

Estruturas de Controle de Fluxo

Em certos algoritmos percebemos que a execução de determinados trechos se faz necessária algumas vezes.

Um exemplo que nós já tratamos é o caso do cálculo da média aritmética entre alguns valores. Vimos que a operação de leitura é repetida de acordo com o número de valores que servirão de base para o cálculo da média. Outra forma de verificar a necessidade de repetição de um determinado trecho do algoritmo é imaginar que ao invés de se desejar apenas obter uma média, o objetivo fosse obter um conjunto de médias. Para o cálculo de cada uma das médias o mesmo conjunto de operações seria executado.

Observações como estas fomentaram a criação das estruturas de repetição as quais veremos a seguir.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – enquanto

Sintaxe:

...

```
enquanto (<expressão-lógica>) faça  
    <sequência-de-comandos>  
fimenquanto
```

...

Obs.: As instruções contidas no enquanto serão executadas durante o tempo em que o resultado da avaliação da expressão lógica resultar verdadeiro. O fato da avaliação da expressão lógica encontrar-se no início do laço faz com que a sequência de comandos só venha a ser executada se ao menos uma vez a avaliação da expressão resultar em verdadeiro.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – enquanto (continuação)

Exemplo:

O pseudocódigo e os fluxogramas a seguir representam algoritmos que escrevem na saída padrão os números inteiros contidos no intervalo $[1, 10]$.

Estruturas de Controle de Fluxo

algoritmo "exemplo 1 laço enquanto"

var

 valor: inteiro

inicio

 valor <- 1

 enquanto (valor <= 10) faça

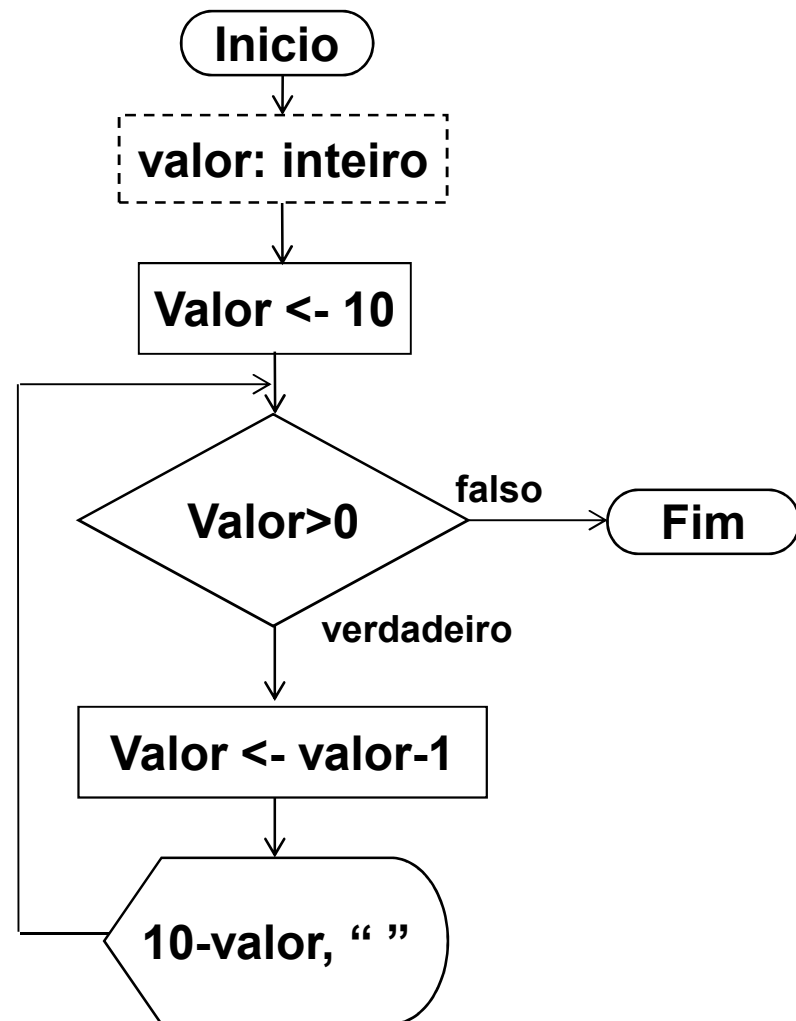
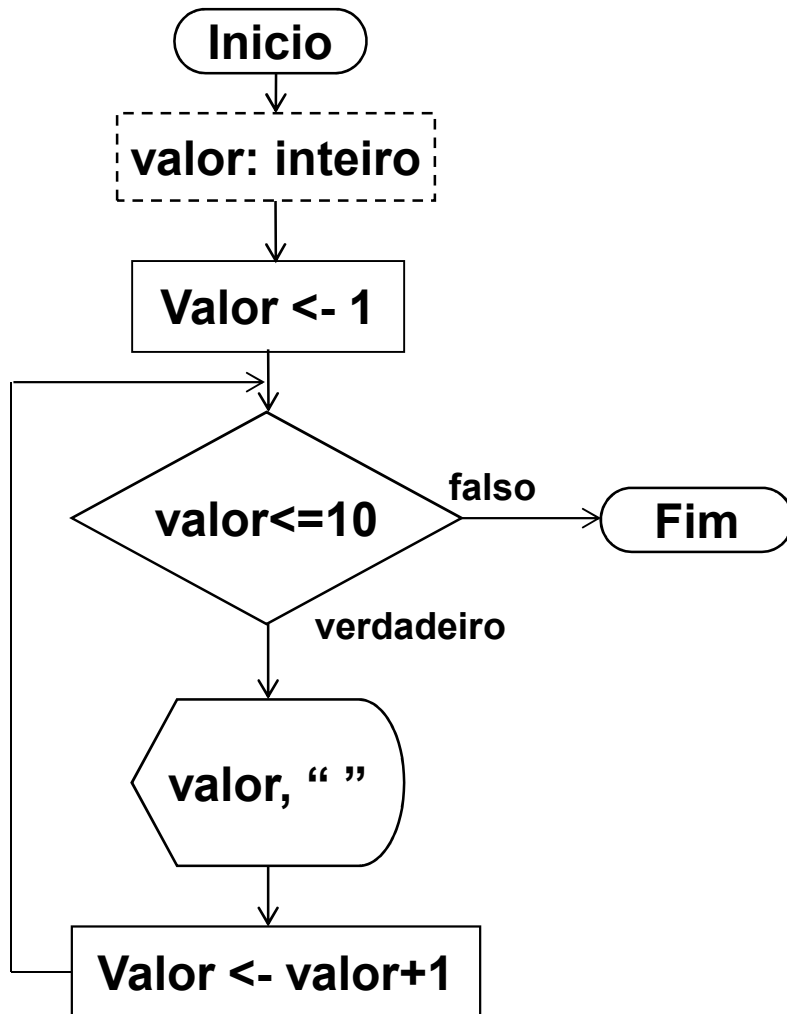
 escreval (valor)

 valor <- valor+1

 fimenquanto

fimalgoritmo

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

2. Estrutura ou laço de repetição – enquanto (continuação)

Exemplo:

Dada uma reta $ax+by+c=0$ e cinco pontos, faça um algoritmo para calcular, para cada ponto, o seguinte: se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Estruturas de Controle de Fluxo

algoritmo "exemplo 2 laço enquanto "

var a,b,c,x,y: real

contador: inteiro

inicio

contador <- 1

escreval ("Equação da reta: $ax+by+c=0$ ")

escreva ("Coeficiente a da reta = ")

leia (a)

escreva ("Coeficiente b da reta = ")

leia (b)

escreva ("Coeficiente c da reta = ")

leia (c)


```

enquanto (contador<=5) faca
    escreval ("Coordenadas do ponto ",contador," :")
    escreva ("Coordenada x do ponto = ")
    leia (x)
    escreva ("Coordenada y do ponto = ")
    leia (y)
    se (x>=0x e yy>=0) entao
        escreval ("A distância do ponto ",contador,
" a reta é: ",((a*x+b*y+c)^2)^0.5/(((a^2)+(b^2))^0.5))
    senao
        escreval ("O ponto não está no primeiro
quadrante! ")
    fimse
    contador <- contador + 1
fimenquanto
fimalgoritmo

```

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – repita

Sintaxe:

...

repita

<sequência-de-comandos>

ate (*<expressão-lógica>*)

...

Obs.: As instruções contidas no repita serão executadas enquanto o resultado da avaliação da expressão lógica resultar em falso. O fato da avaliação da expressão lógica encontrar-se no final do laço faz com que, mesmo no caso da expressão lógica nunca resultar em falso, a sequência de comandos seja executada ao menos uma vez.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – repita (continuação)

Exemplo 5:

O pseudocódigo e os fluxogramas a seguir representam algoritmos que escrevem na saída padrão os números inteiros contidos no intervalo $[1, 10]$.

Estruturas de Controle de Fluxo

algoritmo "exemplo 5"

var

 valor: inteiro

inicio

 valor <- 0

 repita

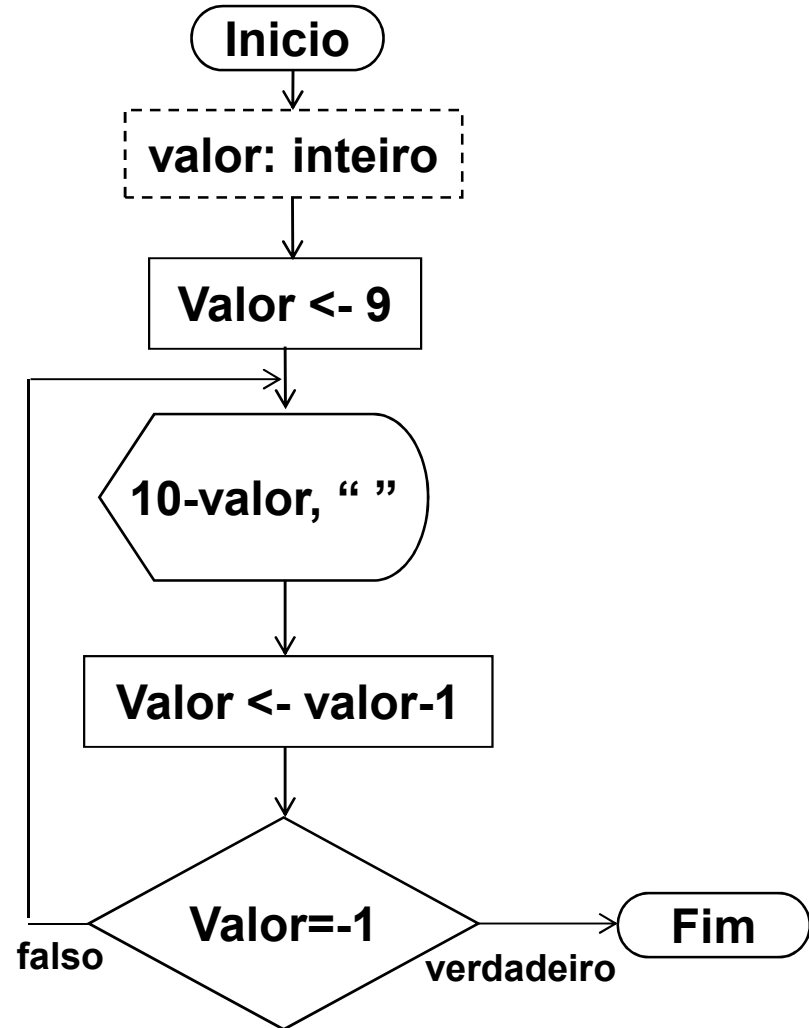
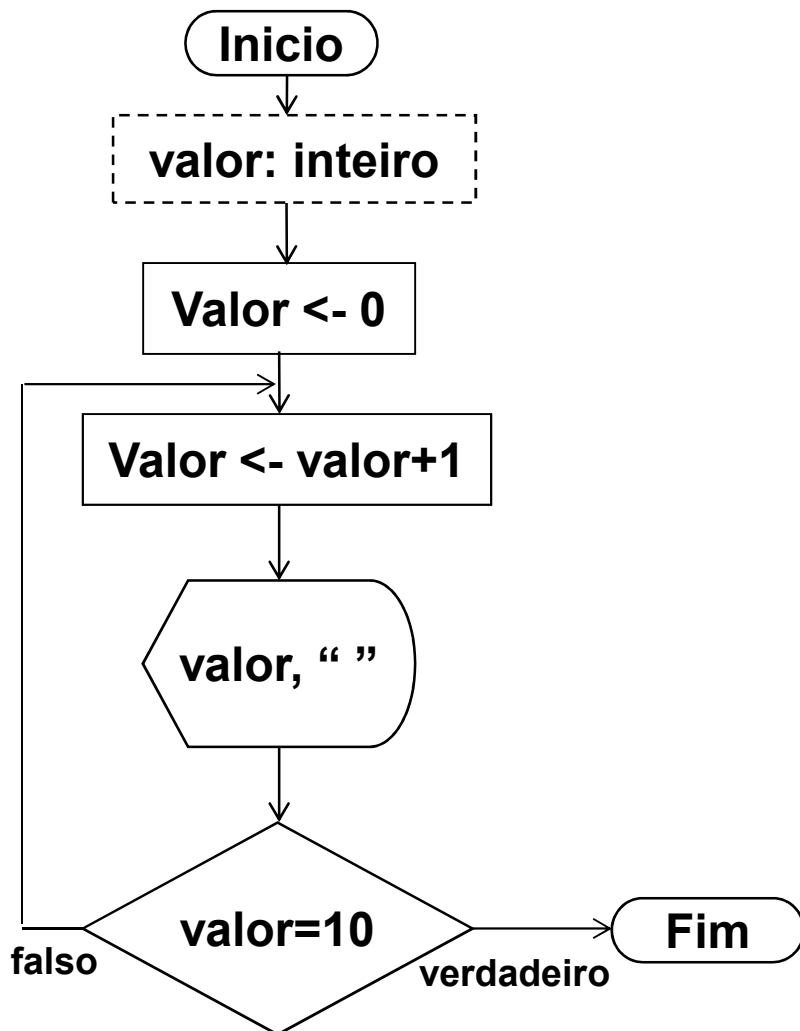
 valor <- valor+1

 escreval (valor)

 ate (valor = 10)

fimalgoritmo

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – **repita** (continuação)

Exemplo:

Dada uma reta $ax+by+c=0$ e cinco pontos, faça um algoritmo para calcular, para cada ponto, o seguinte: se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Estruturas de Controle de Fluxo

algoritmo "exemplo 6"

var a,b,c,x,y: real

contador: inteiro

inicio

contador <- 1

escreval ("Equação da reta: $ax+by+c=0$ ")

escreva ("Coeficiente a da reta = ")

leia (a)

escreva ("Coeficiente b da reta = ")

leia (b)

escreva ("Coeficiente c da reta = ")

leia (c)

repita

escreval ("Coordenadas do ponto ",contador," :")

escreva ("Coordenada x do ponto = ")

leia (x)

escreva ("Coordenada y do ponto = ")

leia (y)

se ((x>=0) e (y>=0)) entao

escreval ("A distância do ponto ",contador,
" a reta é: ",
 $((a*x+b*y+c)^2)^{0.5}/(((a^2)+(b^2))^{0.5})$)

senao

escreval ("O ponto ",contador," não está no
primeiro quadrante! ")

fimse

contador <- contador + 1

ate (contador>5)

fimalgoritmo

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição

Exercício 15:

Faça um algoritmo que recebe números naturais fornecidos pelo usuário, quando o usuário quiser parar a execução do algoritmo, o mesmo fornecerá um número negativo. O algoritmo deve retornar, ao final de seu processamento, a quantidade de números naturais fornecida pelo usuário. Fazer dois algoritmos utilizando em cada um, uma das estruturas de repetição vistas. Os algoritmos desenvolvidos devem ser representados através de um pseudocódigo e de um fluxograma.

Estruturas de Controle de Fluxo

algoritmo "exercício 15 laço de repetição repita a"

var num, contador: inteiro

inicio

contador <- 0

repita

escreva ("Entre com um número natural (entre com um inteiro negativo para sair): ")

leia (num)

se (num >= 0) entao

contador <- contador + 1

fimse

ate (num < 0)

escreva ("Fora fornecidos " , contador, " números naturais pelo usuário ")

141 fimalgoritmo

Estruturas de Controle de Fluxo

algoritmo " exercício 15 laço de repetição repita b"

var num, contador: inteiro

inicio

contador <- -1

repita

escreva ("Entre com um número natural (entre com um inteiro negativo para sair): ")

leia (num)

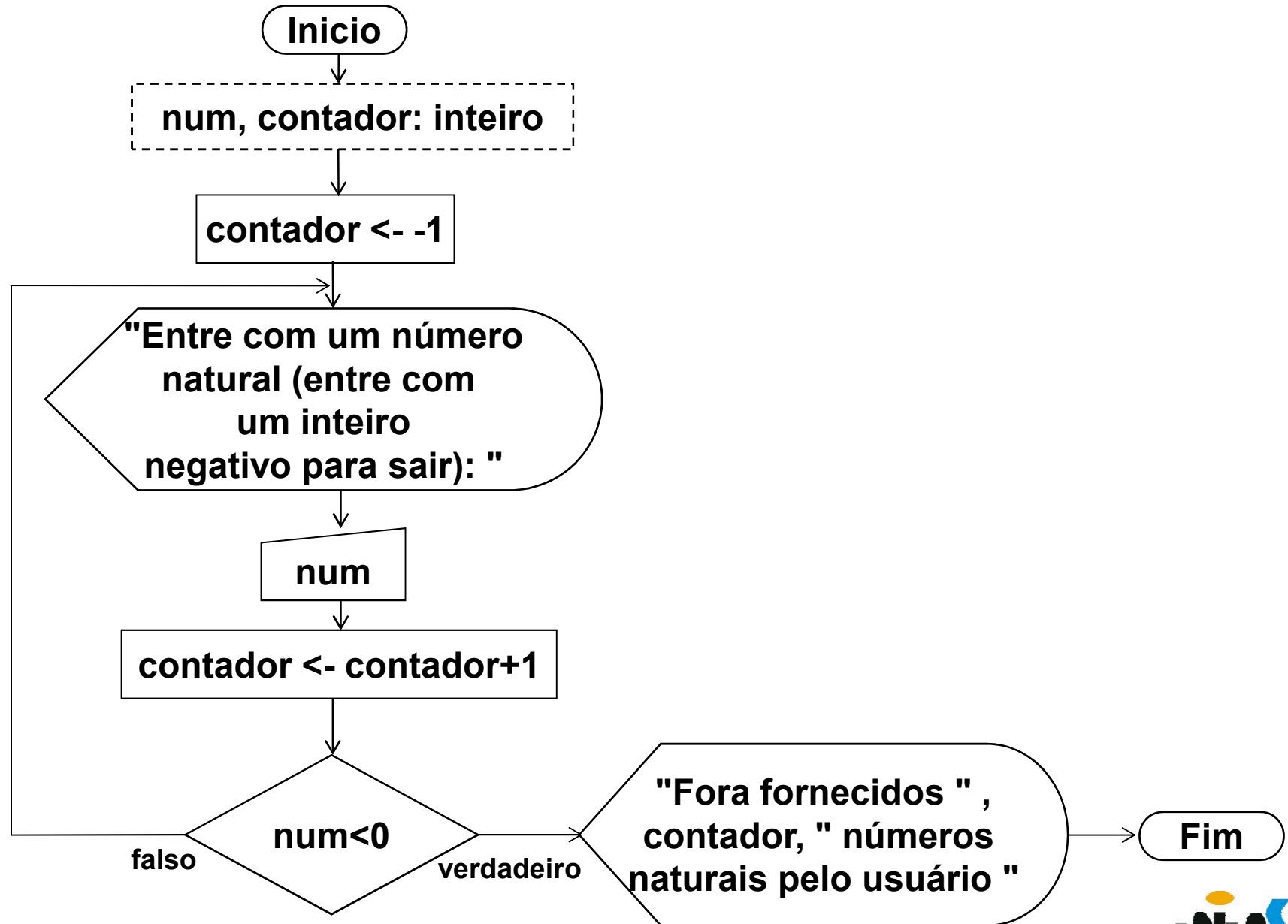
contador <- contador + 1

ate (num<0)

escreva ("Fora fornecidos " ,contador, " números naturais pelo usuário ")

fimalgoritmo

Estruturas de Controle de Fluxo



algoritmo "exercício 15 laço de repetição enquanto a"

var num, contador: inteiro

inicio

contador <- 0

escreva ("Entre com um número natural (entre com um inteiro negativo para sair): ")

leia (num)

enquanto (num >= 0) faça

contador <- contador + 1

escreva ("Entre com um número natural (entre com um inteiro negativo para sair): ")

leia (num)

fimenquanto

escreva ("Fora fornecidos " ,contador, " números naturais pelo usuário")

fimalgoritmo

Estruturas de Controle de Fluxo

algoritmo " exercício 15 laço de repetição enquanto b"

var num, contador: inteiro

Inicio

num <- 1

contador <- -1

enquanto (num>=0) faça

 contador <- contador + 1

 escreva ("Entre com um número natural (entre com um inteiro negativo para sair): ")

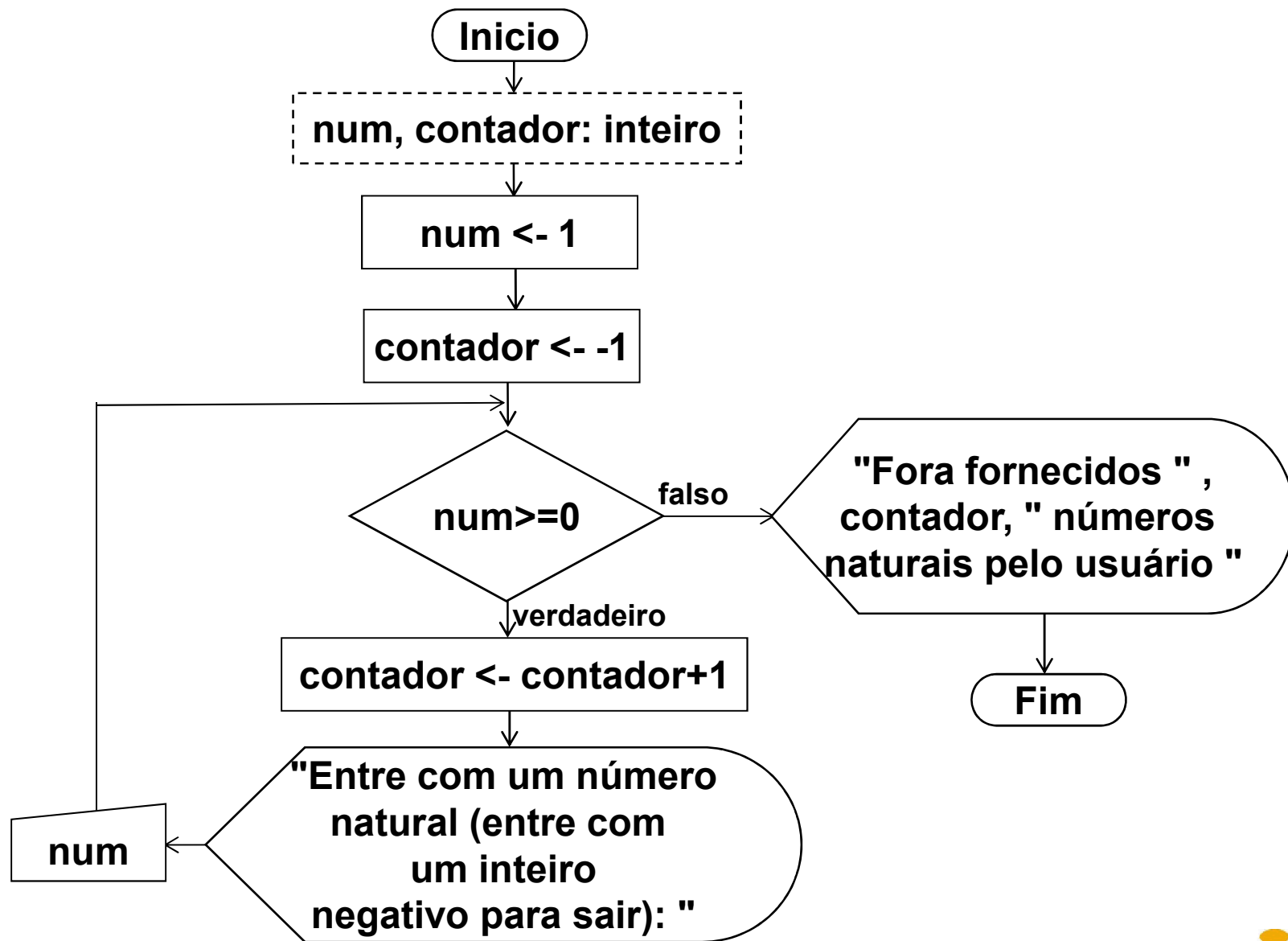
 leia (num)

fimenquanto

escreva ("Fora fornecidos " ,contador, " números naturais pelo usuário")

fimalgoritmo

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

Fluxograma/Exercício 16 – Com base no que foi exposto, construa um fluxograma para obter o resultado da divisão entre dois números. **OBS.:** Caso um dos operandos não seja válido o mesmo deve ser novamente solicitado até um valor válido ser fornecido, ou seja, as entradas devem ser validadas.