Estrutura de controle repetitiva: `Enquanto`

Patrícia de Siqueira Ramos

UNIFAL-MG, *campus* Varginha

25 de Outubro de 2017

- Imagine que queremos imprimir todas as potências de 2 ($2^{**}0, 2^{**}1, \cdots, 2^{**}10$) na tela
- Faríamos algo do tipo:

```
Início
    Escreva(2 ** 0)
    Escreva(2 ** 1)
    Escreva(2 ** 2)
    .
    .
    .
    Escreva(2 ** 10)
Fim
```

- Mas, dessa forma seria muito trabalhoso

Estruturas repetitivas

- Quando queremos repetir uma mesma ação várias vezes (ou com poucas modificações) podemos usar a estrutura repetitiva
- As estruturas repetitivas mais utilizadas são:
  - `enquanto` (*while*)
  - `repita` (*repeat*)
  - `para` (*for*)
- Primeiro veremos o `enquanto`: enquanto uma condição permanece verdadeira, as ações são executadas e, quando ela se torna falsa, o comando é abandonado

Sintaxe do Enquanto

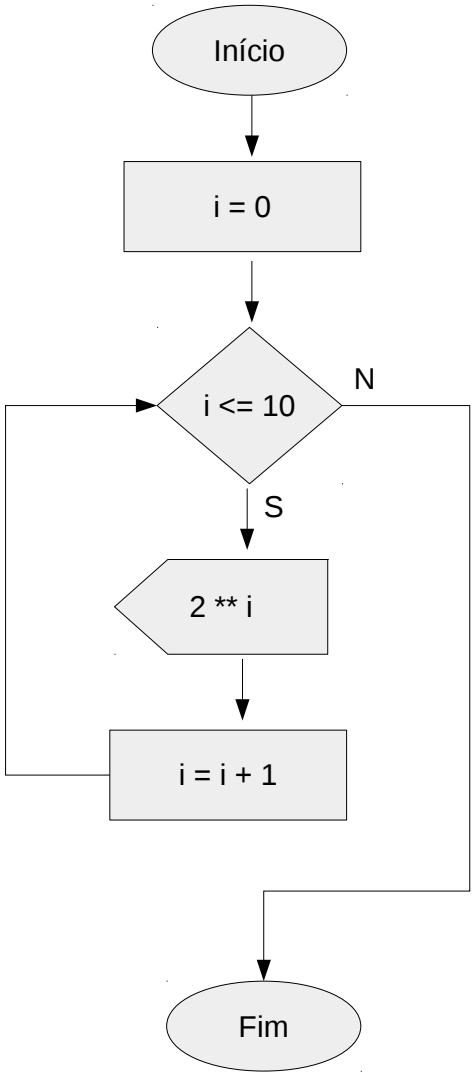
```
Enquanto <condição a ser avaliada> faça
    comando 1
    comando 2
    .
    .
    .
FimEnquanto
```

Ex.: Potências de 2 usando Enquanto

```
Início
    Inteiro: i
    i = 0
    Enquanto i <= 10 faça
        Escreva(2 ** i)
        i = i + 1
    FimEnquanto
Fim
```

Obs.: Como a condição é avaliada antes de entrar no *loop* (laço), o comportamento é parecido com o Se Então

Ex.: Potências de 2 usando `Enquanto`



Situações a se pensar

a) se a condição for falsa da primeira vez?

```
i = 0
Enquanto i != 0 faça
  Escreva(i)
  i = i + 1
FimEnquanto
```

b) se a condição for sempre verdadeira?

```
i = 0
Enquanto i >= 0 faça
    Escreva(i)
    i = i + 1
FimEnquanto
```

Exercício

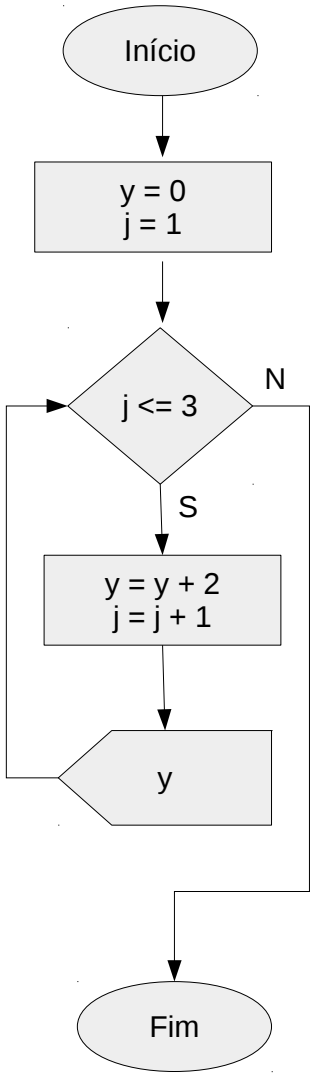
- a) Qual será a saída do seguinte algoritmo?

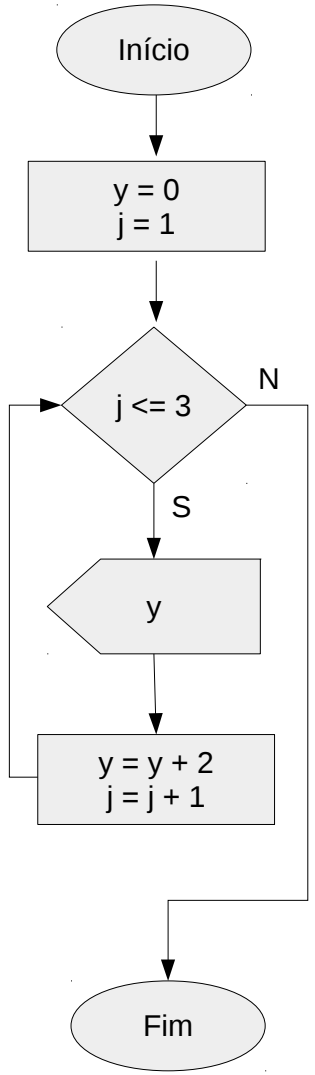
```
Início
    Inteiro: x
    x = 1
    Enquanto x < 6 faça
        Escreva(x)
        x = x + 1
    FimEnquanto
Fim
```

- b) Modifique-o para que o usuário defina o último número a apresentar.

Diferentes saídas do Enquanto

Quais serão as saídas dos dois fluxogramas?





O que esse algoritmo faz? Quando ele pára?

```
Início
    Real: z
    Escreva('Insira um valor:')
    Leia(z)
    Enquanto z >= 0 faça
        Escreva('Raiz =', sqrt(z))
        Escreva('Insira um valor:')
        Leia(z)
    FimEnquanto
Fim
```

Exercício 1

Implemente um algoritmo que faça o seguinte: recebe uma sequência de números terminada em zero informada pelo usuário e retorna a soma dos números. A sequência pode ser de qualquer tamanho.
Por exemplo:
- se o usuário inserir

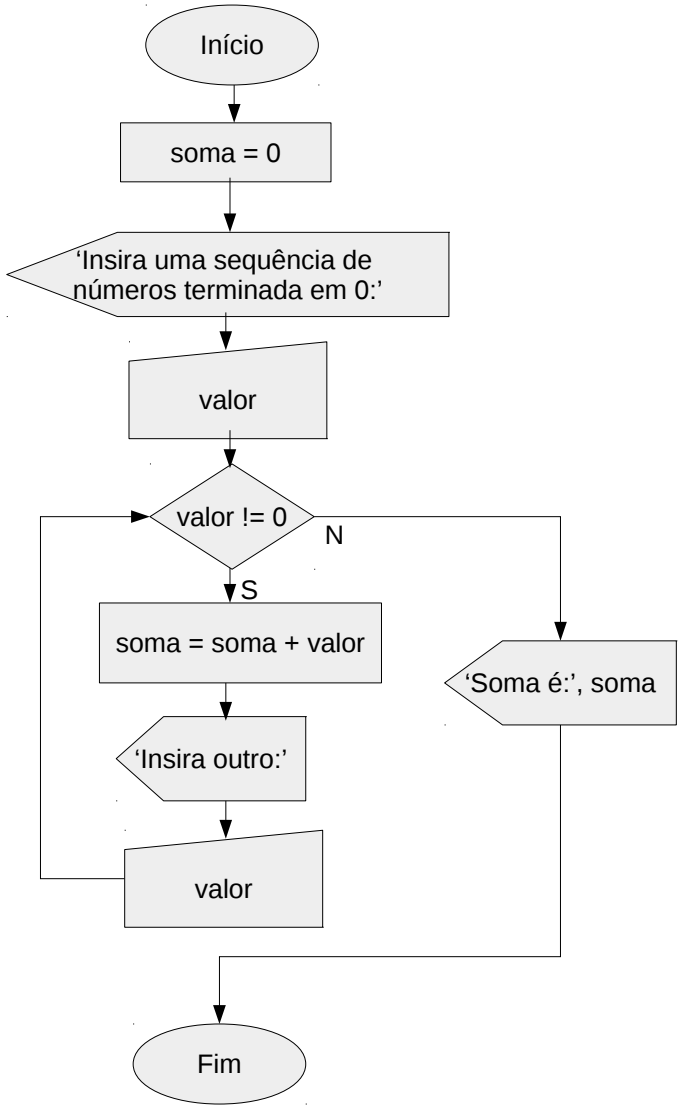
2, 5, 7, -3, 4, 5, 0

o algoritmo deve retornar que a soma é 20
- se o usuário inserir

0

o algoritmo deve retornar 0

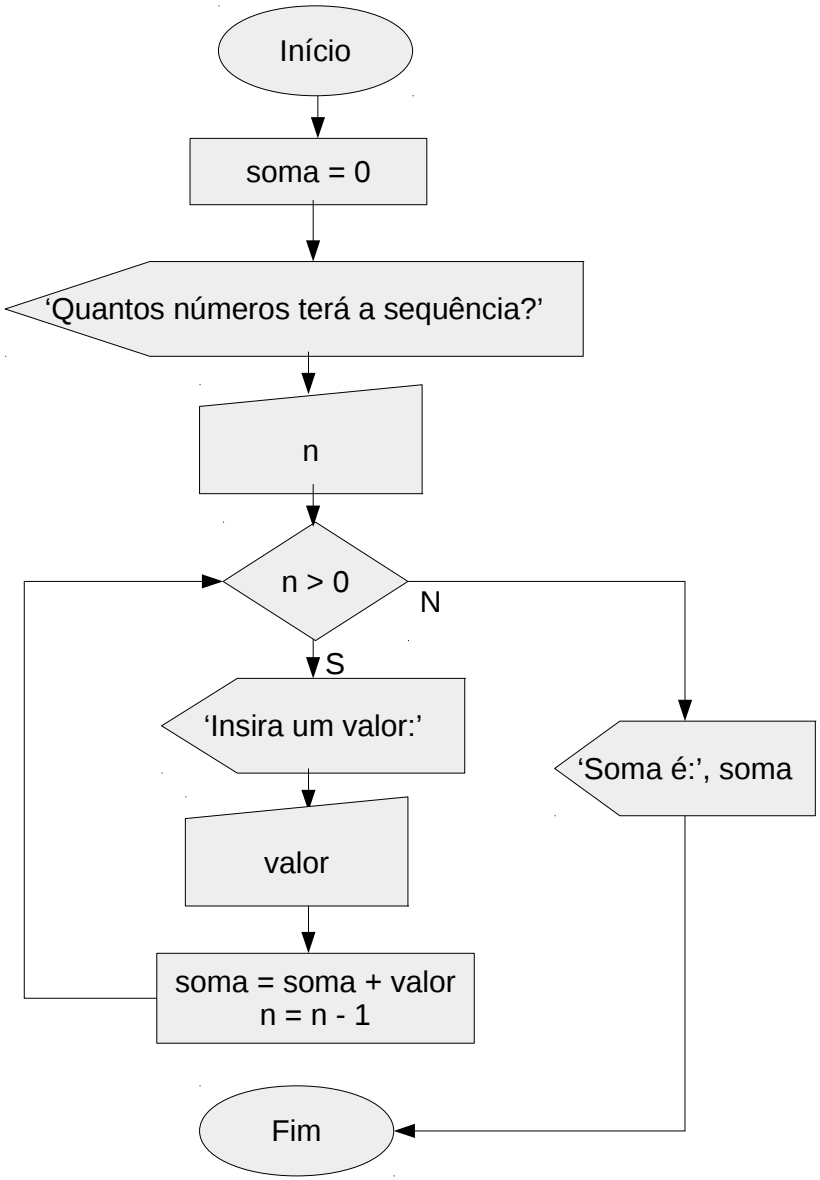
Exercício 1: soma de sequência terminada em 0



Exercício 2

Modifique o algoritmo anterior para que o usuário informe, no início, quantos números vai inserir. Depois ele insere os valores (sem o zero no fim) e o algoritmo retorna a soma.

Exercício 2: soma de sequência (n definido)



Exercício 3

Implemente um algoritmo que faça o seguinte: recebe uma sequência de números terminada em zero informada pelo usuário e retorna o produto dos números, desconsiderando o zero.
Por exemplo:
- se o usuário inserir

3, 2, -3, 0

o algoritmo deve retornar que o produto é -18
- se o usuário inserir

0

o algoritmo deve retornar 0

Exercício 3: produto de sequência terminada em 0

```
Início
    Real: valor, produto
    Escreva('Insira uma sequência terminada por 0:')
    Leia(valor)
    Se valor == 0 Então
        produto = 0
    Senão
        produto = 1
        Enquanto valor != 0 faça
            produto = produto * valor
            Escreva('Insira um valor:')
            Leia(valor)
        FimEnquanto
    FimSe
    Escreva('O produto é', produto)
Fim
```

Desafio

Usando a estrutura `Enquanto`, somar os dígitos de um número inteiro inserido pelo usuário. Por exemplo: se o usuário inserir 25332, o algoritmo deve retornar 15, que é a soma dos dígitos (2 + 5 + 3 + 3 + 2).