

# Programação / Programação I

## LEI/1, LTSI/1, LMAT/1

Hugo Pedro Proença

Universidade da Beira Interior  
Departamento de Informática

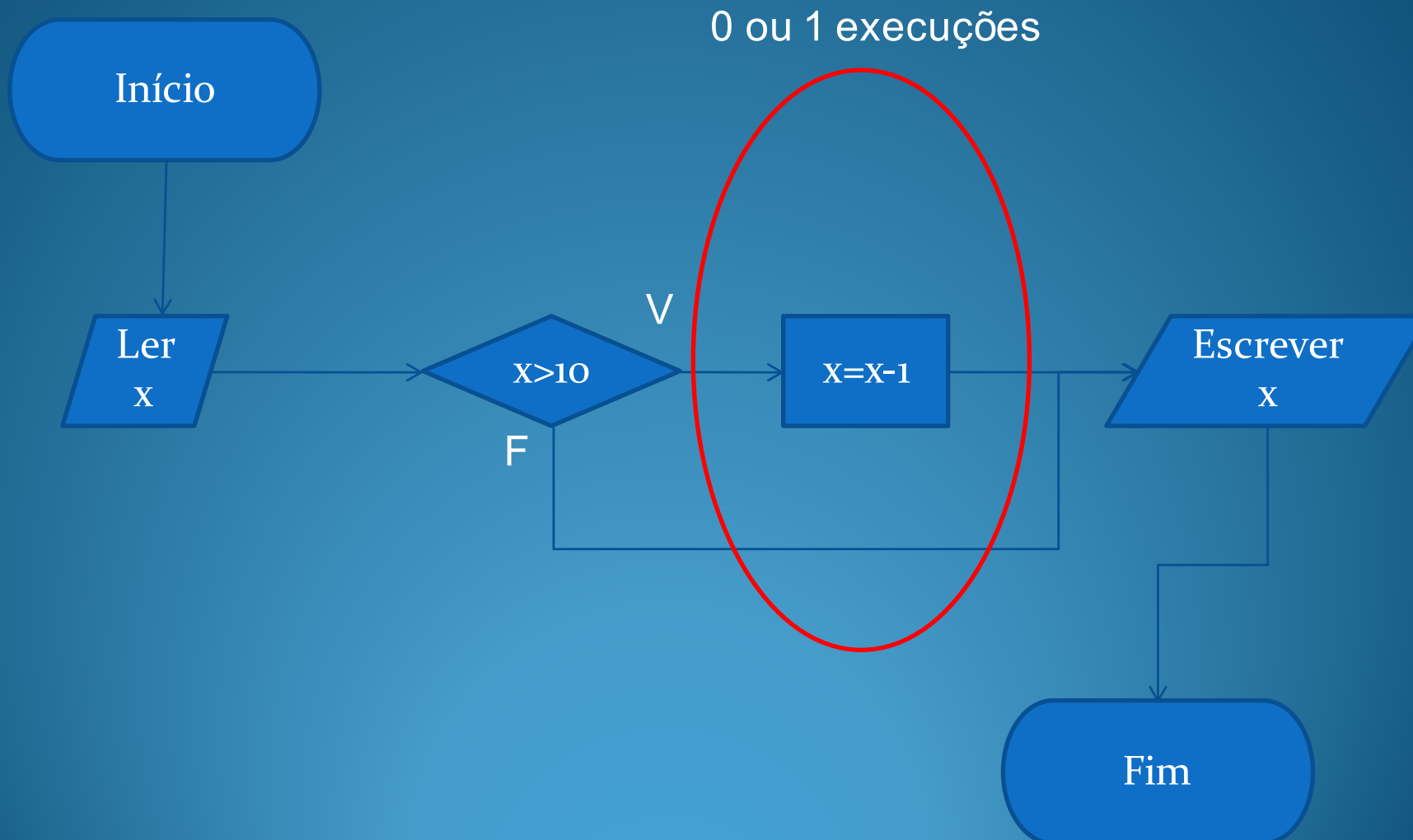
# Resumo

- **Fluxogramas**
  - Blocos iterativos
  - Blocos condicionais
- **Operadores**
  - Aritméticos
  - Relacionais
  - Lógicos
- **Exercícios**

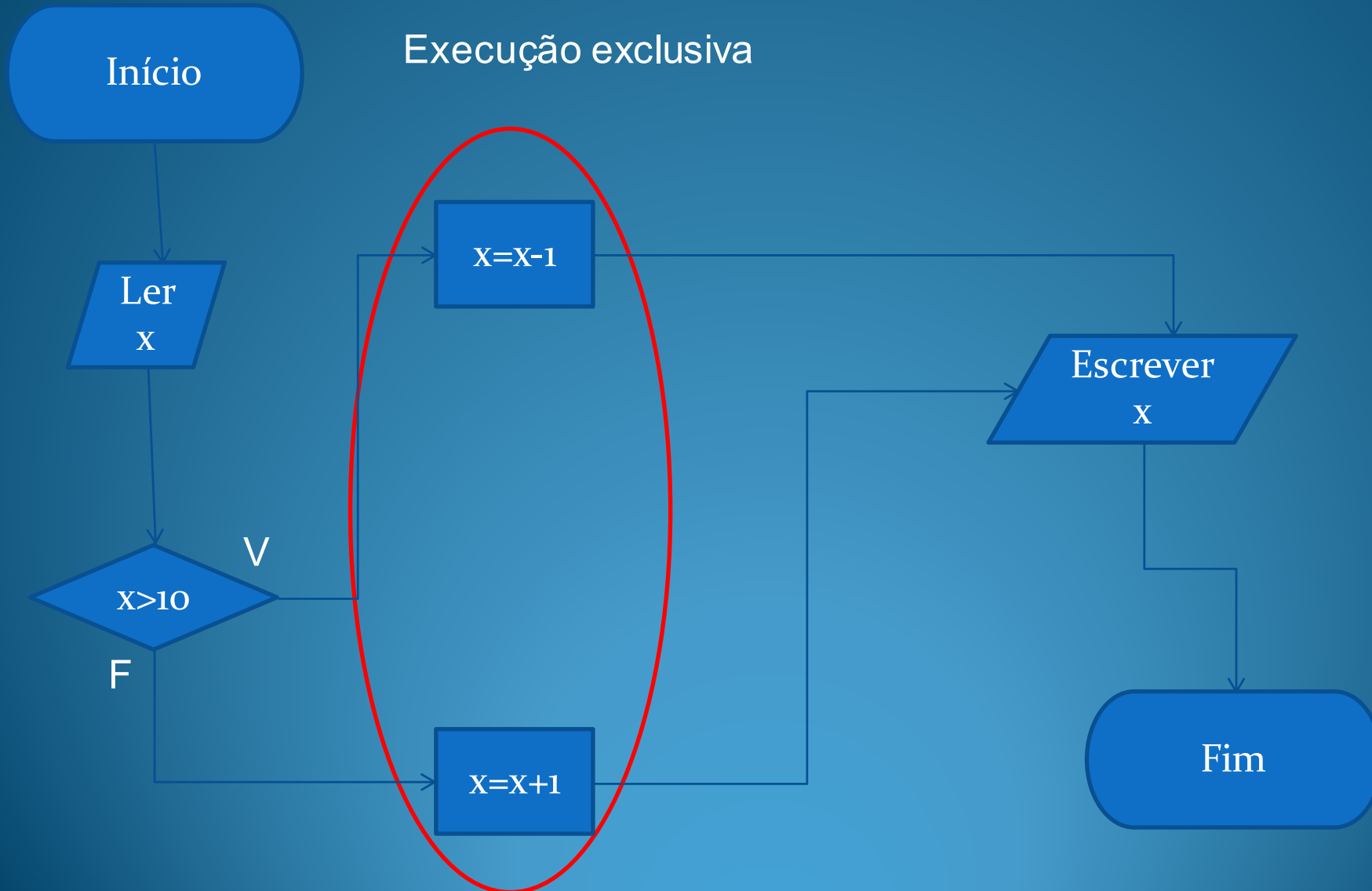
# Fluxogramas

- A partir do símbolo de início do fluxograma, cada figura terá correspondência com 1 instrução e será executada 1 vez. Exceções:
  - Blocos condicionais: 0 ou 1 execuções.
  - Blocos iterativos: “n” execuções.

# Fluxogramas: Bloco Condicional

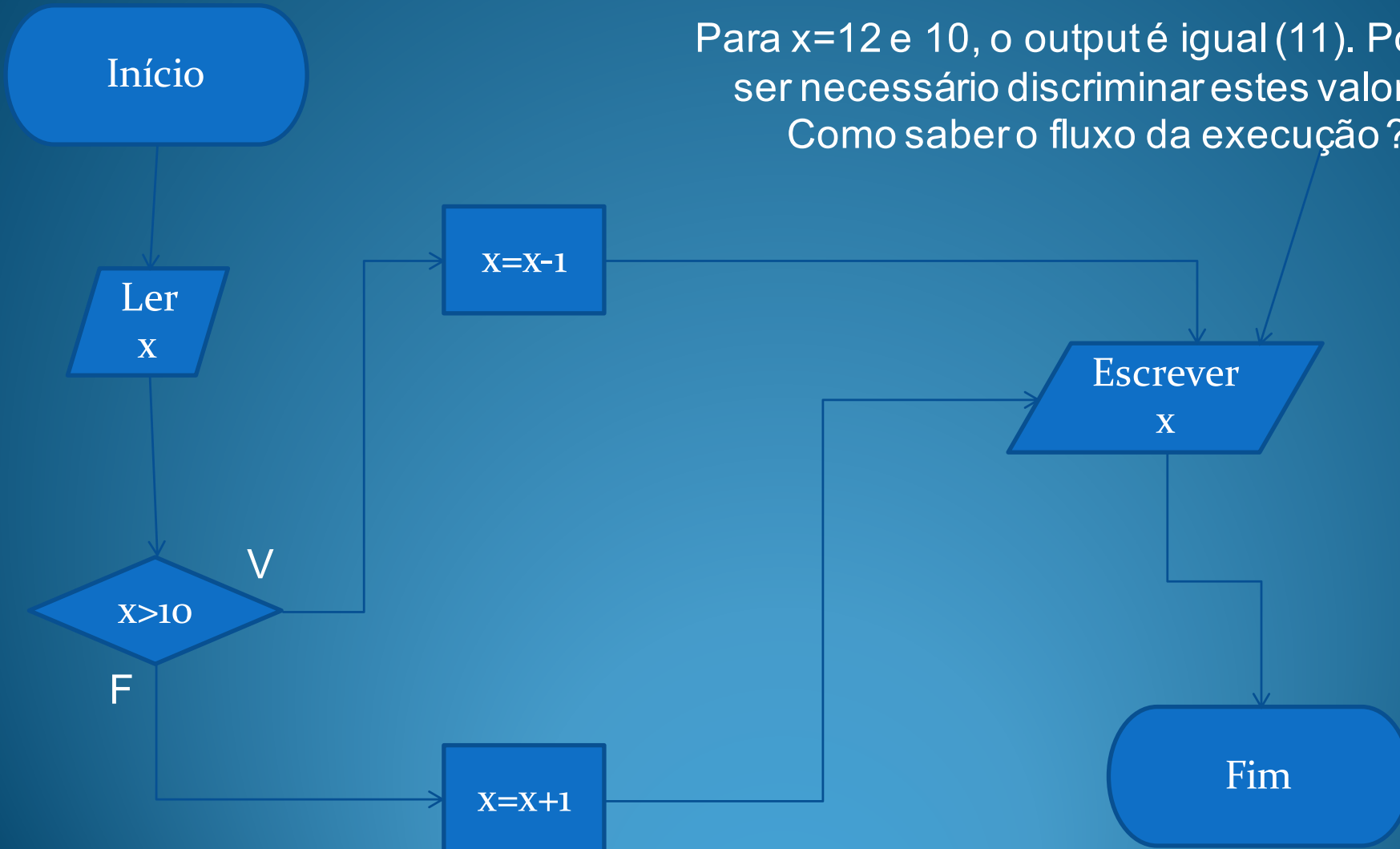


# Fluxogramas: Bloco Cond. Exclusivo



# Fluxogramas: Bloco Cond. Exclusivo

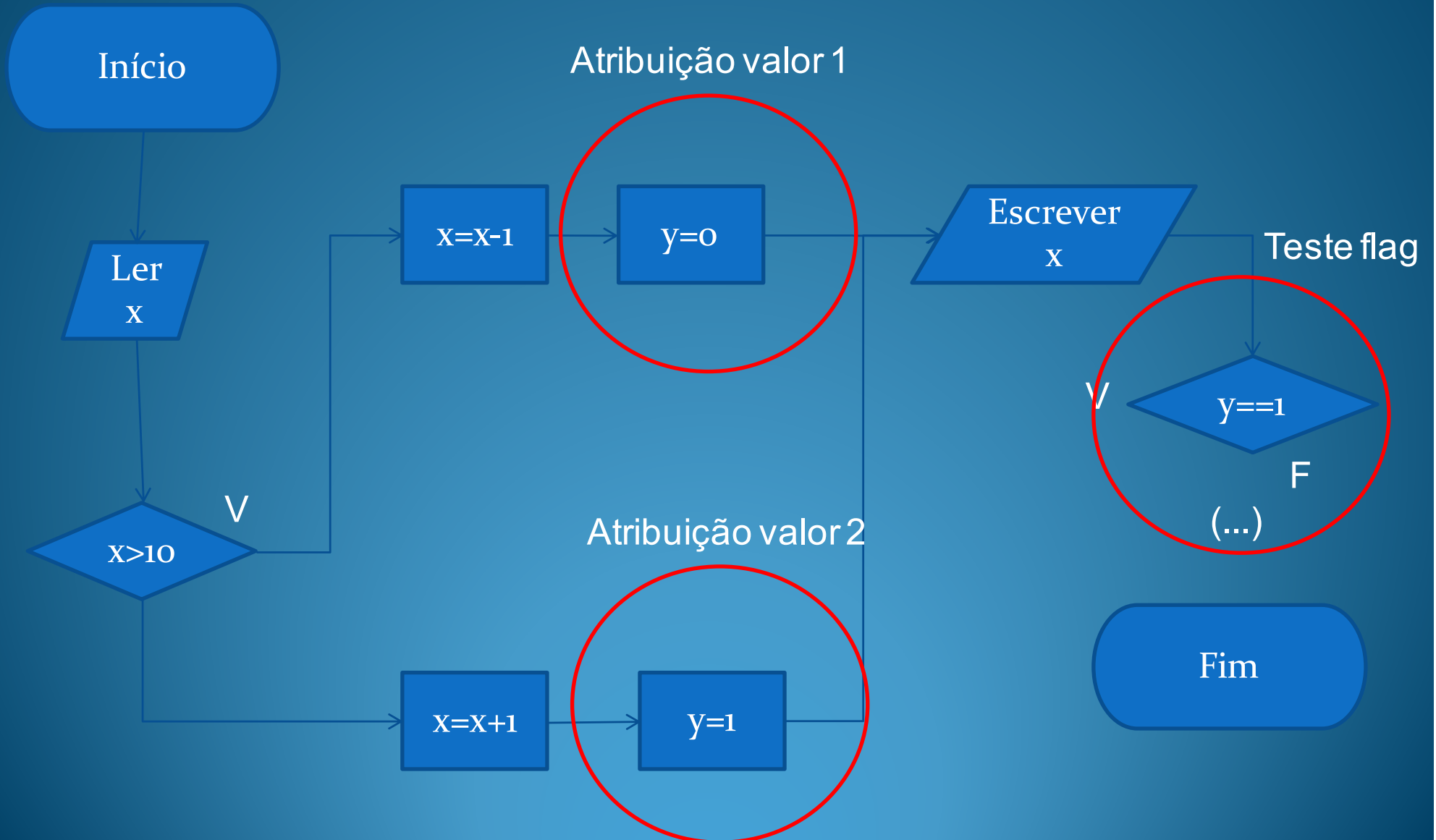
Para  $x=12$  e  $10$ , o output é igual (11). Poderá ser necessário discriminar estes valores. Como saber o fluxo da execução?



# Fluxogramas: Noção de Flag

- Variável (tipicamente inteira).
- Assume valores diferentes em ramos de blocos condicionais.
- O teste (comparação) de valor posterior permite saber qual o ramo de execução.

# Fluxogramas: Noção de Flag





# Flags: Exercício

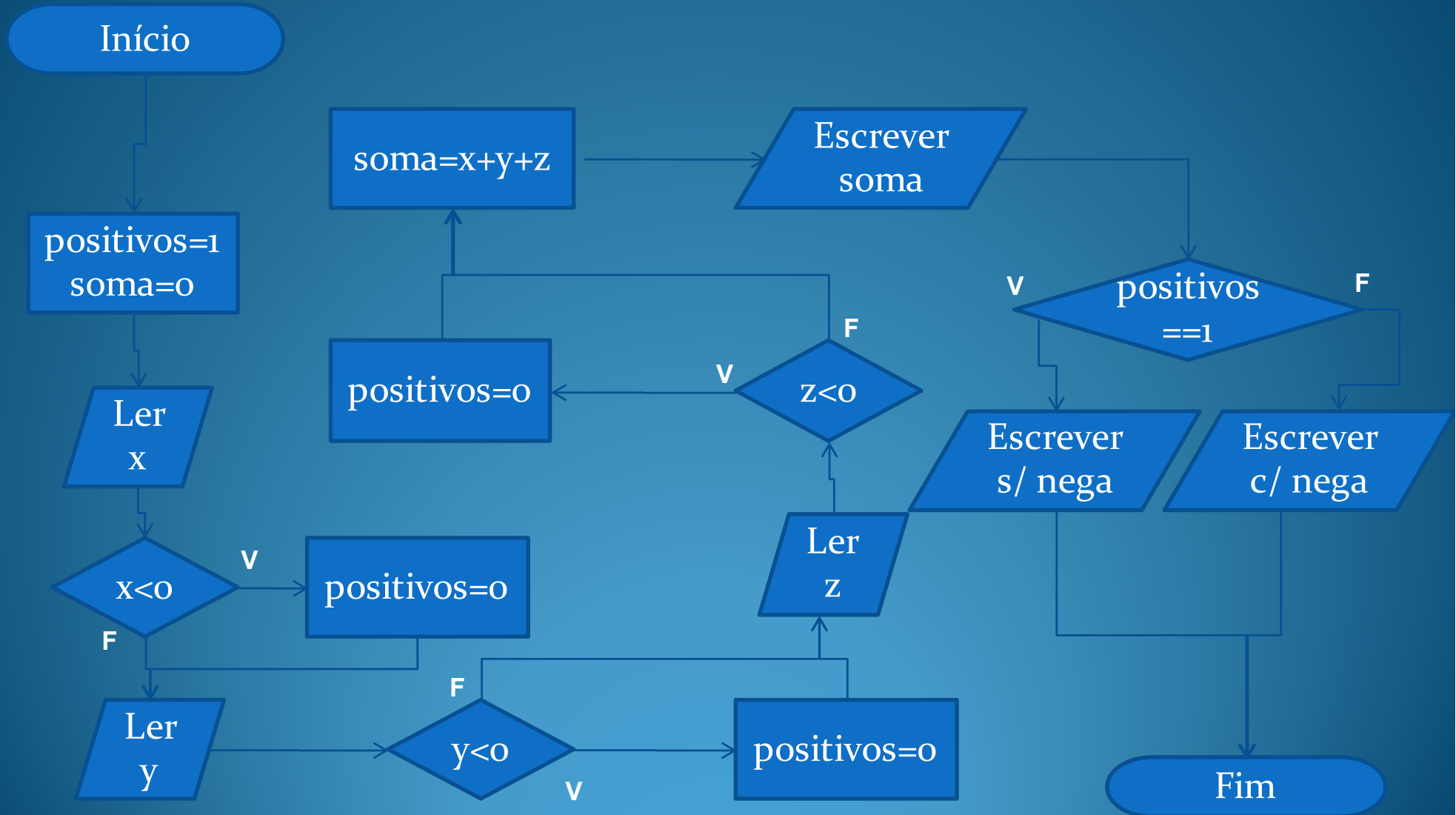
- É necessário um programa que receba 3 valores (x,y,z) do utilizador e imprima a sua soma. Adicionalmente, o programa deverá informar se o utilizador introduziu algum número negativo (utilize a noção de flag).

- Exemplos Output:

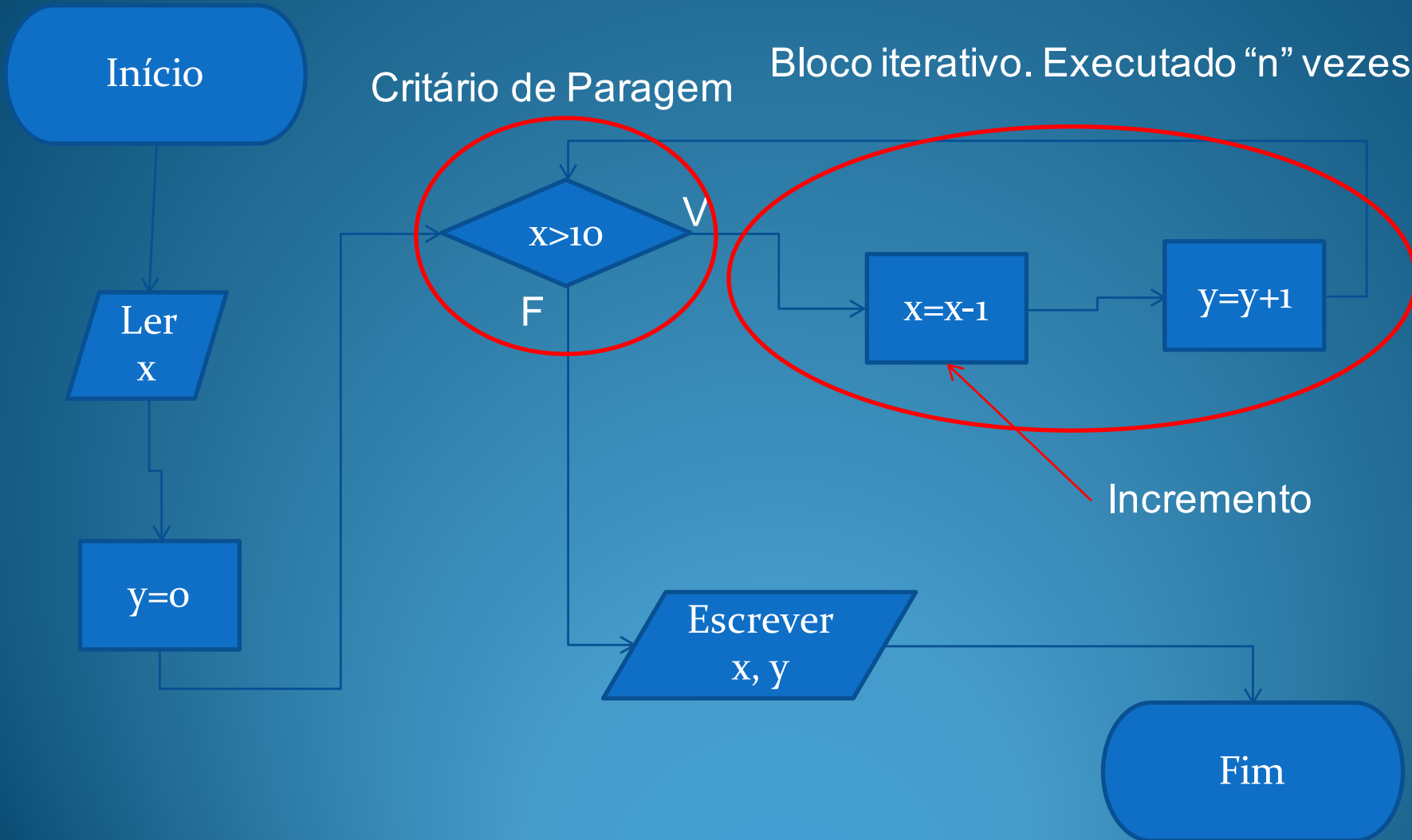
- Soma=25. Não introduziu números negativos

- Soma=31. Introduziu números negativos

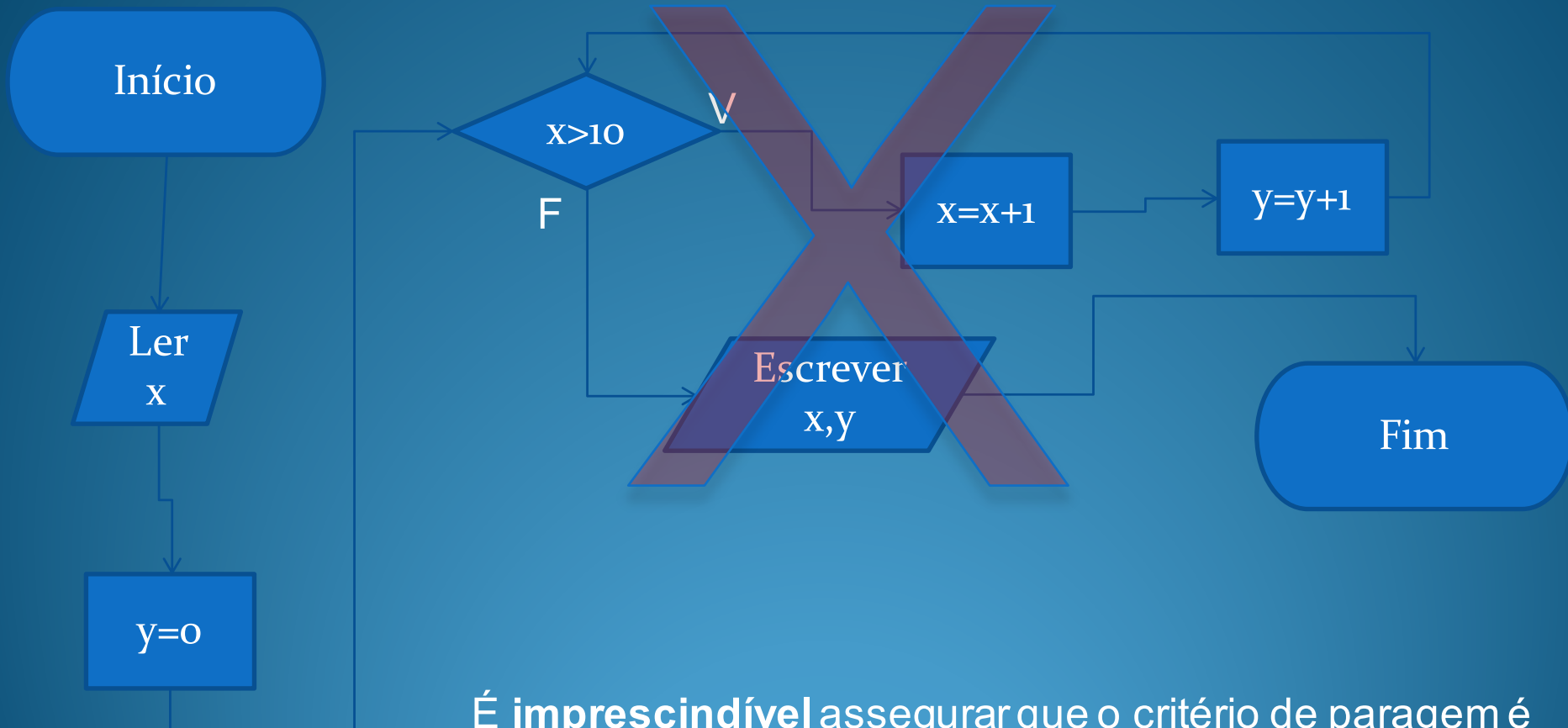
# Flags: Exercício



# Fluxogramas: Bloco Iterativo



# Fluxogramas: Bloco Iterativo

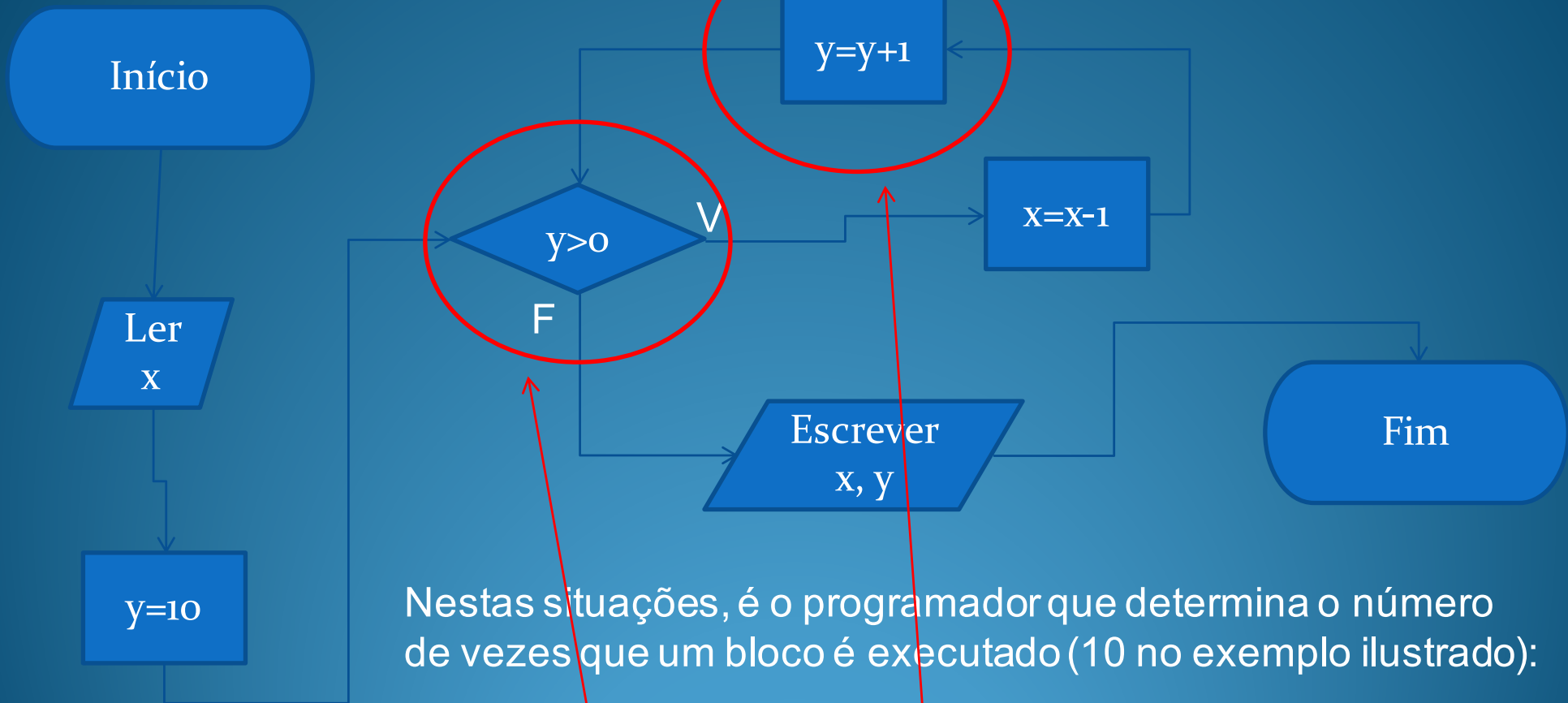


É imprescindível assegurar que o critério de paragem é sempre atingido (através do incremento)!

# Fluxogramas: Bloco Iterativo

- Apesar de incluírem a mesma figura no fluxograma, blocos iterativos e condicionais terão instruções específicas na linguagem. (if, for, while, do..while).
- Específicos para situações (tipos de iteração) diferentes.

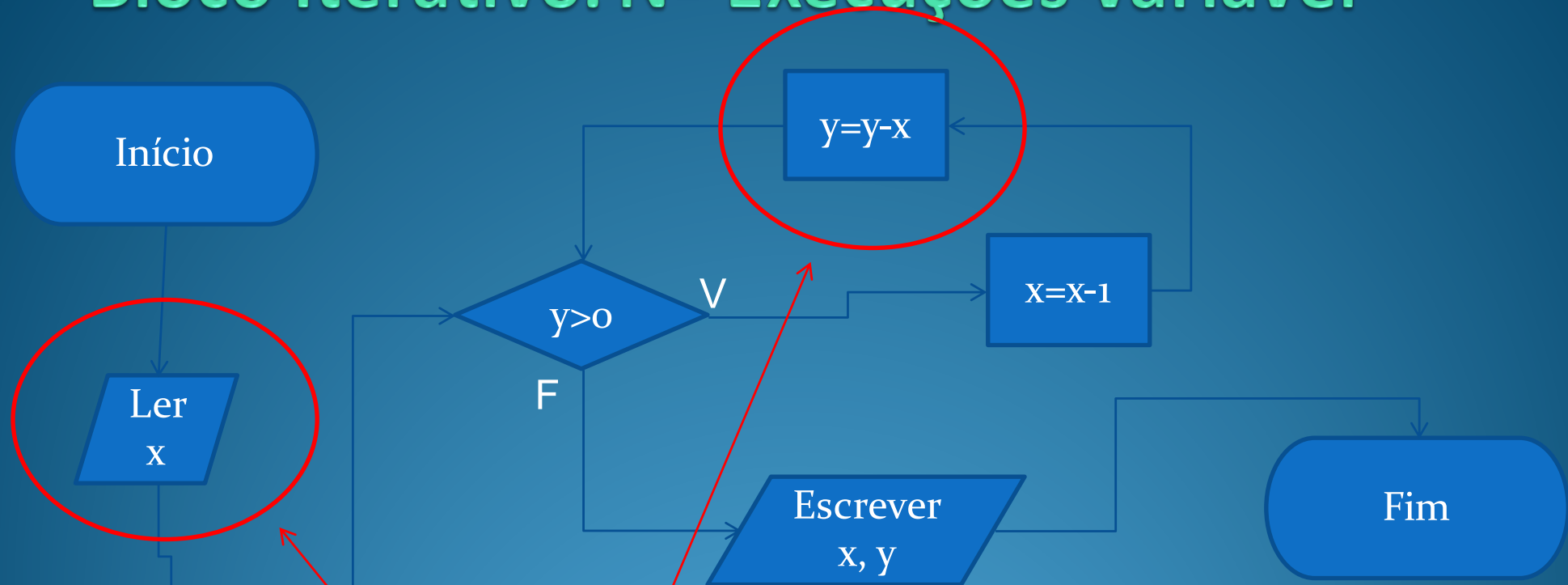
# Bloco Iterativo. Nº Execuções Fixo



Nestas situações, é o programador que determina o número de vezes que um bloco é executado (10 no exemplo ilustrado):

- Critério de paragem e incremento não dependem do utilizador ou de unidades de memória secundária.

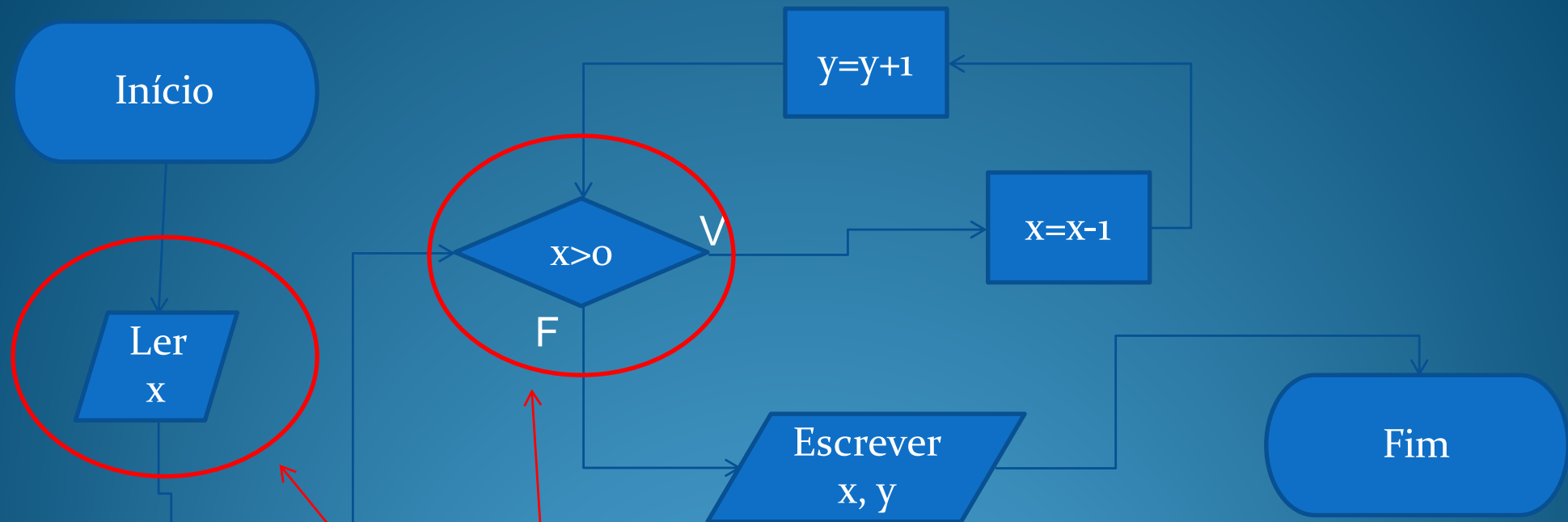
# Bloco Iterativo. Nº Execuções Variável



Nestas situações, o programador não consegue determinar antecipadamente o número de vezes que um bloco será executado.

- O incremento depende de valores provenientes do mundo exterior (utilizador).

# Bloco Iterativo. Nº Execuções Variável



Nestas situações, o programador não consegue determinar antecipadamente o número de vezes que um bloco será executado

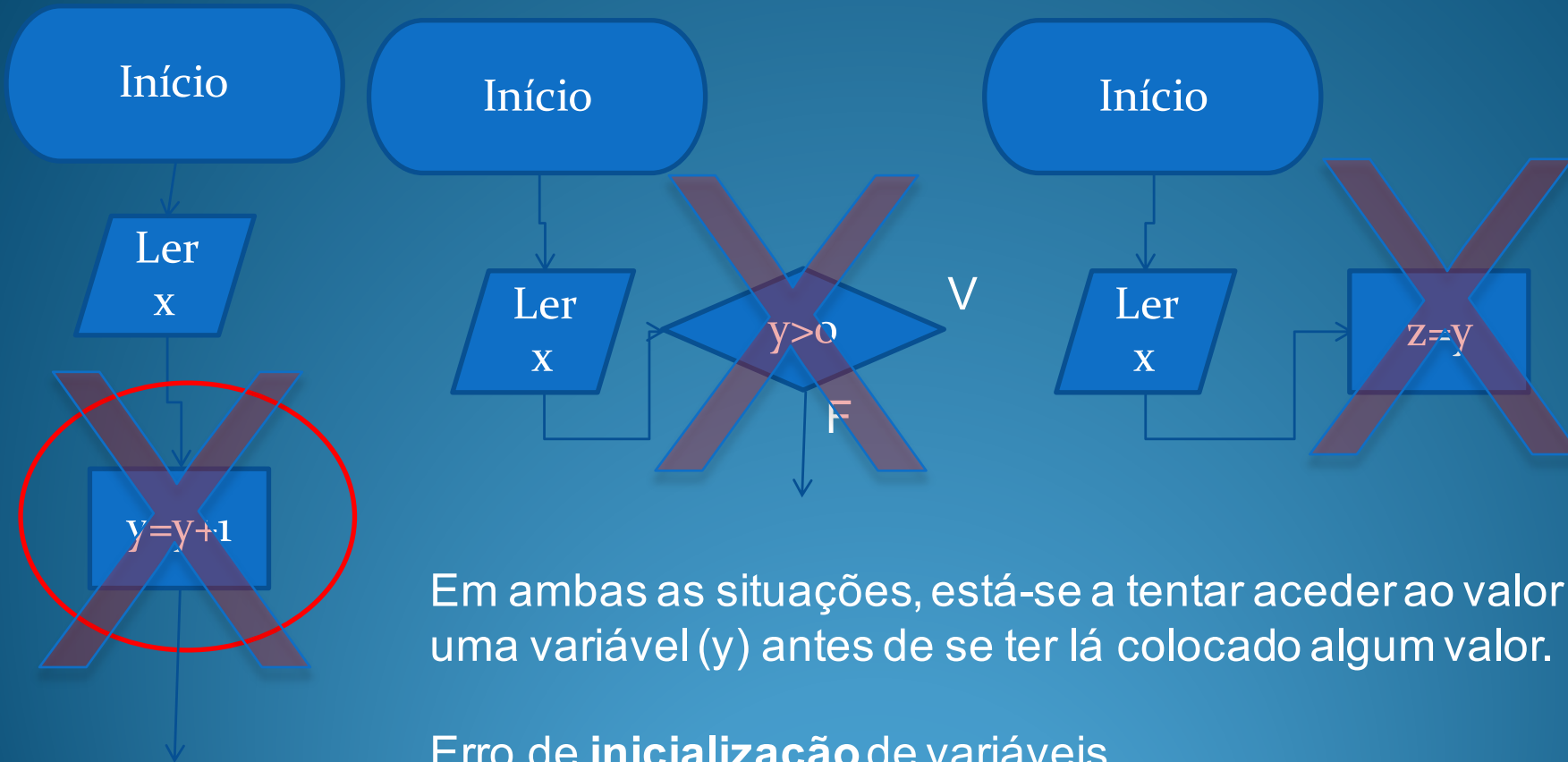
- Critério de paragem depende de valores provenientes do mundo exterior (utilizador).



## Bloco Iterativo. Exercício

- **Desenhe um fluxograma de um programa que imprima no écran os 100 primeiros números pares superiores a um valor introduzido pelo utilizador.**
  - **Analise se para esta situação, o número de execuções é fixo ou variável.**

# Blocos Condicionais e Iterativo. Erros Típicos:

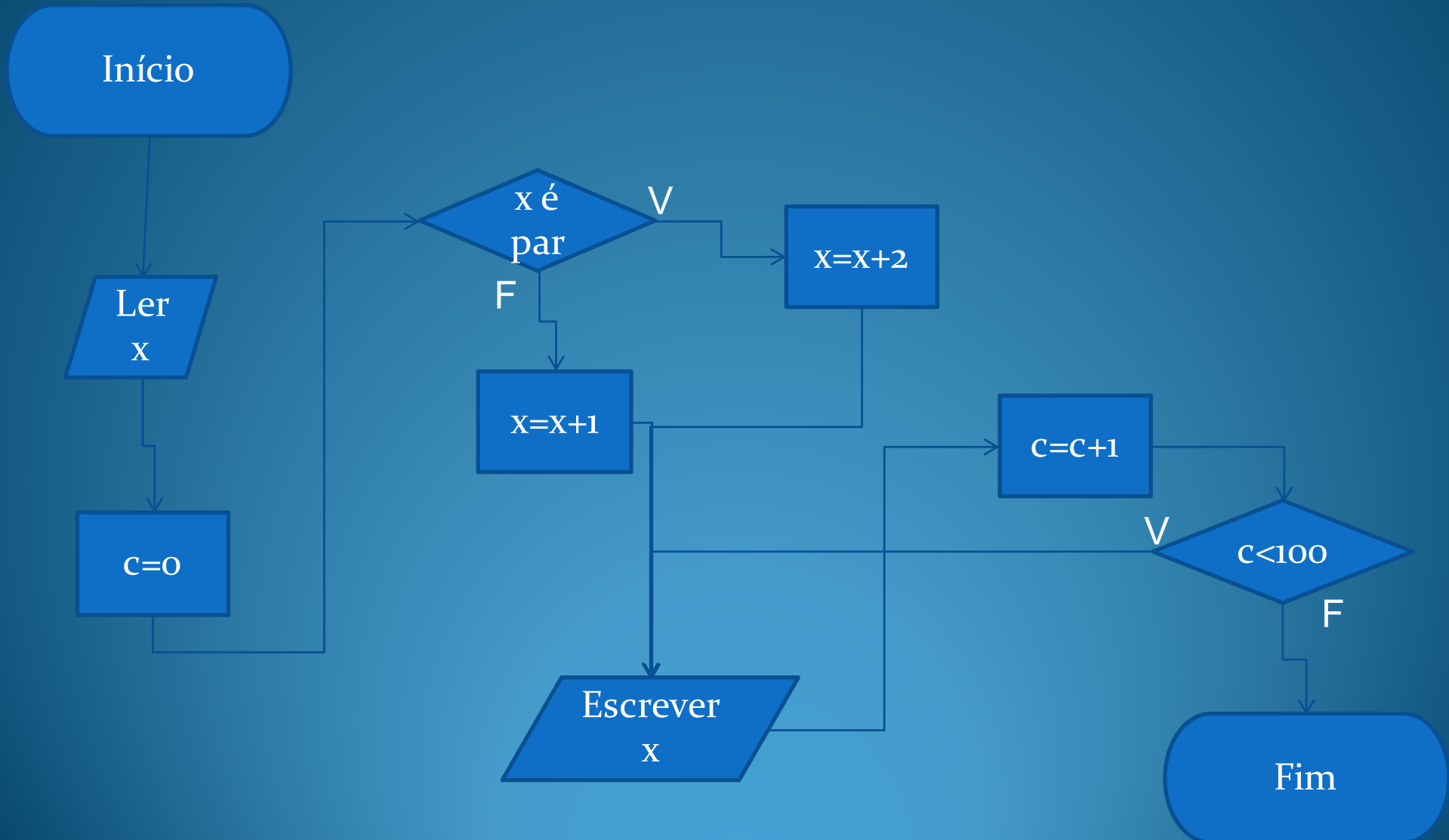


Em ambas as situações, está-se a tentar aceder ao valor de uma variável ( $y$ ) antes de se ter lá colocado algum valor.

Erro de **inicialização** de variáveis.

- O primeiro valor a ser colocado numa variável (através de uma operação de atribuição) designa-se por inicialização da variável.

# Bloco Iterativo. Exercício



# Operadores

- Tal como referido anteriormente, o “processamento” consiste num conjunto de operações relativamente reduzido:

- Operadores Lógicos
- Operadores Aritméticos
- Relacionais/ Atribuição

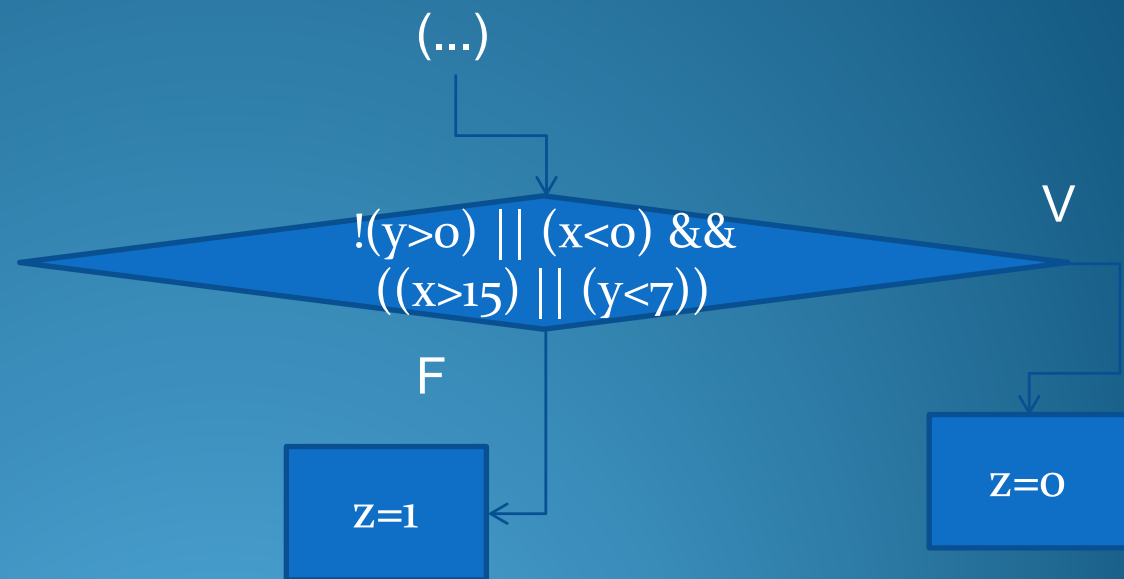
# Operadores Lógicos

• Operações básicas (p/ ordem de precedência:

• NOT: !

• AND: &&

• OR: ||



Exercício: Assinale as regiões do plano cartesiano (x,y), cujos valores das respectivas variáveis na condição acima ilustrada levam à execução da atribuição  $z=0$

# Operadores Aritméticas

- Operações básicas

- Subtração: -

Precedência 0

- Adição: +

---

- Multiplicação: \*

- Divisão: /

Precedência 1

- Módulo: %

# Operadores Relacionais / Atribuição

- Operação de Atribuição:

- = ( $x=y$ , copia o valor de “y” para “x”).

- Operadores Relacionais

- >, maior ( $x>y$  será verdade se o valor de “x” for maior que o de “y”).

- >= maior ou igual.

- <, menor.

- <=, menor ou igual.

- ==, igual ( $x==y$  será verdade se ambas as variáveis tiverem valores iguais).

- !=, diferente