

DESPERTANDO COM KOTLIN

O SABOR DO PRIMEIRO
CÓDIGO COM CAFÉ



LUIZ AUGUSTO

Capítulo

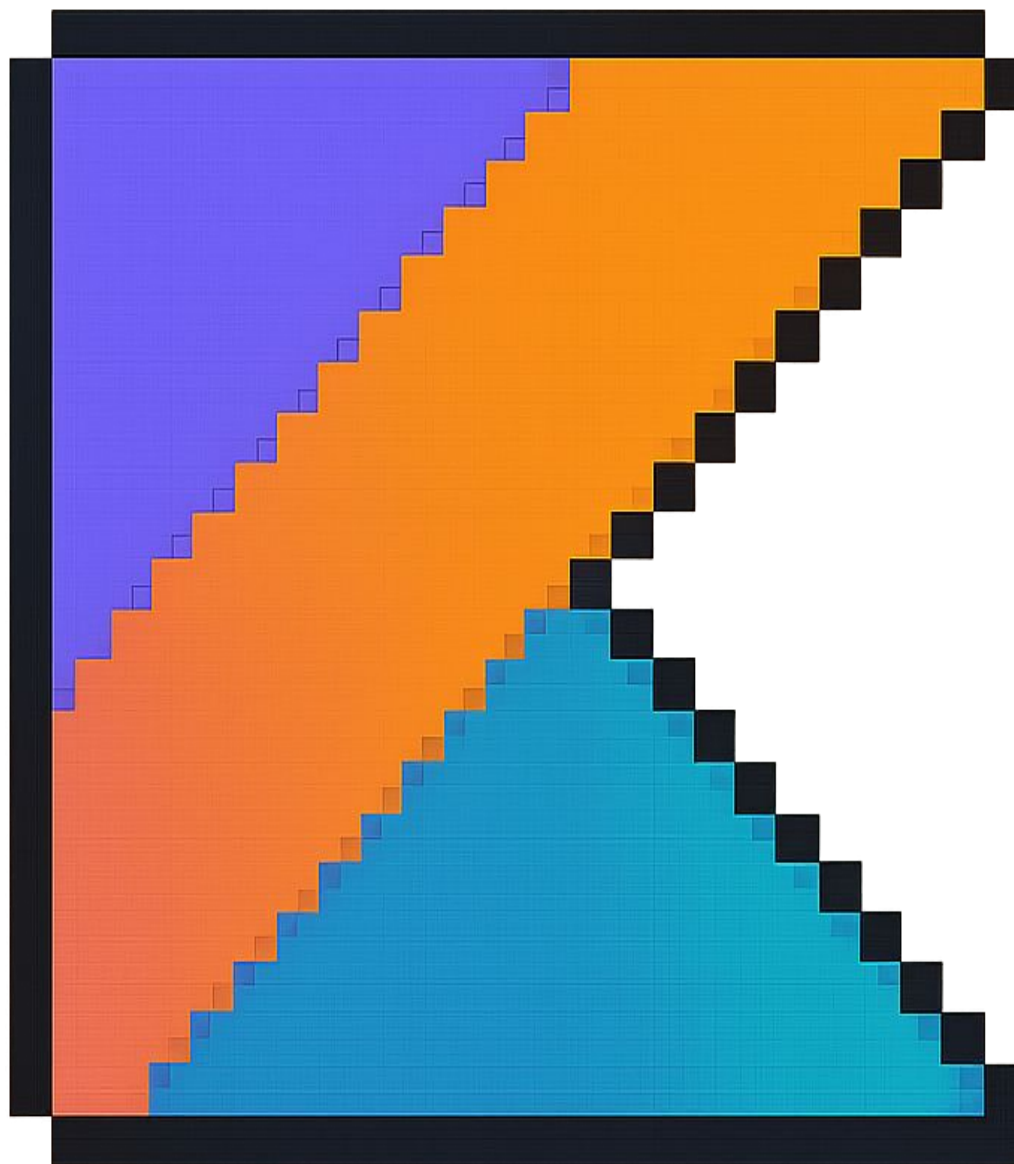
01

A_ORIGEM

De onde veio essa linguagem tão moderna?

A Origem do Kotlin

Kotlin foi criado pela JetBrains para ser uma linguagem moderna, simples e compatível com o Java. Ele surgiu para tornar a programação mais agradável e segura. Desde 2017, é a linguagem oficial para Android, mas vai muito além disso!



Capítulo

02

PRIMEIRO_CÓDIGO

Sua primeira linha de código

Primeiro Código Kotlin

Aprender programação começa com um "Olá, Mundo!". Em Kotlin, esse código é direto e elegante. Com poucas palavras, já vemos como tudo funciona.



main.kt

```
1 fun main() {  
2     println("Olá, Mundo!")  
3 }
```

Com isso, você já criou um programa funcionando!

Capítulo

03

VARIÁVEIS_E_TIPOS

Guardando informações com inteligência

Variáveis e Tipos

Para guardar informações, usamos variáveis. Em Kotlin, usamos `val` para valores fixos e `var` para valores que mudam. Os tipos (como texto, número, etc.) podem ser definidos automaticamente.

main.kt

```
1 val name = "Luiz"      // valor fixo
2 var age = 25           // valor que pode mudar
```

Simples, né?

Capítulo

04

FUNÇÕES

Blocos de código que fazem coisas

Funções

Funções são blocos de código que executam algo. Você pode criar funções simples em uma única linha.



```
1 fun greeting(name: String) = "Olá, $nome!"  
2 println(greeting("Luiz"))
```

Elas ajudam a organizar e reaproveitar o código com facilidade.

Capítulo

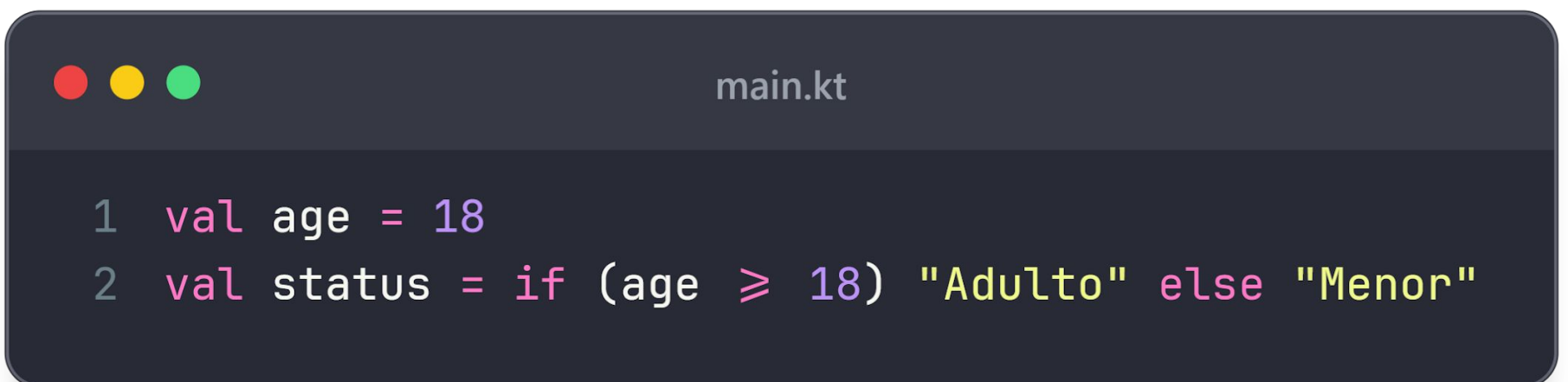
05

CONDIÇÕES

Tomando decisões no código

Condições

Nem tudo é sempre igual. Às vezes, o código precisa tomar decisões. Em Kotlin, usamos `if` e `when` para isso.



```
1  val age = 18
2  val status = if (age ≥ 18) "Adulto" else "Menor"
```

E com `when`, fica ainda mais claro:



```
1  val color = "vermelho"
2  when (color) {
3      "vermelho" → println("Pare")
4      "verde" → println("Siga")
5      else → println("Atenção")
6  }
```

Capítulo

06

LISTAS_E_COLEÇÕES

Guardando vários dados de uma vez

Listas e Coleções

Coleções permitem guardar vários dados juntos, como uma lista de nomes.

```
main.kt  
  
1 val fruits = listOf("Maçã", "Banana", "Uva")  
2 println(fruits[1]) // Banana
```

Você também pode adicionar itens usando listas mutáveis:

```
main.kt  
  
1 val list = mutableListOf("A", "B")  
2 list.add("C")
```

Capítulo

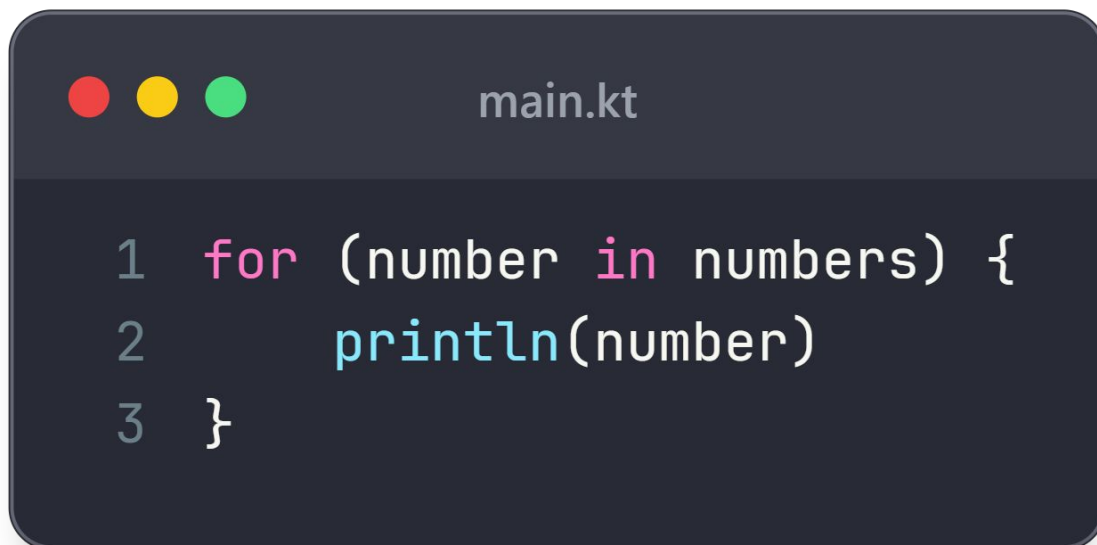
07

LAÇOS_DE_REPETIÇÃO

Fazendo algo várias vezes

Laços de Repetição

E se quisermos repetir algo várias vezes? É aí que entram os laços.



```
1 for (number in numbers) {  
2     println(number)  
3 }
```

Também existe o while e o repeat:



```
1 // repeat  
2 repeat(3) { println("Oi!") }  
3  
4 // while  
5 var number = 0  
6 while (number < 13) {  
7     println(number)  
8 }
```

Capítulo


08

CLASSES_E_OBJETOS

Representando o mundo real no código

Classes e Objetos

Podemos criar "modelos" com classes, e depois gerar objetos a partir delas.



```
1 class Person(val name: String)
2
3 val person = Person("Luiz")
4 println(person.name)
```

Isso ajuda a organizar dados e comportamentos de forma clara.

Capítulo

09

NULL_SAFETY

Evitando erros comuns de forma
inteligente

Null Safety

Kotlin se preocupa com erros causados por valores nulos. Ele te obriga a tratar isso com carinho.



main.kt

```
1 var name: String? = null
2 println(name?.length) // Evita erro!
```

Esse recurso evita muitos bugs comuns em outras linguagens.

Capítulo

10

EXTENSÕES

Dê superpoderes às funções e objetos

Extensões

Você pode adicionar funções a tipos já existentes, sem mexer neles. Parece mágica, mas é Kotlin!



main.kt

```
1 fun String.show() = println(this)
2 "Oi Kotlin!".show()
```

Isso deixa seu código muito mais limpo e poderoso.

Capítulo

11

LAMBDA

Funções como variáveis

Lambdas

Funções podem ser tratadas como valores. Isso é útil para transformar, filtrar e mapear listas, por exemplo.



```
1 val double = { x: Int -> x * 2 }
2 println(double(5)) // 10
```

Kotlin torna o uso de lambdas simples e prático.

Capítulo

12

DATA_CLASSES

Classes prontas para guardar dados

Data Classes

Quer guardar dados? Com data class, o Kotlin gera tudo pra você: toString, equals, hashCode e muito mais!



```
1 data class Product(val name: String, val price: Double)
2
3 val product = Product("Livro", 39.90)
4 println(product)
```

É só escrever e usar.

13

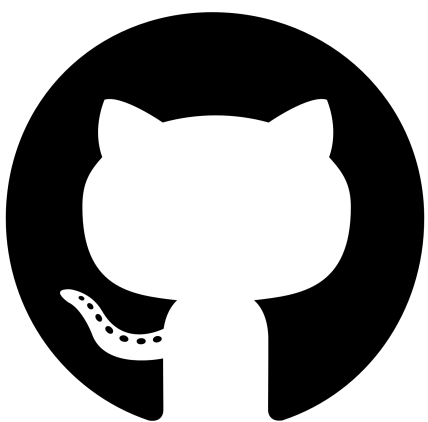
AGRADECIMENTOS

Quem esta comigo nessa?

Agradecimentos

Gostaria de agradecer especialmente ao meu amigo Thiago Kunst, por me motivar e me direcionar no mundo da programação, e também à minha equipe de trabalho atual: Fernando Kanada, Francismar Mariano, Lucas Ribeiro e Gabriel Panza – pessoas excepcionais que contribuíram demais para a minha evolução como dev. Obrigado!

Redes - Conexões



[GitHub](#)



[LinkdIn](#)