

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

Portugal

Equivalências de estruturas entre
Portugal e Lua

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

Decode Team

INSTITUTO POLITÉCNICO DE TOMAR

Índice

Nota Geral:	3
Algumas notas sobre Lua:	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	4
Input – Ler	4
Output – Escrever	5
Estruturas de Decisão	5
Condição “if” e “if else”	5
Condição “while”	7
Condição “do while”	8
Exemplos práticos	8
Estrutura Conector.....	9
Conector	9
Funções	10
Definir funções.....	10
Definir função	10
Chamada de funções	11
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 Lua:

- É case sensitive.
- As funções são habitualmente definidas na parte inicial do programa, antes das instruções que farão a execução principal do programa.
- O código deve ser guardado num ficheiro com extensão .lua
- Para fazer comentários em Python utiliza-se --
 - Exemplo: --Comentário

Estrutura Início

Início:



Início

O Lua não tem equivalência ao Início.

Estrutura Fim

Fim:



Fim

O Lua não tem equivalência ao Fim.

Variáveis

Equivalência entre TIPOS de variáveis

TIPO	Portugol	Python
Inteiro	Inteiro	Number
Real	Real	Number
Texto	Texto	String
Lógico	Logico	boolean

Tabela 1 - Tipos de variáveis

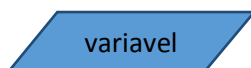
Nota: Lua não diferencia valores inteiros de valores reais.

Definição e atribuição de variáveis

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

Estruturas input/output

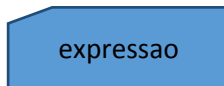
Input – Ler



```
io.write'<texto> '
```

```
variavel = io.read()
```

Output – Escrever



Para escrever no ecrã:

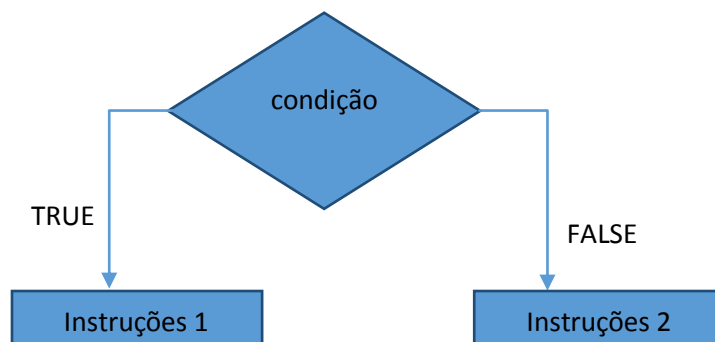
Print '<texto>'

Ou

Print ('<texto>',<variável>)

Estruturas de Decisão

Condição “if” e “if else”



Para TRUE, escrever:

if <condição> **then**

<instruções>

end

Para FALSE:

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

Senão, escrever:

else

<instruções>

Exemplos práticos

Condição “if”

if <condição> **then**

<instruções>

end

Condição “if else”

```
io.write'Digite um número:'
```

```
n = io.read()
```

```
if n%2==0 then
```

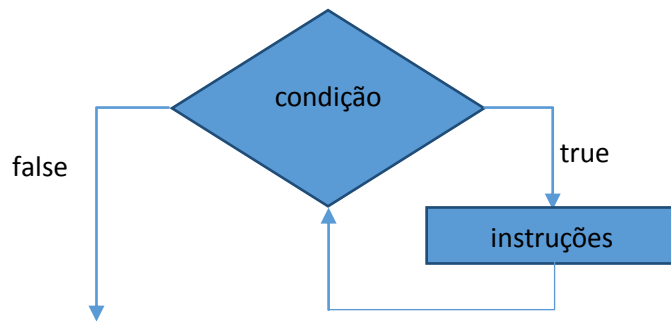
```
    print 'Numero Par'
```

```
else
```

```
    print 'Numero Impar'
```

```
end
```

Condição “while”



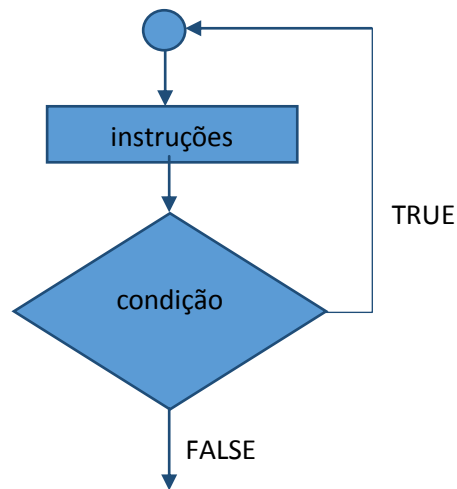
while <condição> **do**

<instruções>

end

Nota: Em Lua o ciclo While é composto sempre pelo 'do'.

Condição “do while”



Instruções

Nota: Não tem equivalência

Exemplos práticos

Condição “while”

```
i=1
while i<=10 do
    print(i)
    i=i+1
end
```

Condição “do while”

```
i=0
repeat
    io.write'Digite um número: '
    i = io.read()
until tonumber(i)>=0
```

Estrutura Conector

Conector



Se for uma condição “*do while*” escrever:

repeat

Funções

Definir funções

Exemplo(a , b , ...)

Definir função

Function<nome_função>

<Instruções>

end

Chamada de funções

Function(Nome)

NOME(PARAMETRO);

Exemplos do uso de funções

```
function fact(k)
    if tonumber(k) > 2 then
        return k * fact(k - 1)
    else
        return k
    end
end
```

```
i=5
j=fact(i)
print(j)
```

Estrutura de retorno

Return

expressao

return expressao;

Operadores

Aritméticos

Nome	Portugol	Lua
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		\wedge

Tabela 2 - Equivalência de operadores aritméticos

Lógicos

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

Tabela 3 - Equivalência de operadores lógicos

Relacionais

Nome	Portugol	Lua
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

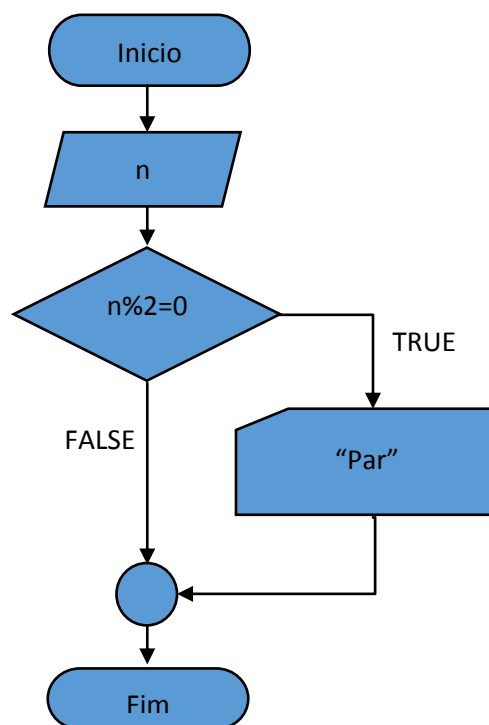
ANEXO

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

```
io.write'Digite um número:'
```

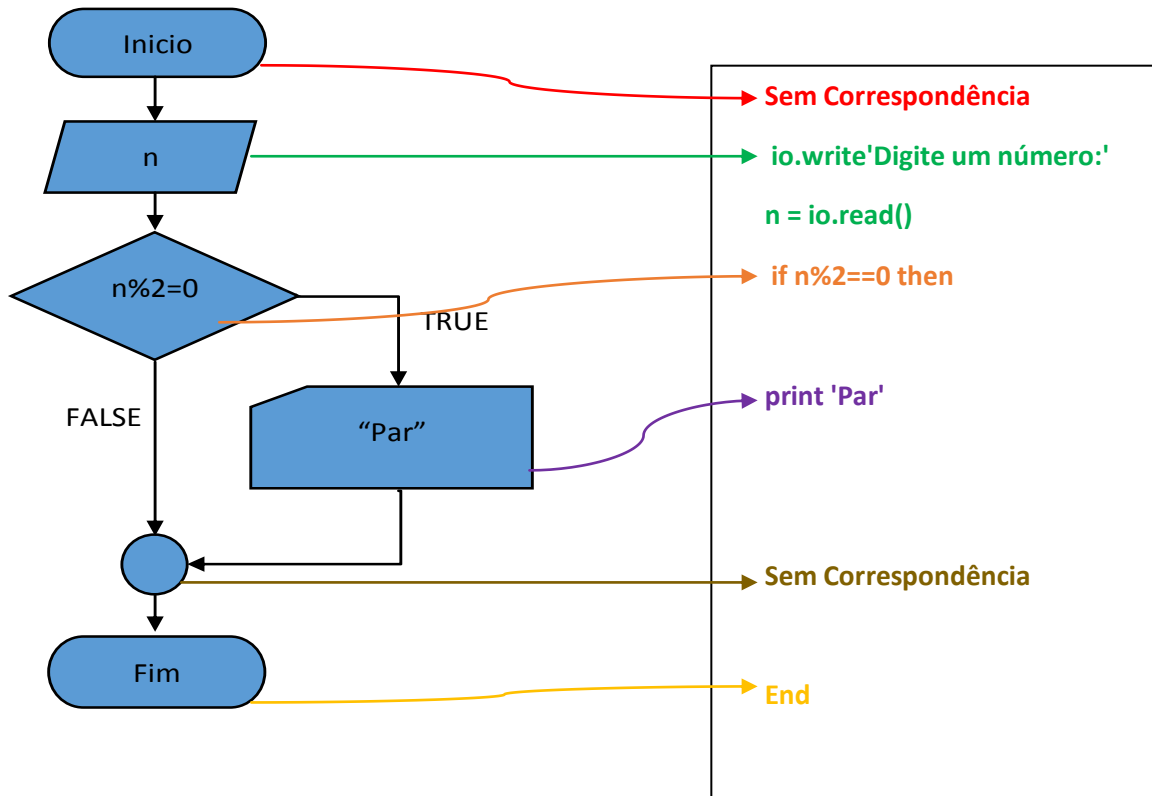
```
n = io.read()
```

```
if n%2==0 then
```

```
    print 'Par'
```

```
end
```

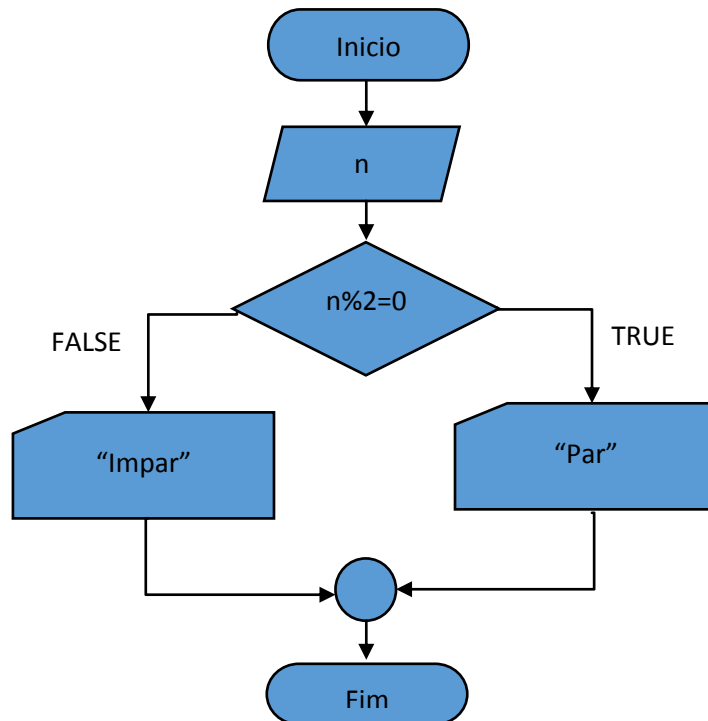
Esquema detalhado



Algoritmo com o uso da condição “if else”

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

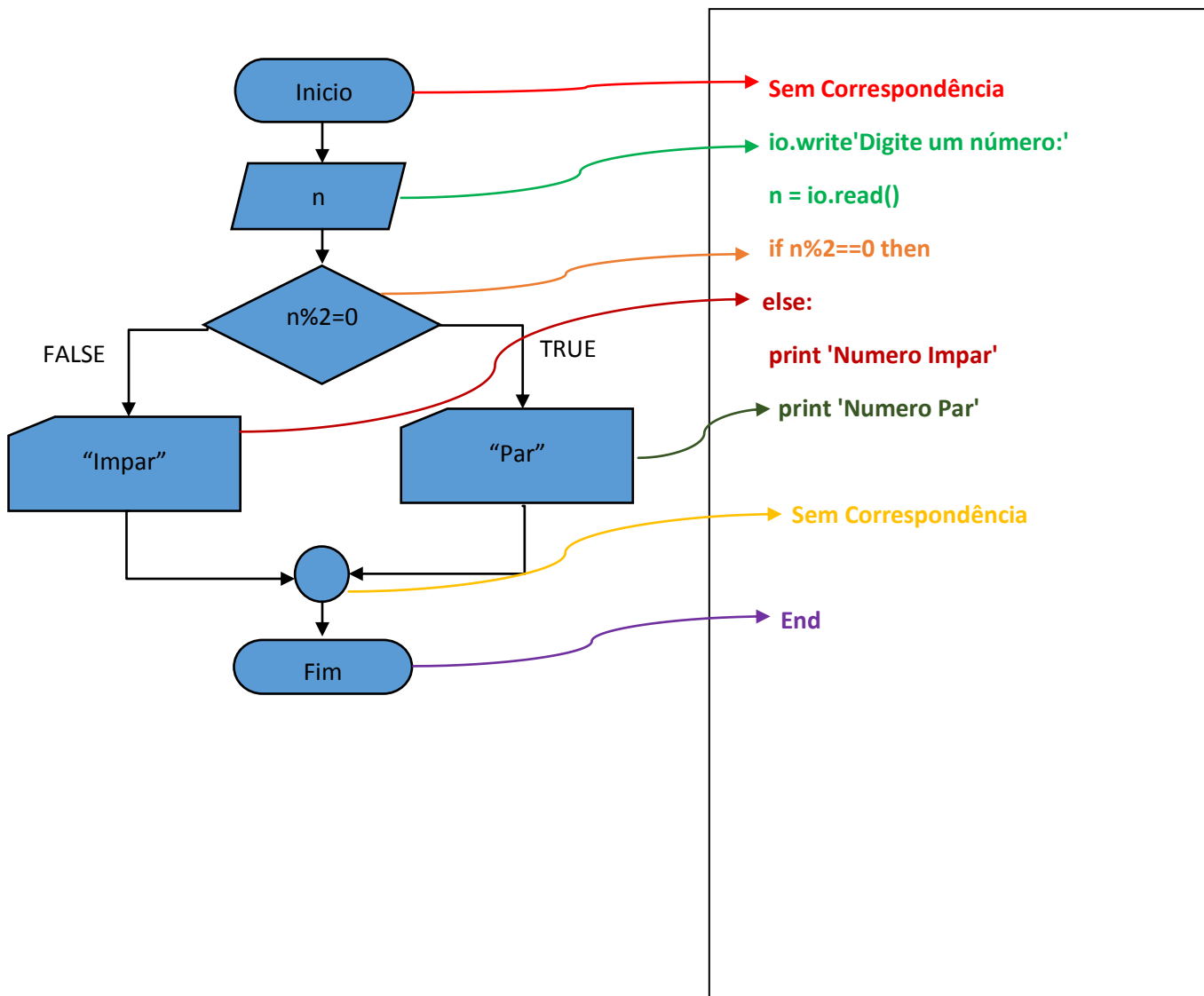
Fluxograma



Código:

```
io.write'Digite um número:'  
n = io.read()  
if n%2==0 then  
    print 'Numero Par'  
else  
    print 'Numero Impar'  
end
```

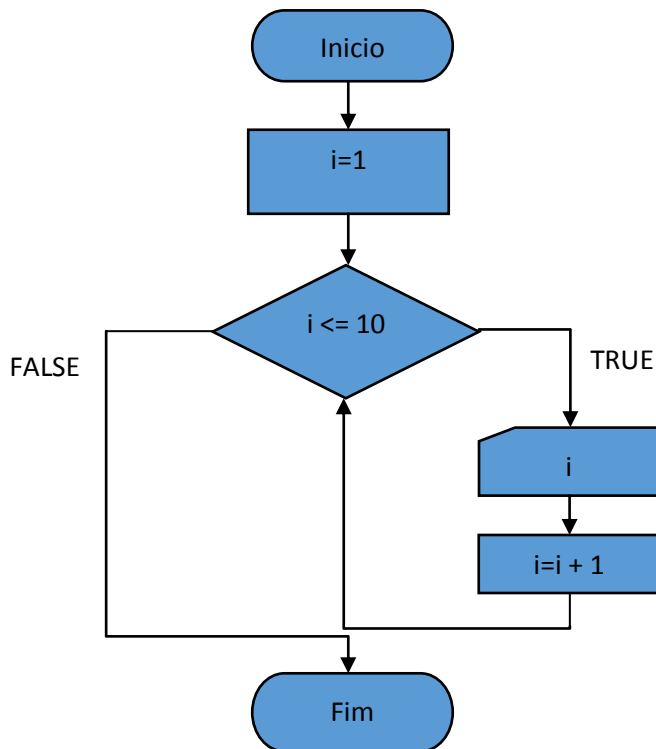

Esquema detalhado



Algoritmo com o uso da condição “while”

Problema: Escrever um número de 1 a 10.

Fluxograma



Código

```
i=1
```

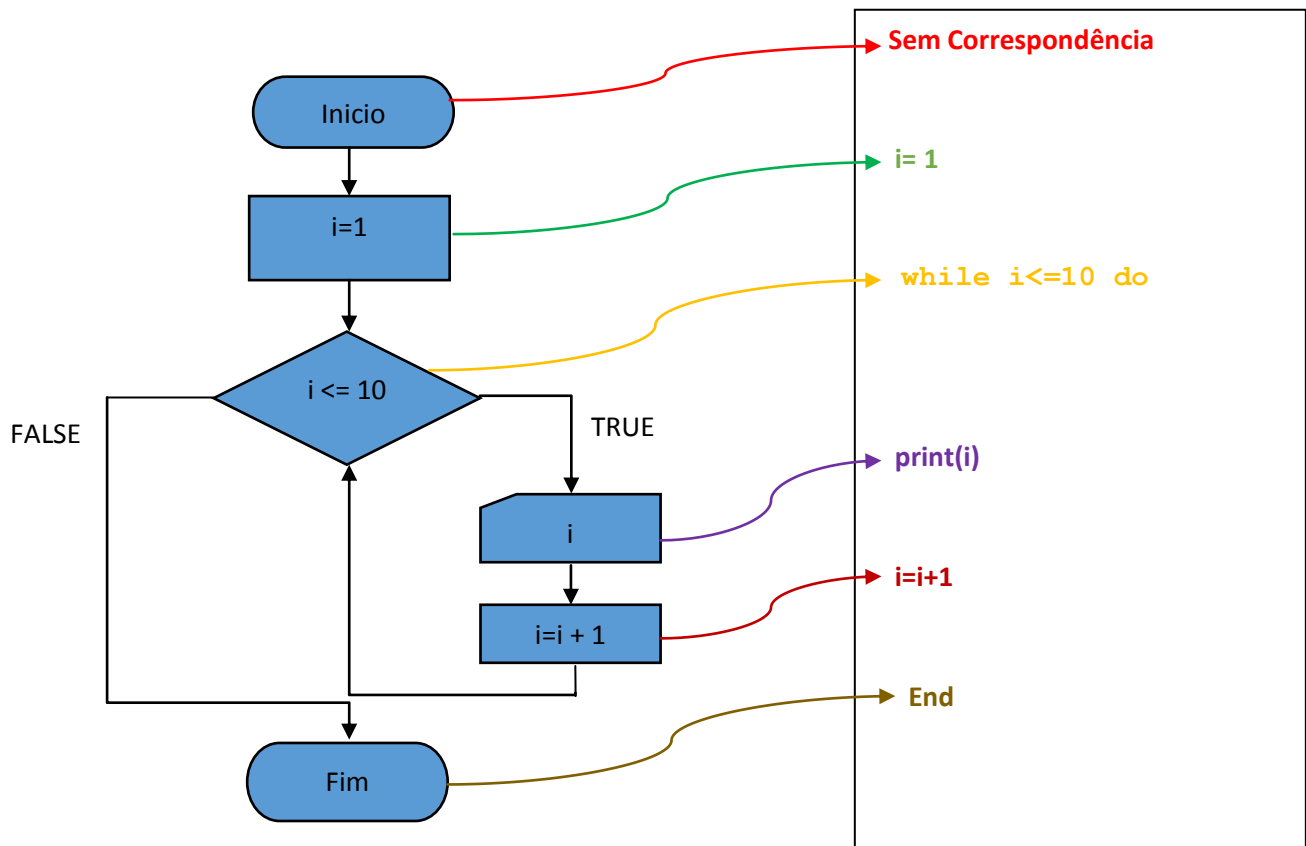
```
while i<=10 do
```

```
    print(i)
```

```
    i=i+1
```

```
end
```

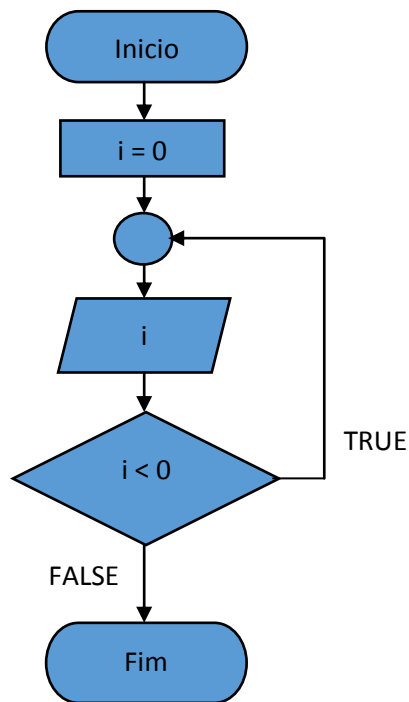
Esquema detalhado



Algoritmo com o uso da condição “do while”

Problema: Pedir um número positivo.

Fluxograma



Código

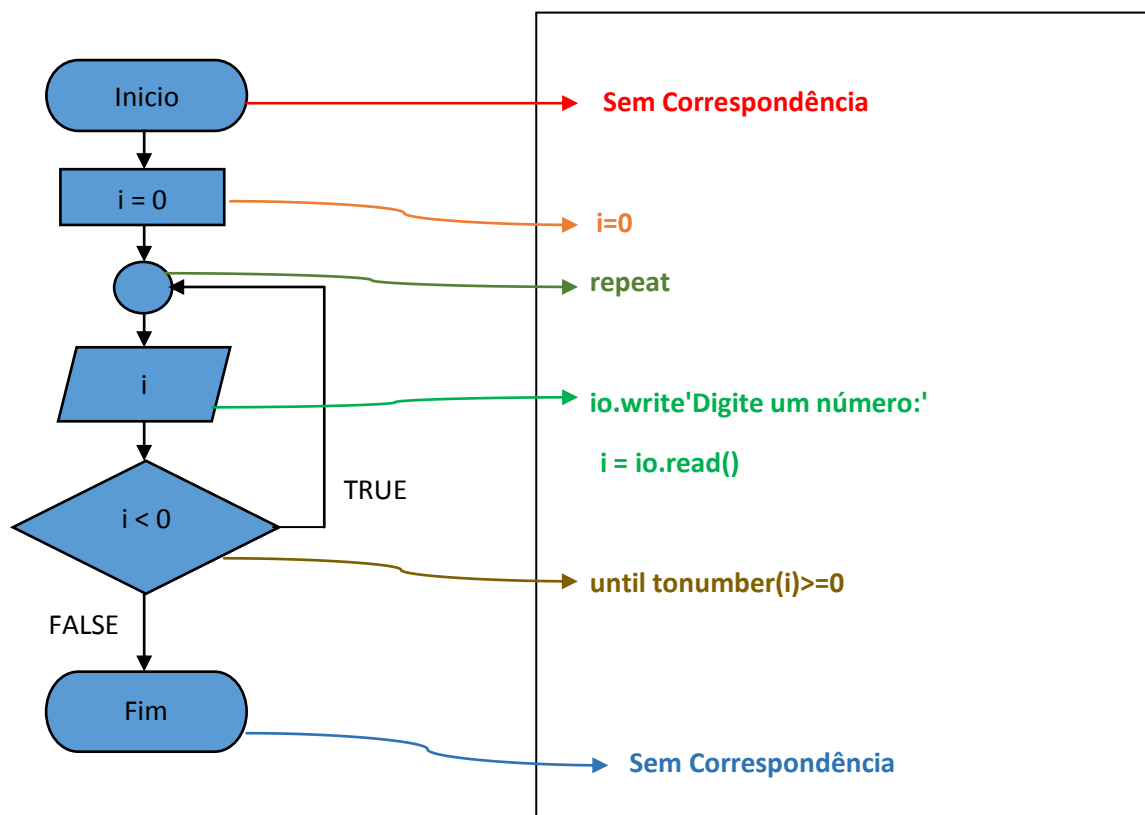
```
i=0

repeat
    io.write'Digite um número: '

    i = io.read()

until tonumber(i)>=0
```

Esquema detalhado

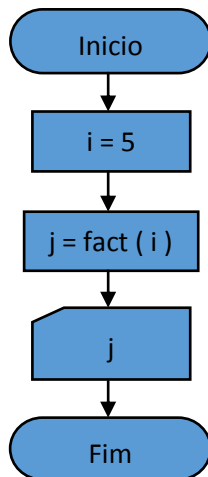


Algoritmo com o uso de uma função

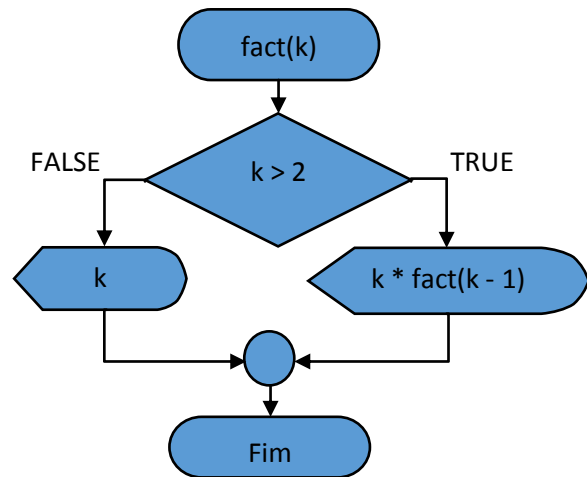
Problema: Factorial de um número.

Fluxogramas

Código principal



Função fact(k)



Código

```
function fact(k)

    if tonumber(k) > 2 then

        return k * fact(k - 1)

    else

        return k

    end

end

i=5

j=fact(i)

print(j)
```

Esquema detalhado

