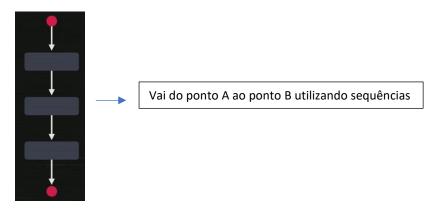
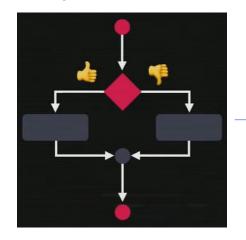
AULA 13 - REPETIÇÕES (PARTE 1)

→ Existem algumas estruturas de controle

SEQUÊNCIAS

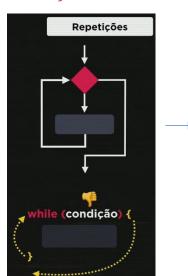


CONDIÇÕES



Onde tenho um teste lógico (representado pelo losango) onde ele tem duas possibilidades: verdadeiro ou falso. De acordo com esses valores, vão ser executados tarefas específicas

REPETIÇÕES



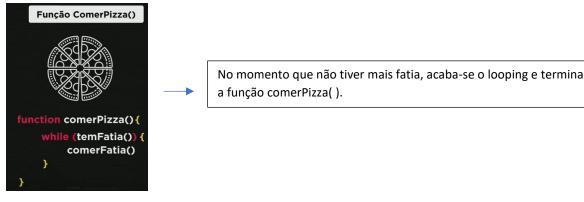
Repetições (ou laços) começa exatamente com uma condição: ele vai testar uma expressão. Assim como nas estruturas de condições, ele tem duas possibilidades: verdadeiro ou falso, porém, o comportamento é diferente!

Teste lógico verdade: vai executar o bloco, porém, ele não vai seguir desse bloco até o ponto final, mas sim vai voltar, realizar um looping. Voltando ao losango, ele vai testar de novo essa condição, e ele vai continuar esse looping **enquanto a expressão for verdadeira**. A partir do momento em que essa expressão for mentira, o laço é quebrado e eu sigo meu fluxo natural

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

Exemplo da pizza:

- Por repetição



- Por sequência

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

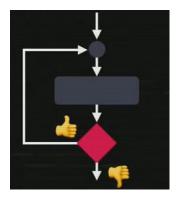
Exemplos no Visual Studio Code:

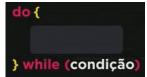
```
var c = 1
while (c < 6) {
    console.log('Tudo bem')
    c++
}
Info: Start process (2:07:55 PM)
Tudo bem
Tudo bem
Tudo bem
Tudo bem
Tudo bem
Tudo bem
Info: End process (2:07:55 PM)</pre>
```

- → Aqui eu criei uma variável c que serie 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
- \rightarrow Se eu quisesse escrever "Tudo bem" 6 vezes, eu poderia fazer esse contador começar com 0 ou estão escrever a expressão em c <= 6

```
Var c = 1
While (c <= 6) {
    console.log(`Passo ${C}`)
    C++
}</pre>
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





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

Exemplos no Visual Studio Code:

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

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>
```