

Algoritmos Iterativos

Comandos Enquanto (While)

Usando estruturas de controle de execução para criar algoritmos com repetição de comandos

Estrutura lógica de um algoritmo

- Contém uma sequência lógica e finita de ações (**comandos**), claras e precisas, com fluxo de execução baseado em:

- sequência;

Programação Estruturada

- seleção condicional (seleção de ações);

- **iteração (repetição de ações);**



Na aula de hoje!

Considere o enunciado

- Escreva um algoritmo para calcular o **fatorial** de um número informado pelo usuário
 - **Entradas:** um número
 - **Saída:** fatorial do número
 - **Processamento:** realizar operações necessárias para calcular o fatorial

Fatorial

- O fatorial de um número **n** é dado pela seguinte equação matemática:

$$f = n \times (n - 1) \times (n - 2) \times \dots \times 1$$

- A quantidade de termos da equação depende do valor de **n**

Exemplo: o fatorial de 4 é dado pelo que segue

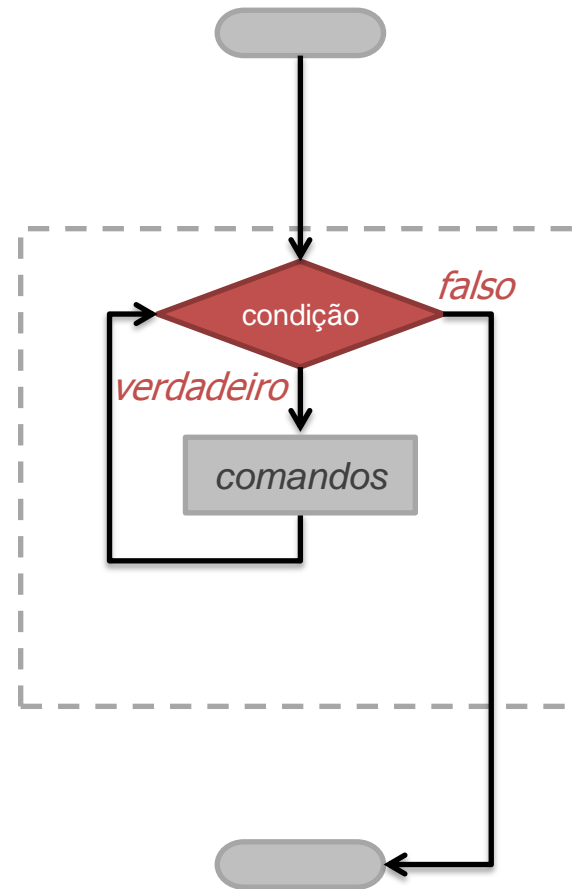
$$4! = 4 \times 3 \times 2 \times 1 = \mathbf{24}$$

Fatoriais

{	0!	=	1
	1!	=	1
	2!	=	2
	3!	=	6
	4!	=	24

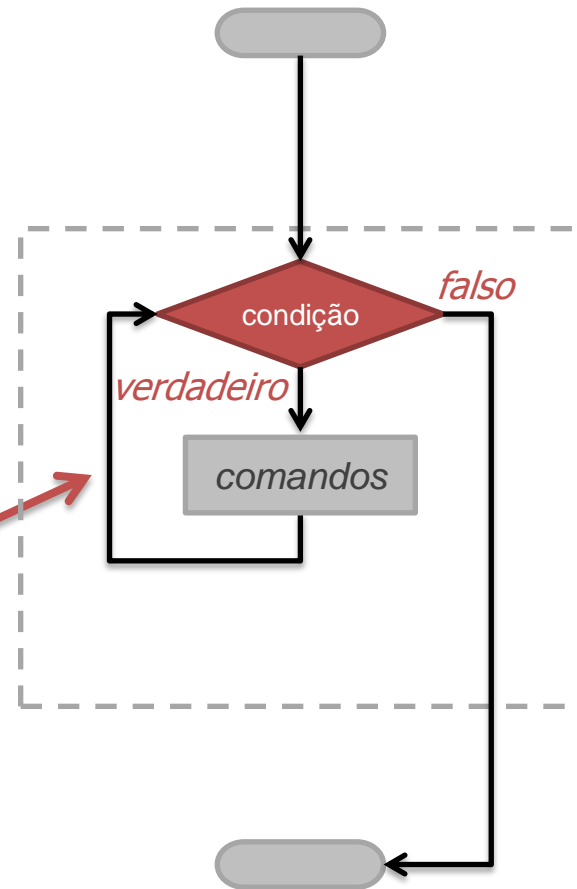
Repetição em Fluxograma

- Uma repetição executa várias vezes os comandos dentro do caminho **verdadeiro**
- A quantidade de execuções depende de uma **condição de parada**
- Quando condição for **falsa** o algoritmo sai do laço



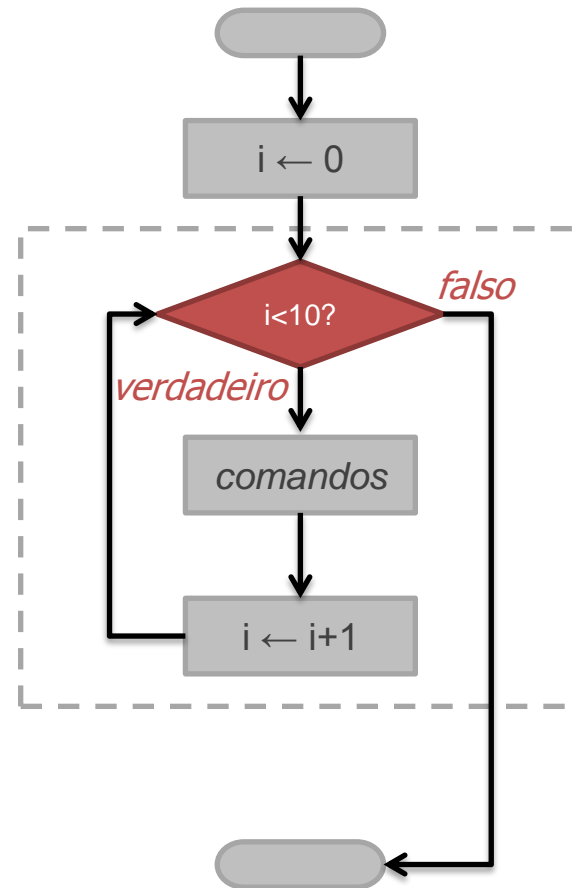
Repetição em Fluxograma

- Note que não existe uma forma específica para representar repetição em fluxograma
- O que indica que existe uma repetição é a **seta de volta** para um ponto anterior



Repetição em Fluxograma

- Por exemplo, um fluxograma que executa 10 vezes uma série de comandos



Repetição em Fluxograma

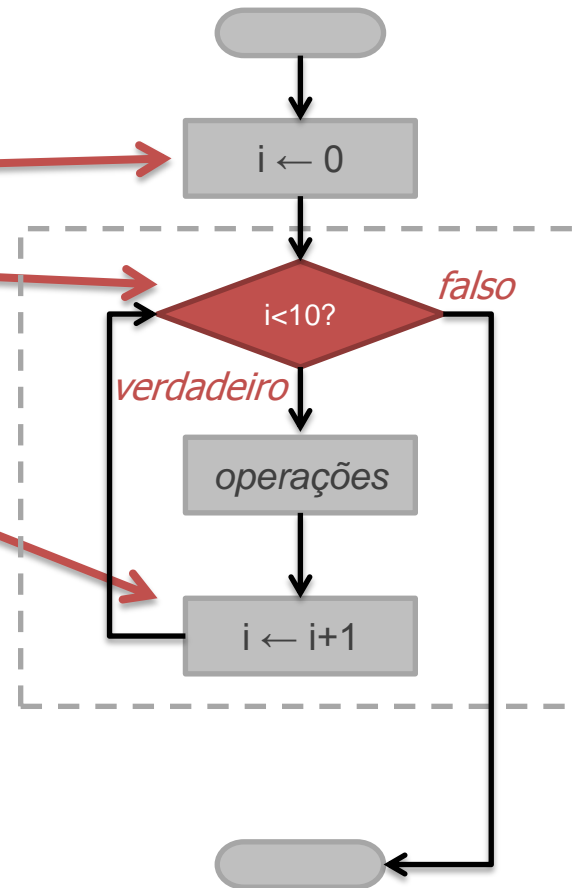
- **Variável de controle**

- Nesse caso usamos i

- Inicialização
 - Condição
 - Alteração

- **Condição de parada**

- Algum dos comandos no caminho verdadeiro precisa **alterar a variável de controle** para que o laço não execute infinitamente



Repetição em Pseudo-Código com Enquanto

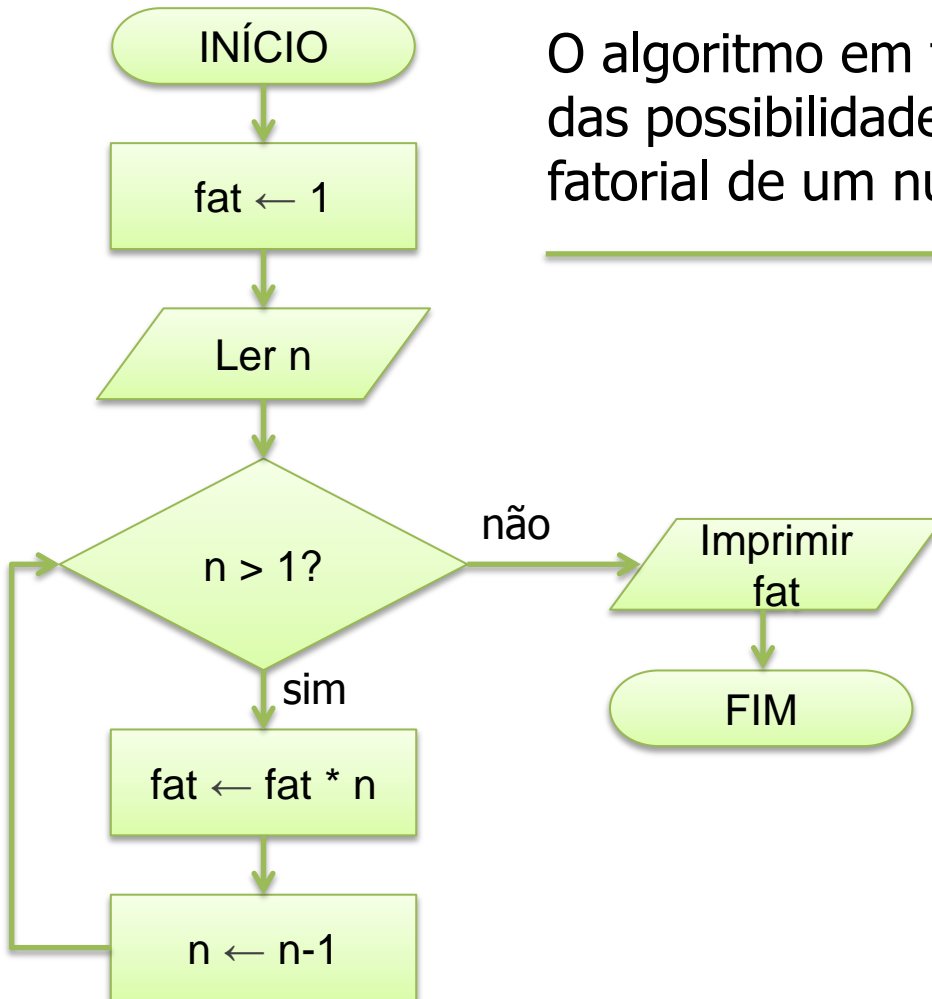
```
1. Principal ()
2. Início
3.   i ← 0;
4.   Enquanto (i < 10)
5.   Faça
6.     Comandos ;
7.     i++;
8.   Fim
9. Fim
```

- Repete os comandos enquanto a **condição** for verdadeira
- A condição pode ser qualquer condição lógica
- $i < 10$ é apenas um exemplo

Repetição em C com `While`

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i;
6
7      i = 0;
8      while(i<10) {
9          printf("Iteração: %d\n", i);
10         i++;
11     }
12
13     return 0;
14 }
```

Voltando ao cálculo o fatorial



O algoritmo em fluxograma ao lado, mostra uma das possibilidades de solução de como fazer o fatorial de um número fornecido ao computador

Teste de mesa: seja $n = 4$ o valor fornecido ao computador

fat	n
1	

Proposta

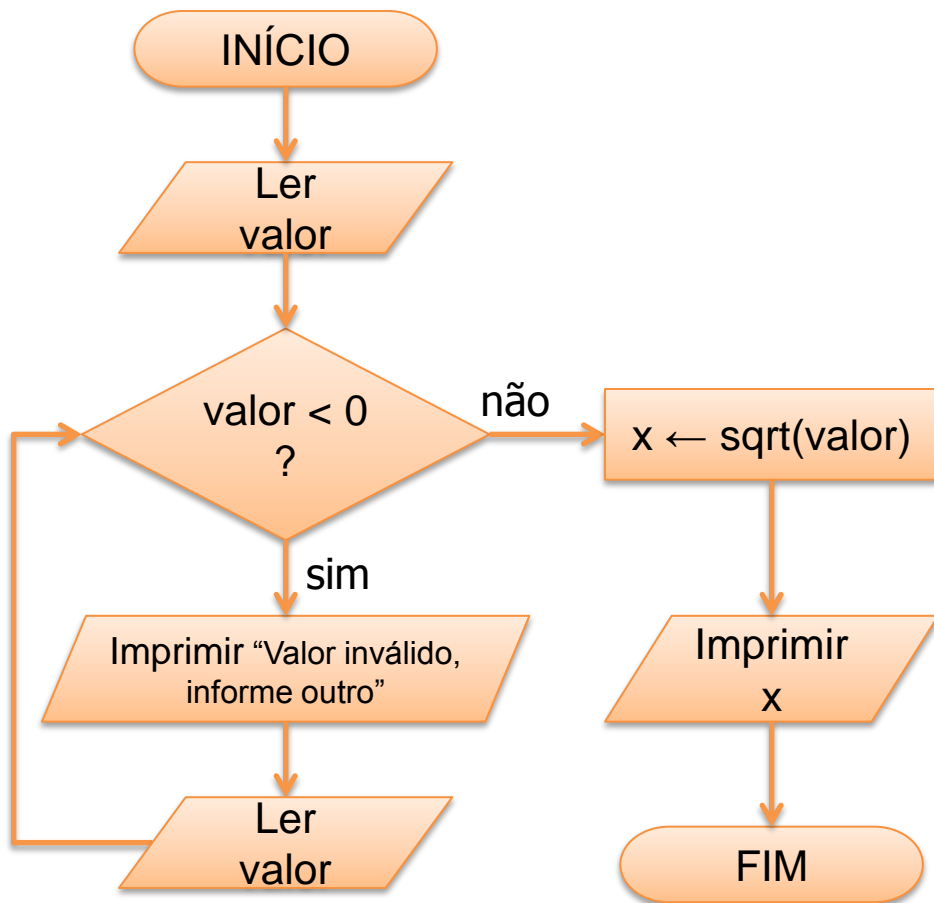
- Tente implementar o exemplo do fatorial em pseudo-código
- Depois em casa tente fazer o mesmo exemplo em linguagem C

Outro exemplo de problema com repetição

- Faça um algoritmo que receba um valor e imprima a sua raiz quadrada
- O programa deve aceitar apenas valores de entrada **positivos**
- Caso o usuário digite um valor negativo deve ser solicitado outro valor

Como implementar essa consistência?

Usando repetição



- Condição de parada
 - $valor \geq 0$
- Variável de controle
 - $valor$
- Note que, uma vez dentro do laço, a variável $valor$ precisa ser alterada para que o algoritmo saia da repetição
- Nesse caso a variável de controle é **alterada pelo usuário**, não por um incremento

Exemplos com pseudo-código

Quais são as saídas geradas?

```
1. Programa ()
2. Início
3.   a ← 0;
4.   Enquanto (a < 5)
5.     Faça
6.       Imprime a;
7.       a ← a + 1;
8.     Fim
9.   Imprime a;
10. Fim
```

```
1. Programa ()
2. Início
3.   a ← 1;
4.   Enquanto (a <= 5)
5.     Faça
6.       a ← a + 1;
7.       Imprime a;
8.     Fim
9.   Imprime a;
10. Fim
```

Exemplos com pseudo-código

Quantas vezes os laços repetem?

```
1. Programa ()
2. Início
3.   y ← 0;
4.   Enquanto (y <= 10)
5.     Faça
6.       Imprime y;
7.       y ← y + 2;
8.     Fim
9.   Imprime y;
10. Fim
```

```
1. Programa ()
2. Início
3.   x ← 10;
4.   Enquanto (x < 10)
5.     Faça
6.       x ← x - 1;
7.       Imprime x;
8.     Fim
9. Fim
```


Exemplos com pseudo-código

Quantas vezes os laços repetem?

```
1. Programa ()
2. Início
3.   op ← 's' ;
4.   Enquanto (op != 'n' )
5.     Faça
6.       Lê val;
7.       Imprime val;
8.       Imprime "Deseja
          digitar outro valor";
9.       Lê op;
10.  Fim
11. Fim
```

- E esse laço?
- Quantas vezes repete?

Exemplos com pseudo-código

Quantas vezes os laços repetem?

```
1. Programa ()
2. Início
3.   op ← 's' ;
4.   Enquanto (op != 'n')
5.     Faça
6.       Lê val;
7.       Imprime val;
8.       Imprime "Deseja
          digitar outro valor";
9.       Lê op;
10.  Fim
11. Fim
```

- E esse laço?
- Quantas vezes repete?

Nem sempre se sabe de antemão quantas vezes um laço vai executar