\text{escreva}(n)$

$n \leftarrow n - 1$

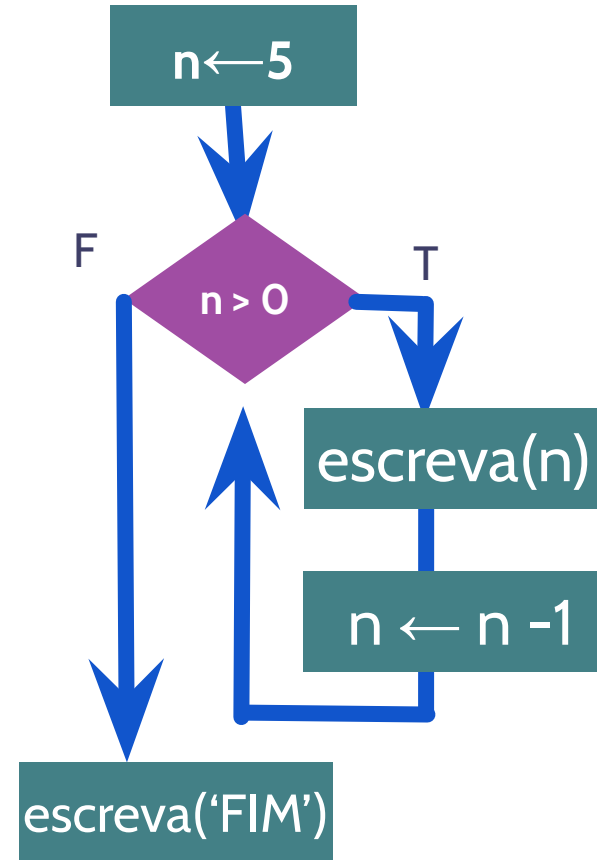
fimenquanto

$\text{escreva}(\text{"FIM"})$



Comando: enquanto (condição) faça

- Cuidados:
 - A condição tem que se tornar falsa em algum momento, caso contrário, o *loop* (repetição) 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

$n \leftarrow 5$

enquanto $(n > 0)$ **faça**

$\text{escreva}(n)$

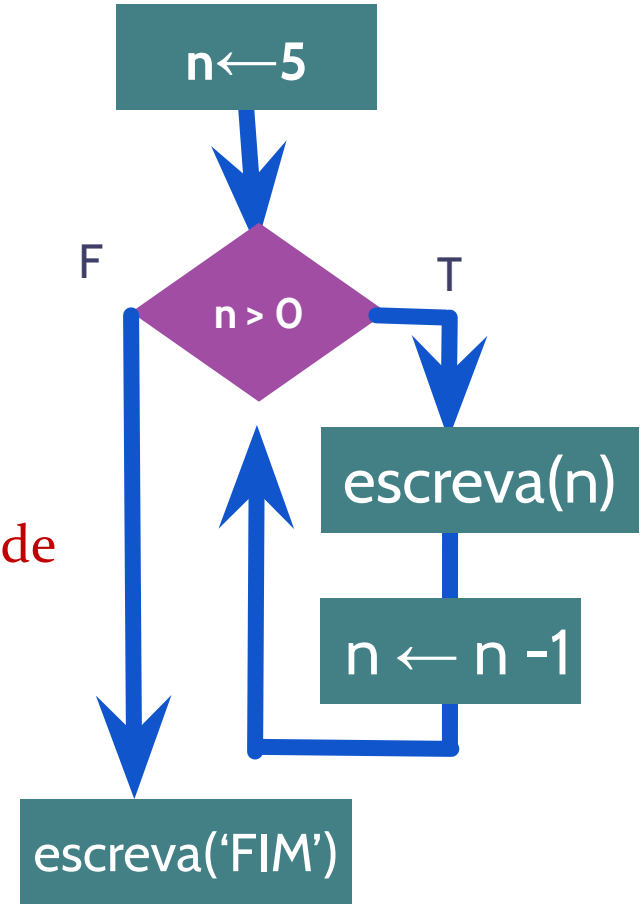
$n \leftarrow n - 1$

fimenquanto

$\text{escreva}(\text{"FIM"})$

Condição de
parada

Atualização



Exemplo

$n \leftarrow 5$

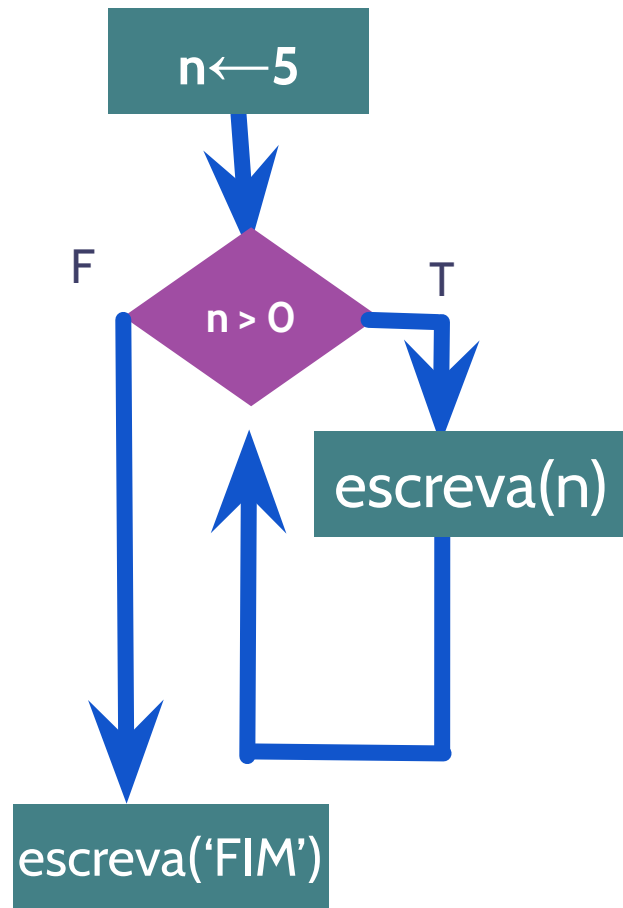
enquanto ($n > 0$) **faça**

$\text{escreva}(n)$

fimenquanto

$\text{escreva}(\text{"FIM"})$

Temos um *loop* infinito!!!

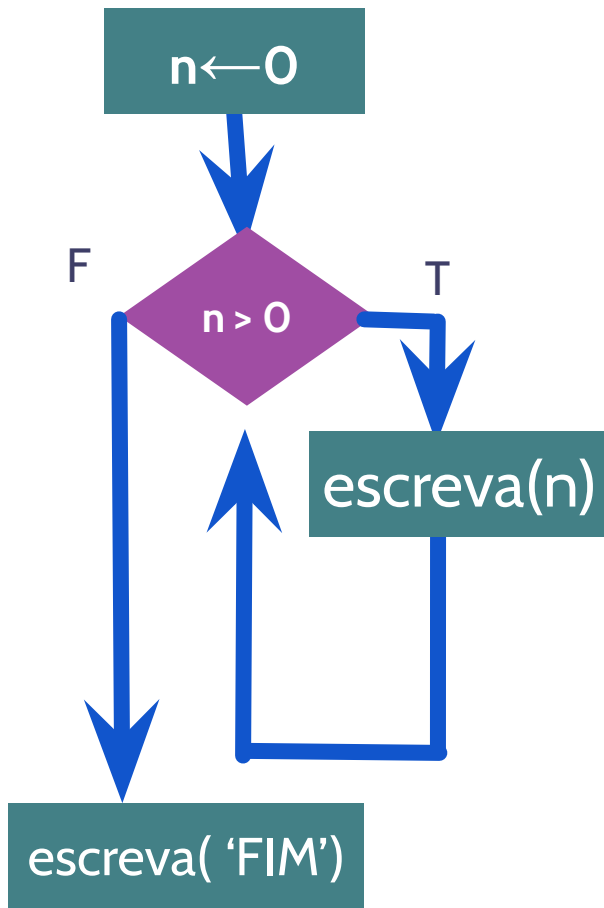


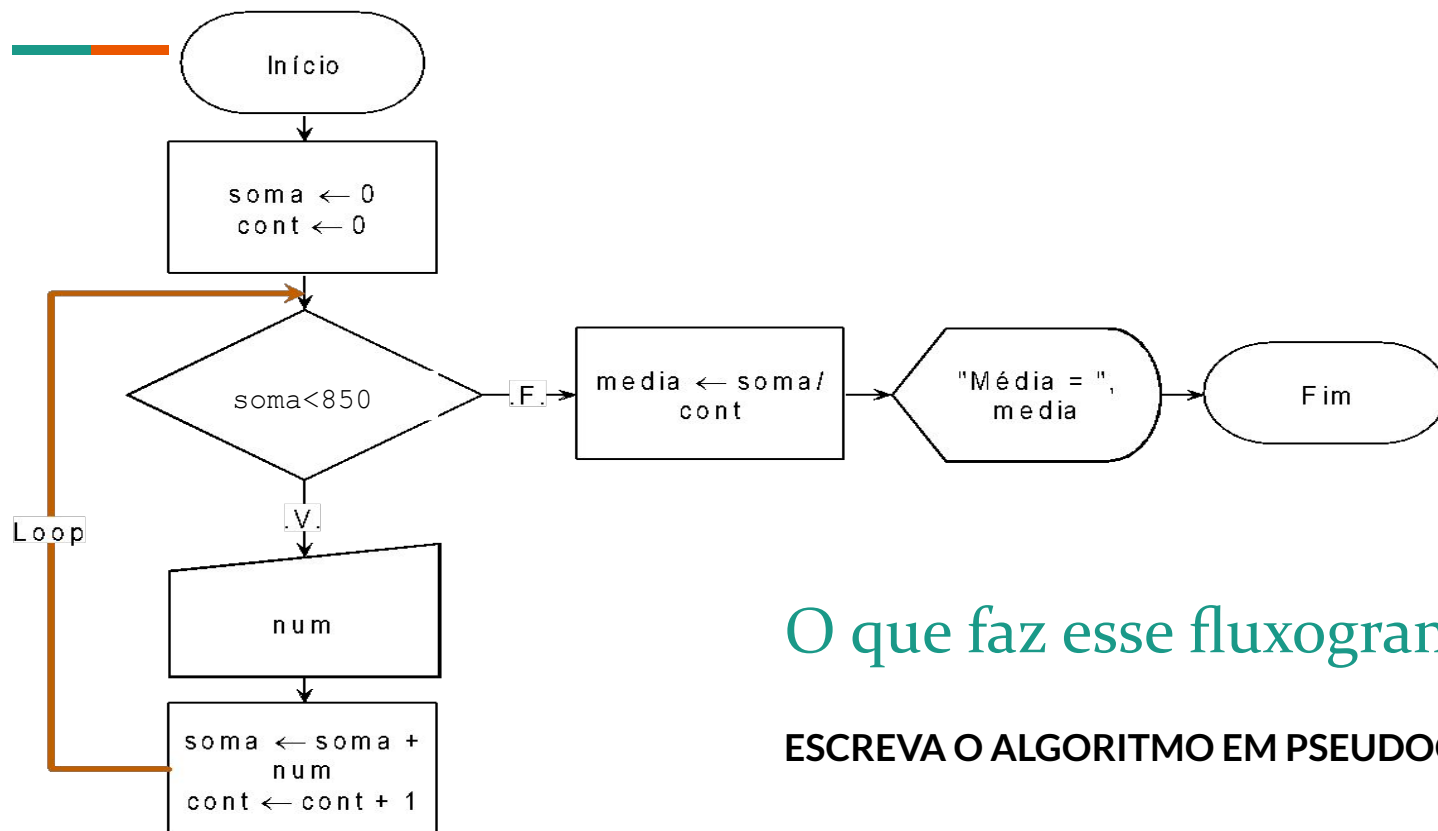
Exemplo

$n \leftarrow 0$

```
enquanto (n > 0) faça  
    escreva(n)  
fimenquanto  
    escreva("FIM")
```

Não entra no *loop*!!!



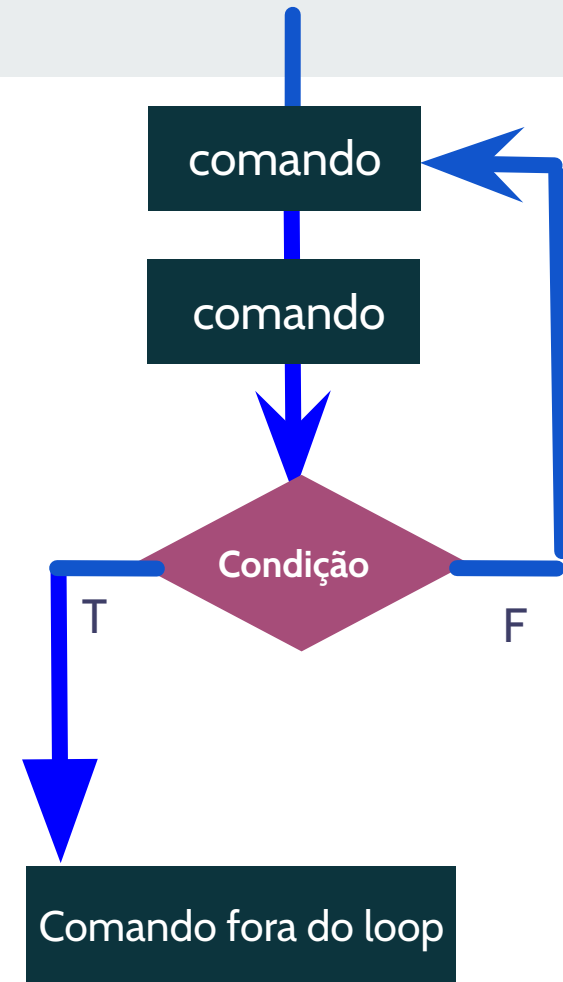


O que faz esse fluxograma?

ESCREVA O ALGORITMO EM PSEUDOCÓDIGO

Comando: repita até (condição)

- 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**

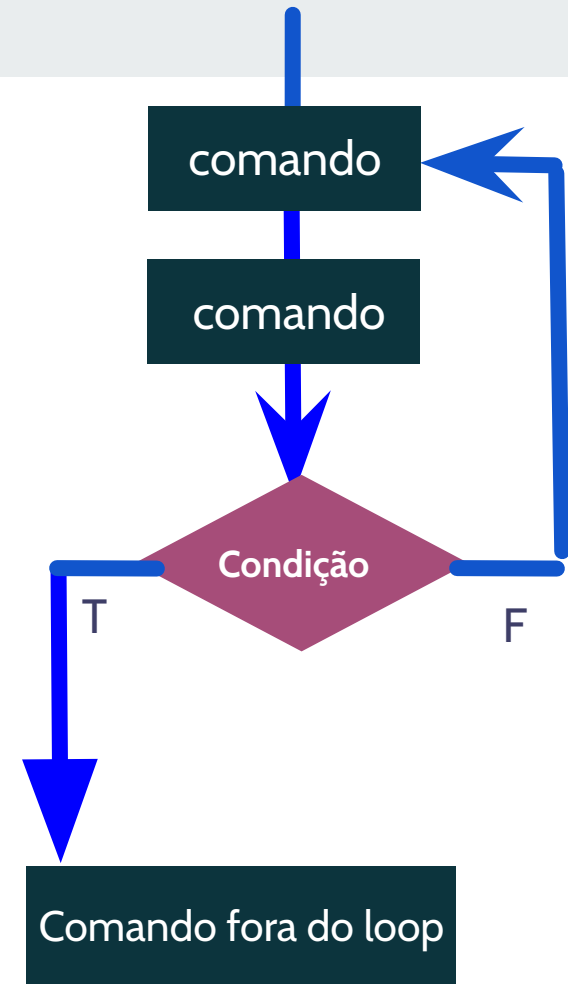


Comando: repita até (condição)

repita

Bloco de comandos repetido
até a condição se tornar
verdadeira

até (condição)



Exemplo

$n \leftarrow 5$

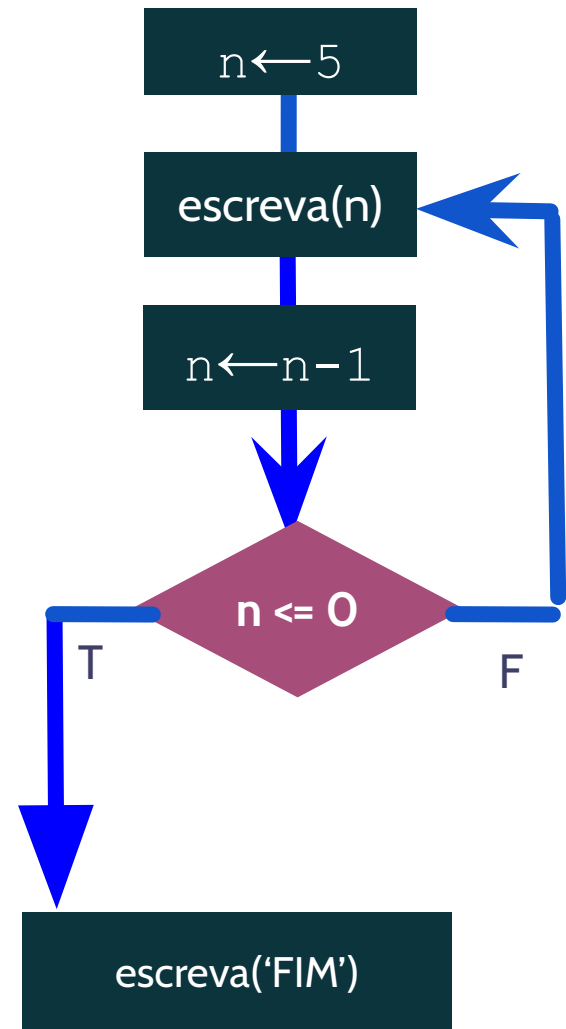
repita

`escreva(n)`

$n \leftarrow n - 1$

até ($n \leq 0$)

`escreva("FIM")`



Exemplo

$n \leftarrow 5$

repita

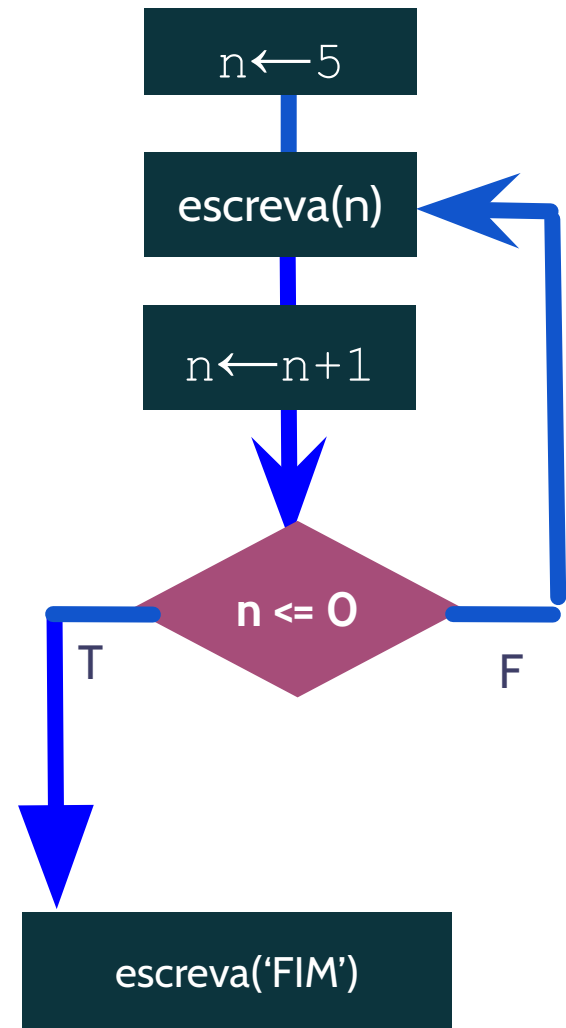
`escreva(n)`

$n \leftarrow n + 1$

até ($n \leq 0$)

`escreva("FIM")`

Temos um loop infinito!!!





Exercício

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



Exercício:

- Escreva um algoritmo para controlar uma pequena máquina registradora. Para cada produto, 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



Loop indefinido

- Os comandos **enquanto .. faça** e **repita .. até** permitem a criação de *loops* indefinidos
 - **enquanto** (condição) **faça**
repete **enquanto** a condição for verdadeira
 - **repita até** (condição)
repete **até que** a condição de parada se torne verdadeira
- Nem sempre é possível prever quando a condição de parada ocorrerá - **loop indefinido**

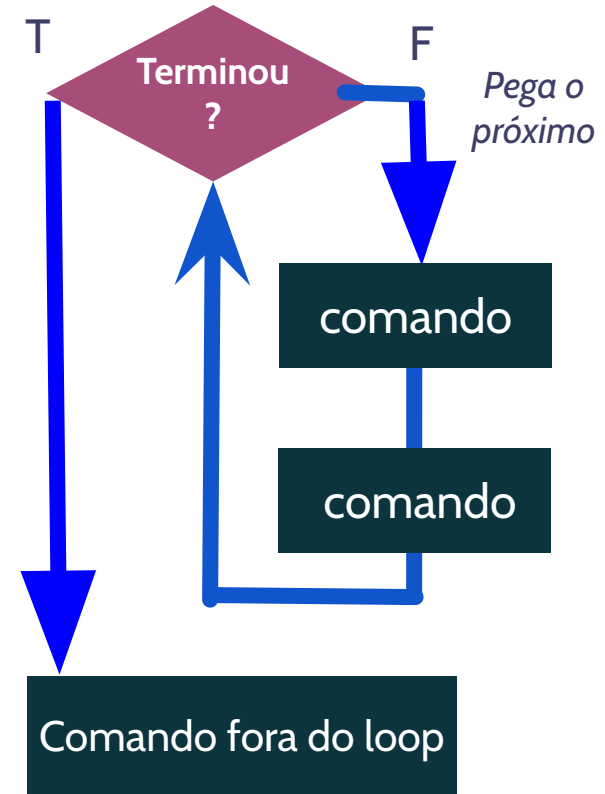


Exercícios

- Exercícios 1, 5 e 6

Comando: para ... faça

- 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)

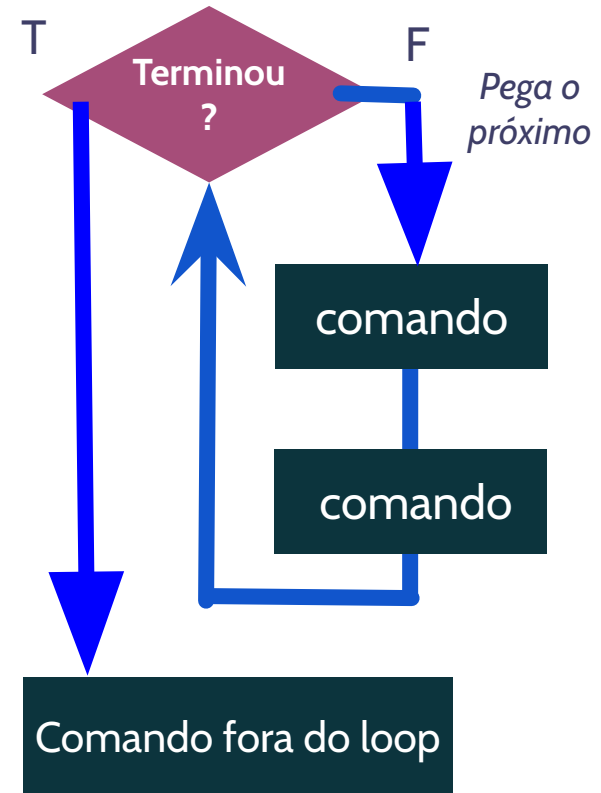


Comando: para .. faça

para <especificação do conjunto*> faça
 Bloco de comandos repetido a cada
 elemento do conjunto
fimpara

* <variavel controle> de <valor-inicial>
até <valor-max> [passo incremento]

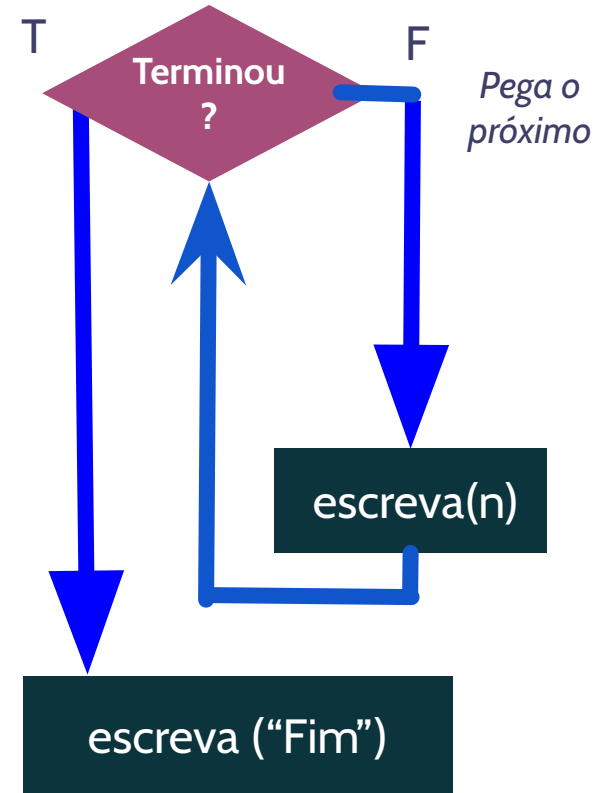
[passo incremento] é opcional
Caso não presente, incremento = 1



Comando: para .. faça

```
para n de 1 até 5 faça  
  escreva(n)  
fimpara  
escreva("Fim")
```

```
para n de 1 até 5 passo 2 faça  
  escreva(n)  
fimpara  
escreva("Fim")
```

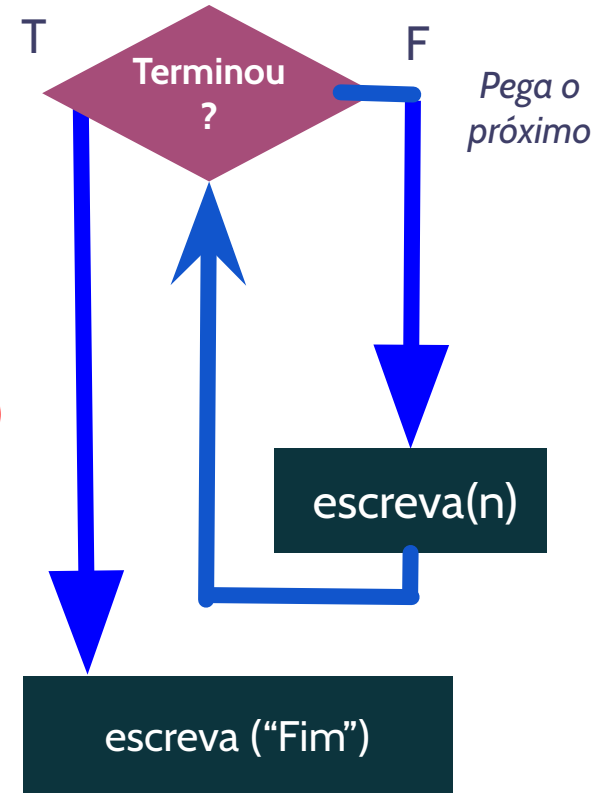


Comando: para .. faça

```
para n de 5 até 1 faça  
  escreva(n)  
fimpara  
escreva("Fim")
```

NÃO FUNCIONA !!!
NECESSÁRIO DEFINIR O PASSO
INCREMENTO (VALOR PADRÃO 1)

```
para n de 5 até 1 passo -1 faça  
  escreva(n)  
fimpara  
escreva("Fim")
```





Exercícios

1. Faça um algoritmo que imprima os números pares de 1 a 100, inclusive.
2. Faça um algoritmo que acumule a soma dos números ímpares de 1 a 100.
3. Faça um algoritmo que leia um natural n , seguido da leitura de n números inteiros e calcule a média dos valores lidos.
4. Faça um algoritmo que leia um natural n , seguido da leitura de n números naturais e verifique o maior valor lido.



para .. até X repita .. até/enquanto

- Por que os exercícios do slide anterior puderam ser solucionados com o comando **para..até?**
- Poderiam ser solucionados com **repita..até/enquanto..faça?**
- Os exercícios dos slides 11 e 17 poderiam ser solucionados com o comando **para..até??**



Estrutura de repetição

- enquanto..faça / repita até
 - repete enquanto ou até condição ser verdadeira
- para..faça
 - *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 das 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

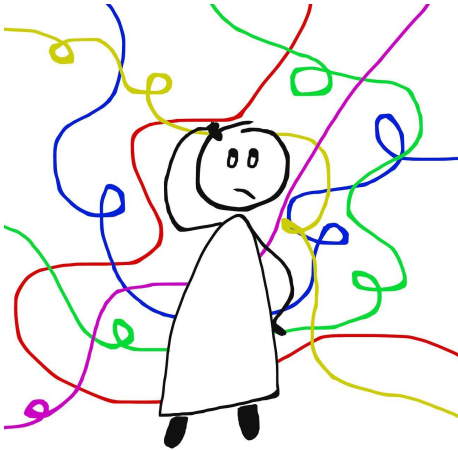
2 x 2 = 4 ...



Exercício

- Solicite que o usuário informe um número N (entre 1 e 10) ou 0 para encerrar. Qualquer outra entrada deverá ser informada como “Entrada inválida!”
- Para cada número N válido, imprima a tabuada do número informado.
- Utilize o comando **enquanto..faça/ repita...ate** apenas para *loops* indefinidos e utilize o comando **para..faça** para *loops* definidos.

Dúvidas?



Compartilhe no Fórum de Dúvidas
Participe dos atendimentos (Monitorias)