



# **INTRODUÇÃO AO ANDROID STUDIO**

**PROF. ME. HÉLIO ESPERDIÃO**

# ANDROID STUDIO

- É um ambiente de desenvolvimento integrado para desenvolver para a plataforma Android. Foi anunciado em 16 de Maio de 2013 na conferência Google I/O. Android Studio é disponibilizado gratuitamente

# LINGUAGENS DE PROGRAMAÇÃO

- Java
- Kotlin
- C++

# KOTLIN

- Kotlin é uma Linguagem de programação multiplataforma, orientada a objetos e funcional, concisa e estaticamente tipada, desenvolvida pela JetBrains em 2011, que compila para a Máquina virtual Java e que também pode ser traduzida para a linguagem JavaScript e compilada para código nativo



# DECLARAÇÃO DE VARIÁVEL

- O Kotlin usa duas palavras-chave diferentes para declarar variáveis: val e var.
  - Use val para uma variável cujo valor nunca muda. Não é possível reatribuir um valor a uma variável que tenha sido declarada usando val.
  - Use var para uma variável cujo valor possa ser mudado

# EXEMPLO



No exemplo abaixo, count é uma variável do tipo Int que recebe um valor inicial de 10:



```
var count: Int = 10
```

# CONSTANTES



Você quiser garantir que `languageName` sempre tenha o valor "Kotlin", poderá declarar `languageName` usando a palavra-chave `val`:



```
val languageName: String = "Kotlin"
```

# INT



Int é um tipo que representa um número inteiro, um dos muitos tipos numéricos que podem ser representados em Kotlin.



Assim como acontece com outras linguagens, você também pode usar Byte, Short, Long, Float e Double, dependendo dos seus dados numéricos.



# STRINGS

- Strings podem ser criadas com aspas duplas ou aspas triplas. Além disso, os caracteres de escape podem ser usados com aspas duplas;

```
val myString = "Meu Texto"  
val escapeString = "Hélio Esperidião \n"
```

# MANIPULAÇÃO DE STRINGS SIMPLES

```
val name = "Hélio"
```

```
val message = "A primeira letra do meu nome é: ${name.first()}"
```

# TIPOS NÃO COMPATÍVEIS

```
var age = 12
```

```
age = "12 years old" // Erro
```

# TIPOS BÁSICOS

Double - 64 bits

Float - 32 bits

Long - 64 bits

Int - 32 bits

Short - 16 bits

Byte - 8 bits

# UTILIZANDO MÚLTIPLOS TIPOS

```
val myInt = 35  
val myLong = 5000000000L  
val myFloat = 35.05F  
val myDouble = 45.7845645646  
val myHexadecimal = 0x10  
val myBinary = 0b1010
```

# CONVERSÃO DE TIPOS

```
myInt = 500
```

```
myLong = myInt.to
```

```
teIdade(idad
```

```
resposta =
```

```
"Maior de i
```

```
lse {
```

```
"Menor de i
```

- f to(that: B) for A in kotlin Pair<
- m toLong()
- f toULong() for Int in kotlin
- m toByte()
- m toChar()
- m toDouble()
- m toFloat()
- m toInt()
- m toShort()
- m toString()
- f toBigDecimal() for Int in kotlin Big
- f toBigDecimal(mathContext: MathContext) for Int in ko... Big
- f toBigInteger() for Int in kotlin Big
- f toString(radix: Int) for Int in kotlin.text
- f toUByte() for Int in kotlin
- f toUInt() for Int in kotlin
- f toUShort() for Int in kotlin
- f toColorLong() for Int in androidx.core.graphics
- f toColor() for Int in androidx.core.graphics
- f toDrawable() for Int in androidx.core.graphics.dr... ColorD
- f toDuration(units: DurationUnit /\* = TimeUnit \*/) for Int in D

Press Enter to insert, Guia to replace

# EXEMPLOS DE CONVERSÃO

- `val stringNumber = "101"`
- `val intValue = stringNumber.toInt()`
- `val intValue = "101".toInt()`

# O TIPO BOOLEANO

```
val meuBooleano = false
val x = 1
val y = 3
val w = 4
val z = 6
val n = x < z && z > w
```



# INFERÊNCIA DE TIPO

- No exemplo a seguir, `languageName` é inferido como `String`.
- Portanto, não é possível chamar nenhuma função que não faça parte da classe `String`:
- `val languageName = "Kotlin"`
- `val upperCaseName = languageName.toUpperCase()`

# OPERADOR ARITMÉTICO: +

```
var x: Int = 4
```

```
var y: Int = 6
```

```
var soma: Int
```

```
soma = x + y // Resultado: 10.
```

```
// Operador + em strings
```

```
var texto1: String = "Ola"
```

```
var texto2: String = "Mundo"
```

```
var concat: String
```

```
concat = texto1 + " " + texto2 // Resultado: "ola Mundo".
```

# OPERADOR ARITMÉTICO: %

```
var x: Int = 3  
var y: Int = 2  
var resto: Int  
resto = x % y
```

```
// Resultado: 1.
```

# OPERADORES LÓGICOS

Operação	Operador	Operador
e	&&	and
Ou		or
Não	!	

# EXPRESSÕES ARITMÉTICAS

Operador	Descrição	Hierarquia da Operação
+	Soma	3°
-	Subtração	3°
/	Multiplicação	2°
*	Divisão	2°
(%)	Resto da divisão	2°
( )	Parênteses	1° 1°

$$7+2/2=?$$
$$(7+2)/2=?$$

# OPERADORES RELACIONAIS

Operador	Função	Exemplo	Resultado
>	Maior	5>3	?
<	Menor	5<3	?
>=	Maior ou igual	5>=6	?
<=	Menor ou igual	7<=3	?
!=	Diferente de	5!=5	?
==	Igual a	5==5	?
!	Inversor	!5==5	?

# EXEMPLOS

```
var x: Boolean = false
var y: Boolean = false
var conjuncion: Boolean
conjuncion = x && y // Resultado: false
```

```
var x1: Boolean = false
var y1: Boolean = true
var conjuncion1: Boolean
conjuncion1 = x and y // Resultado: false
```

```
var x2: Boolean = true
var y2: Boolean = false
var conjuncion2: Boolean
conjuncion2 = x2 && y2 // Resultado: false
```

```
var x3: Boolean = true
var y3: Boolean = true
var conjuncion3: Boolean
conjuncion3 = x && y // Resultado: true
```

# COMENTÁRIOS

```
/*  
  Bloco de comentário  
*/
```

```
// Linha de comentário
```



# CONDICIONAIS

```
if (i == 42) {  
    println("oi")  
} else {  
    println("oi 2")  
}
```

# ELSE IF

```
if (idade > 17) {  
    println("maior de idade ")  
} else if (idade > 65) {  
    println("e idoso")  
} else {  
    println("menor de idade")  
}
```

# EVITAR REPETIÇÃO



As instruções condicionais são úteis para representar a lógica com estado, mas você pode se repetir ao gravá-las.



No exemplo anterior, você simplesmente imprime um String em cada ramificação.



Para evitar essa repetição, o Kotlin oferece expressões condicionais.

# EVITE REPETIÇÕES

```
val answerString: String = if (idade > 17) {  
    "maior de idade "  
} else if (idade > 65) {  
    " e idoso "  
} else {  
    "menor de idade"  
}  
  
println(answerString)
```

# WHEN

- Conforme a complexidade da instrução condicional aumenta, é recomendável substituir a expressão if-else por uma expressão when, conforme mostrado neste exemplo:

```
val answerString = when {  
    count == 42 -> "I have the answer."  
    count > 35 -> "The answer is close."  
    else -> "The answer eludes me."  
}  
  
println(answerString)
```

# EXEMPLOS DA ESTRUTURA FOR

```
for (i in 6 downTo 0 step 2) {  
    |   println(i)  
}
```

```
for (i in 1..3) {  
    |   println(i)  
}
```

```
var t:Int=50  
for (i in 1..t) {  
    |   println(i)  
}
```

# EXEMPLOS DA ESTRUTURA WHILE

```
var x: Int=0
while (x > 0) {
|   x--
}
}
```

```
var y:Int =0
do {
|   y--
} while (y >5)
```

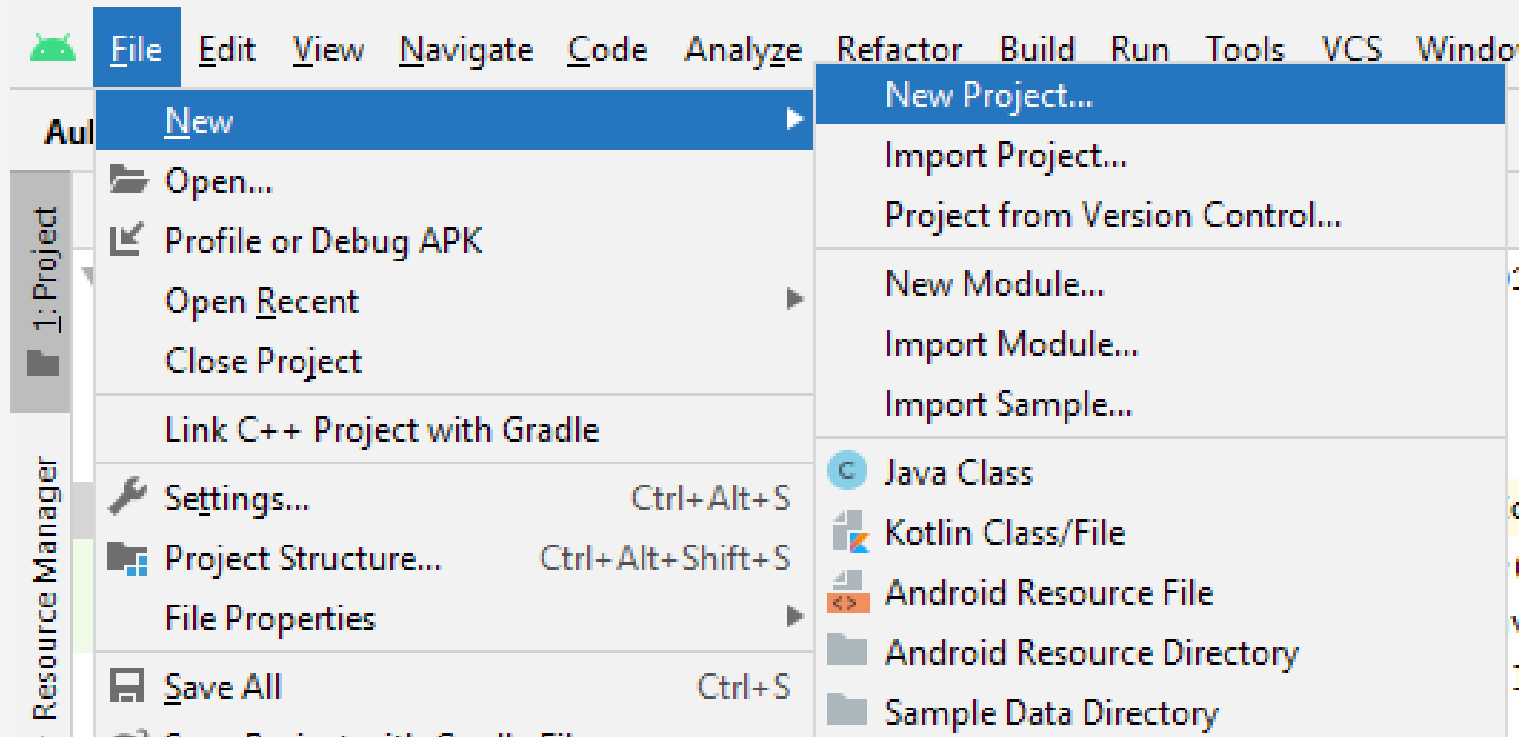
# FUNÇÕES

- Você pode agrupar uma ou mais expressões em uma *função*. Em vez de repetir a mesma série de expressões sempre que precisar de um resultado, você pode unir as expressões em uma função e chamar essa função.



```
fun testeIdade(idade:Int): String {  
    val resposta = if (idade >=18) {  
        "Maior de idade."  
    } else {  
        "Menor de idade"  
    }  
  
    return resposta  
}
```

# CRIAR UM PROJETO NO ANDROID STUDIO





## Select a Project Template

Phone and Tablet

Wear OS

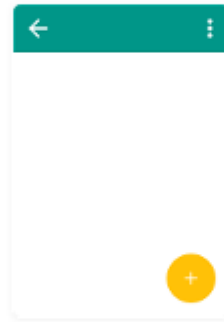
Android TV

Automotive

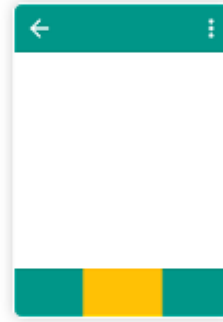
Android Things



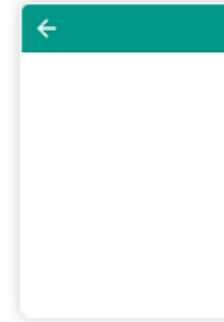
No Activity



Basic Activity



Bottom Navigation Activity



Empty Activity



### Empty Activity

Creates a new empty activity

Previous

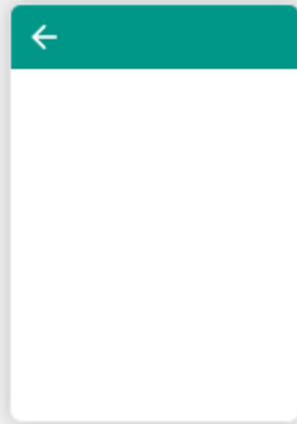
Next

Cancel

Finish



## Configure Your Project



Empty Activity

Name

My Application

Package name

com.example.myapplication

Save location

C:\Users\aluno\AndroidStudioProjects\MyApplication

Language

Kotlin

Minimum SDK

API 16: Android 4.1 (Jelly Bean)

**i** Your app will run on approximately **99,8%** of devices.

[Help me choose](#)

Previous

Next

Cancel

Finish

Project

- Android
  - app
    - manifests
    - java
      - com.example.aula01
        - MainActivity
        - com.example.aula01 (androidTest)
        - com.example.aula01 (test)
      - java (generated)
      - res
      - res (generated)
    - Gradle Scripts

Resource Manager

Structure

Build Variants

2: Favorites

Build Variants

```
1 package com.example.aula01
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10
11
12
13     }
14 }
```

Gradle

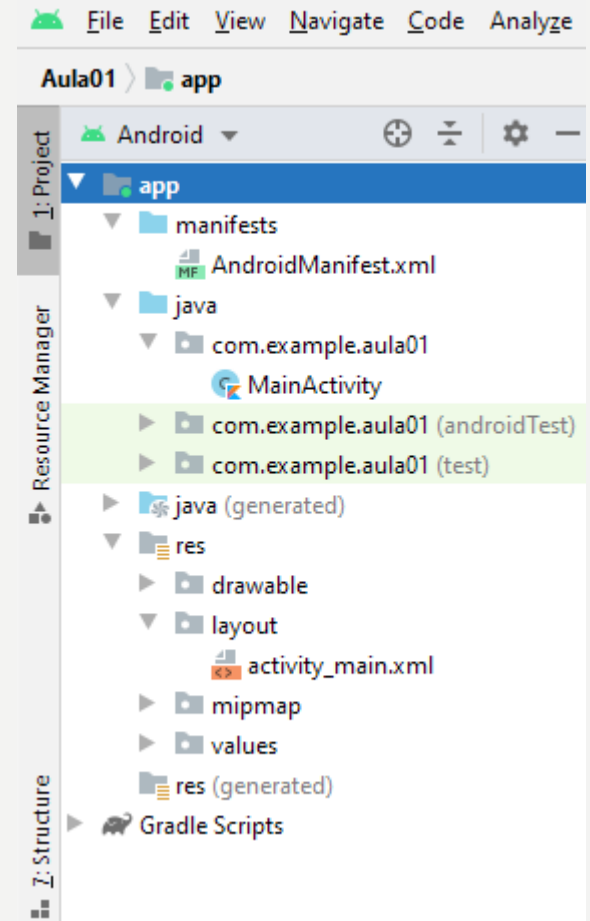
Device File Explorer

Emulator

**Android Studio 4.2 Beta 4 available**  
[Update...](#)

# ARQUIVOS DE PROJETO

- AndroidManifest.xml
  - Permissões especiais de uso do dispositivo
- MainActivity.kt
  - Programa Kotlin responsavel pela Activity
- activity\_main.xml
  - Layout da activity.



# MAINACTIVITY

```
package com.example.aula01

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

# ACTIVITY\_MAIN.XML

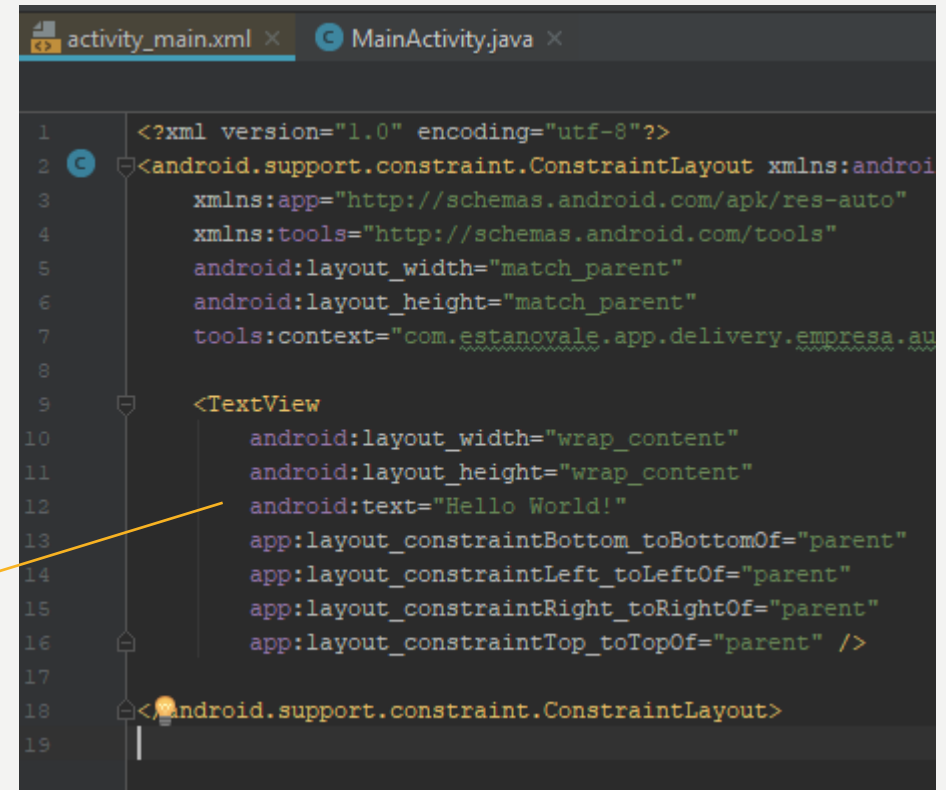
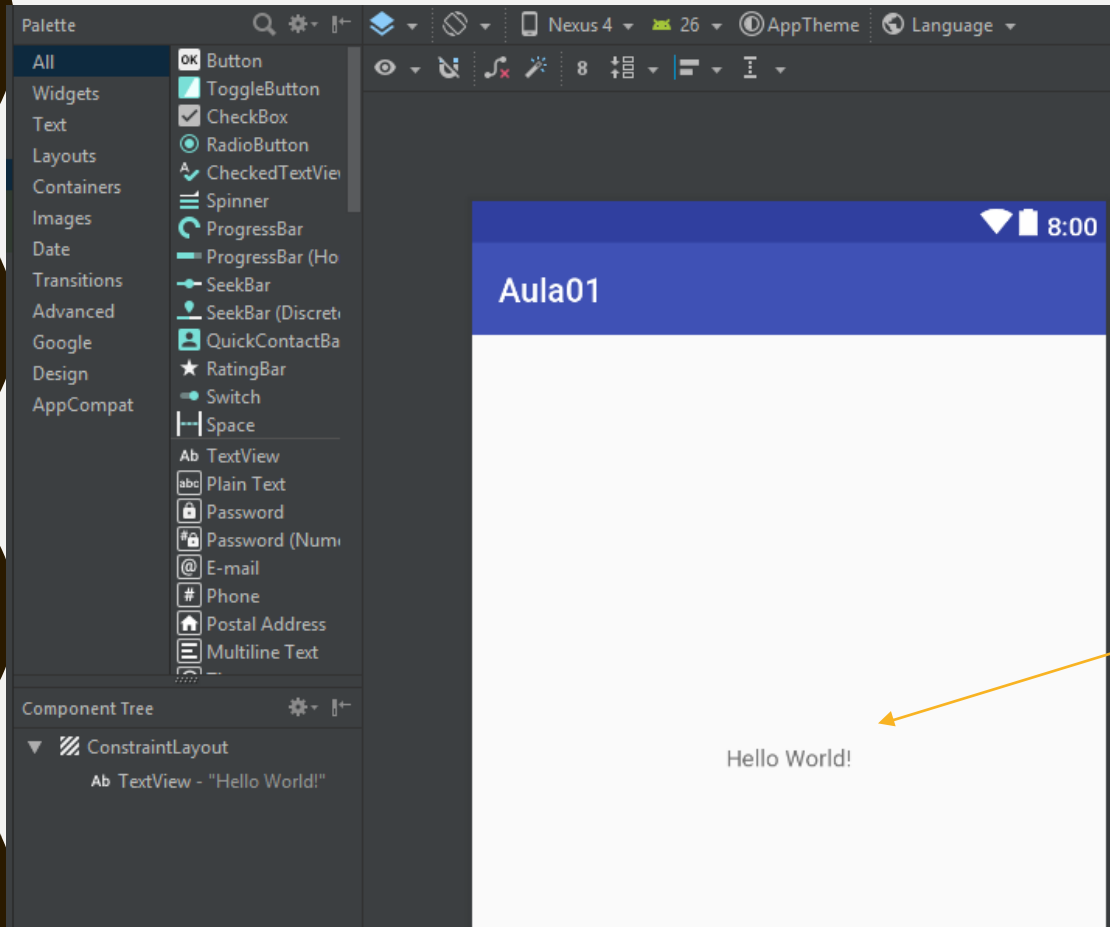
- O arquivo xml é responsável pelo layout do aplicativo, nele são definidas as posições de todos os componentes visuais.

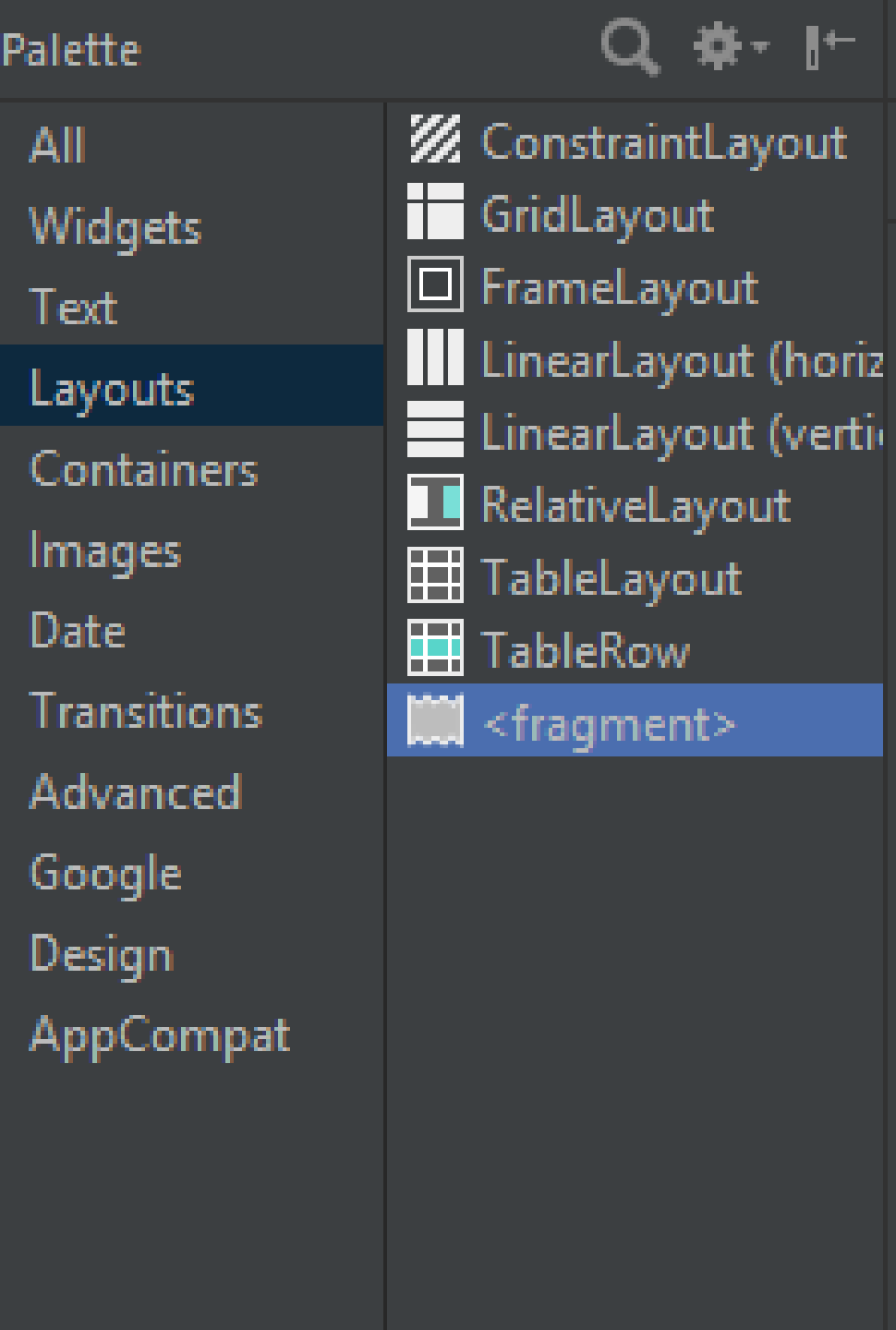
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

</androidx.constraintlayout.widget.ConstraintLayout>
```



# EDITOR GRÁFICO





# PALETA DE COMPONENTES VISUAIS

- Os layouts determinam como os componentes visuais serão distribuídos na janela.
- O Layout LinearLayout é o mais simples pois permite que os componentes sejam posicionados um abaixo do outro ou lado a lado.

# ESTRUTURA BÁSICA DO XML

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.estanovale.app.delivery.empresa.aula01.MainActivity">
    |
</android.support.constraint.ConstraintLayout>
```

Programa aqui seus componentes visuais.

# EDITTEXT

- Um elemento da interface do usuário para inserir e modificar texto. Ao definir um widget de edição de texto, você deve especificar o `R.styleable.TextView_inputType` atributo.
- Por exemplo, para entrada de texto simples, defina `inputType` como "text":

```
<EditText  
    android:id = "@ + id /name"  
    android:layout_height = "wrap_content"  
    android:layout_width = "match_parent"  
    android:inputType = "text" />
```

# EDITTEXT: PASSWORD

```
<EditText
```

```
    android:id="@+id/editTextTextPassword"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:inputType="textPassword" />
```

# EDITTEXT: EMAIL ADDRESS

```
<EditText  
    android:id="@+id/editTextTextEmailAddress"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
  
    android:inputType="textEmailAddress" />
```

# TEXTVIEW\_INPUTTYPE

date	Data
datetime	data e hora
None	Não pode ser editado pelo usuário
number	Número
numberDecimal	Número decimal
numberPassword	Senha numérica
numberSigned	Número com sinal
phone	Número de telefone

# TEXTVIEW\_INPUTTYPE

textPersonName	Nome da pessoa
time	Hora
text	Texto puro
textCapCharacters	Texto em maiúsculas.
textEmailAddress	Endereço de email
textPassword	Senha alfanumérica



# PRIMEIRO APP: ACTIVITY\_MAIN.XML

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/lblNome"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView" />

    <EditText
        android:id="@+id/txtNome"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:text="Name" />

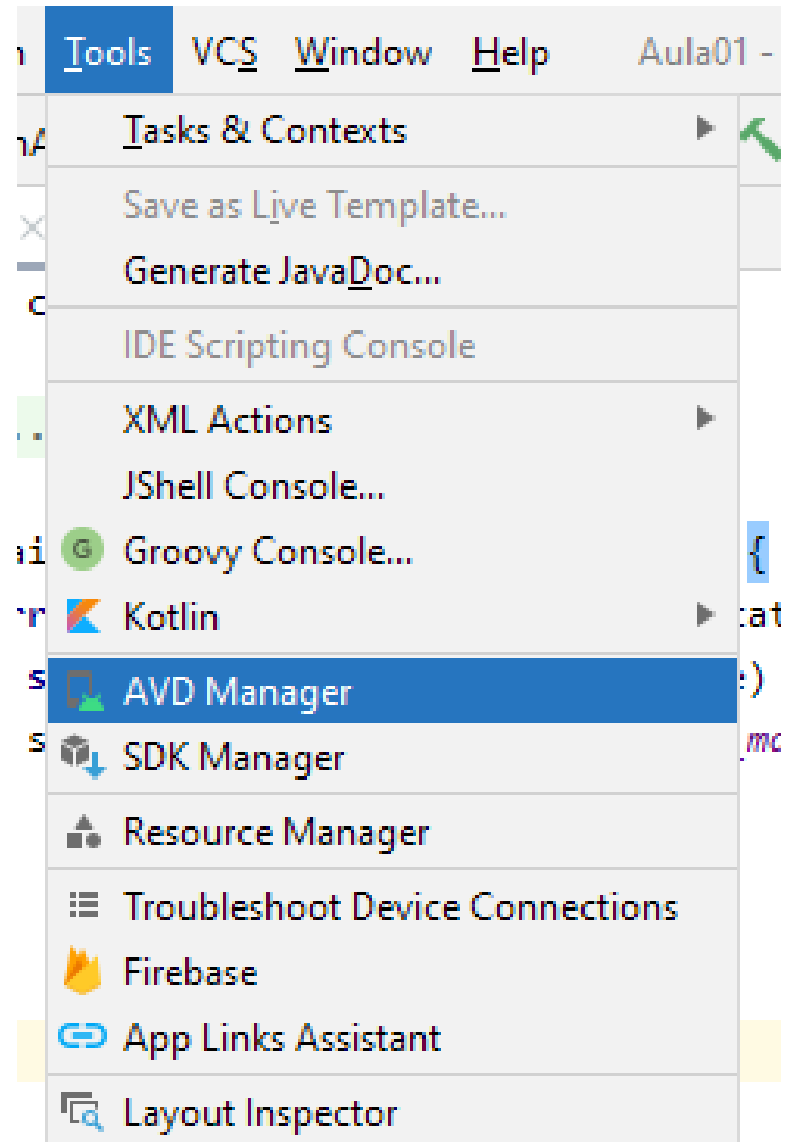
    <Button
        android:id="@+id/btn1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />

</LinearLayout>
```

# PRIMEIRO APP: MAINACTIVITY.KT

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        var txtNome: EditText = findViewById(R.id.txtNome)  
        var lblNome: TextView = findViewById(R.id.lblNome)  
        var btn1: Button = findViewById(R.id.btn1)  
  
        btn1.setOnClickListener { it: View!  
            var nome = txtNome.text.toString()  
            lblNome.text = nome  
        }  
    }  
}
```

# ADV MANAGER



# SEUS DISPOSITIVOS

Android Virtual Device Manager



## Your Virtual Devices

Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 5X API 27 - Oreo		1080 × 1920: 420dpi	27	Android 8.1 (Google Pl...	x86	9,0 GB	
	Pixel_3a_API_30_x86		1080 × 2220: 440dpi	30	Android 11.0 (Google A...	x86	801 MB	

+ Create Virtual Device...





## Select Hardware

## Choose a device definition

Category	Name	Play Store	Size	Resolution	Density
TV	Nexus 4		4,7"	768x1280	xhdpi
Phone	Nexus 5X		5,2"	1080x1920	420dpi
Wear OS	Nexus 5		4,95"	1080x1920	xxhdpi
Tablet	Nexus 4		4,7"	768x1280	xhdpi
Automotive	Galaxy Nexus		4,65"	720x1280	xhdpi
	8" Foldable		8,03"	2200x2480	420dpi

New Hardware Profile

Import Hardware Profiles



## Nexus 4



Size: normal  
Ratio: long  
Density: xhdpi

Clone Device...



Previous

Next

Cancel

Finish



## System Image

## Select a system image

Recommended [x86 Images](#) [Other Images](#)

Release Name	API Level ▾	ABI	Target
<b>R</b>	30	x86	Android 11.0 (Google APIs)
<a href="#">Q Download</a>	29	x86	Android 10.0 (Google APIs)
<a href="#">Pie Download</a>	28	x86	Android 9.0 (Google APIs)
<a href="#">Oreo Download</a>	27	x86	Android 8.1 (Google APIs)
<a href="#">Oreo Download</a>	26	x86	Android 8.0 (Google APIs)
<a href="#">Nougat Download</a>	25	x86	Android 7.1.1 (Google APIs)
<a href="#">Nougat Download</a>	24	x86	Android 7.0 (Google APIs)
<a href="#">Marshmallow Download</a>	23	x86	Android 6.0 (Google APIs)
<a href="#">Lollipop Download</a>	22	x86	Android 5.1 (Google APIs)

R



API Level

**30**

Android

**11.0****Google Inc.**

System Image

**x86**

We recommend these images because they run the fastest and support Google APIs.



[Previous](#)[Next](#)[Cancel](#)[Finish](#)





## Android Virtual Device (AVD)

### Verify Configuration

AVD Name

 Nexus 4	4.7 768x1280 xhdpi	<input type="button" value="Change..."/>
 R	Android 11.0 x86	<input type="button" value="Change..."/>

Startup orientation

 Portrait	 Landscape
-----------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------

### AVD Name

The name of this AVD.



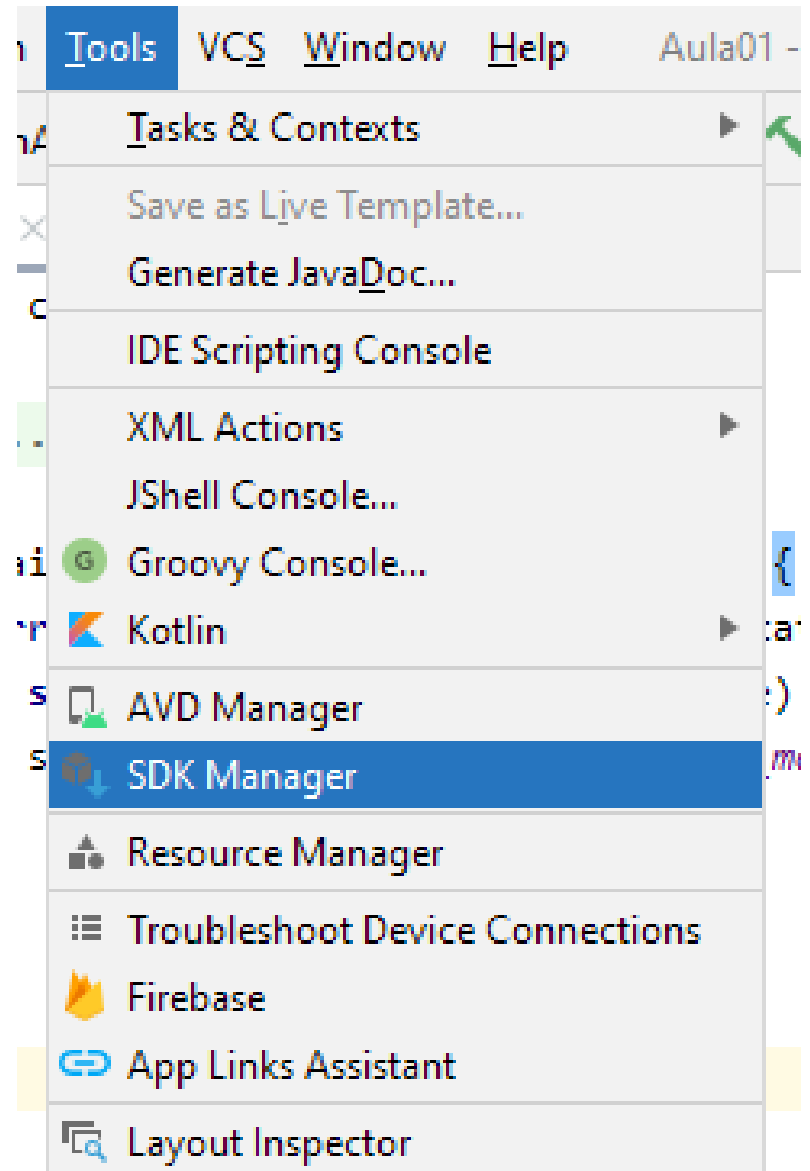
Previous

Next

Cancel

Finish

# SDK MENAGER







## ▼ Appearance &amp; Behavior

Appearance

Menus and Toolbars

## ▼ System Settings

Passwords

HTTP Proxy

Data Sharing

Date Formats

Updates

Android SDK

Memory Settings

Notifications

Quick Lists

Path Variables

## Keymap

## ▶ Editor

## Plugins

## ▶ Build, Execution, Deployment

## Kotlin

## ▶ Tools

## Appearance &amp; Behavior &gt; System Settings &gt; Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: [Edit](#) [Optimize disk space](#)

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Android 11.0 (R)	30	3	Not installed
<input type="checkbox"/>	Android 10.0 (Q)	29	5	Partially installed
<input checked="" type="checkbox"/>	Android 9.0 (Pie)	28	6	Installed
<input type="checkbox"/>	Android 8.1 (Oreo)	27	3	Partially installed
<input type="checkbox"/>	Android 8.0 (Oreo)	26	2	Not installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Not installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Partially installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Not installed
<input type="checkbox"/>	Android 4.4 (KitKat)	19	4	Not installed
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3	Not installed
<input type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3	Not installed
<input type="checkbox"/>	Android 4.1 (Jelly Bean)	16	5	Not installed
<input type="checkbox"/>	Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
<input type="checkbox"/>	Android 4.0 (IceCreamSandwich)	14	4	Not installed
<input type="checkbox"/>	Android 3.2 (Honeycomb)	13	1	Not installed

 Hide Obsolete Packages  Show Package Details

OK

Cancel

Apply



## ▼ Appearance &amp; Behavior

Appearance

Menus and Toolbars

## ▼ System Settings

Passwords

HTTP Proxy

Data Sharing

Date Formats

Updates

Android SDK

Memory Settings

Notifications

Quick Lists

Path Variables

## Keymap

## ▶ Editor

## Plugins

## ▶ Build, Execution, Deployment

## Kotlin

## ▶ Tools

## Appearance &amp; Behavior &gt; System Settings &gt; Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\aluno\AppData\Local\Android\Sdk

[Edit](#) [Optimize disk space](#)

SDK Platforms SDK Tools SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates.

Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
<input checked="" type="checkbox"/> Android SDK Build-Tools		Installed
<input type="checkbox"/> NDK (Side by side)		Not Installed
<input type="checkbox"/> Android SDK Command-line Tools (latest)		Not Installed
<input type="checkbox"/> CMake		Not Installed
<input type="checkbox"/> Android Auto API Simulators	1	Not installed
<input type="checkbox"/> Android Auto Desktop Head Unit Emulator	2.0.0 rc1	Not installed
<input checked="" type="checkbox"/> Android Emulator	30.3.5	Installed
<input type="checkbox"/> Android Emulator Hypervisor Driver for AMD Processors (installer)	1.6.0	Not installed
<input checked="" type="checkbox"/> Android SDK Platform-Tools	30.0.5	Installed
<input checked="" type="checkbox"/> Android SDK Tools	26.1.1	Installed
<input type="checkbox"/> Google Play APK Expansion library	1	Not installed
<input type="checkbox"/> Google Play Instant Development SDK	1.9.0	Not installed
<input type="checkbox"/> Google Play Licensing Library	1	Not installed
<input type="checkbox"/> Google Play services	49	Not installed
<input checked="" type="checkbox"/> Google USB Driver	13	Installed
<input type="checkbox"/> Google Web Driver	2	Not installed
<input checked="" type="checkbox"/> Intel x86 Emulator Accelerator (HAXM installer)	7.5.6	Installed
<input type="checkbox"/> Layout Inspector image server for API 29-30	6	Not installed

 Hide Obsolete Packages  Show Package Details

OK

Cancel

Apply