

Kotlin Básico: A Evolução da JVM

**Dos fundamentos da
sintaxe à segurança
do Null Safety.**

Uma introdução técnica para
desenvolvedores que buscam código
conciso, seguro e interoperável.



O Berço: A Busca pela Eficiência (2010)

A história começa na JetBrains (criadores do IntelliJ e Android Studio). Em 2010, seus engenheiros enfrentavam um dilema: amavam a robustez da JVM (Java Virtual Machine), mas o Java da época era verboso e lento.

Os Três Pilares

- ✓ 1. **Concisa:** Menos linhas de código para fazer a mesma tarefa.
- ✓ 2. **Segura:** Eliminar o erro 'NullPointerException' (o erro de um milhão de dólares).
- ✓ 3. **Interoperável:** Funcionar 100% com o código Java existente.



A Ascensão ao Trono do Mobile

2011-2012

O nome 'Kotlin' é escolhido (uma ilha russa). O projeto torna-se Open Source.



2017 (O Divisor de Águas)

Google I/O: Kotlin é anunciado como linguagem oficial para Android. A plateia aplaude de pé.

2019 (Kotlin First)

Google declara que todo novo desenvolvimento Android será pensado primeiro para Kotlin.

Futuro: Com o **Kotlin Multiplatform** (KMP), a linguagem vai além do Android, rodando em iOS, Web e Backend.



Por que o Kotlin Venceu?

Antes (Java Legacy)

- Código longo e repetitivo (Verboso).
- Erros de “valor nulo” são comuns e perigosos.
- Evolução lenta da sintaxe.

Depois (Kotlin)

- ✓ **Código curto** e expressivo.
- ✓ **Segurança** contra nulos (Null Safety) embutida.
- ✓ **Moderno** e com evolução rápida.

Variáveis: A Arte de Guardar Dados

Em Kotlin, criar uma variável é como preparar uma etiqueta para uma caixa na memória.



A Decisão de Mutabilidade.

Antes de decidir
O QUE vai na caixa,
você deve decidir se
ela será lacrada ou
deixada aberta.



As Duas Palavras Mágicas

val

(Imutável / Value)



- Apenas leitura. Uma vez atribuído, nunca mais muda. Comparável ao 'final' do Java.
- **Uso Ideal:** Constantes, IDs de banco, Nomes de usuário.



var

(Mutável / Variable)

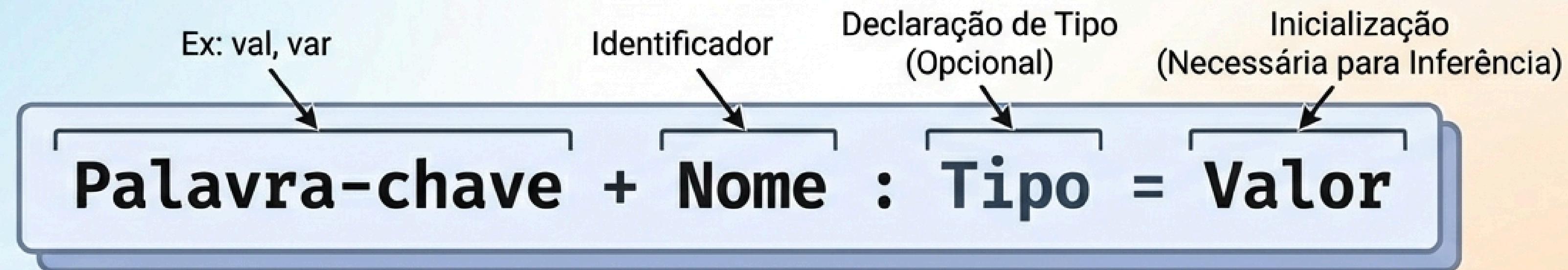


- Leitura e escrita. O valor pode ser alterado quantas vezes for necessário.
- **Uso Ideal:** Contadores, preços, status de jogo.



Anatomia e a Magia da Inferência

A Fórmula



Exemplo Explícito

Palavra-chave Nome Tipo Valor

val nome: String = "Anderson"

A Magia (Inferência)

O Kotlin é inteligente. Se você atribui o valor na criação, ele 'adivinha' o tipo.

val cidade = "Barbacena"

Compilador sabe que é **String**

Palavra-chave Nome Tipo Valor

var idade: Int = 25

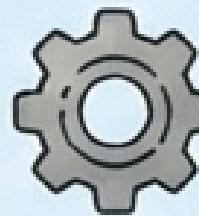
var temperatura = 22.5

Compilador sabe que é **Double**



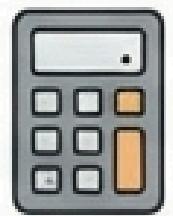
O Ecossistema de Tipos

Tipos otimizados da JVM.



Inteiros

Int (10), **Long** (1500L)



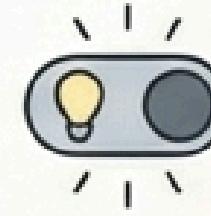
Decimais

Double (99.90), **Float** (5.5f)



Textos

String ("Kotlin"), **Char** ('K')



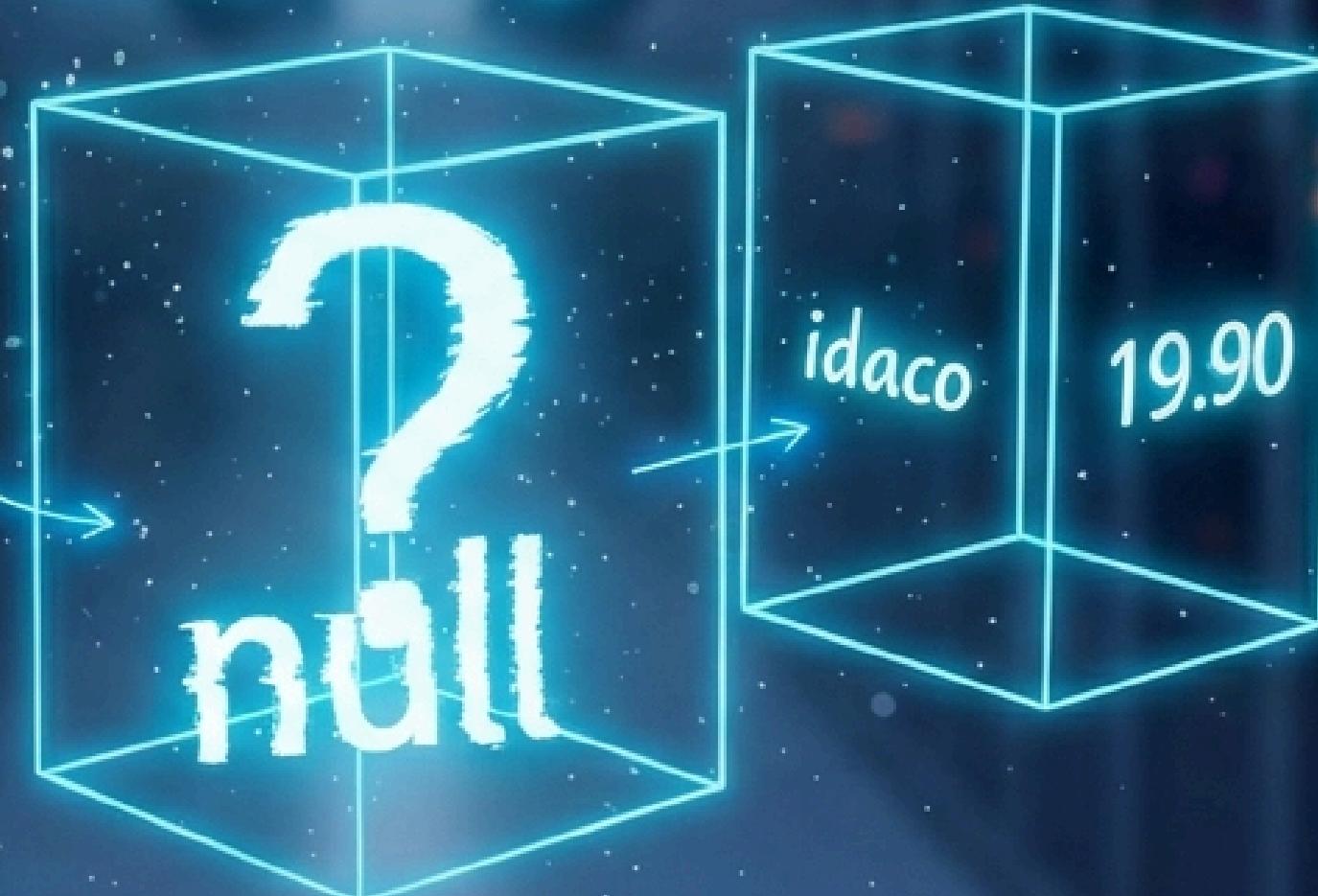
Lógicos

Boolean (true, false)

O Perigo do Null (O Erro de 1 Milhão de Dólares)

Ponteiro para
o nada.

nome: String?



0 (Zero) = Uma caixa com o número zero.

"" (Vazio) = Uma caixa com um papel em branco.

Null = Não existe caixa.
Ausência total de valor.



Null Safety: O Escudo do Kotlin

Por padrão, o Kotlin NÃO aceita nulo. O erro é pego antes de rodar.

Padrão (Non-nullable)

```
var nome: String = "Anderson"  
  
nome = null
```



COMPILER ERROR

Nullable (Aceita Nulo)

```
var nome: String? = "Anderson"  
  
nome = null
```



Valid Code



Falando com o Mundo (Output)

Funções Básicas

- **print()**: Mantém na mesma linha.
- **println()**: Pula para a próxima linha.

String Templates (\$)

A forma moderna de concatenar texto.

```
val idade = 25
println("Sua idade é $idade anos")
```



Substituição
automática do valor



Ouvindo o Usuário: readln()

Lê uma linha inteira de texto digitada no console.



Simplicidade (Sem imports)

Nenhum nuto de são necessitanou importos adicionais.



Pausa (Espera o Enter)

O programa pausa para uma intrada no usuário, seguido a lxa do Enter.



Retorno (Sempre String)

A salda é sempre uma uma string.

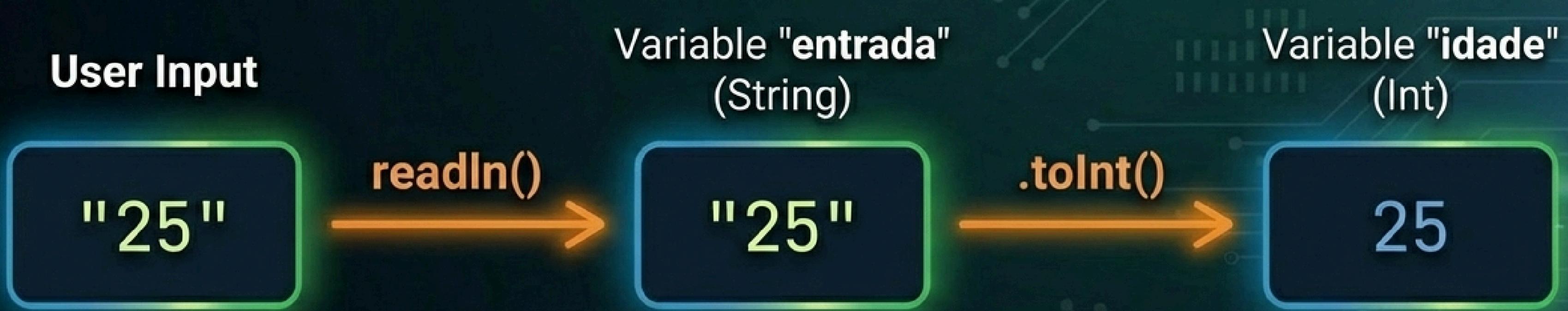
Code Walkthrough

```
print("Digite seu nome: ")
val nome = readln()
println("Olá, $nome!")
```



Convertendo Entradas

O `readln()` sempre entrega texto. Para matemática, converta.



```
val idade = readln().toInt()
```



A Abordagem Clássica: Scanner

Ferramenta do Java disponível no Kotlin. Útil para interoperabilidade.

```
import java.util.Scanner  
  
val leitor = Scanner(System.`in`)
```

Atenção: 'in' é palavra reservada no Kotlin. Use crases para escapar.



O Grande Resumo: Exemplo Prático

Cadastro de Aluno

```
CadastroAluno.kt
```

```
1 fun main() {
2     // Entrada
3     print("Digite seu nome: ")
4     val nome = readln() // Inferência
5
6     print("Digite sua nota: ")
7     val nota = readln().toDouble() // Conversão
8
9     // Processamento
10    val passou = nota >= 6.0
11
12    // Saída com Template
13    println("---- Resultado ----")
14    println("Aluno: $nome | Nota: $nota")
15    println("Aprovado: $passou")
16 }
```

