

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the year 2013.

2013

# Portugol

Equivalências de estruturas entre  
Portugol e Python

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Decode Team

INSTITUTO POLITÉCNICO DE TOMAR

---

## Índice

---

Nota Geral: .....	3
Algumas notas sobre Python: .....	3
Estrutura Início .....	3
Início: .....	3
Estrutura Fim .....	3
Fim: .....	3
Variáveis .....	4
Equivalência entre TIPOS de variáveis .....	4
Definição e atribuição de variáveis .....	4
Estruturas input/output .....	5
Input – Ler .....	5
Output – Escrever .....	5
Estruturas de Decisão .....	6
Condição “if” e “if else” .....	6
Exemplos práticos .....	6
Condição “while” .....	7
Condição “do while” .....	8
Exemplos práticos .....	8
Estrutura Conector .....	9
Conector .....	9
Funções .....	10
Definir funções .....	10
Definir função .....	10
Exemplos do uso de funções .....	11
Estrutura de retorno .....	11
Return .....	11
Operadores .....	12
Aritméticos .....	12
Lógicos .....	12
Relacionais .....	12
ANEXO .....	13
Algoritmo com o uso da condição “if” .....	13
Fluxograma .....	13
Código .....	13

Esquema detalhado.....	14
Algoritmo com o uso da condição “if else” .....	15
Fluxograma .....	15
Código:.....	15
Esquema detalhado.....	16
Algoritmo com o uso da condição “while” .....	17
Fluxograma .....	17
Código.....	17
Esquema detalhado.....	18
Algoritmo com o uso da condição “do while” .....	19
Fluxograma .....	19
Código.....	19
Esquema detalhado.....	20
Algoritmo com o uso de uma função .....	21
Fluxogramas.....	21
Código.....	21
Esquema detalhado.....	22

---

### ***Nota Geral:***

---

Devido à especificação da linguagem, a tradução só é possível depois de ser executado o fluxograma.

---

### ***Algumas notas sobre Python:***

---

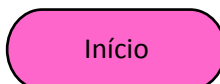
- É case sensitive.
- Necessita sempre do código indentado pois é a única maneira de o Python saber o que faz está dentro e fora dos ciclos porque não possui um 'End'.
- As funções são habitualmente definidas na parte inicial do programa, antes das instruções que farão a execução principal do programa.
- Para fazer comentários em Python utiliza-se o #
  - Exemplo: #comentário

---

### ***Estrutura Início***

---

Início:



O Python não tem equivalência ao Início.

---

### ***Estrutura Fim***

---

Fim:



O Python não tem equivalência ao Fim.

## Variáveis

### Equivalência entre TIPOS de variáveis

TIPO	Portugol	Python
Inteiro	Inteiro	Int
Real	Real	Float
Texto	Texto	String
Lógico	Logico	boolean

Tabela 1 - Tipos de variáveis

### Definição e atribuição de variáveis

Nota: Ao utilizar variáveis em Python não é necessário informar o tipo desta variável como fazemos em outras linguagens.

Mas por vezes e para evitar que o Python confunda o nome de uma variável que tem um valor inteiro com uma string, basta antes do colocar antes do nome da variável `int` ou se for outro tipo colocar o respectivo tipo antes( `real`, `boolean` ou `float`)

Exemplo:

```
if int(n)%2==0:
```

- Neste caso indicamos que a variável *n* é do tipo *int*(inteiro) para que o programa não confuda com uma string.

Int(n)

Exemplo:

```
n= input('Digite um numero: ')
if int(n)%2==0:
    print('Numero Par')
```

---

## *Estruturas input/output*

---

### Input – Ler



<variável>= input('<texto>')

### Output – Escrever



Para escrever no ecrã:

Print '<texto>'

Ou

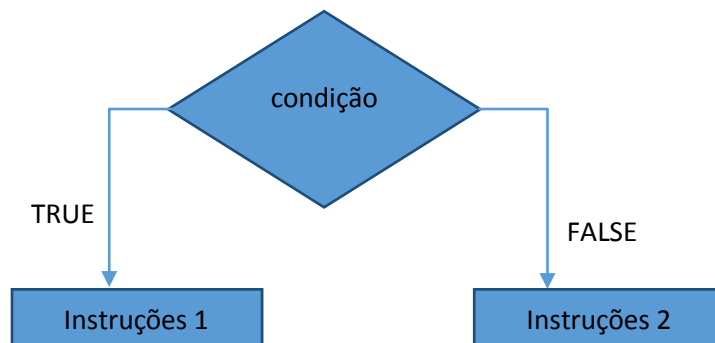
Print ( '<texto>',<variável>)

---

## *Estruturas de Decisão*

---

### Condição “if” e “if else”



Para TRUE, escrever:

```
if <condição>:  
    <instruções>
```

Para FALSE:

Se Instruções 2 for igual a  (conector) não fazer nada.

Senão, escrever:

```
else:  
    <instruções>
```

### Exemplos práticos

#### Condição “if”

```
n= input('Digite um numero: ')  
if int(n)%2==0:  
    print('Par')
```

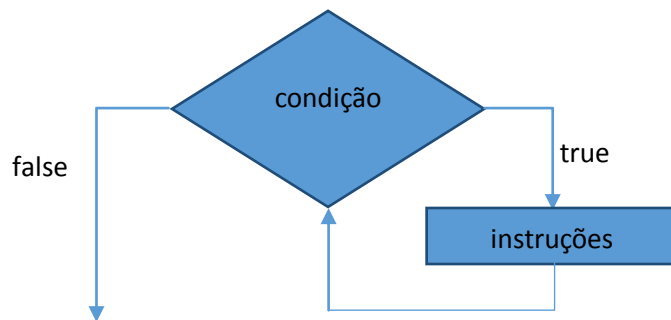
#### Condição “if else”

```
n= input('Digite um numero: ')  
if int(n)%2==0:  
    print('Numero Par')
```

else:

```
print('Numero Impar')
```

## Condição “while”

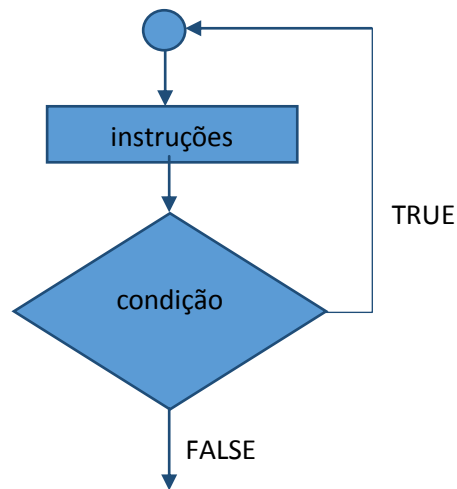


```
while <condição>:
```

```
<instruções>
```



## Condição “do while”



Instruções

**While True:**

<Instruções>

**If**<condição>:

<Instruções>

**break**

## Exemplos práticos

### Condição “while”

```
i=1
while int(i)<=10:
    print(i)
    i=i+1
```

### Condição “do while”

```
while True:

    i=input('Digite um número:')

    int(i)<0

    if int(i)>=0:

        break
```

---

## ***Estrutura Conector***

---

### Conector



Se for uma condição “do while” escrever:

```
While True:

    instrução
```

Senão, escrever:

Break

---

## *Funções*

---

### Definir funções

Exemplo( a , b , ... )

#### Definir função

**def**<nome\_função>:

<instruções>

Chamada de funções

Fact(Nome)

NOME(PARAMETRO);

### Exemplos do uso de funções

```
k=input('Digite um numero: ')
```

```
def fact(k):  
    if k == 0:  
        return 1  
    else:  
        return (int(k) * fact( int(k) - 1 ))
```

```
j=fact(k)
```

```
print(fact(k))
```

---

### *Estrutura de retorno*

---

### Return

expressao

```
return expressao;
```

## Operadores

### Aritméticos

Nome	Portugol	Python
Adição	$a + b$	$a + b$
Subtração	$a - b$	$a - b$
Divisão	$a / b$	$a / b$
Multiplicação	$a * b$	$a * b$
Resto da divisão inteira		$a \% b$
Potenciação		$**$

Tabela 2 - Equivalência de operadores aritméticos

### Lógicos

Nome	Portugol	Python
Disjunção	$a \&\& b$	and
Conjunção	$a    b$	or
Negação	Não	not

Tabela 3 - Equivalência de operadores lógicos

### Relacionais

Nome	Portugol	Python
Igual	$a == b$	$a == b$
Diferente	$a != b$	$a != b$
Maior	$a > b$	$a > b$
Maior ou igual	$a >= b$	$a >= b$
Menor	$a < b$	$a < b$
Menor ou igual	$a <= b$	$a <= b$

Tabela 4 - Equivalência de operadores relacionais

---

## ANEXO

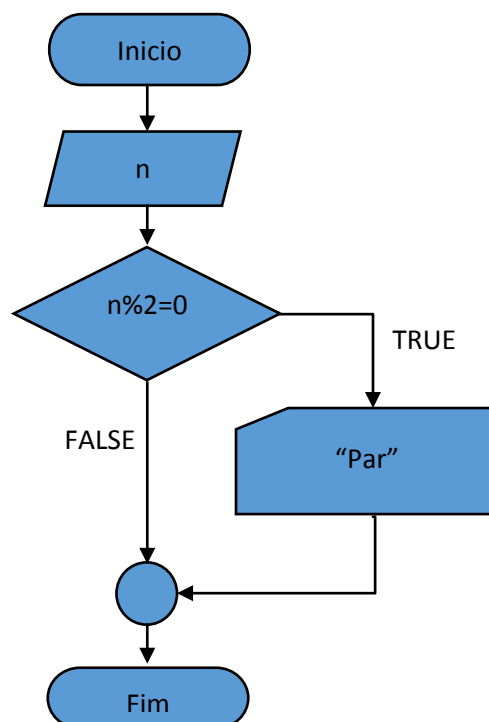
---

Para uma compreensão mais abrangente do uso das estruturas, ficam alguns exemplos mais extensivos, com o uso de várias estruturas em algoritmos completos.

### Algoritmo com o uso da condição “if”

**Problema:** Verificar se um número introduzido pelo utilizador é par.

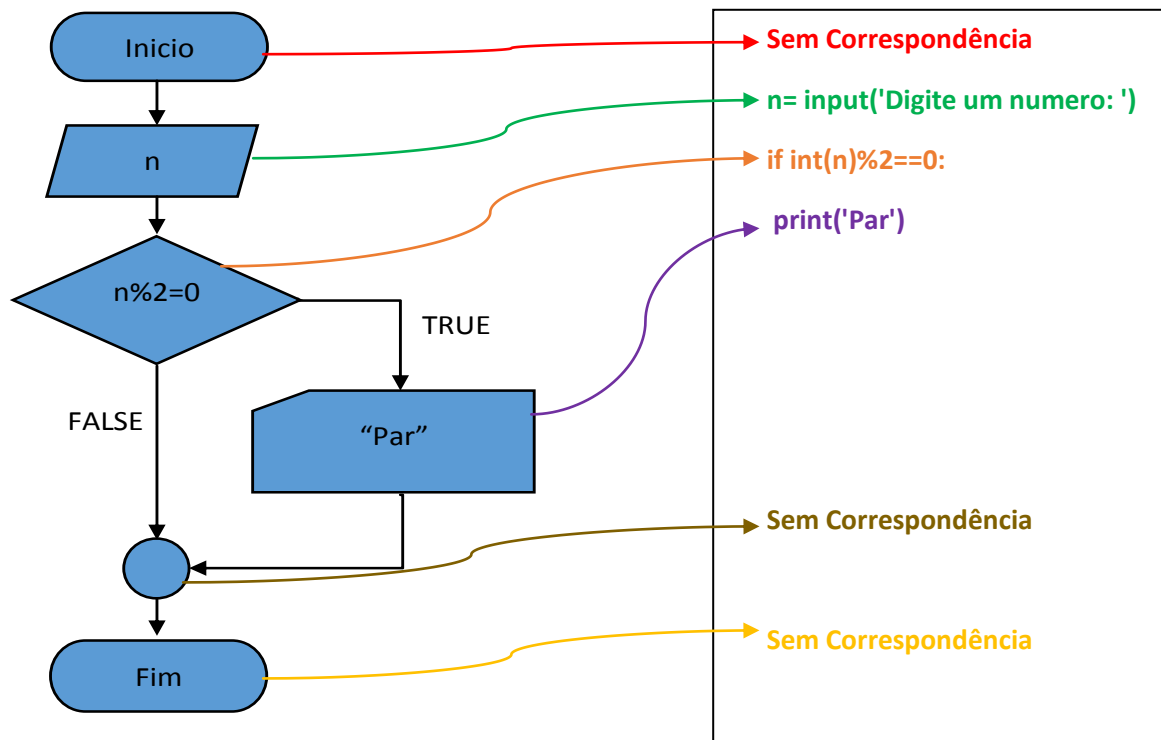
#### Fluxograma



#### Código

```
n= input('Digite um numero: ')
if int(n)%2==0:
    print('Par')
```

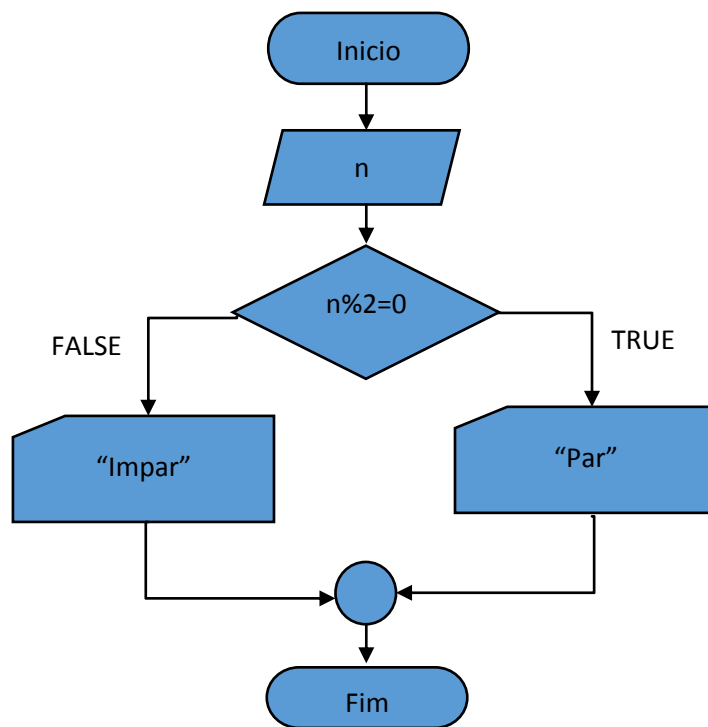
Esquema detalhado



## Algoritmo com o uso da condição “if else”

**Problema:** Verificar se um número introduzido pelo utilizador é par ou ímpar.

### Fluxograma

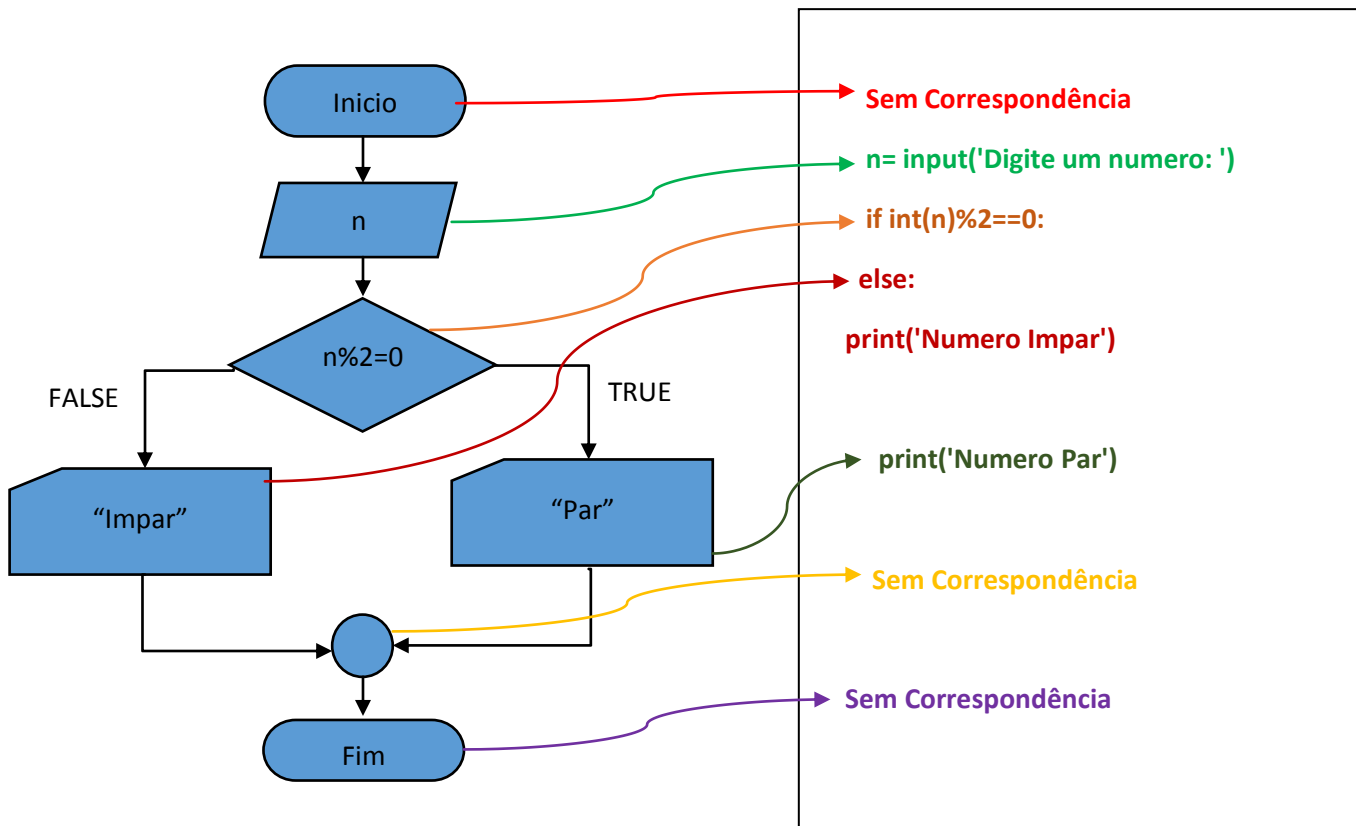


### Código:

```
n= input('Digite um numero: ')
if int(n)%2==0:
    print('Numero Par')
else:
    print('Numero Impar')
```



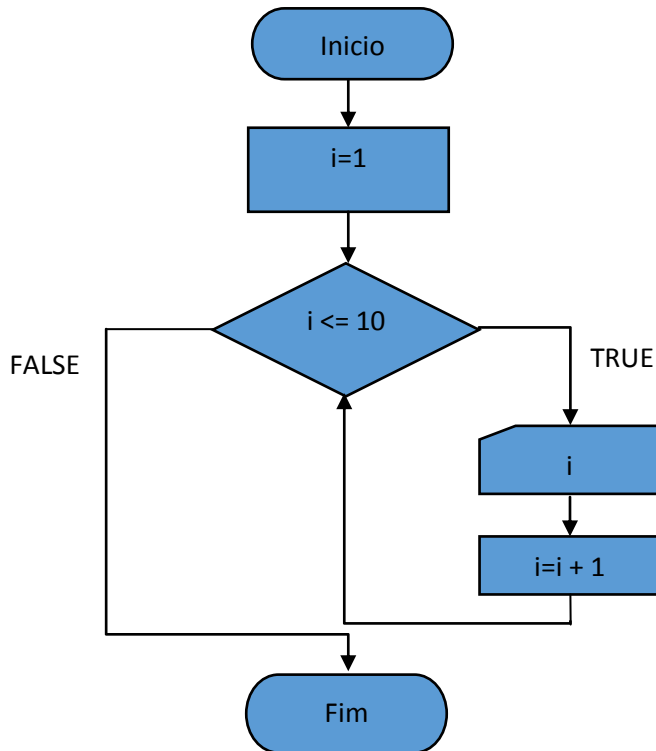
Esquema detalhado



## Algoritmo com o uso da condição “while”

**Problema:** Escrever um número de 1 a 10.

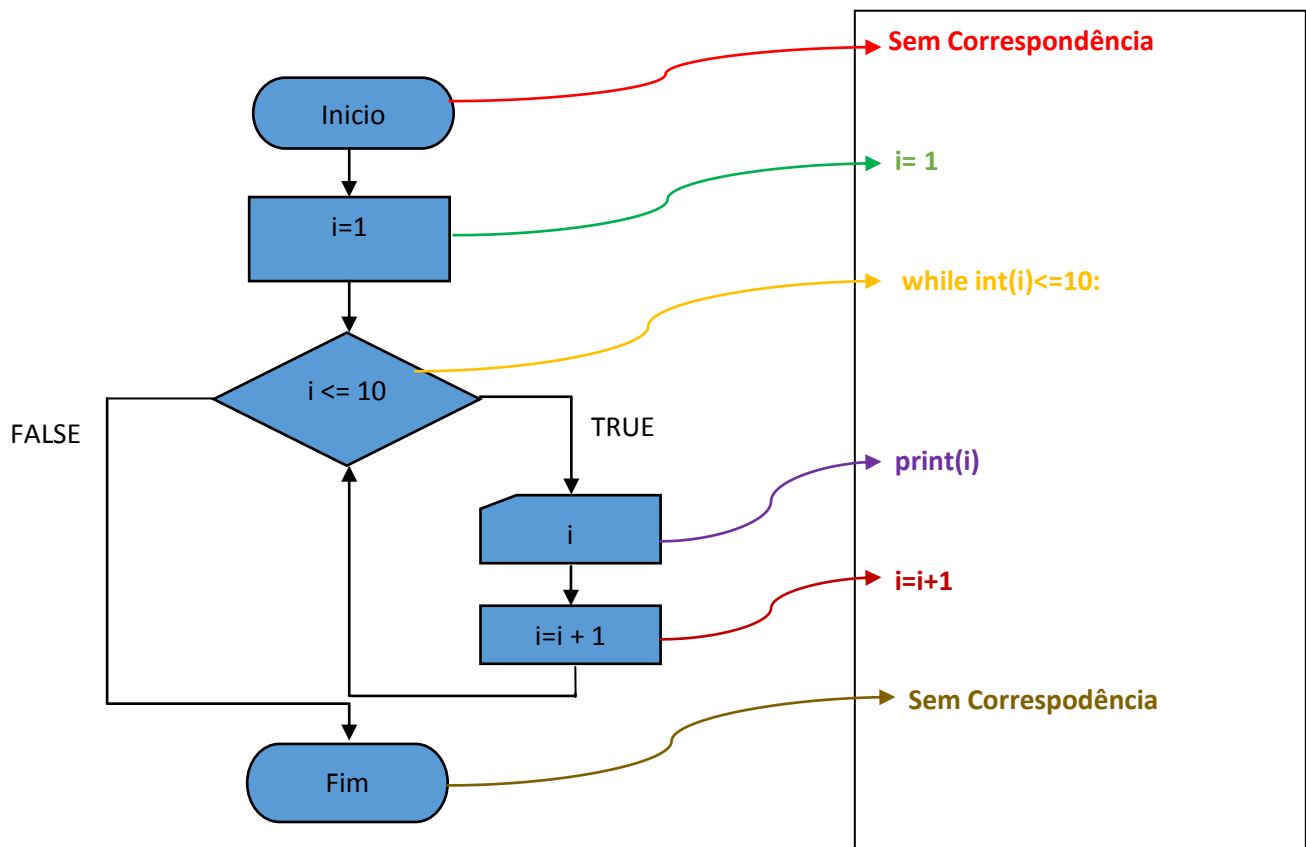
### Fluxograma



### Código

```
i=1
while int(i)<=10:
    print(i)
    i=i+1
```

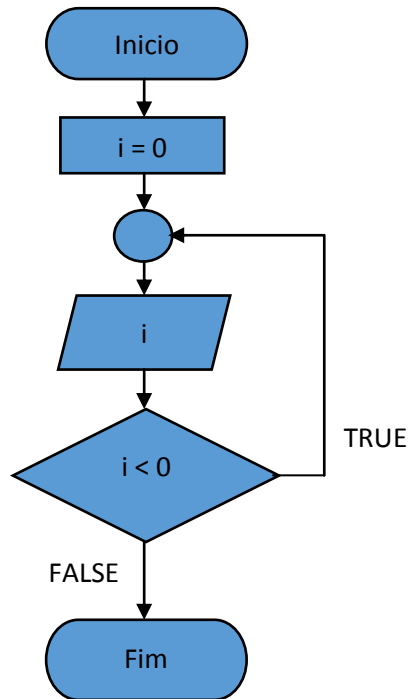
Esquema detalhado



## Algoritmo com o uso da condição “do while”

**Problema:** Pedir um número positivo.

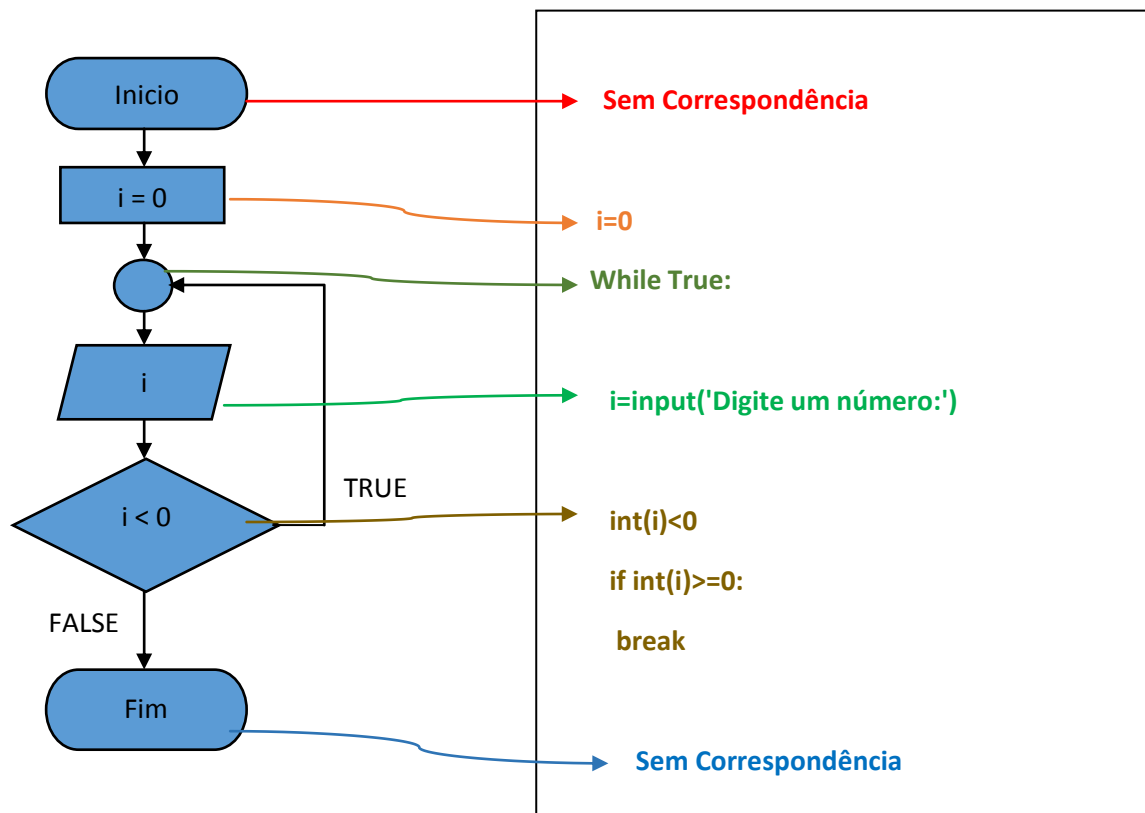
### Fluxograma



### Código

```
i=0
while True:
    i=input('Digite um número:')
    int(i)<0
    if int(i)>=0:
        break
```

Esquema detalhado

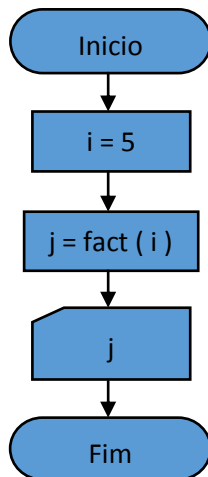


## Algoritmo com o uso de uma função

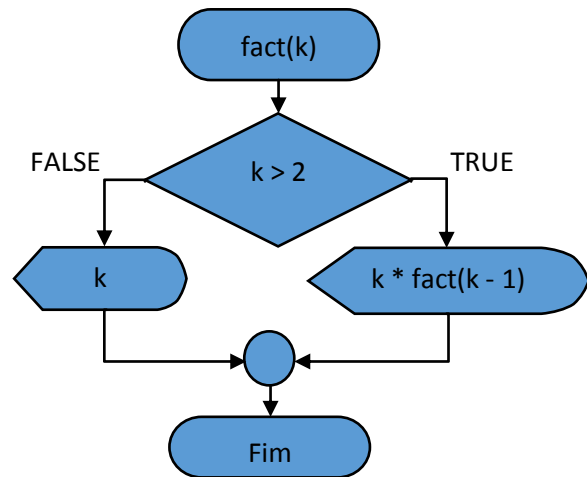
**Problema:** Factorial de um número.

### Fluxogramas

**Código principal**



**Função fact(k)**



### Código

```
def fact(k):  
    if k > 2 :  
        return (k * fact(k - 1) )  
    else:  
        return k  
  
i=5  
  
j=fact(i)  
  
print (j)
```

Esquema detalhado

