

IOS – Instituto de
Oportunidade Social

Aula JS 05 - Laços de repetição



- > Laço while
- > Laço do-while
- > Laço For

IOS – Instituto de
Oportunidade Social

Laços de repetição



Os **laços de repetição** são comandos utilizados para executar instruções mais de uma vez, ou seja, cria loops de repetição de uma ou mais instruções com um número limitado de vezes. Eles possuem variáveis de controle/acumuladora.

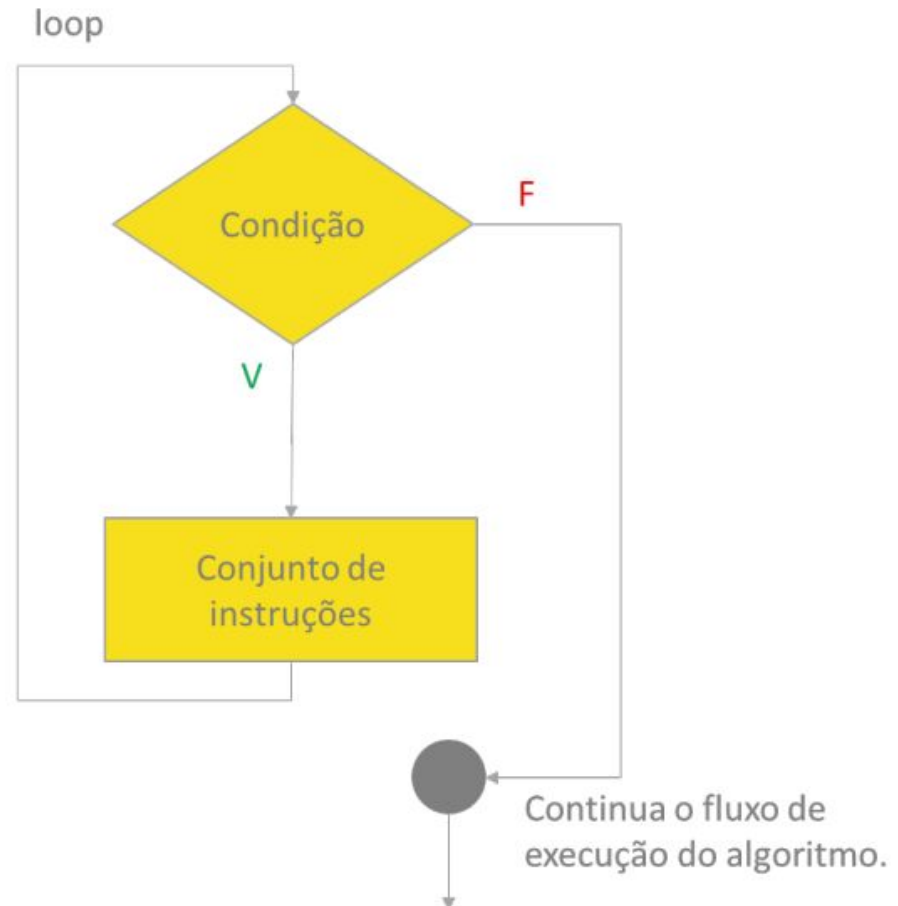
Para trabalhar com laços de repetições, sempre será necessário **testar uma condição que irá resultar em um valor booleano (true ou false)**. Por isso, quase sempre, fazemos usar de **operadores de comparação e/ou lógicos**:

Categoria	Operador	Descrição
Operadores de comparação	==	Igual
	!=	Diferente
	<	Menor que
	<=	Menor ou igual
	>	Maior que
	>=	Maior ou igual
	===	Triplo igual

Categoria	Operador	Descrição
Operadores lógicos	&&	Lógica “and” ou “e”, que retorna verdadeiro se todos os operandos forem verdadeiros.
	 	Lógica “or” ou “ou”, que retorna verdadeiro se pelo menos um operando for verdadeiro.
	!	Lógica “not” ou “não”, que inverte o valor lógico se é verdadeiro, retorna falso e se é falso retorna verdadeiro.

> Laço while

O laço **while** é uma repetição com teste no início do comando. Assim, o bloco de instruções será repetidamente executado enquanto a condição **for verdadeira**. Quando a condição **for falsa**, a execução do bloco de comandos será interrompido:



> Laço while

A sintaxe da estrutura while é:

```
while ( condição )  
{  
    instrucao1;  
    instrucao2;  
    instrucao3;  
    instrucaoN;  
}  
proximaInstrucao;
```


> Laço while

Todo **laço de repetição** pode ter um contador com **variável de controle da condição**. Um contador é uma variável do tipo inteiro usada para contar a quantidade de vezes que um bloco de instruções é repetido e deve ser inicializado antes de ser utilizado.

> Laço while

Inicialização;

```
while(condição){  
    Bloco de instruções;  
    incremento ou decremento;  
}
```

```
let contador = 1;
```

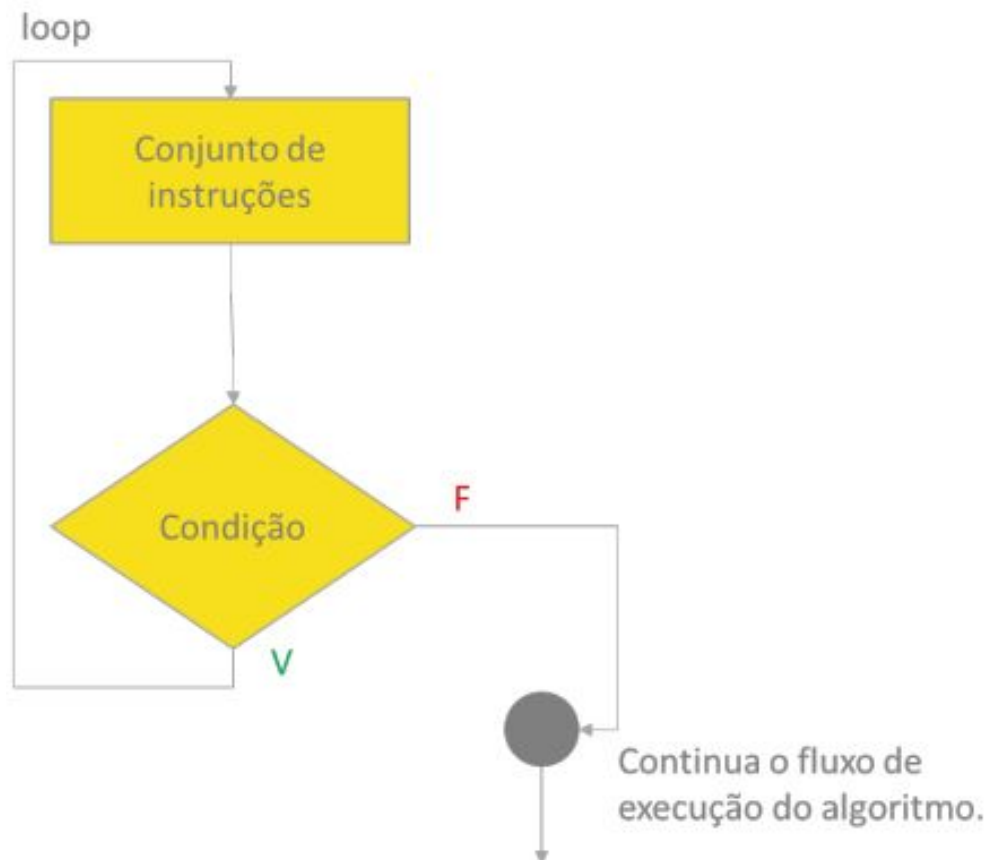
```
while(contador <= 10){  
    console.log(`Valor = ${contador}`);  
    contador ++;  
}
```

> Vamos Praticar

> Laço do-while

Esse tipo de estrutura de repetição é caracterizado por fazer o teste de controle no final do bloco de comandos. Os comandos repetidos **são executados pelos menos uma vez antes da condição ser testada**. A condição é sempre testada no final do laço e após a execução do bloco de comandos dele

> Laço do-while



> Laço do-while

A diferença básica para o comando while é que o bloco a ser repetido sempre executa ao menos uma vez. A sintaxe do do-while é:

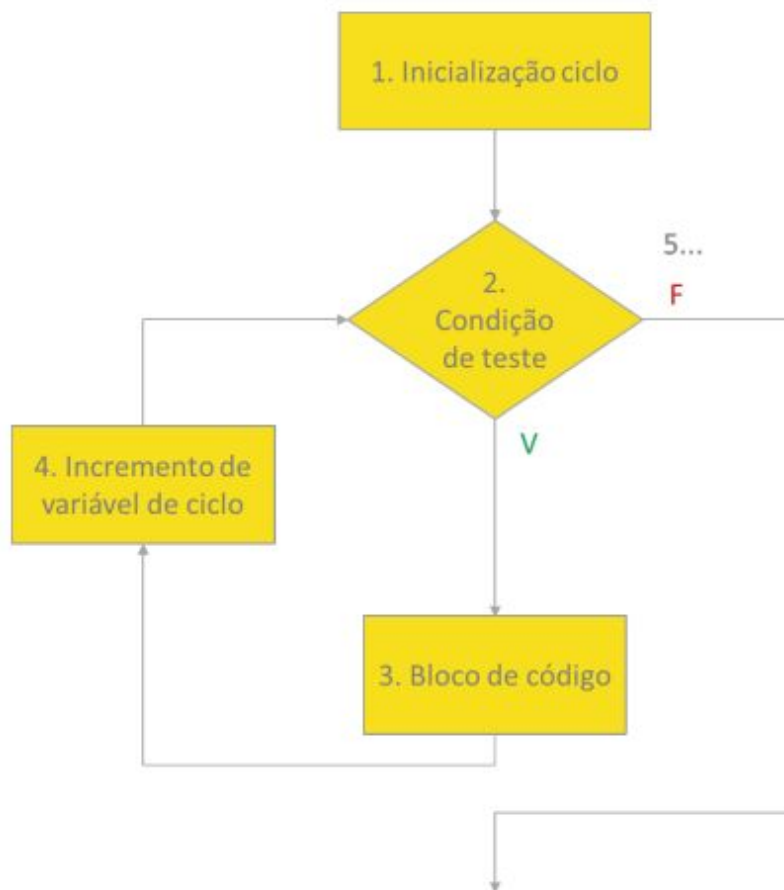
```
do{  
    instrucao1;  
    instrucao2;  
    instrucao3;  
    instrucaoN;  
} while( condição );  
proximaInstrucao;
```

> Vamos Praticar

> Laço For

O comando **for** também realiza o teste lógico no início do **laço**. Ele tem uma estrutura um pouco diferente do while, mas sua execução é similar. O programa não executará nenhuma repetição (ações programadas) sem antes testar a condição:

> Laço For



> Laço For

A sintaxe do comando for é:

```
1)      2)      3)  
for(inicialização; condição; incremento){  
    instrução1;  
    instrução2;  
    instruçãoN;  
}  
proximaInstrução;
```


> Laço For

No comando for temos:

`for` (Inicialização `let i = 0;` Condição `i < 10;` Incremento `i++`)

IOS – Instituto de
Oportunidade Social

Exercício



Em um arquivo `main.js` você terá que criar um algoritmo utilizando laço de repetição e estrutura condicional para criar uma tabuada, seguindo as seguintes regras:

Se a tabuada for 2, imprima a tabuada do número 2.

Se a tabuada for 3, imprima a tabuada do número 3.