

# Estrutura de controle repetitiva: Para (*for*)

Patrícia de Siqueira Ramos

UNIFAL-MG, *campus* Varginha

6 de Abril de 2020

# Estrutura de repetição Para

- Estrutura de repetição que utiliza um contador que já predefine quantas vezes os comandos serão executados
- A sintaxe utilizada no pseudocódigo é a seguinte (exemplo que repete o comando `Escreva(i)` 5 vezes):

```
Para i de 1 até 5 faça  
    Escreva(i)  
FimPara
```

# Estrutura Para no Python: for

Imprimir 5 números na tela. Veja a diferença entre as três formas:

```
for i in range(5):  
    print(i)
```

```
for i in range(5):  
    print(i + 1)
```

```
for i in range(1, 6):  
    print(i)
```

# Estrutura Para no Python: for

Imprimir 5 números na tela. Veja a diferença entre as três formas:

```
for i in range(5):  
    print(i)
```

```
for i in range(5):  
    print(i + 1)
```

```
for i in range(1, 6):  
    print(i)
```

- *O Python sempre inicia a contagem com 0*
- *O Python considera o intervalo fechado à esquerda e aberto à direita: [3,6) mostra de 3 a 5, por exemplo)*

# Exemplo de Para

Início

Inteiro: i, n

n = 5

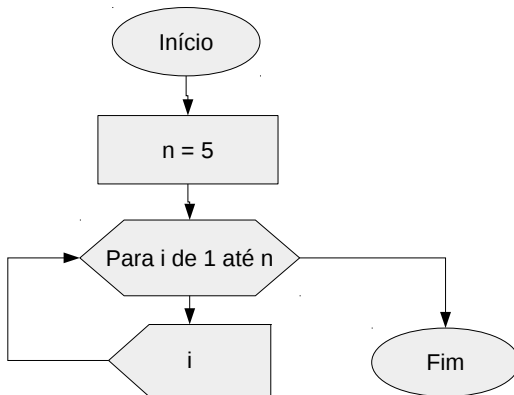
Para i de 1 até n faça

    Escreva(i)

FimPara

Fim

# Exemplo de Para



## Ex.: Potências de 2 usando Para

Início

    Inteiro: j

    Para j de 0 até 10 faça

        Escreva(2 \*\* j)

    FimPara

Fim

# Semelhança entre Para e Enquanto

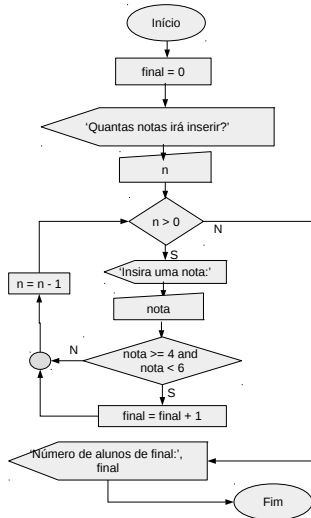
- Em várias situações em que usamos o Enquanto é possível usar o Para
- Essa conversão é possível quando sabemos de antemão quantas vezes as instruções devem ser executadas
- Já em outras situações, em que não sabemos quantas vezes as instruções serão repetidas, é mais recomendável utilizar o Enquanto (contar dígitos de um número, se o usuário tiver que ir inserindo números enquanto eles forem pares etc.)



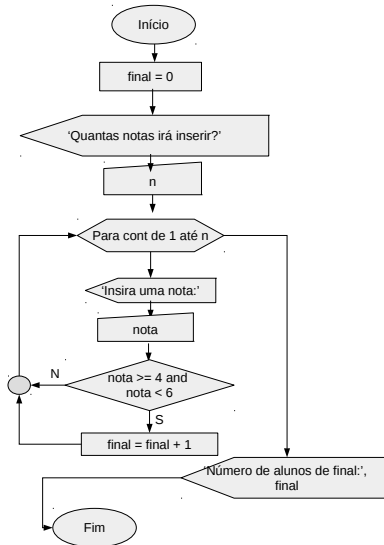
# Exemplo 1

Dados um número inteiro  $n > 0$  e as notas finais de  $n$  alunos, determinar quantos alunos ficaram de final. Um aluno está de final se sua média estiver no intervalo  $[4.0; 6.0)$ .

# Exemplo 1 usando Enquanto



# Exemplo 1 usando Para



## Exemplo 2

- Ler um número inteiro positivo  $n$  e verificar se ele é primo.
- Se o número for primo, imprime 'primo'. Caso contrário, imprime 'não primo'.
- Nota: um número é primo se ele só possui dois divisores: o 1 e o próprio número.
- Assim, uma solução seria verificar o resto da divisão por todos os números de 1 até  $n$  e utilizar uma variável que conte quantos divisores esse número possui. Se ele possuir apenas dois, ele é primo.

## Exemplo 2 usando Enquanto

Início

Inteiro: n, i, div

div = 0

i = 1

Escreva('Digite o número')

Leia(n)

Enquanto i <= n faça

Se n % i == 0 Então

div = div + 1

FimSe

i = i + 1

FimEnquanto

Se div == 2 Então

Escreva('O número', n, 'é primo')

Senão

Escreva('O número', n, 'não é primo')

FimSe

Fim

## Exemplo 2 usando Para

Início

Inteiro: n, i, div

div = 0

Escreva('Digite o número')

Leia(n)

Para i de 1 até n faça

Se  $n \% i == 0$  Então

div = div + 1

FimSe

FimPara

Se div == 2 Então

Escreva('O número', n, 'é primo')

Senão

Escreva('O número', n, 'não é primo')

FimSe

Fim

# Exercício 1

Faça um algoritmo em pseudocódigo que receba a idade de 10 pessoas e informe quantas são maiores de 18 anos.

## Exercício 2

A série de RICCI difere da série de FIBONACCI porque os dois primeiros termos são fornecidos pelo usuário. Os demais termos são gerados da mesma forma que a série de FIBONACCI.

Criar um algoritmo em fluxograma que imprima os N primeiros termos da série de RICCI.



## Exercício 3

Modifique o exercício 2 de forma a garantir que o primeiro termo inserido seja menor do que o segundo (os dois também não devem ser iguais).

Há duas opções:

- verificar se o primeiro é menor do que o segundo número. Se não for, trocar os dois, forçando o primeiro a ser menor
- usar uma estrutura de repetição que vá pedindo para o usuário inserir dois números até que ele insira o primeiro menor do que o segundo