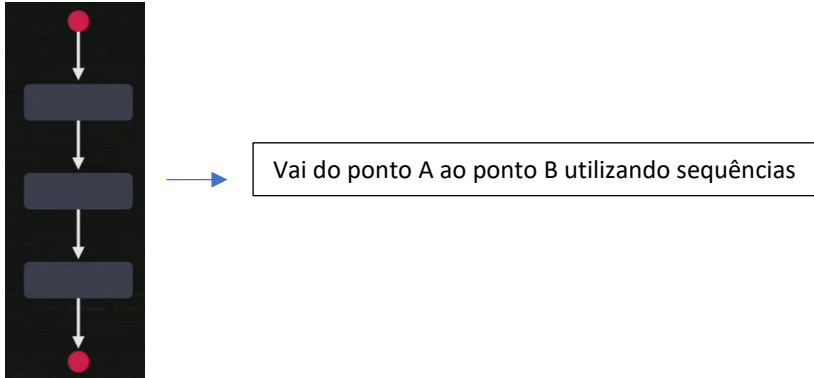


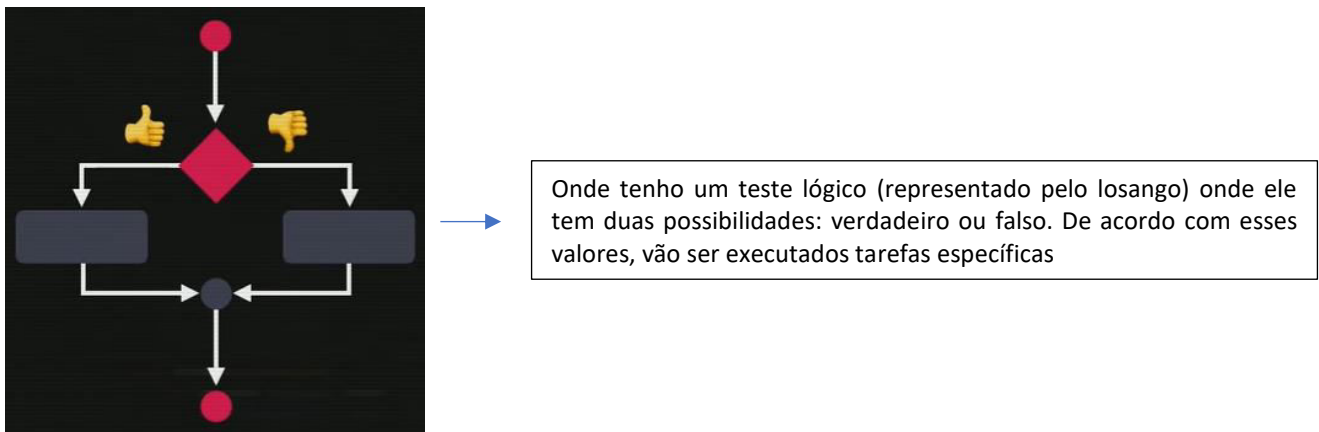
AULA 13 – REPETIÇÕES (PARTE 1)

→ Existem algumas **estruturas de controle**

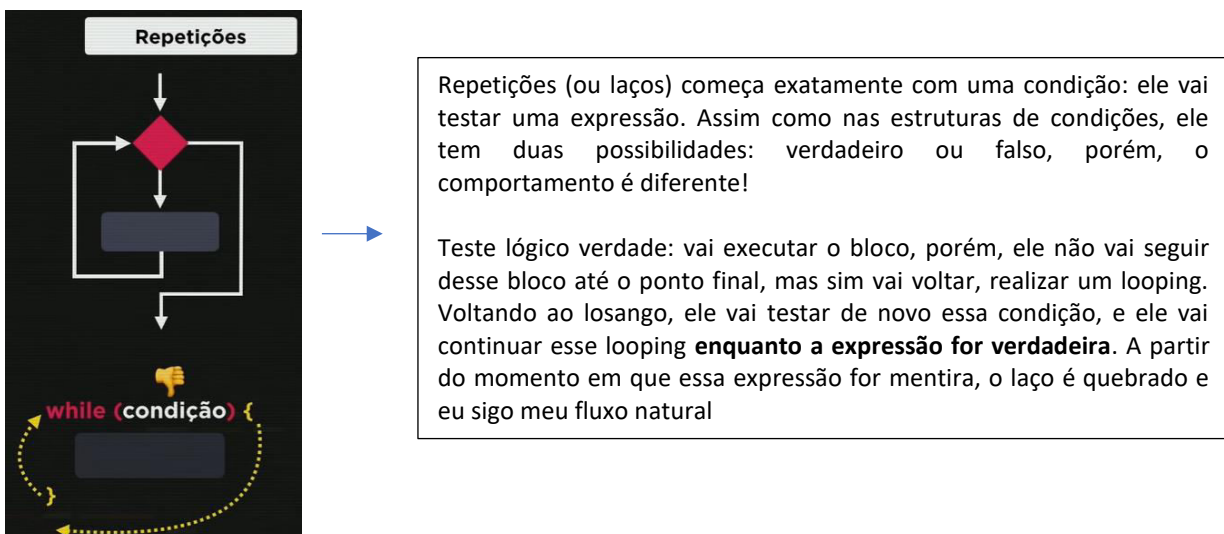
SEQUÊNCIAS



CONDIÇÕES



REPETIÇÕES



→ Essa estrutura é classificada como **estrutura de repetição com teste lógico no início**

Exemplo da pizza:

- Por repetição



No momento que não tiver mais fatia, acaba-se o looping e termina a função comerPizza().

- Por sequência

```
function comerPizza() {  
  comerFatia()  
  comerFatia()  
  comerFatia()  
  comerFatia()  
  comerFatia()  
  comerFatia()  
  comerFatia()  
  comerFatia()  
}
```

Esse exemplo só serve quando eu tenho 8 fatias!

Exemplos no Visual Studio Code:

```
var c = 1  
while (c < 6) {  
  console.log('Tudo bem')  
  c++  
}
```

Info: Start process (2:07:55 PM)
Tudo bem
Tudo bem
Tudo bem
Tudo bem
Tudo bem
Tudo bem
Info: End process (2:07:55 PM)

→ Aqui eu criei uma variável c que seria meu contador começando com o valor 1

→ Enquanto esse contador for menor do que 6, eu escrevo na tela “Tudo bem” e após esse contador vai receber mais um

→ Quando esse contador for maior ou igual a 6, esse comando termina, por isso eu escrevo na tela “Tudo bem” um total de 5 vezes

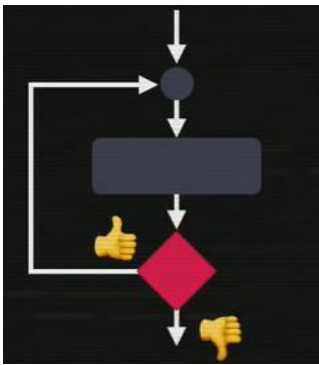
→ Se eu quisesse escrever “Tudo bem” 6 vezes, eu poderia fazer esse contador começar com 0 ou então escrever a expressão em `c <= 6`

```
var c = 1  
while (c <= 6) {  
  console.log(`Passo ${c}`)  
  c++  
}
```

Info: Start process (2:13:23 PM)
Passo 1
Passo 2
Passo 3
Passo 4
Passo 5
Passo 6
Info: End process (2:13:23 PM)

→ Além dessa estrutura com teste lógico no início, eu posso também fazer a estrutura ao contrário: primeiro eu executo o bloco e depois eu faço o teste

→ **Estrutura de repetição com teste lógico no final**



```
do {  
    [Process Block]  
} while (condição)
```

→ Ambos os exemplos são essencialmente estruturas de repetição, porém, tem **comportamentos diferentes**

Exemplos no Visual Studio Code:

<pre>var c = 1 do { console.log(`Passo \${c}`) c++ } while (c <= 6)</pre>	<pre>Info: Start process (2:20:57 PM) Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Passo 6 Info: End process (2:20:57 PM)</pre>
--	--