

Algoritmia e Programação

Instruções de repetição

Organização da apresentação

- Tipos de instruções de repetição em JavaScript:
 - do { ... } while (condição);
 - while (condição) { ... }
 - for (inicialização; condição; atualização) { ... }
 - for...of para percorrer colecções (arrays, strings, etc.)
 - for...in para percorrer propriedades de objetos
- Controlo de ciclos:
 - break, continue
- Boas práticas e erros comuns

Exemplo

Implemente um programa que recolha a idade de 10 pessoas e, no fim da execução, mostre a soma das idades superiores a 20 anos.

```
Variaveis  
    idade, soma: inteiro  
Inicio  
  
    soma <- 0  
  
    ler(idade)  
  
    se idade > 20 entao  
        soma <- soma + idade  
    fim-se  
  
    ler(idade)  
  
    se idade > 20 entao  
        soma <- soma + idade  
    fim-se  
  
    ...  
  
    escrever(soma)  
Fim
```

```
const prompt = require("prompt-sync")();  
  
let soma = 0;  
  
let idade = Number(prompt(`Idade:`));  
if (idade > 20) {  
    soma += idade;  
}  
  
idade = Number(prompt(`Idade:`));  
if (idade > 20) {  
    soma += idade;  
}  
  
idade = Number(prompt(`Idade:`));  
if (idade > 20) {  
    soma += idade;  
}  
  
//....  
console.log(`Soma das idades > 20: ${soma}`);
```

Exemplo

Implemente um programa que recolha a idade de 10 pessoas e, no fim da execução, mostre a soma das idades superiores a 20 anos.

```
const prompt = require("prompt-sync")();

let soma = 0;

for (let i = 0; i < 10; i++) {
    let idade = Number(prompt(`Idade:`));

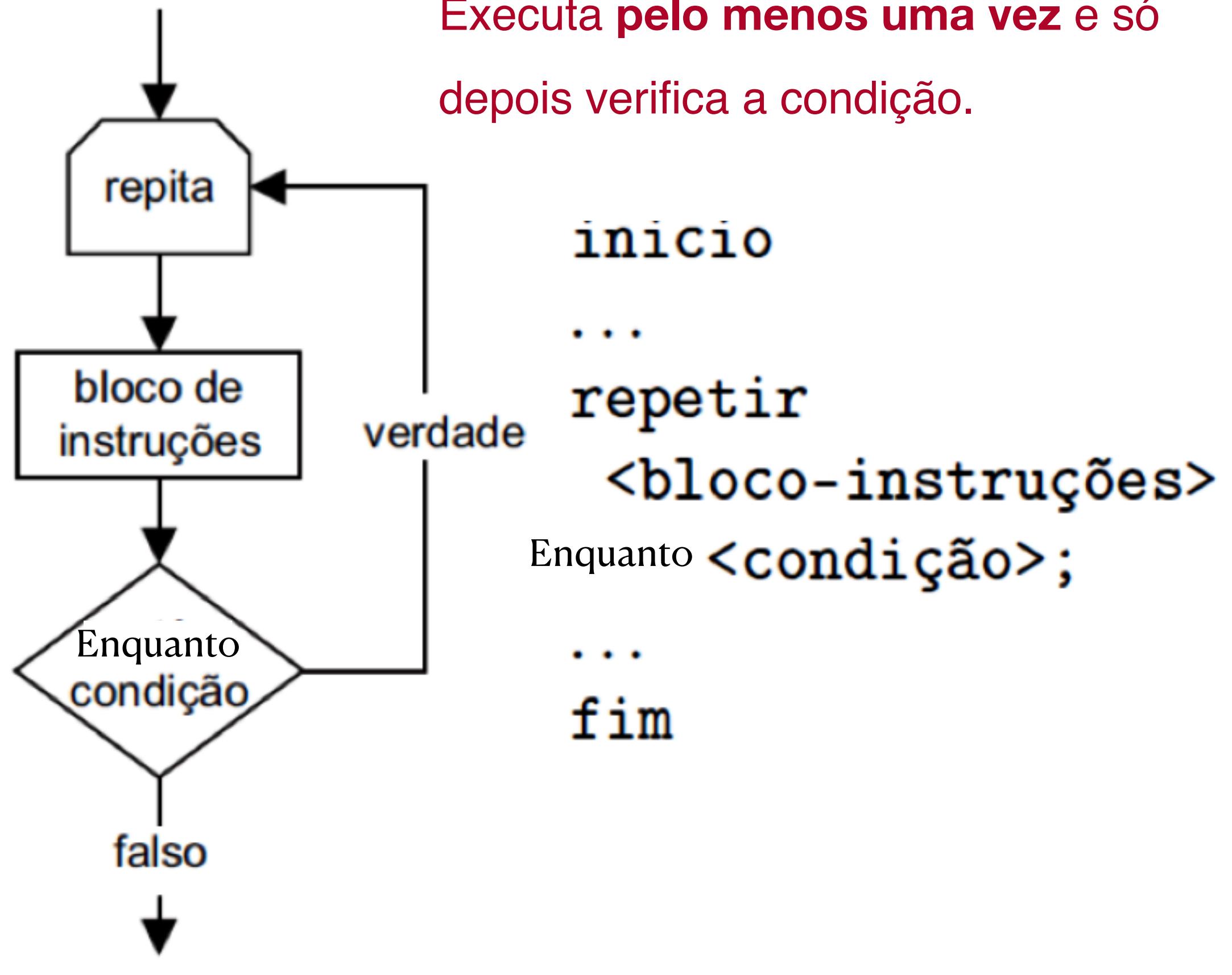
    if (idade > 20) {
        soma += idade;
    }
}

console.log(`Soma das idades > 20: ${soma}`);
```

Instruções de repetição - Conceito geral

- Uma instrução de repetição (ou ciclo) permite executar várias vezes o mesmo bloco de código.
- A repetição é controlada por uma condição lógica (verdadeiro / falso).
- Enquanto a condição for verdadeira, o bloco é executado; quando passa a falsa, o ciclo termina.
- Em JavaScript temos várias formas de repetir:
 - do...while
 - while
 - for
 - for...of, for...in
 - (e ainda métodos como forEach, map, ... sobre arrays)

do...while (repetir..enquanto)



Variaveis

counter: inteiro
Inicio

counter <- 0

repetir

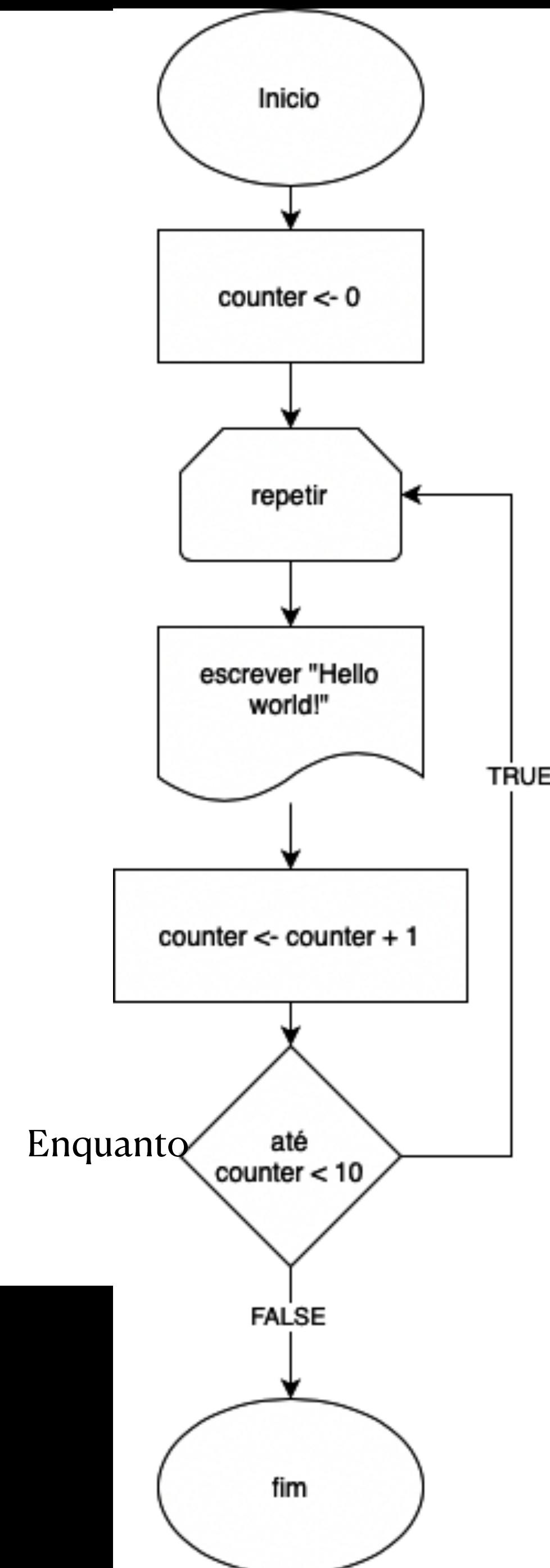
escrever("Hello world\n")
 counter <- counter + 1
enquanto counter < 10

Fim

A execução do bloco de instruções é controlada por uma expressão lógica efetuada no final da execução do bloco. Este facto leva a que o bloco de instruções seja executado pelo menos uma vez.

Repetir..enquanto

```
let counter = 0;  
  
do {  
    console.log("Hello World");  
    counter++;  
} while (counter < 10);
```



Exercício do...while (números pares)

- Desenvolva um programa que recolha 10 números e no final da execução deverá ser mostrado no ecrã a quantidade de números pares inseridos.

```
algoritmo "QuantidadePares"
// Função : Calcular a quantidade de pares naturais inseridos
// Autor : Célio Carvalho
// Versão : 1.0
var
    // declarar variáveis
    contador, numero : inteiro
inicio

    // inicialização de variáveis
    contador <- 0

    // informar condição de paragem
    escreval("insira inteiro negativo para parar.")

repita
    // solicitar novo número
    escreva("insira um numero natural: ")
    leia(numero)

    // verificar cumprimento de regras:
    //     não negativo, não zero e é par
    se (numero > 0) E (numero MOD 2 = 0) entao
        contador <- contador + 1
    fimse

    ate (numero < 0)

    // informar o resultado
    escreval("quantidade de pares inseridos: ", contador)

fimalgoritmo
```

Exercício do...while (números pares)

- Desenvolva um programa que recolha 10 números e no final da execução deverá ser mostrado no ecrã a quantidade de números pares inseridos.

```
const prompt = require("prompt-sync")();

let contadorPares = 0;
let i = 0;

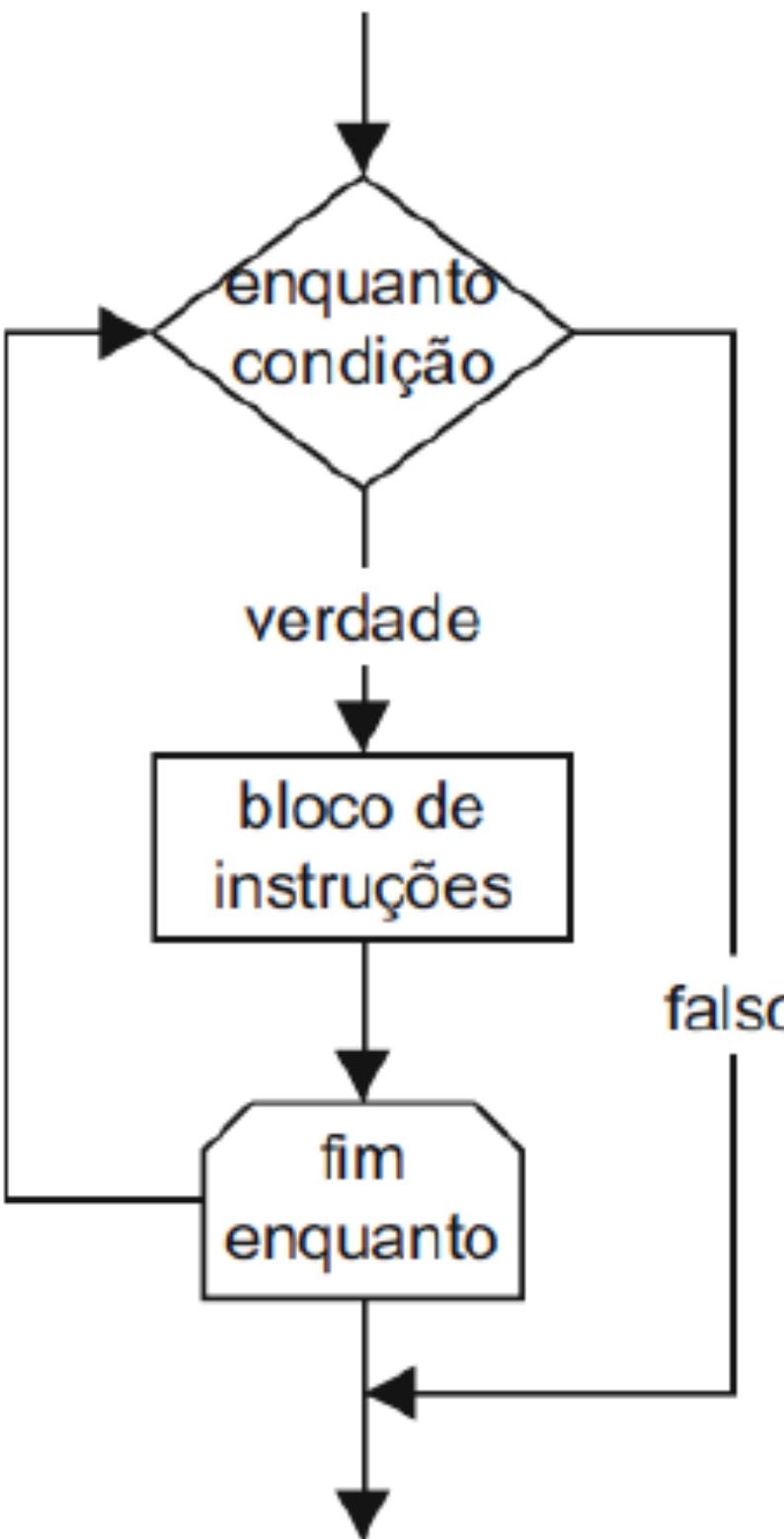
do {
    const numero = Number(prompt(`Insira o número ${i + 1}:`));

    if (numero % 2 === 0) {
        contadorPares++;
    }

    i++;
} while (i < 10);

console.log(`Foram inseridos ${contadorPares} números pares.');
```

while (enquanto..fazer)



Verifica a condição antes de executar o bloco. Pode não executar nenhuma vez.

```
inicio  
...  
enquanto <condição> fazer  
  <bloco-instruções>  
fim-enquanto  
...  
fim
```

Variaveis

counter: inteiro

Início

counter <- 0

```
enquanto counter < 10 fazer  
  escrever("Hello world\n")  
  counter <- counter + 1  
fim enquanto
```

Fim

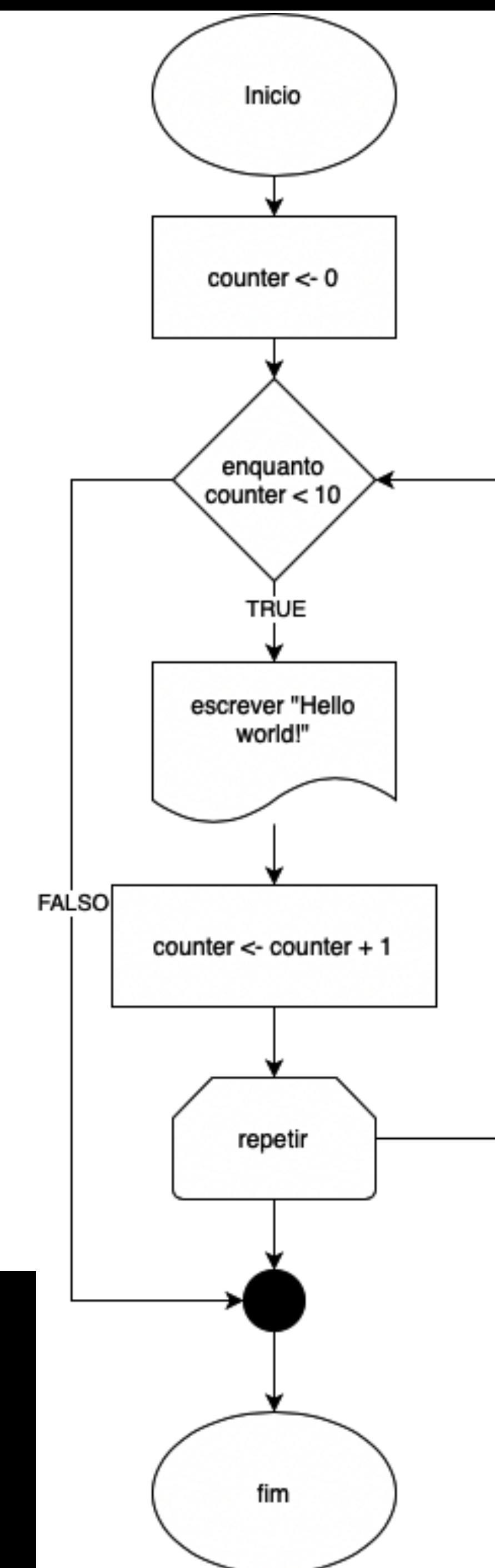
A execução do bloco de instruções é controlada por uma expressão lógica efetuada antes da execução do bloco.

Desta forma o bloco de instruções pode nunca ser executado.

while (enquanto..fazer)

```
let counter = 0;

while (counter < 10) {
    console.log("Hello World");
    counter++;
}
```



while (enquanto..fazer)

O organismo público responsável pela meteorologia em Portugal necessita de fazer um estudo acerca da temperatura na cidade do Porto.

Crie uma aplicação capaz de recolher durante n dias a temperatura, e no final, informar qual o pico de calor ocorrido, assim como a média das temperaturas recolhidas.

Em cada dia apenas será efetuada uma única recolha.

```
algoritmo "LeituraTemperatura"
// Função : calcular media e maximo de leituras inseridas
// Autor : Célio Carvalho
// Versão : 1.0
var
    // variáveis necessárias
    dias : inteiro
    leitura, maximo, acumuladoLeituras : real

início
    // inicialização de valores
    dias <- 0
    maximo <- -999
    acumuladoLeituras <- 0

    // informar condição de paragem
    escreval("para terminar, insira a temperatura 999")

    // primeira leitura
    escreva("Insira a temperatura do dia 1: ")
    leia(leitura)

    // executar ciclo até que inserida a temperatura 999
    enquanto (leitura < 999) faça
        // atualizar variaveis
        dias <- dias + 1
        acumuladoLeituras <- acumuladoLeituras + leitura

        // verificar se existe novo maximo
        se (leitura > maximo) entao
            maximo <- leitura
        fimse

        // solicitar nova temperatura
        escreva("Insira a temperatura do dia ", dias + 1, ": ")
        leia(leitura)

    fimenquanto

    // mostrar resultados
    escreval("RESULTADOS")
    escreval(" > Dias recolhidos: ", dias)

    // tratamento da divisão por zero
    se (dias > 0) entao
        escreval("Temperatura maxima: ", maximo)
        escreval("Temperatura media: ", acumuladoLeituras / dias)
    senao
        escreval("sem leituras")
        escreval("sem leituras")
    fimse

fimalgoritmo
```

while (enquanto..fazer)

O organismo público responsável pela meteorologia em Portugal necessita de fazer um estudo acerca da temperatura na cidade do Porto.

Crie uma aplicação capaz de recolher durante n dias a temperatura, e no final, informar qual o pico de calor ocorrido, assim como a média das temperaturas recolhidas.

Em cada dia apenas será efetuada uma única recolha.

```
const prompt = require("prompt-sync")();

let dias = 0;
let soma = 0;
let maxTemp = -999;

let temperatura = Number(prompt(`Temperatura:`));

while (temperatura != 999) {

    soma += temperatura;

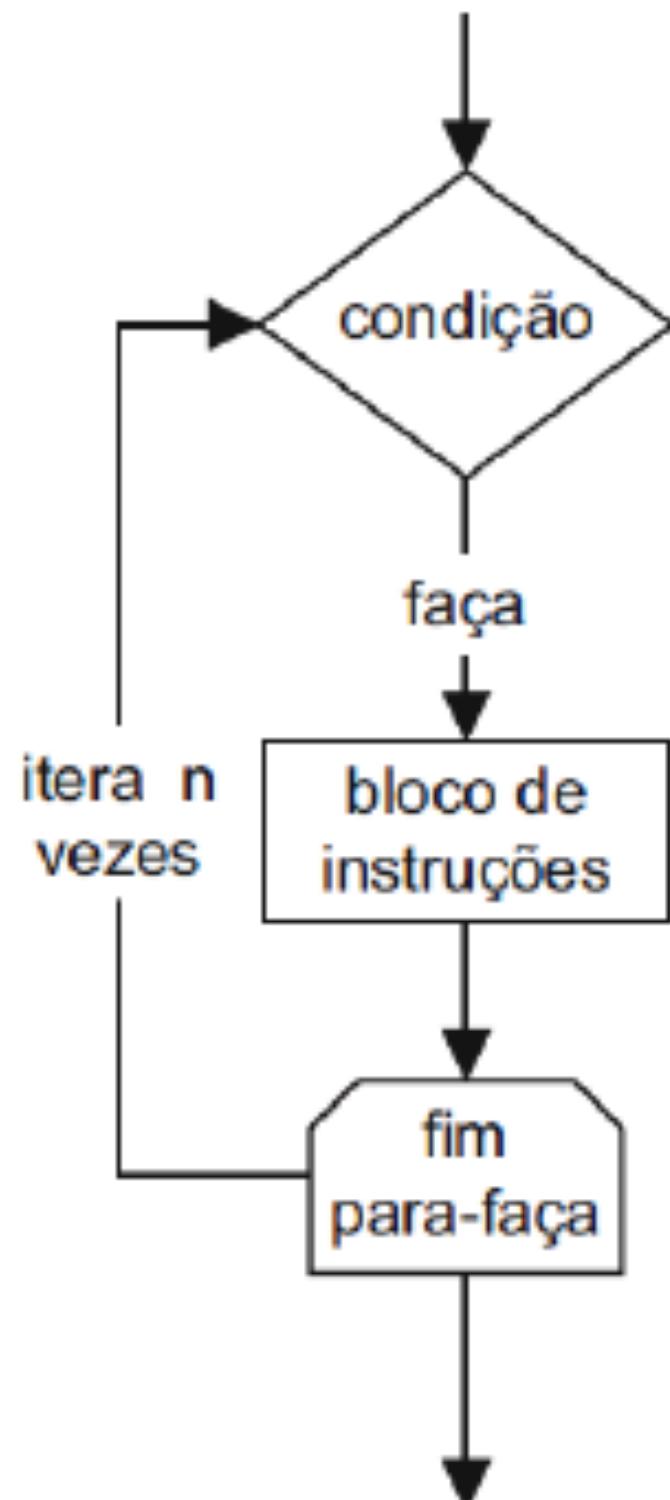
    if (temperatura > maxTemp) {
        maxTemp = t;
    }

    dias++;
    temperatura = Number(prompt(`Temperatura:`));
}

if (dias > 0) {
    const media = soma / dias;

    console.log(`Pico de calor: ${maxTemp} °C`);
    console.log(`Média das temperaturas: ${media.toFixed(2)} °C`);
}
else {
    console.log("Sem leituras")
}
```

for (para..fazer)



Indicado quando sabemos à partida quantas iterações queremos.

inicio

...

para <bloco-início>;<condi

<bloco-instruções>

fim-para

...

fim

Variaveis

i: inteiro

Inicio

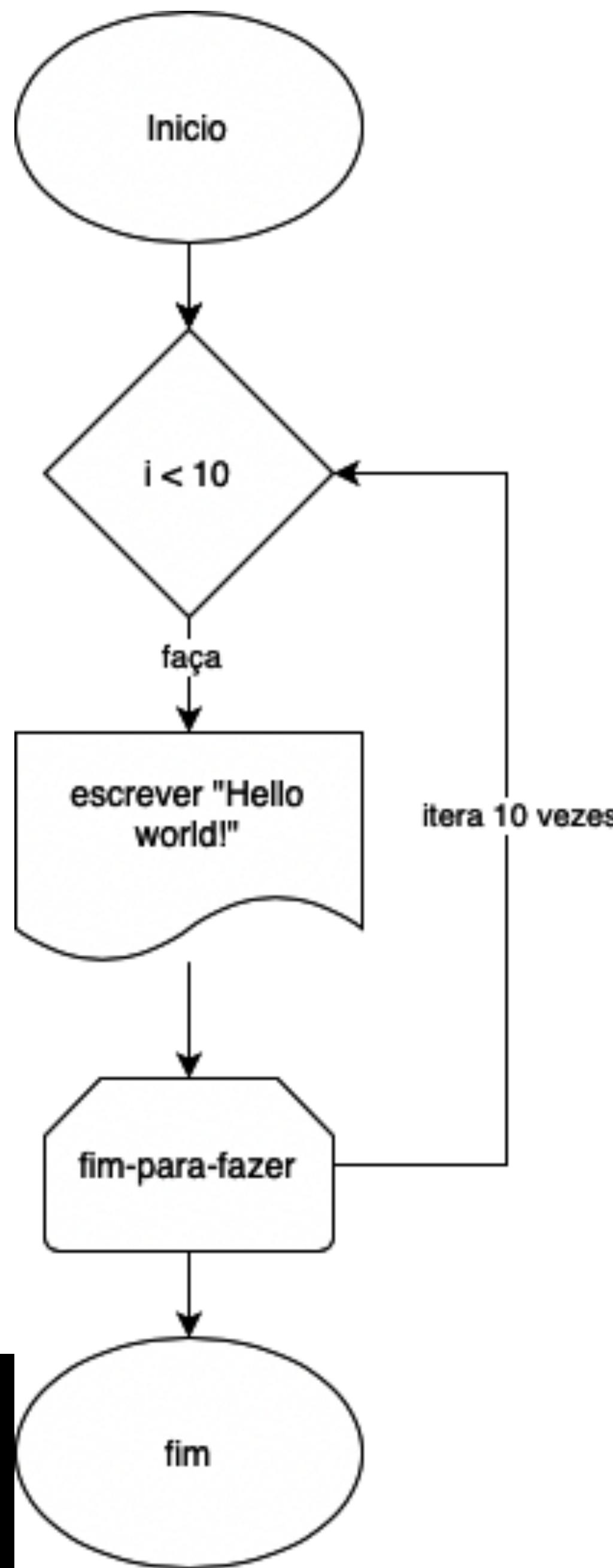
```
para i <- 1 até 10 passo 1 fazer  
    escrever("Hello world\n")  
fim-para
```

Fim

Instrução particularmente indicada para execução de um bloco de instruções repetidamente quando é conhecida à partida o número de vezes que se pretende executar esse bloco.

for (para..fazer)

```
for (let i = 0; i < 10; i++) {  
    console.log("Hello World");  
}
```



Exercício exemplo for (para..fazer)

- Desenvolva um programa capaz de calcular o factorial de um dado numero (ver formula seguinte).

$$\text{factorial}(n) = \begin{cases} n = 0 & \rightarrow 1 \\ n \geq 1 & \rightarrow n * \text{factorial}(n - 1) \end{cases}$$

Exemplo: $\text{factorial}(5) = 5 * 4 * 3 * 2 * 1 = 120$

```
algoritmo "Fatorial"
// Função : Calcula o factorial de um numero
// Autor : Célio Carvalho
// Versão : 1.0
var
    // variaveis
    numero, factorial, i : inteiro

inicio

    // Solicitar dados
    escreval("Insira o numero que pretende calcular factorial: ")
    leia(numero)

    // inicializar variaveis
    factorial <- 1

    // calcular o factorial
    para i de 1 ate numero passo 1 faca
        factorial <- factorial * i
    fimpara

    // apresentar resultado
    escreval("o factorial de ", numero, " é ", factorial)

fimalgoritmo
```

Exercício exemplo for (para..fazer)

- Desenvolva um programa capaz de calcular o fatorial de um dado numero (ver formula seguinte).

$$\text{factorial}(n) = \begin{cases} n = 0 & \rightarrow 1 \\ n \geq 1 & \rightarrow n * \text{factorial}(n - 1) \end{cases}$$

Exemplo: `factorial(5)=5*4*3*2*1=120`

```
const prompt = require("prompt-sync")();

const n = Number(prompt("Introduza um número inteiro positivo:"));

let factorial = 1;

for (let i = 1; i <= n; i++) {
    factorial *= i;
}

console.log(` ${n}! = ${factorial}`);
```

break e continue

- break → termina o ciclo imediatamente.
- continue → salta o resto do bloco e passa à próxima iteração.

```
for (let i = 1; i <= 10; i++) {  
    if (i === 5) continue; // salta o 5  
    if (i === 9) break; // pára no 9  
  
    console.log(i);  
}
```

Exercício

- A Associação de estudantes de Escola Secundária “Só Crânios” está a organizar um concurso de matemática com o objetivo de encontrar os 3 melhores alunos à disciplina. A equipa organizadora do concurso necessita de uma aplicação que faça as inscrições de alunos, através da recolha do nome e da nota de 11º ano a matemática. O número de alunos inscritos no concurso nunca poderá ser inferior a 5 e superior a 10. No entanto, as inscrições terminam se a média das notas dos alunos inscritos atingir os 18 valores, desde que cumprido o preceito do mínimo de 5 alunos inscritos. Caso um candidato tenha nota inferior a 14 valores, a sua inscrição é automaticamente recusada.

Boas práticas e erros comuns em JS com ciclos

- Usar sempre let ou const, evitar var (problemas de scope).
- Garantir que a condição do ciclo eventualmente se torna falsa → evitar ciclos infinitos.
- Atenção aos “off-by-one errors”:
 - $i < array.length$ vs $i \leq array.length$.
- Em for...in:
 - Não usar para percorrer arrays (pode apanhar propriedades extra).
 - Preferir for clássico ou for...of.

Algoritmia e Programação

Instruções de repetição