



FILIP



# Variáveis

Mobile App Development



# Introdução a Variáveis

## Objetivo

Conhecer variáveis em `Kotlin`, como declará-las, inicializá-las, tipos de dados disponíveis e a importância das variáveis na programação.

## Referência de Memória:

Variáveis apontam para uma área específica da memória onde os valores são armazenados.

## Recipiente de Dados:

Uma variável é um recipiente que armazena dados, identificado por um nome e associado a um tipo.

- Nome: Identificador único que refere-se ao valor armazenado pela variável.
- Tipo: Define o tipo de dados que a variável pode armazenar e as operações permitidas.
- Valor: O dado real armazenado na variável.
- Escopo: A região do código onde a variável pode ser acessada.

# Introdução a Variáveis

## Principais tipos de dados

Tipo	Descrição	Intervalo	Exemplo
Char	Armazena um único caractere	-	'A', 'b'
Int	Armazena números inteiros	-2,147,483,648 a 2,147,483,647 (4 bytes)	42, -100
Long	Armazena números inteiros maiores	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 (8 bytes)	123456789L, -9876543210L
Float	Armazena números de ponto flutuante (precisão simples)	6 a 7 dígitos decimais (4 bytes)	3.14f, 2.71828f
Double	Armazena números de ponto flutuante (precisão dupla)	15 a 16 dígitos decimais (8 bytes)	3.141592653589793, 2.718281828459045

# Introdução a Variáveis

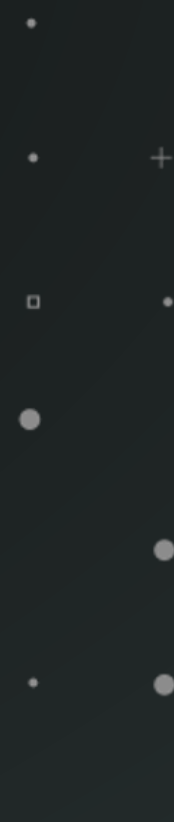
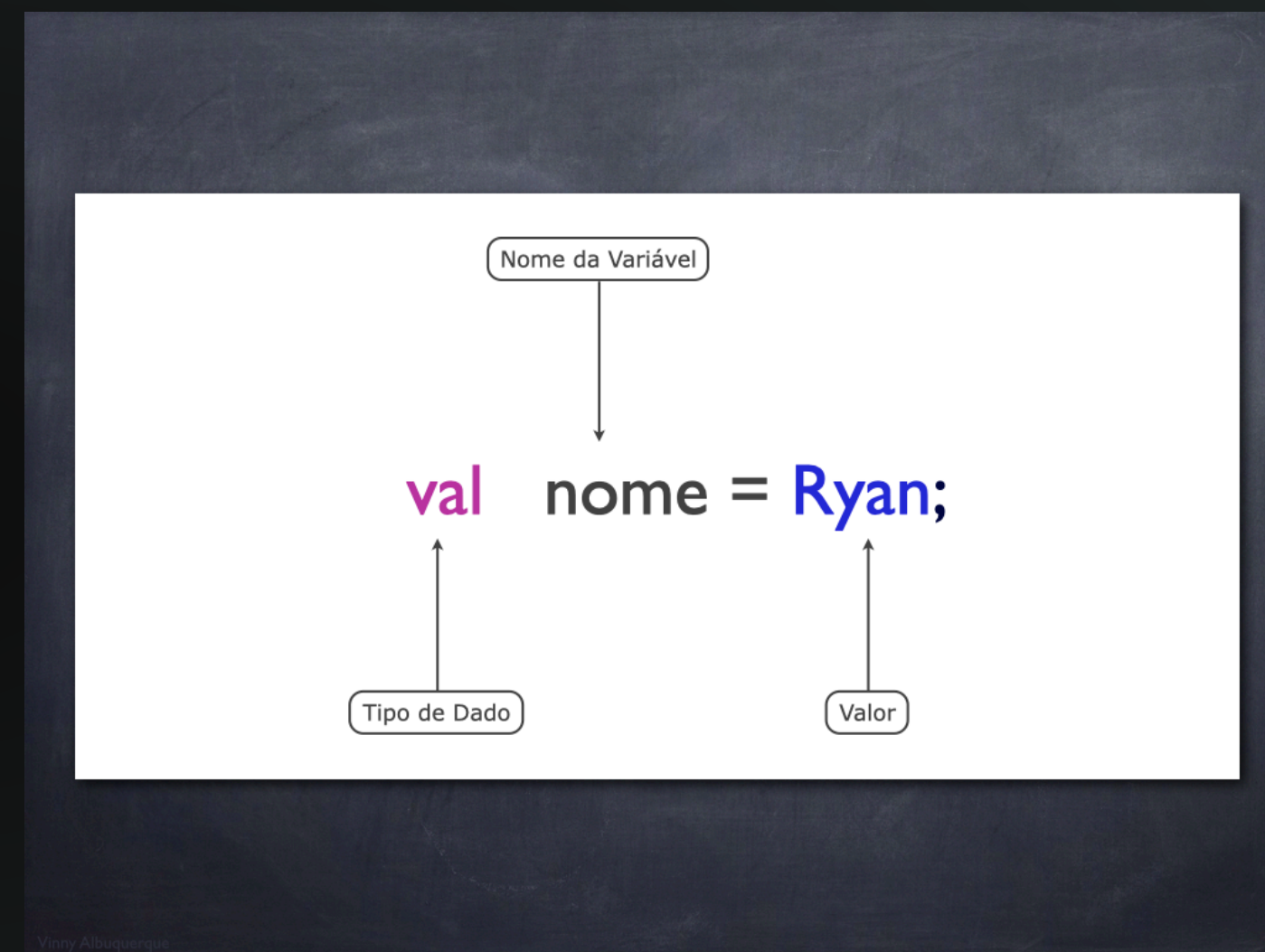
Tipo	Tamanho
Char	2 bytes
Int	4 bytes
Long	8 bytes
Float	4 bytes
Double	8 bytes



# Variáveis

|  
+

Na imagem ao lado podemos conferir um exemplo mais detalhando do que significa, **tipo de dado**, **nome da variável** e **valor**.





# Variáveis

## Escpecificadores de Formato

Especificador	Descrição
%d	Para inteiros decimais
%i	Para inteiros decimais
%f	Para números de ponto flutuante
%.2f	Para números de ponto flutuante com 2 casas decimais
%s	Para strings
%c	Para caracteres
%x	Para inteiros em formato hexadecimal

# Variáveis

## Sintaxe Kotlin

```
// Exemplo de variável em Kotlin
fun main() {
    val nome = "Maria" //Variavel tipo string
    var idade = 25 //Variavel tipo inteira

    println("Nome: $nome, Idade: $idade") // imprimindo variáveis
}
```



# Variáveis

## Declaração e Inicialização de Variáveis

- Sintaxe para declarar variáveis em Kotlin
- Na prática diferentes tipos de dados (`Int`, `Double`, `String`, `Boolean`)

```
// Exemplos de declaração e inicialização de variáveis
fun main() {
    val numeroInteiro: Int = 10
    var numeroDecimal: Double = 3.14
    val texto: String = "Olá, mundo!"
    var verdadeiro: Boolean = true

    println("Número Inteiro: $numeroInteiro")
    println("Número Decimal: $numeroDecimal")
    println("Texto: $texto")
    println("Valor Booleano: $verdadeiro")
}
```

# Variáveis

## Tipos de Dados

- Numéricos ( `Int` , `Long` , `Double` , `Float` )
- Caractere ( `Char` ) e cadeia de caracteres ( `String` )
- Tipo lógico ( `Boolean` )

```
// Exemplos de diferentes tipos de dados
fun main() {
    val numeroInteiro: Int = 10
    val numeroLong: Long = 100000000000L
    val numeroDecimal: Double = 3.14
    val caractere: Char = 'A'
    val texto: String = "Kotlin é incrível!"
    val verdadeiro: Boolean = true

    println("Número Inteiro: $numeroInteiro")
    println("Número Long: $numeroLong")
    println("Número Decimal: $numeroDecimal")
    println("Caractere: $caractere")
    println("Texto: $texto")
    println("Valor Booleano: $verdadeiro")
}
```

# Variáveis

## Mutabilidade e Imutabilidade

Variáveis servem como recipientes de dados, podendo armazenar valores que podem ser alterados ou mantidos constantes ao longo do tempo.

### 1. **var** mutável:

- Declara uma variável que pode ser alterada posteriormente.

```
var nome: String = "João"  
nome = "Maria" // Atribuindo novo valor
```

### 2. **val** imutável:

- Declara uma variável que não pode ser alterada após ser atribuída.

```
val idade: Int = 30  
// idade = 31 // Isso resultaria em um erro de compilação
```

Nota: **var** para dados que podem mudar e **val** para dados que permanecerão constantes após a atribuição inicial.

# Variáveis

## Mutabilidade e Imutabilidade

- Diferença entre variáveis `mutáveis` e `imutáveis`
- Vantagens da imutabilidade
- Exemplos de uso de `val` (`imutável`) e `var` (`mutável`)

```
// Exemplos de mutabilidade e imutabilidade
fun main() {
    val imutavel: Int = 5 // imutavel
    var mutavel: Int = 10 // mutável

    mutavel = 15 // OK
    imutavel = 8 // Erro! Não é permitido reatribuir valor a uma variável imutável

    println("Valor da variável mutavel: $mutavel")
    println("Valor da variável imutavel: $imutavel")
}
```

# Variáveis

## Convenções de Nomenclatura

- Convenções para nomear variáveis em Kotlin
- Padrões de nomenclatura ( `camelCase` , `snake_case` )

```
// Exemplos de nomenclatura de variáveis
fun main() {
    val nomeCompleto: String = "João Silva"
    val idadeDoUsuario: Int = 30
    val valor_Total: Double = 50.0 // Evite usar underscores em nomes de variáveis

    println("Nome Completo: $nomeCompleto")
    println("Idade do Usuário: $idadeDoUsuario")
    println("Valor Total: $valor_Total")
}
```

# Variáveis

## Escopo de Variáveis

- Explicação sobre o escopo de variáveis
- Variáveis locais vs. variáveis de escopo global

```
// Exemplo de escopo de variáveis
fun main() {
    val nome = "Alice"

    if (true) {
        val sobrenome = "Silva" // Variável sobrenome só é visível dentro deste bloco
        println("$nome $sobrenome") // Alice Silva
    }

    // println("$nome $sobrenome") // Erro! Variável sobrenome não é visível aqui
}
```

# Variáveis

## Operadores

Elementos para manipular dados em Kotlin, categorizados por tipo de operação.

Tipo	Descrição	Exemplo
Atribuição	Atribuir ou atualizar valores em variáveis.	= , += , -= , *= , /=
Matemáticos	Realizar operações aritméticas.	+ , - , * , / , %
Comparação	Comparar valores e determinar relações.	== , != , < , > , <= , >=



# Variáveis

## **Desafio** - Média Escolar

Crie um programa para Calculadora de Média Escolar considere 3 notas.

Considere: Nota1, Nota2, Nota3 e média.

# Variáveis

## Desafio - Temperatura

- Crie um programa que faça a Conversão de Temperatura.
- Considere que: `Celsius` = 25.0 °C
- Fahrenheit = `Celsius` \* 9/5 + 32
- `Kelvin` é `Celsius` + 273.15

# Variáveis

## Desafio - Perfil do usuário

Crie um Perfil de Usuário em Kotlin, vamos aplicar os conceitos aprendidos sobre `variáveis` em `Kotlin` para criar um programa simples que simule o armazenamento e exibição de informações de um perfil de usuário.

- Crie um arquivo Kotlin (com extensão `.kt`)
- Declare variáveis para armazenar informações de um perfil de usuário, como `nome`, `idade`, `email` e se o usuário está `ativo` ou `inativo`.
- Atribua valores às variáveis de acordo com os dados que você escolher.
- Utilize `println()` para exibir as informações do perfil do usuário no console.

# Variáveis

## Importante! Publique no GitHub

Após criar seu programa conforme desafios, publique em sua conta do github, abra uma **ISSUE**.  
No repositório. <https://github.com/mvalbuquerque/mad-android>, coloque na **ISSUE** o link do seu repositório do **git** com seu desafio concluído.

# Dúvidas

 Divirta-se!

 Prof<sup>º</sup> Vinny Albuquerque

 [profvinny.albuquerque@fiap.com.br](mailto:profvinny.albuquerque@fiap.com.br)

 [@mvalbuquerque](#)