

A dark blue vertical bar runs down 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 JavaScript

Decode Team

INSTITUTO POLITÉCNICO DE TOMAR

Índice

Nota Geral:	3
Algumas notas sobre JavaScript:	3
Estrutura Início	4
Início:	4
Estrutura Fim	4
Fim:	4
Variáveis	5
Equivalência entre TIPOS de variáveis	5
Definição e atribuição de variáveis	5
Se a variável não estiver definida em memória	5
Se a variável estiver definida em memória	5
Alguns exemplos de definição e atribuição de variáveis	5
Estruturas input/output	6
Input – Ler	6
Ler do teclado	6
Output – Escrever	6
Estruturas de Decisão	7
Condição “if” e “if else”	7
Exemplos práticos	8
Condição “while”	9
Condição “do while”	9
Exemplos práticos	10
Estrutura Conector	11
Conector	11
Funções	12
Definir funções	12
Chamada de funções	12
Estrutura de retorno	13
Return	13
Operadores	14
Aritméticos	14
Lógicos	14
Relacionais	14
ANEXO	15

Algoritmo com o uso da condição “if”	15
Fluxograma	15
Código.....	15
Esquema detalhado.....	16
Algoritmo com o uso da condição “if else”	17
Fluxograma	17
.....	17
Código:	17
Esquema detalhado.....	18
Algoritmo com o uso da condição “while”	19
Fluxograma	19
Código.....	19
Esquema detalhado.....	20
Algoritmo com o uso da condição “ do while”	21
Fluxograma	21
Código.....	21
Esquema detalhado.....	22
Algoritmo com o uso de uma função	23
Fluxogramas	23
Código.....	23
Esquema detalhado.....	24

Nota Geral:

O JavaScript é uma linguagem de programação interpretada, pelo que não usa um compilador e deve ser interpretada por browsers.

Algumas notas sobre JavaScript:

- É uma Linguagem dinâmica. Isto significa que a mesma variável pode ser usada como diferentes tipos.
- Para terminar uma linha de código pode ser usado o ponto e virgula (;) ou apenas a mudança de linha.
- As funções podem ser definidas em qualquer zona do código, desde que se encontrem dentro das tags `<script></script>`.

Estrutura Início

Início:



```
<script>
```

Resto do código

```
</script>
```

Estrutura Fim

Fim:



Nota: Só tem equivalência quando são usadas funções

```
}
```

Variáveis

Equivalência entre TIPOS de variáveis

Não existem definições dos tipos de variáveis

Definição e atribuição de variáveis

```
variavel <- expressao
```

Se a variável não estiver definida em memória

As variáveis não necessitam ser definidas, mas para uma melhor compreensão do código deve fazer-se através da expressão:

```
var variavel = expressao
```

Se a variável estiver definida em memória

```
variavel = expressao;
```

Alguns exemplos de definição e atribuição de variáveis

O JavaScript consegue fazer a distinção entre variáveis numéricas e de texto. Isto é necessário quando existem somas em que os valores a somar são inseridos pelo utilizador. Neste caso a linguagem assume o input sempre como string. Para que o input passe a ser assumido como número, deve ser feito um *parseFloat(expressao)*.

Exemplo:

```
a = 1;
```

```
x = prompt("insira valor"); -> este valor, mesmo sendo numérico, vai ser tomado como string.
```

```
b = a + parseFloat(x); -> é necessário este passo para que b = 2 (assumir que utilizador introduziu o valor 1), em vez de 11.
```

Exemplo de funcionamento das variáveis:

```
var x;          // agora x é um tipo indefinido  
var x = 5;      // agora x é um número  
var x = "John"; // agora x é uma String
```

Estruturas input/output

Input – Ler



Ler do teclado

Para ler do teclado é necessário lançar uma prompt (género de alerta com zona de input). É sempre feito da mesma forma:

```
variavel = prompt("expressão");
```

Output – Escrever

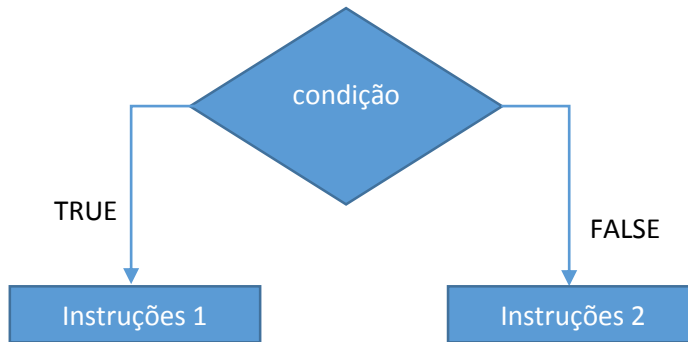


Para escrever no ecrã:

```
document.write("expressao");
```

Estruturas de Decisão

Condição “if” e “if else”



Para TRUE, escrever:

```
If (condição) {  
    Instruções 1
```

Para FALSE:

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

Senão, escrever:

```
    } else {  
        Instruções 2
```


Exemplos práticos

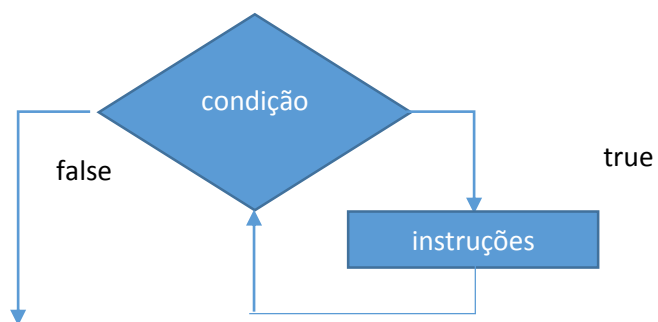
Condição “if”

```
if (n%2==0) {  
    document.write("Par");  
}
```

Condição “if else”

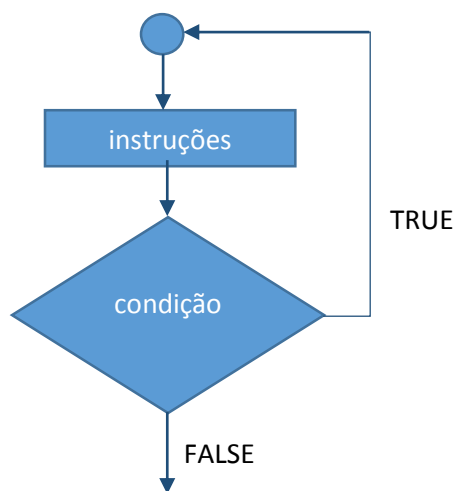
```
if (n % 2 == 0) {  
    document.write ("Par");  
} else {  
    document.write ("Impar");  
}
```

Condição “while”



```
while (condição){  
    Instruções  
}
```

Condição “do while”



```
Instruções  
}while(condição);
```

Exemplos práticos

Condição “while”

```
while(i<=10){  
    document.write (i);  
    i++;  
}
```

Condição “do while”

```
do{  
    i++;  
}while(i<0);
```

Estrutura Conector

Conector



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

```
do {
```

Senão, escrever:

```
}
```

Funções

Definir funções

Exemplo(a , b , ...)

A função pode ser definida em qualquer parte do código, desde que dentro das tags `<script></script>`, e não tem tipo de parâmetros.

```
function Exemplo(a, b, ...)  
{  
}  
}
```

Chamada de funções

variavel = NOME(PARAMETRO)

variavel = NOME(PARAMETRO);

Estrutura de retorno

Return



`return expressao;`

Operadores

Aritméticos

Nome	Portugol	JavaScript
Adição	$a + b$	$a + b$
Subtração	$a - b$	$a - b$
Divisão	a / b	a / b
Multiplificação	$a * b$	$a * b$
Resto da divisão inteira	$a \% b$	$a \% b$
Potenciação	$a ^ b$	$\text{Math.pow}(\text{base}, \text{expoente});$
Concatenação de texto	a, b	$+$

Tabela 1 - Equivalência de operadores aritméticos

Lógicos

Nome	Portugol	JavaScript
Disjunção	$a \text{ E } b$	$a \&\& b$
Conjunção	$a \text{ OU } b$	$a b$
Conjunção Exclusiva	$a \text{ XO } b$	$a ^ b$
Negação	$\text{NAO } a$	$! a$

Tabela 2 - Equivalência de operadores lógicos

Relacionais

Nome	Portugol	JavaScript
Igual	$a = b$	$a == b$
Diferente	$a \neq b$	$a != b$
Maior	$a > b$	$a > b$
Maior ou igual	$a \geq b$	$a \geq b$
Menor	$a < b$	$a < b$
Menor ou igual	$a \leq b$	$a \leq b$

Tabela 3 - Equivalência de operadores relacionais

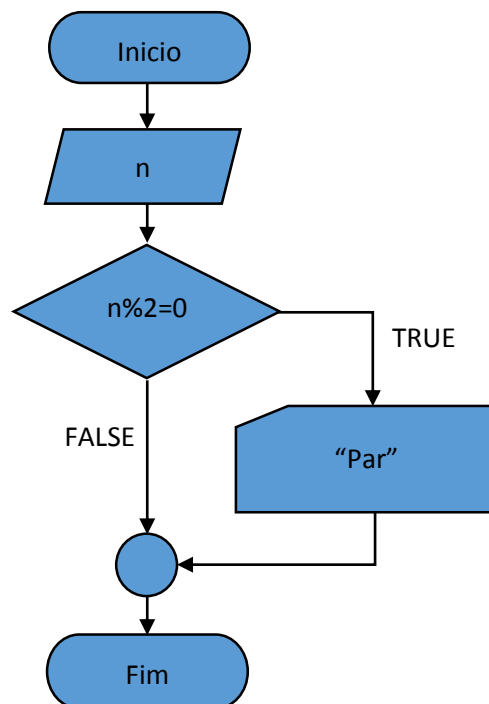
ANEXO

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.

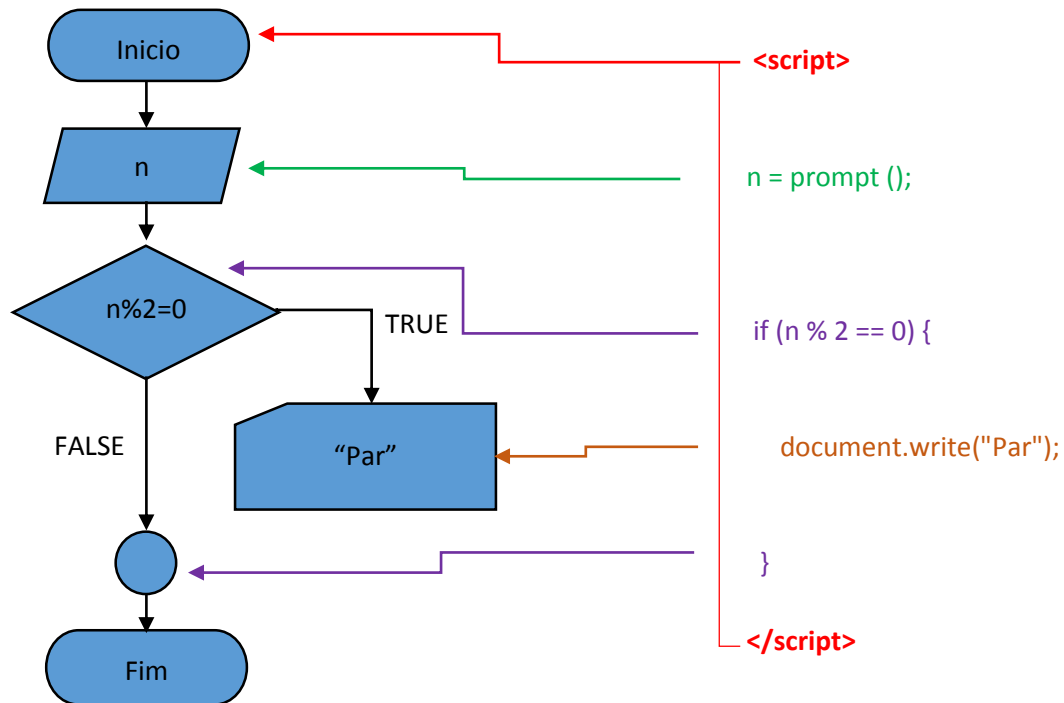
Fluxograma



Código

```
<script>
  n = prompt ();
  if (n % 2 == 0) {
    document.write("Par");
  }
</script>
```

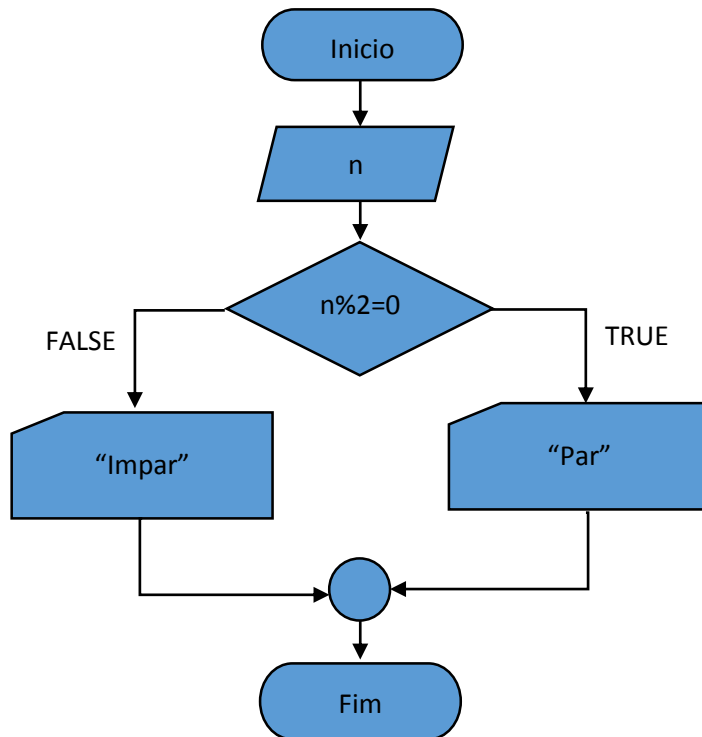

Esquema detalhado



Algoritmo com o uso da condição “if else”

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

Fluxograma



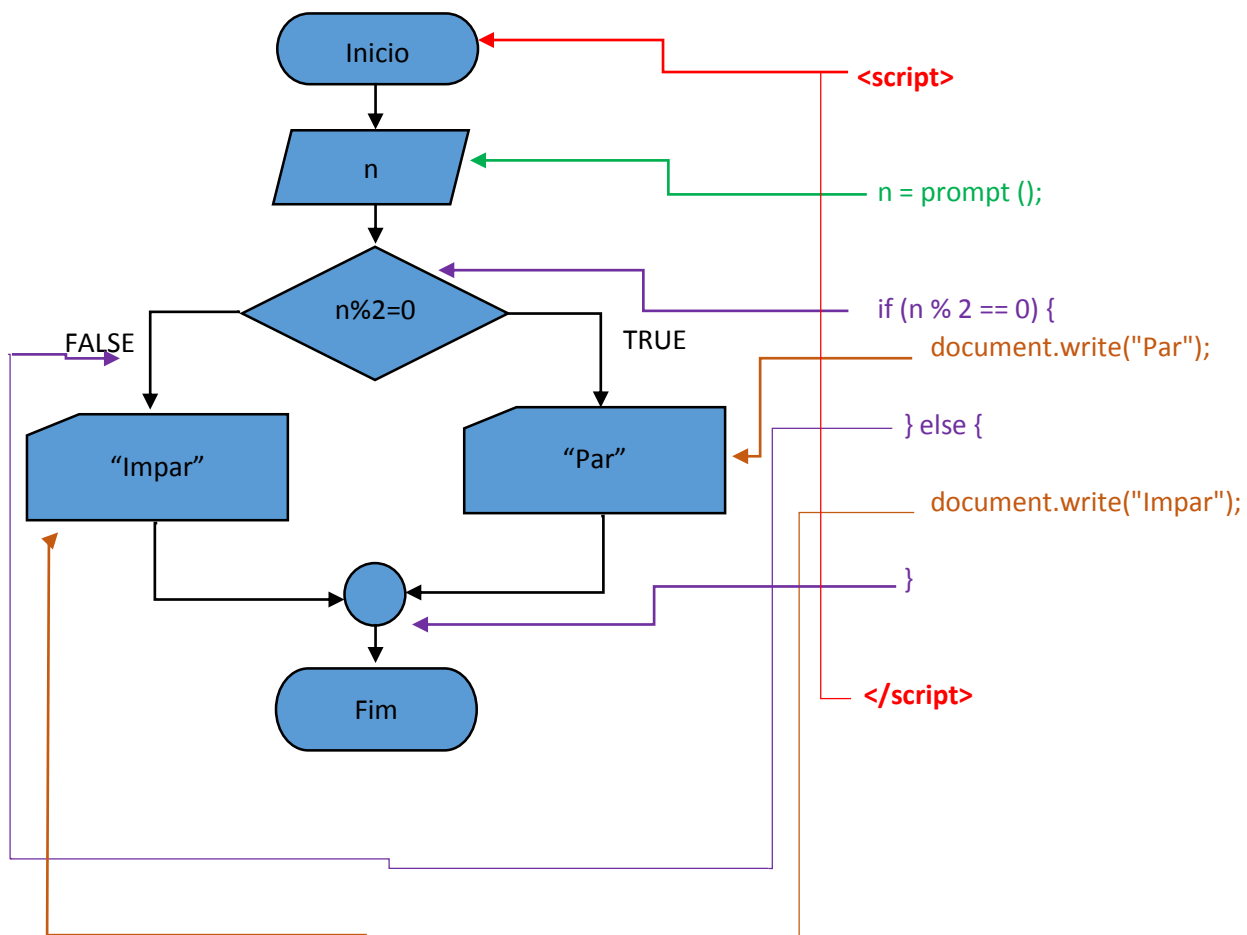
Código:

<script>

```
n = prompt ();  
if (n%2==0) {  
    document.write("Par");  
}  
else{  
    document.write("Impar");  
}
```

</script>

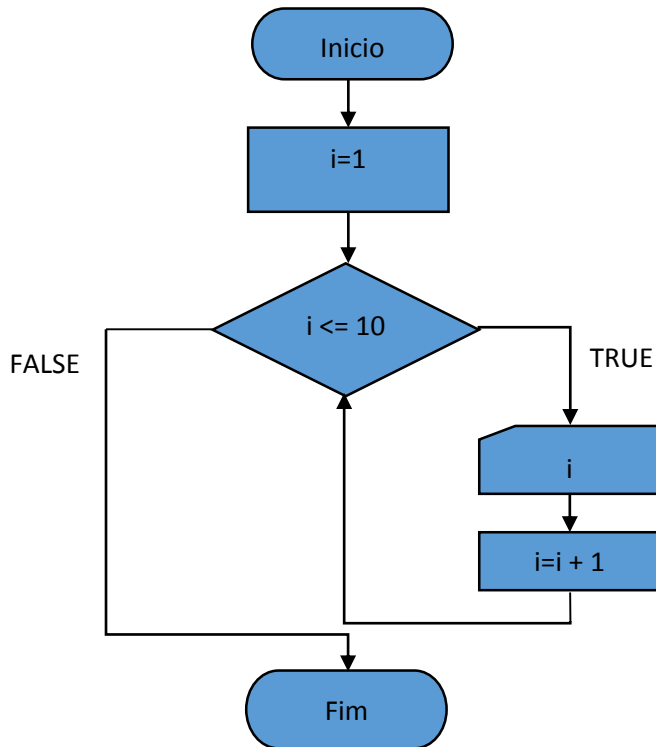
Esquema detalhado



Algoritmo com o uso da condição “while”

Problema: Escrever um número de 1 a 10.

Fluxograma



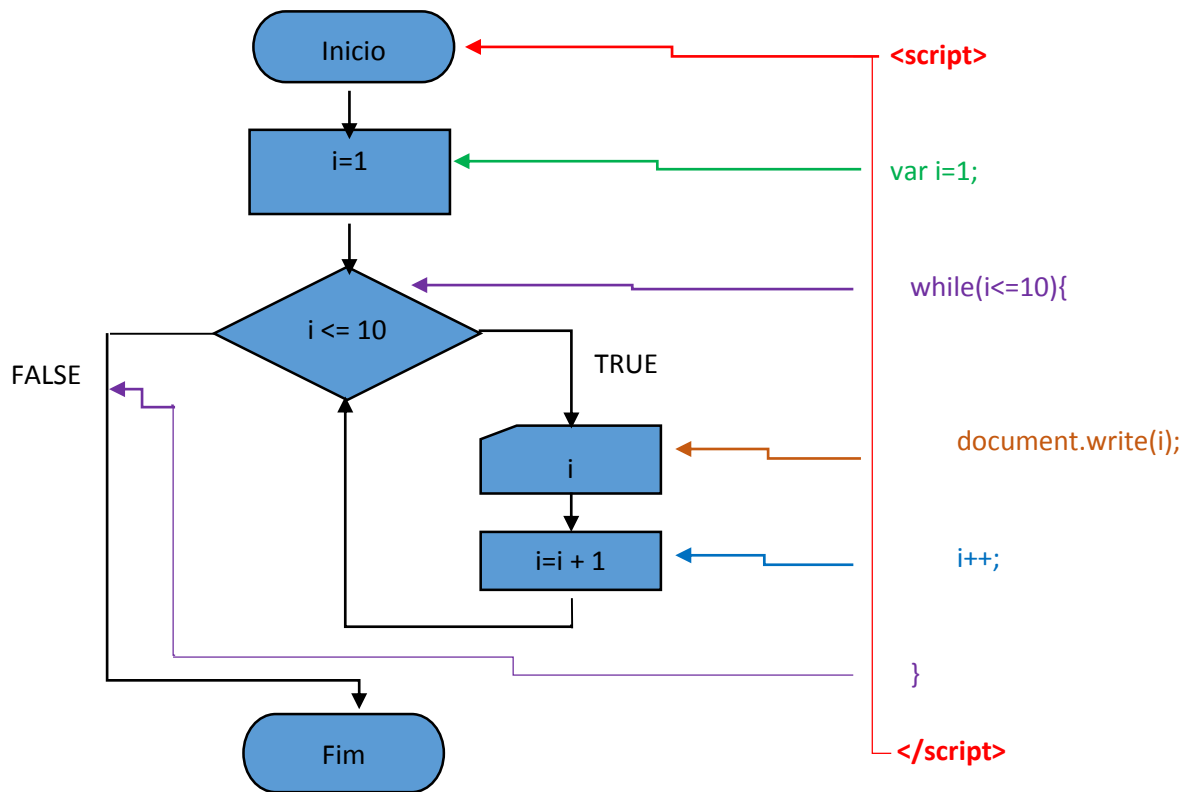
Código

<script>

```
var i=1;
while (i<=10)
{
    document.write(i);
    i++;
}
```

</script>

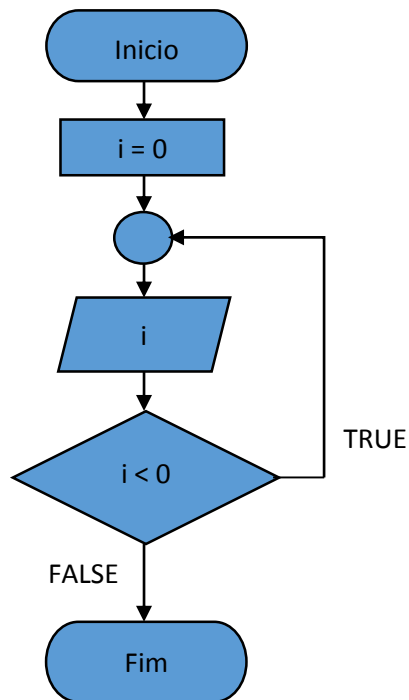
Esquema detalhado



Algoritmo com o uso da condição “do while”

Problema: Pedir um número positivo.

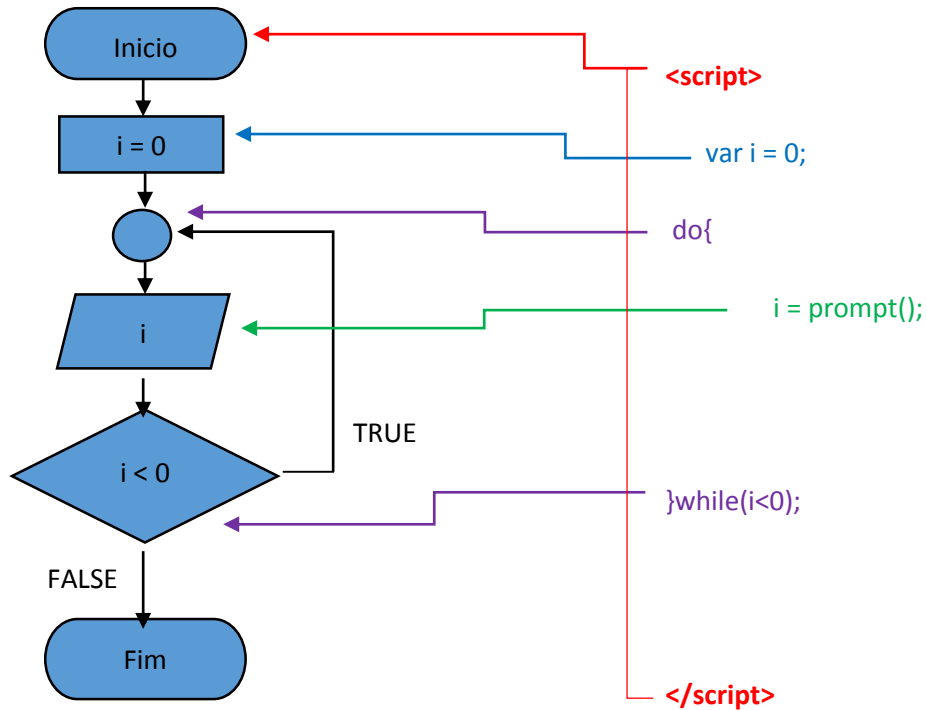
Fluxograma



Código

```
<script>
    var i = 0;
    do
    {
        i = prompt();
    }
    while (i<0)
</script>
```

Esquema detalhado

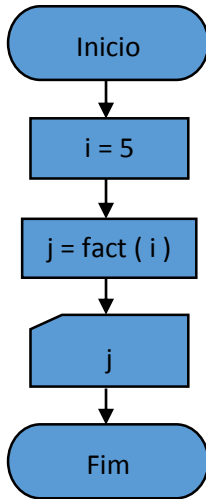


Algoritmo com o uso de uma função

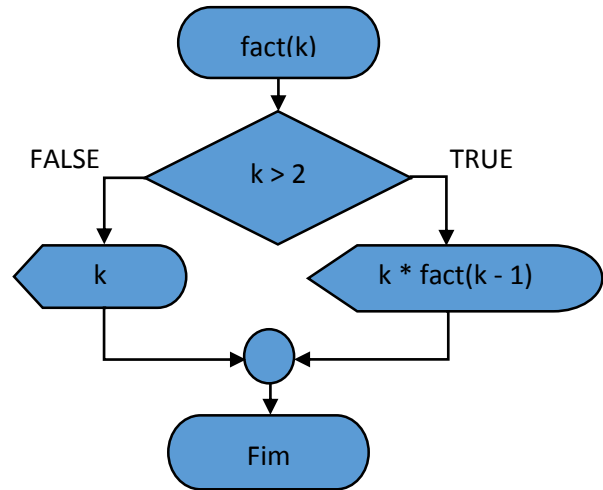
Problema: Factorial de um número.

Fluxogramas

Código principal



Função fact(k)



Código

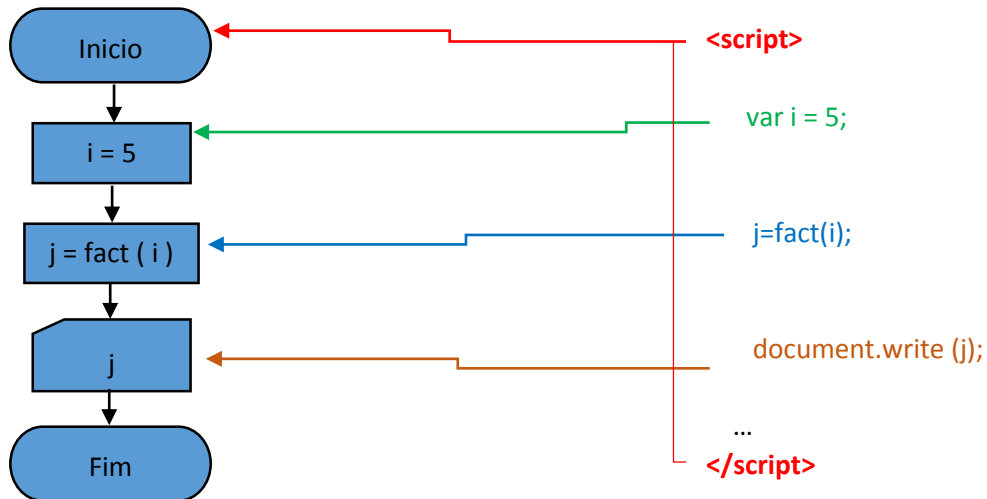
<script>

```
var i = 5;
j = fact(i);
document.write (j);
function fact(k)
{
    if (k>2)
    {
        return (k * fact(k-1));
    }
    else
    {
        return k;
    }
}
```

</script>

Esquema detalhado

Função Main



Função fact

