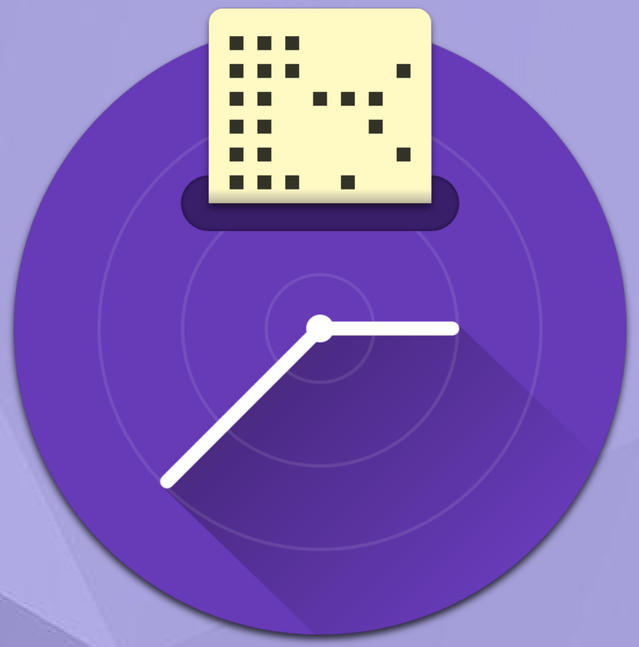
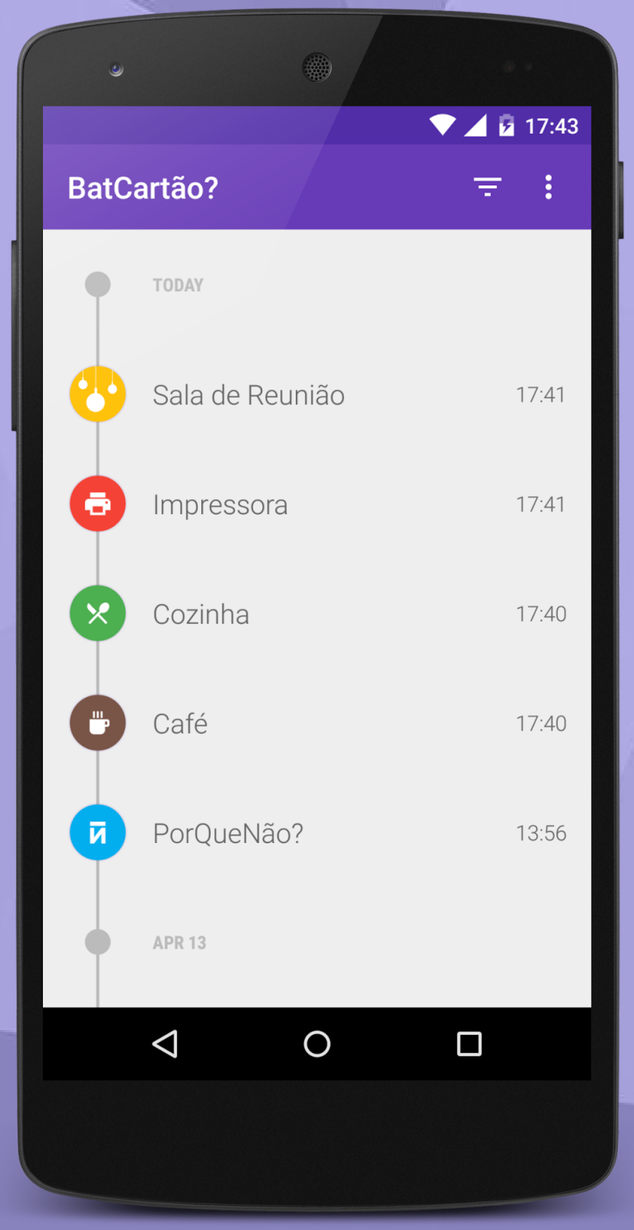
Kotlin

O DESPERTAR DA FORÇA

Ӣ




17:43
BatCartão?
TODAY
Sala de Reunião
17:41
Impressora
17:41
Cozinha
17:40
Café
17:40
PorQueNão?
13:56
APR 13


Kotlin

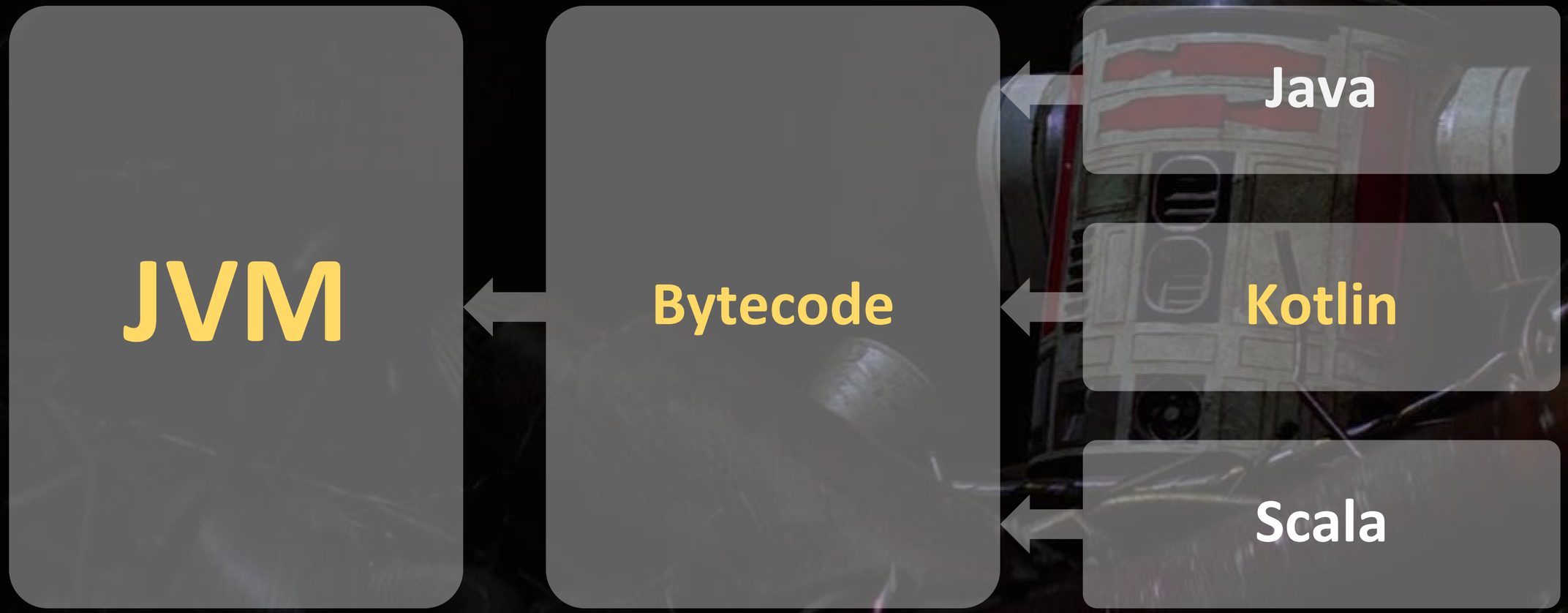
JetBRAINS

Open Source

JVM

100% Interoperável com Java

Arquitetura



Motivação

Uso Industrial
Design "Efetivo"
Moderna

Segura | Simples | Concisa

PorQueKotlin?

Desenvolvimento Ativo
Próximo da versão final
Funciona AGORA

Sintaxe

Sintaxe docs

```kotlin
package mobi.porquenao.poc.kotlin

import java.util.*

fun main() {
  println("Hello World!")
}
```

Sintaxe docs

```
package mobi.porquenao.poc.kotlin

import java.util.*

fun sum(n1: Int, n2: Int): Int {
  return n1 + n2
}
```

Sintaxe docs

```
package mobi.porquenao.poc.kotlin

import java.util.*

fun sum(n1: Int, n2: Int) = n1 + n2
```

Variáveis docs

```
val i: Int = 0
```

Variáveis docs

```
val i: Int = 0

val i = 0  // Tipo Int é inferido
```

Variáveis mutabilidade docs

```
val i = 0

i = 1 // Erro, i é imutável
```

Variáveis mutabilidade docs

```
val i = 0

i = 1 // Erro, i é imutável

var i = 0

i = 1 // Ok!
```

Null Safety docs

```
var text: String = "Hi"
```

Null Safety docs

```
var text: String = "Hi"

var text: String = null // Erro
```

Null Safety docs

```
var text: String = "Hi"

var text: String = null // Erro

var text: String? = null // Nullable!
```

Null Safety docs

```
var text: String? = "Hi"
```

Null Safety docs

```
var text: String? = "Hi"

println(text.length) // Não compila, text pode ser null
```

Null Safety docs

```
var text: String? = "Hi"

if (text != null) { // Smart-cast

  println(text.length) // "2"

}
```

Null Safety docs

```
var text: String? = "Hi"

println(text?.length) // "2"
```

Null Safety docs

```
var text: String? = "Hi"

println(text?.length) // "2"

var text: String? = null

println(text?.length) // "null"
```

Null Safety docs

```
var text: String? = "Hi"

println(text?.length) // "2"

var text: String? = null

println(text?.length) // "null"

var text: String? = null

println(text?.length ?: 0) // "0"
```

Null Safety docs

```
var text: String? = "Hi"

text!!.length
```

Class docs

```
class User {

  val name: String = "Luke"
  var surname: String = "Skywalker"

  fun fullName(): String {
    return "$name $surname"
  }

}
```

Class docs

```
class User(val name: String, var surname: String = "Skywalker") {

  fun fullName() = "$name $surname"

}

val user = User("Luke")
println(user.fullName()) // "Luke Skywalker"
```

Getters and Setters docs

```
public class MainActivity : BaseActivity() {

  var somePreference: Boolean
    get() {
      return sharedPreferences.getBoolean("SomePreference", true)
    }
    set(value) {
      sharedPreferences.edit()
        .putBoolean("SomePreference", value)
        .apply()
    }

}
```

Delegates.lazy docs

```
val sharedPreferences by Delegates.lazy {
  getSharedPreferences("SharedPreference", Context.MODE_PRIVATE)
}
```

Delegates.lazy docs

```
val sharedPreferences by Delegates.lazy {
  getSharedPreferences("SharedPreference", Context.MODE_PRIVATE)
}

val textView: TextView by Delegates.lazy {
  findViewById(R.id.text) as TextView
}
```

When docs

```
var text: CharSequence? = null

when (text) {
  null          -> println("Null")
  "Strong"      -> println("is the Force")
  is String     -> println(text.length)
  else          -> println(text)
}
```

Lambdas java docs

```
void calculate(Runnable callback) {
  // Calcula algo
  callback.run();
}
```

Lambdas java docs

```
void calculate(Runnable callback) {
  // Calcula algo
  callback.run();
}

calculate(new Runnable() {
  @Override
  public void run() {
    println("Fim!");
  }
});
```

Lambdas docs

```
fun calculate(callback: () -> Unit) {
  // Calcula algo
  callback()
}
```

Lambdas docs

```
fun calculate(callback: () -> Unit) {
  // Calcula algo
  callback()
}

calculate({
  println("Fim!")
})
```

Lambdas docs

```
fun calculate(callback: () -> Unit) {
  // Calcula algo
  callback()
}

calculate({
  println("Fim!")
})

calculate {
  println("Fim!")
}
```

Lambdas docs

```
fun sum(n1: Int, n2: Int, callback: (Int) -> Unit) {
  callback(n1 + n2)
}
```

Lambdas docs

```
fun sum(n1: Int, n2: Int, callback: (Int) -> Unit) {
  callback(n1 + n2)
}

sum(1, 2, { result ->
  println(result)
})
```

Lambdas docs

```
fun sum(n1: Int, n2: Int, callback: (Int) -> Unit) {
  callback(n1 + n2)
}

sum(1, 2) { result ->
  println(result)
}
```

Lambdas docs

```
fun sum(n1: Int, n2: Int, callback: (Int) -> Unit) {
  callback(n1 + n2)
}

sum(1, 2) {
  println(it)
}
```

Lambdas android docs

```
button.setOnClickListener {
  println("Hi!")
}
```

Lambdas android docs

```
button.setOnClickListener {
  println("Hi!")
}

textView.addTextChangedListener(object : TextWatcher {

  override fun onTextChanged(s: CharSequence, start: Int, before: Int, count: Int) {}
  override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {}
  override fun afterTextChanged(s: Editable?) {}

})
```

Interfaces docs

```
interface Clickable {

    fun onClick()

}
```

Interfaces docs

```
interface Clickable {

    val enabled: Boolean

    fun onClick()

}
```

Interfaces docs

```
interface Clickable {

    val enabled: Boolean

    fun onClick()

    fun click() {
        if (enabled) onClick()
    }

}
```

Interfaces docs

```
class Button : Clickable {

    override val enabled: Boolean = true

    override fun onClick() {
        println("Click!")
    }

}
```

Interfaces docs

```
class Button : Clickable {

   override val enabled: Boolean = true

   override fun onClick() {
       println("Click!")
   }

}

Button().click() // "Click!"
```

Extensions docs

```
fun String.first(): Char {
  return this[0]
}
```

Extensions docs

```
fun String.first(): Char {
  return this[0]
}

println("Hey!".first()) // "H"
```

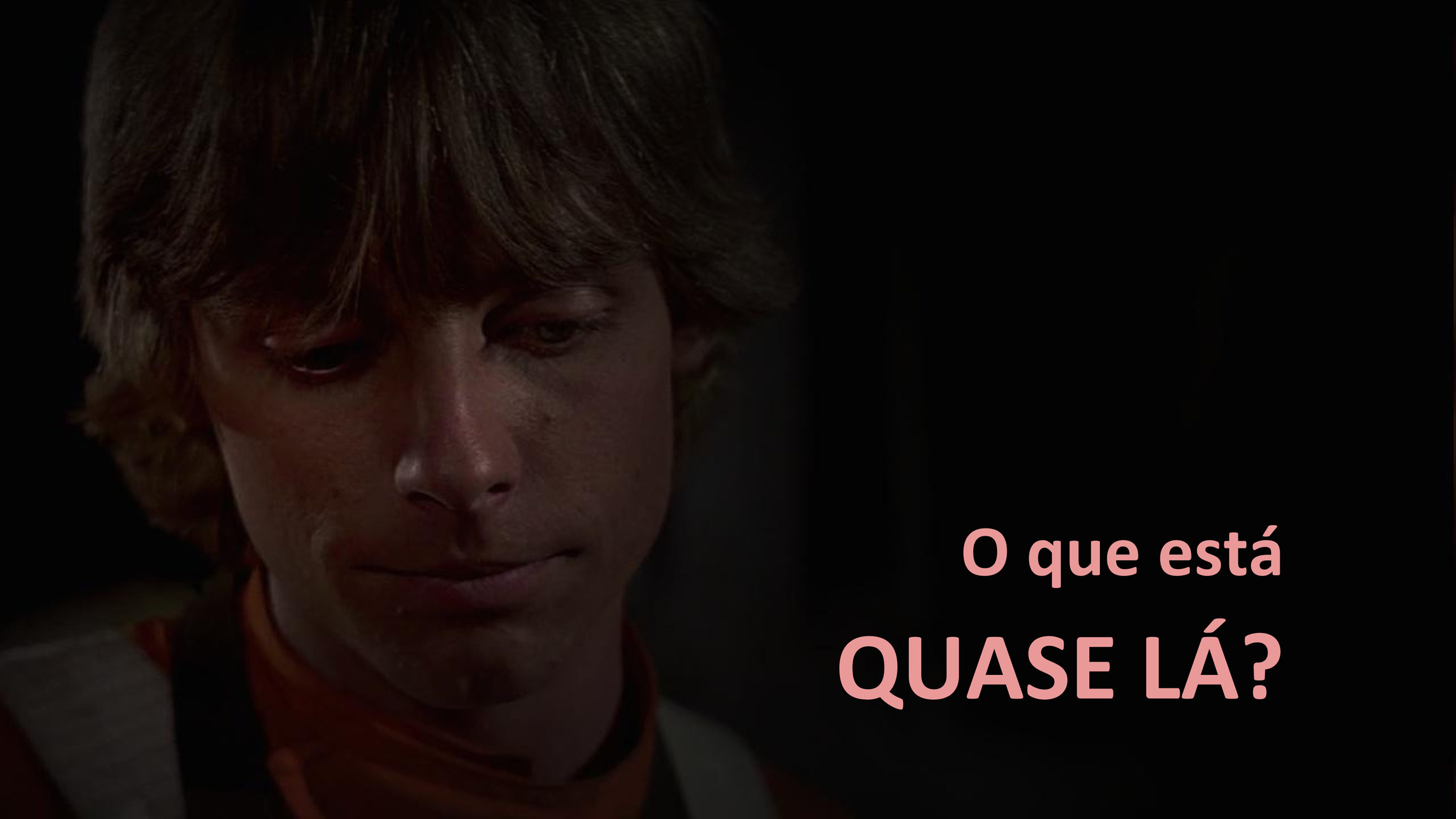
Extensions java docs

```
package ext

fun String.first(): Char {
  return this[0]
}

System.out.println(ExtPackage.first("Hey!")) // "H"
```


Performance?



O que está
QUASE LÁ?

Android-APT

- Introduzido recentemente via kapt
- Não está 100%
- Aumenta o tempo de build

Constantes

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
  // Do something
}
```

Constantes

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
  // Do something
}

if (Build.VERSION.SDK_INT >= 21) {
  // Do something
}
```

Métodos Estáticos

```
getResources().getColor(R.color.color_primary)
```

Métodos Estáticos

```
getResources().getColor(R.color.color_primary)

ActivityCompat.getColor(this, R.color.color_primary)
```

Métodos Estáticos

```
getResources().getColor(R.color.color_primary)

ActivityCompat.getColor(this, R.color.color_primary)

ContextCompat.getColor(this, R.color.color_primary)
```

Tooling

- Auto-complete um pouco mais lento
- Tempo de build maior

Configurando Android Studio

http://kotlinlang.org/docs/tutorials/kotlin-android.html

1. Instalar Plugin "Kotlin"

Android Studio

- Preferences > Plugins
- Browse repositories…
- Procurar: "Kotlin"
- Instalar
- Reiniciar Android Studio (2015?)

2. Configurar Gradle Plugin

Editar: build.gradle

```
buildscript {
  dependencies {
    classpath 'com.android.tools.build:gradle:1.3.0'
    classpath 'org.jetbrains.kotlin:kotlin-gradle-plugin:+'
  }
}
```

3. Configurar Dependência

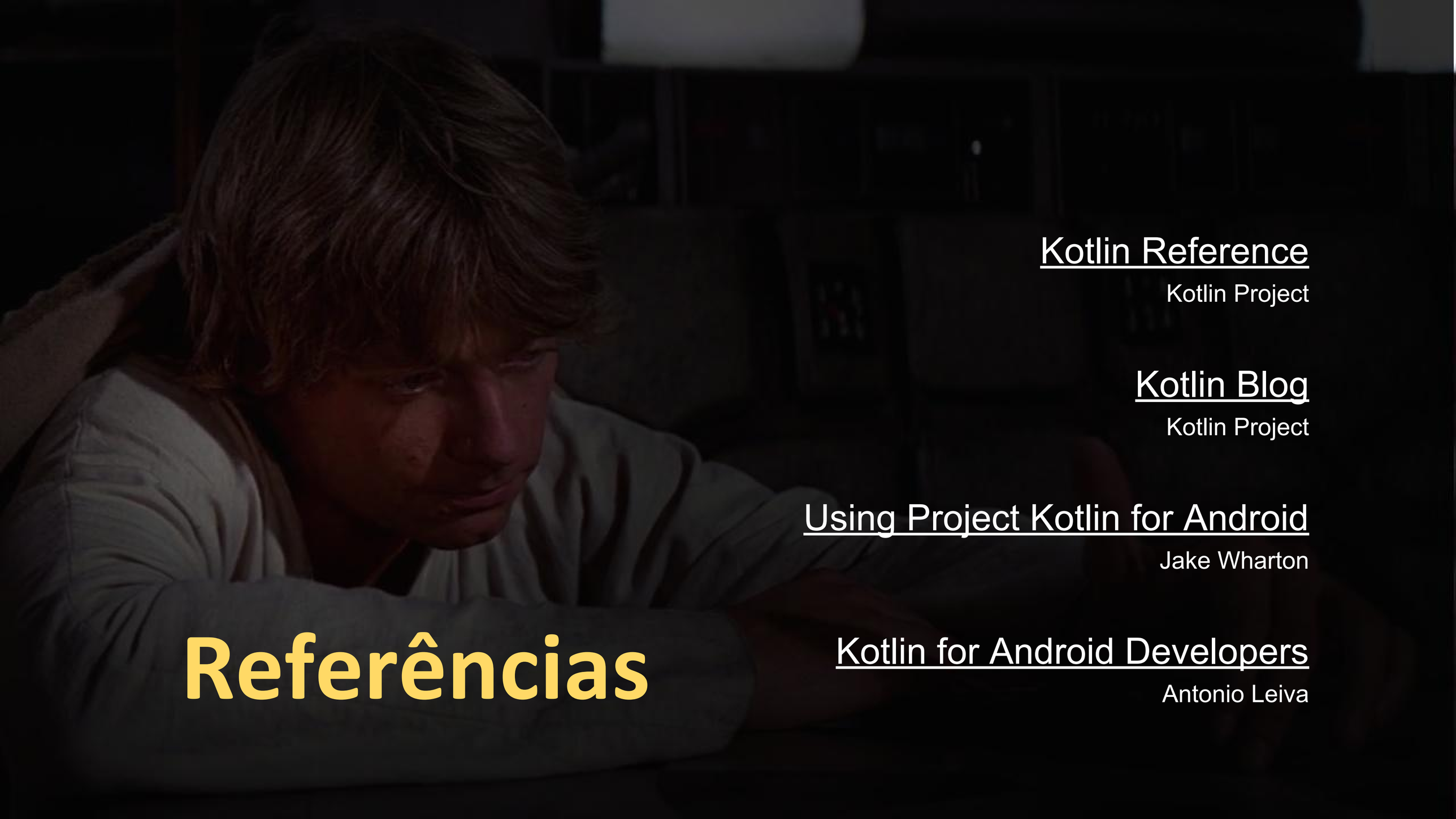
Editar: app/build.gradle

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
[...]
dependencies {
  [...]
  compile 'org.jetbrains.kotlin:kotlin-stdlib:+'
}
```

4. Profit

Acessar qualquer *.java

- Code > Convert Java File to Kotlin File



Referências

Kotlin Reference
Kotlin Project

Kotlin Blog
Kotlin Project

Using Project Kotlin for Android
Jake Wharton

Kotlin for Android Developers
Antonio Leiva

Conhecimento

NÃO MORRE

Obrigado

https://bit.ly/kotlin-despertar

Mickele Moriconi
@mickele

Ademar Oliveira
@ademar111190