

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição

Exercício:

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 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 ")

103 fimalgoritmo

Estruturas de Controle de Fluxo

algoritmo " exercício 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)

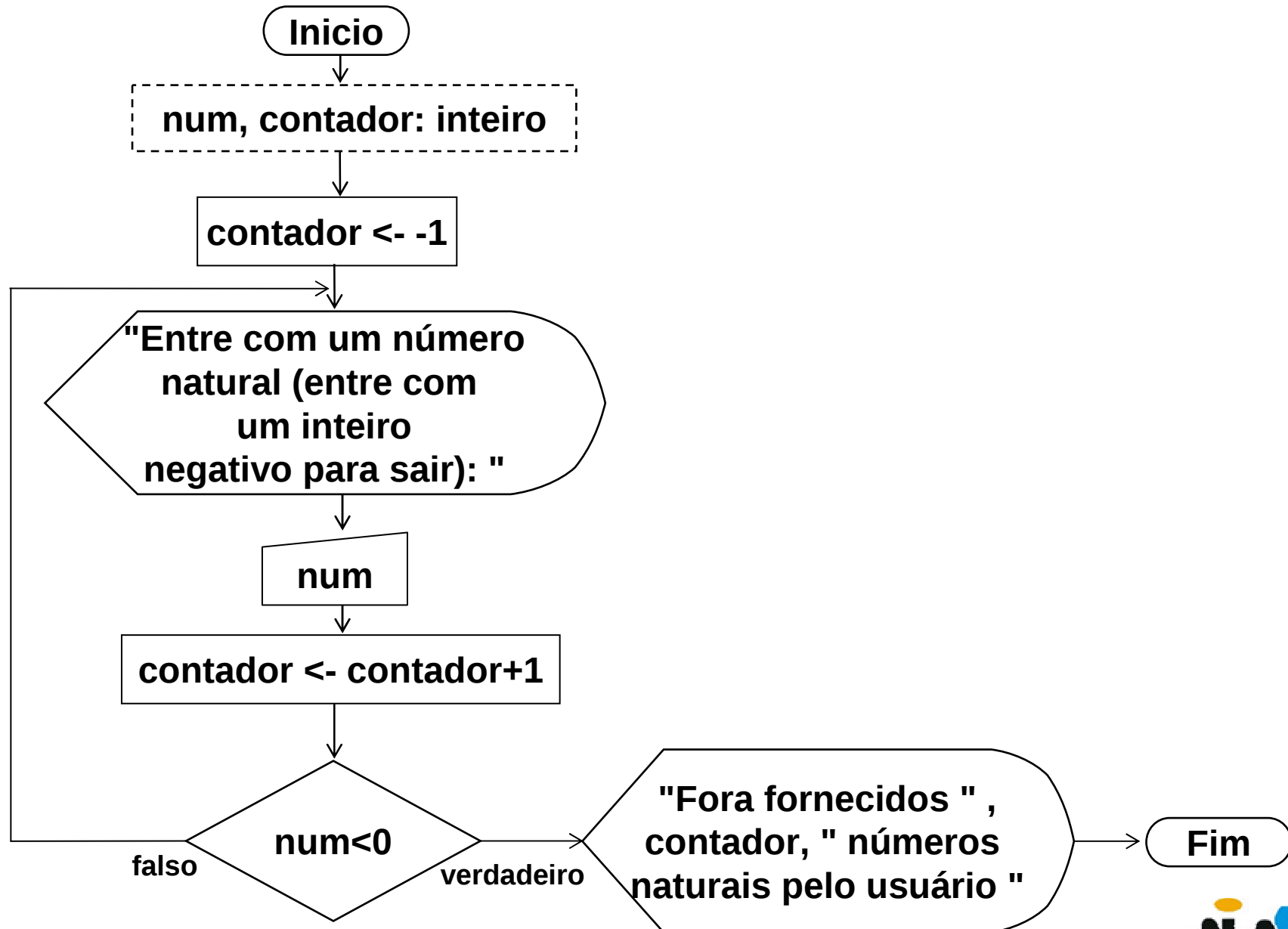
contador <- contador + 1

ate (num<0)

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

fimalgoritmo

Estruturas de Controle de Fluxo



algoritmo "exercício 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 laço de repetição enquanto b"

var num, contador: inteiro

Início

num <- 1

contador <- -1

enquanto (num>=0) faça

 contador <- contador + 1

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

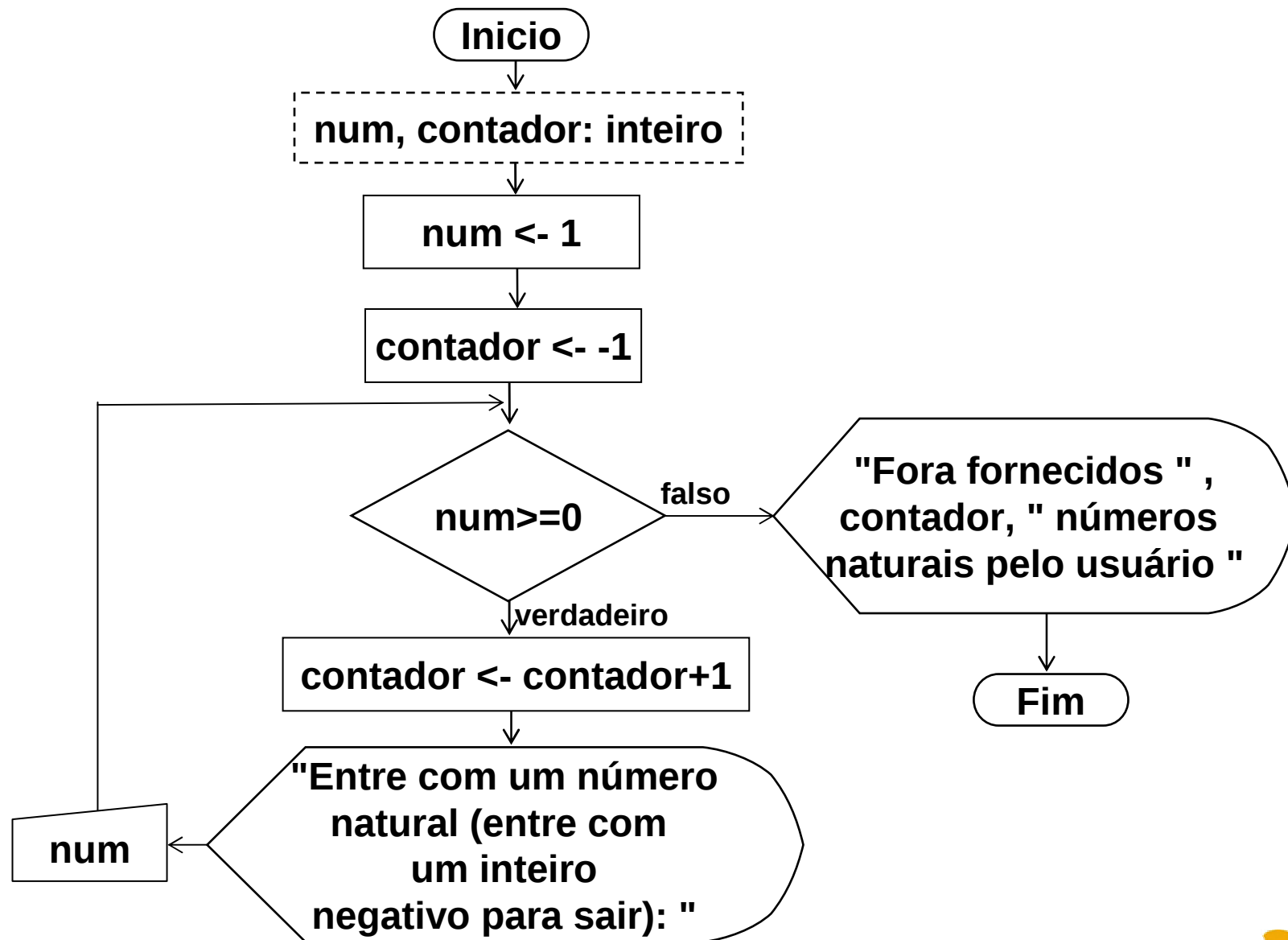
 leia (num)

fimenquanto

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

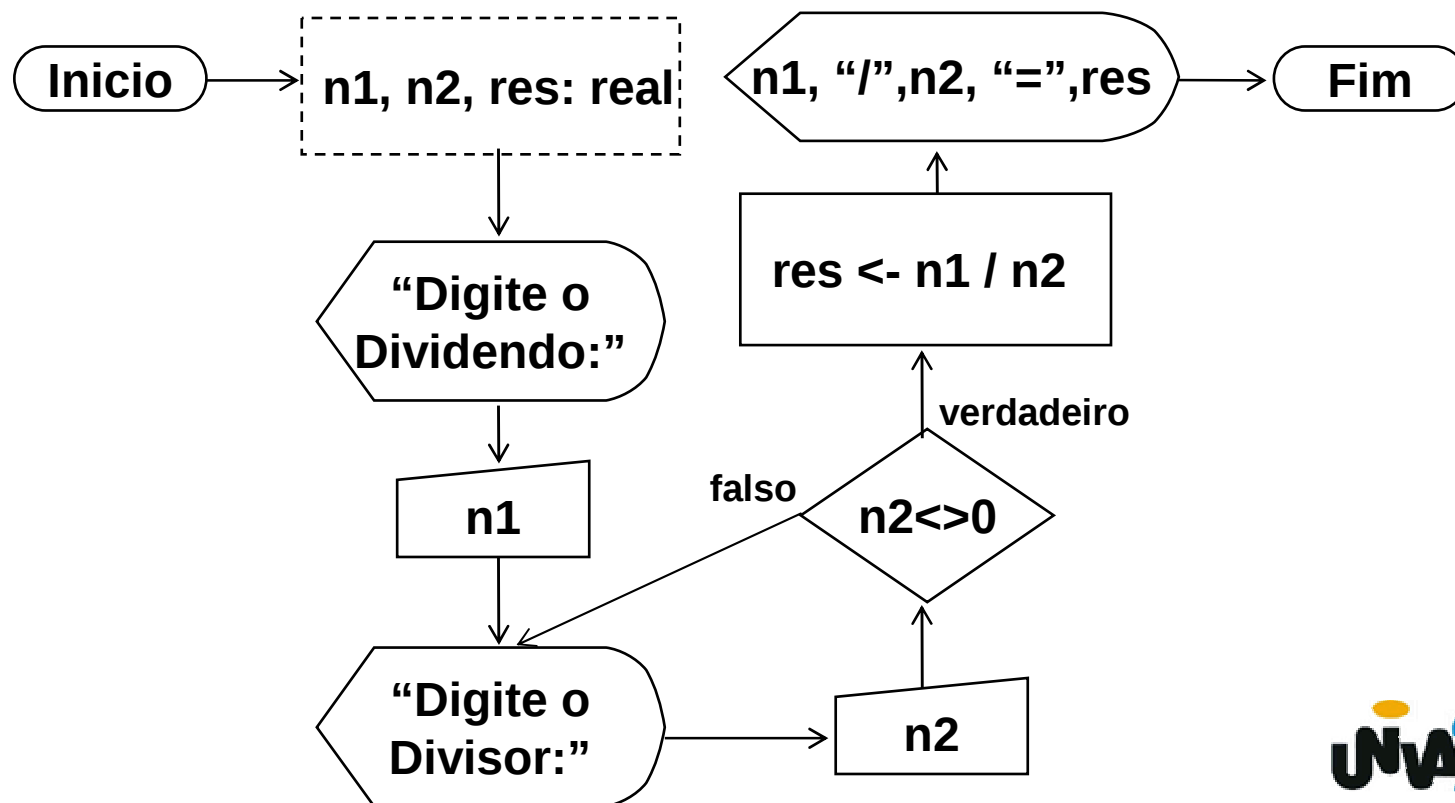
fimalgoritmo

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

Fluxograma/Exercício – 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.



Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição

Ao analisarmos o que ocorre nos laços de repetição estudados, enquanto se repita, perceberemos que, normalmente, ocorre uma inicialização de uma variável, envolvida na expressão lógica que controla o número de repetições, e dentro do laço ocorre uma atualização no valor da variável mencionada, fazendo com que esta venha a tornar o resultado da avaliação da expressão lógica falso, finalizando assim a execução do laço de repetição.

Com base nesta observação foi criado o laço de repetição **para**.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – **para**

Sintaxe:

...

para *<variável>* de *<valor-inicial>* ate *<valor-limite>*
passo <incremento> faca

<seqüência-de-comandos>

fimpara

...

<variável > É a variável contadora do número de repetições do laço (deve ser necessariamente uma variável do tipo inteiro)

Estruturas de Controle de Fluxo

<valor-inicial> É uma expressão que especifica o valor de inicialização da variável contadora.

<valor-limite> É uma expressão que especifica o valor máximo que a variável contadora pode alcançar.

<incremento> É opcional. Quando presente, é precedido pela palavra-reservada *passo*, constitui-se de uma expressão que especifica valor do incremento que será acrescentado à variável contadora em cada repetição do laço. O valor padrão, assumido por omissão, de *<incremento>* é 1. É possível especificar valores negativos para *<incremento>*.

fimpara Indica o fim da seqüência de comandos a serem repetidos. Cada vez que o programa chega neste ponto, é acrescentado à variável contadora o valor de *<incremento>*, e comparado a *<valor-limite>*. Se for menor ou igual (ou maior ou igual, quando *<incremento>* for negativo), a seqüência de comandos será executada mais uma vez; caso contrário, a execução prosseguirá a partir do primeiro comando que esteja após o fimpara.

<valor-inicial>, *<valor-limite>* e *<incremento>* são avaliados uma **única vez** antes da execução da primeira repetição, e **não se alteram durante a execução do laço**, mesmo que variáveis eventualmente presentes nessas expressões tenham seus valores alterados.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição (continuação)

Exemplo:

O pseudocódigo e o fluxograma a seguir escrevem na saída padrão os números inteiros contidos no intervalo $[1, 10]$.

Estruturas de Controle de Fluxo

algoritmo "exemplo"

var

 valor: inteiro

inicio

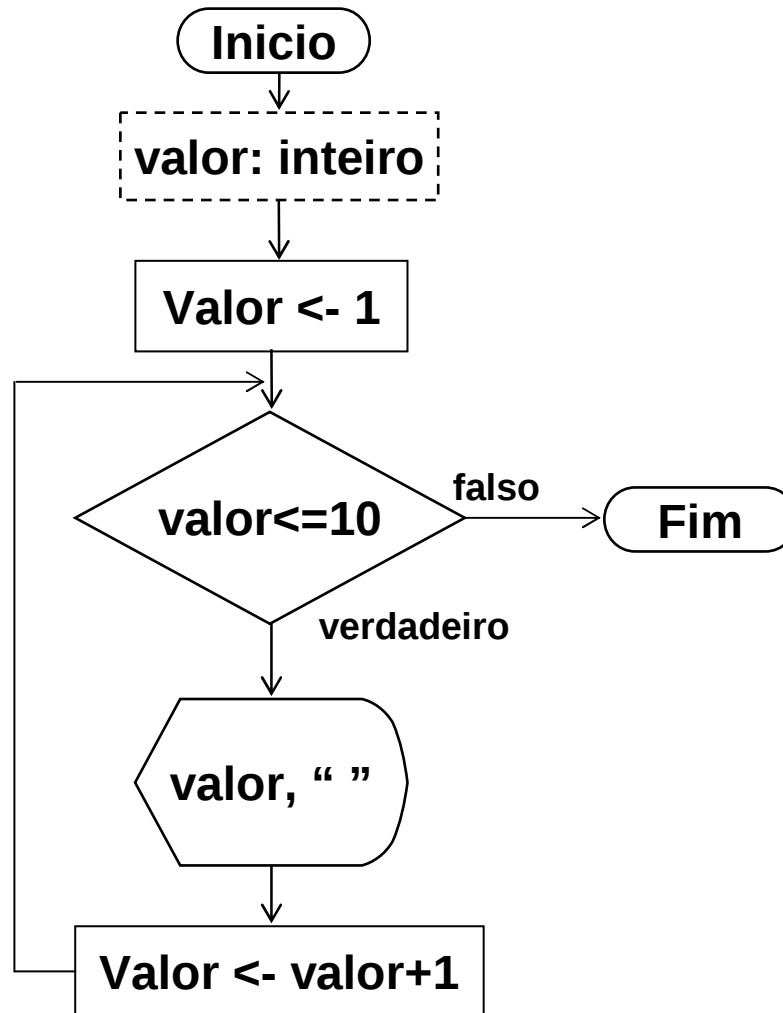
 para valor de 1 ate 10 faca

 escreval (valor)

 fimpara

fimalgoritmo

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição (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"

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

contador: inteiro

inicio

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)

```
para contador de 1 ate 5 faca
    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
fimpara
finalgoritmo
```

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição (continuação)

Exercício:

Construa um pseudocódigo para um algoritmo que exiba em um monitor uma contagem decrescente do valor 30 até o valor 1.

Estruturas de Controle de Fluxo

algoritmo "exercício"

var

 valor: inteiro

inicio

 para valor de 30 ate 1 passo -1 faca

 escreval (valor)

 fimpara

fimalgoritmo

Estruturas de Controle de Fluxo

algoritmo "exercício – resposta alternativa"

var

 valor: inteiro

inicio

 para valor de 0 ate 29 faca

 escreval (30-valor)

 fimpara

fimalgoritmo

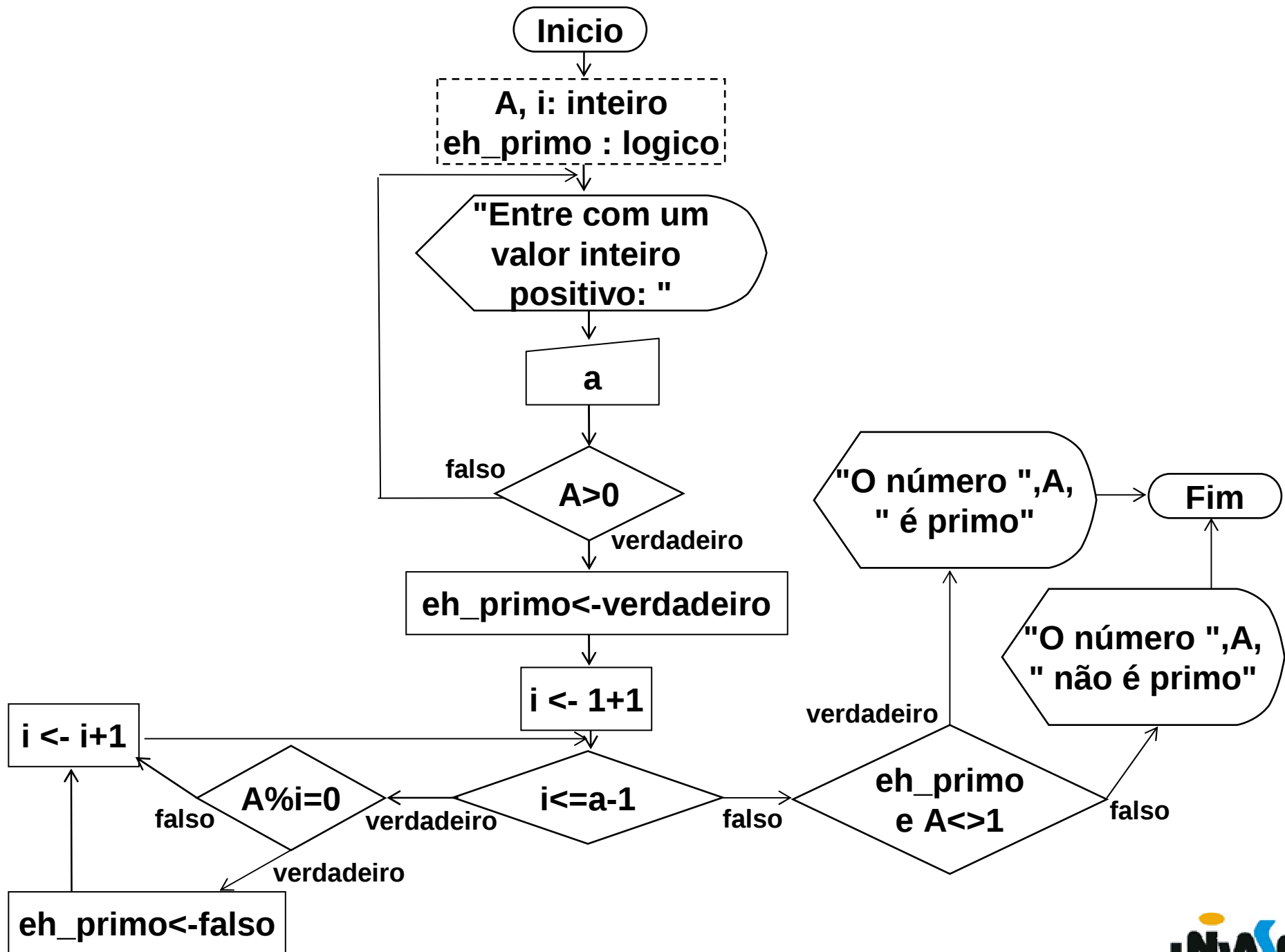
Estruturas de Controle de Fluxo

3. Laços de repetição (continuação)

Exercício:

Construa um algoritmo, representando-o através de um pseudocódigo e de um fluxograma, que leia da entrada padrão um número inteiro positivo e retorne na saída padrão uma mensagem indicando se o número lido é ou não um número primo. As entradas devem ser validadas.

```
algoritmo "exercício"
var   A, i: inteiro
      eh_primo: logico
inicio
  repita
    escreva ("Entre com um valor inteiro positivo: ")
    leia (A)
  ate (A>0)
  eh_primo <- verdadeiro
  para i de 1+1 ate A-1 faca
    se (A%i=0) entao
      eh_primo <- falso
    fimse
  fimpara
  se (eh_primo e A<>1) entao
    escreva ("O número ",A," é primo")
  senao
    escreva ("O número ",A," não é primo")
  fimse
fimalgoritmo
```



```

algoritmo "exercício - usando interrompa"
var   A, i: inteiro
        eh_primo: logico
inicio
    repita
        escreva ("Entre com um valor inteiro positivo: ")
        leia (A)
    ate (A>0)
    eh_primo <- verdadeiro
    para i de 1+1 ate A-1 faca
        se (A%i=0) entao
            eh_primo <- falso
            interrompa //causa uma saída imediata do laço
        fimse
    fimpara
    se (eh_primo e A<>1) entao
        escreva ("O número ",A," é primo")
    senao
        escreva ("O número ",A," não é primo")
    fimse
fimalgoritmo

```

