

# Kotlin

## Fluxo de Execução



Softblue  
cursos online

---

---

---


---

---

---

---

### Tópicos Abordados



- Estruturas de controle
  - *if / else*
  - *when*
- Estruturas de repetição
  - *for*
  - *while*
  - *do..while*
- Desvios de fluxo

---

---

---


---

---

---

---

### Estrutura *if / else*



- Funciona da forma como você já conhece
- Em Kotlin, a estrutura *if / else* é uma expressão, portanto pode retornar um valor

```
var situacao: String
var nota: Double = 8.0

if (nota >= 7) {
    situacao = "Aprovado"
} else {
    situacao = "Reprovado"
}
```

```
var nota: Double = 8.0
var situacao: String = if (nota >= 7) {
    "Aprovado"
} else {
    "Reprovado"
}
```

```
var nota: Double = 8.0
var situacao: String = if (nota >= 7) "Aprovado" else "Reprovado"
```

---

---

---

---

---

---

---

## Estrutura *when*



- Substitui a estrutura *switch* do Java

```
var x: Int = 5;

when (x) {
  0 -> print("Valor é 0")
  1 -> print("Valor é 1")
  else -> print("Outro valor")
}
```

```
var x: Int = 0

var b: Boolean = when (x) {
  0 -> false
  else -> true
}
```

```
var x: Int = 5;

when (x) {
  0, 1 -> print("Valor é 0 ou 1")
  in 2..10 -> print("Valor está entre 2 e 10")
  else -> print("Outro valor")
}
```

---

---

---

---

---

---

---

## Estrutura *for*



- Funcionamento similar ao Java

```
var numbers: IntArray = intArrayOf(1, 3, 5, 7)

for (i in numbers.indices) {
  print(numbers[i]);
}
```

```
var numbers = listOf(1, 3, 5, 7)

for (n in numbers) {
  print(n);
}
```

```
for (i in 1..10 step 2) {
  print(i);
}
```

---

---

---

---

---

---

---

## Estruturas *while* e *do..while*



- Funcionam de forma igual ao Java

```
var x: Int = 0;

while (x < 10) {
  print(x);
  x++;
}
```

→ O teste é feito no início do bloco

```
var x: Int = 0;

do {
  print(x);
  x++;
} while (x < 10)
```

→ O teste é feito no fim do bloco

---

---

---

---

---

---

---

## Desvios de fluxo



- Kotlin suporta desvios de fluxo
- Funciona de forma semelhante ao Java
- Expressões
  - *return*
    - Retorna a execução para quem fez a chamada da função
  - *break*
    - Termina o loop corrente
  - *continue*
    - Executa o novo passo do loop corrente

---

---

---

---

---

---

---



---

---

---

---

---

---

---