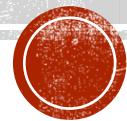


ANDROID

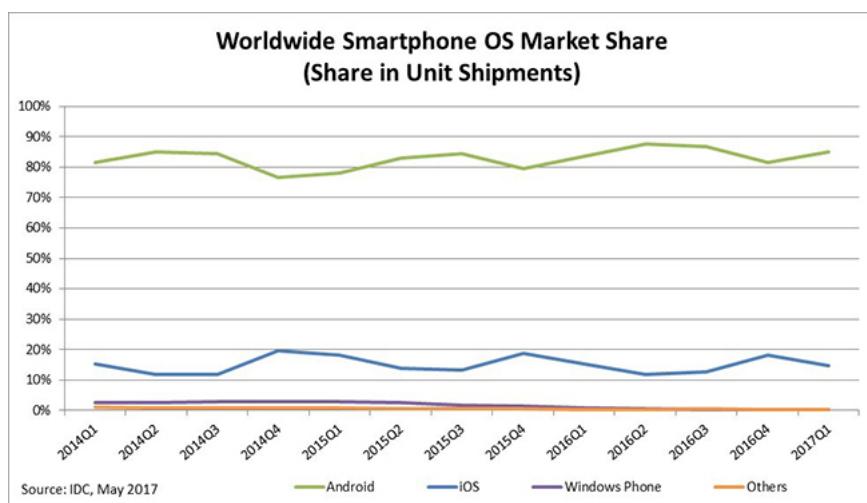
Introdução



Introdução

- SO baseado em Linux
- Máquina Virtual Java (Google) - Dalvik
- Objetivo de criar um concorrente para Symbian, BlackBerry e iPhone
- Desenvolvida pela Android Inc.
- Comprada pela Google em 2005
- Novembro de 2007, criada a Open Handset Alliance (OHA)
 - Aliança entre empresas para criar padrões abertos para telefonia móvel
 - Revelou a plataforma aberta Android
- Novembro/2007 : Uma versão da SDK divulgada para programadores
- Lançado comercialmente em no celular HTC Dream em Outubro/2008

Market Share



Prof. Razer A N R Montaño

<http://www.idc.com/promo/smartphone-market-share/os>

2017

3

Histórico

Versão	API Level	Nome	Ano
Android 1.0	1	Astro	Set/2008
Android 1.1	2	Battenberg	Fev/2009
Android 1.5	3	Cupcake	Abr/2009
Android 1.6	4	Donut	Set/2009
Android 2.0	5	Éclair	Out/2009
Android 2.0.1	6	Éclair	Dez/2009
Android 2.1	7	Éclair	Jan/2010
Android 2.2.X	8	Froyo	Mai/2010
Android 2.3, 2.3.2, 2.3.2	9	Gingerbread	Dez/2010
Android 2.3.3, 2.3.4, .2.3.5, 2.3.6, 2.3.7	10	Gingerbread	Fev/2011
Android 3.0	11	Honeycomb	Fev/2011

Prof. Razer A N R Montaño

2017

4

Histórico

Versão	API Level	Nome	Ano
Android 3.1	12	Honeycomb	Mai/2011
Android 3.2	13	Honeycomb	Jul/2011
Android 4.0, 4.0.1, 4.0.2	14	Ice Cream Sandwitch	Out/2011
Android 4.0.3, 4.0.4	15	Ice Cream Sandwitch	Dez/2011
Android 4.1.X	16	Jelly Bean	Jul/2012
Android 4.2.X	17	Jelly Bean	Nov/2012
Android 4.3.X	18	Jelly Bean	Jul/2013
Android 4.4.X	19, 20	KitKat	Out/2013
Android 5.0, 5.0.1, 5.0.2	21	Lollipop	Nov/2014
Android 5.1, 5.1.1	22	Lollipop	Mar/2015

Prof. Razer A N R Montaño

2017

5

Histórico

Versão	API Level	Nome	Ano
Android 6.0	23	Marshmallow	Out/2015
Android 7.0	24	Nougat	Ago/2016
Android 7.1, 7.1.1, 7.1.2	25	Nougat	Dez/2016
Android 8.0	26	Oreo	Ago/2017
Android 8.1			Dez/2017

Prof. Razer A N R Montaño

2017

6

VM Dalvik

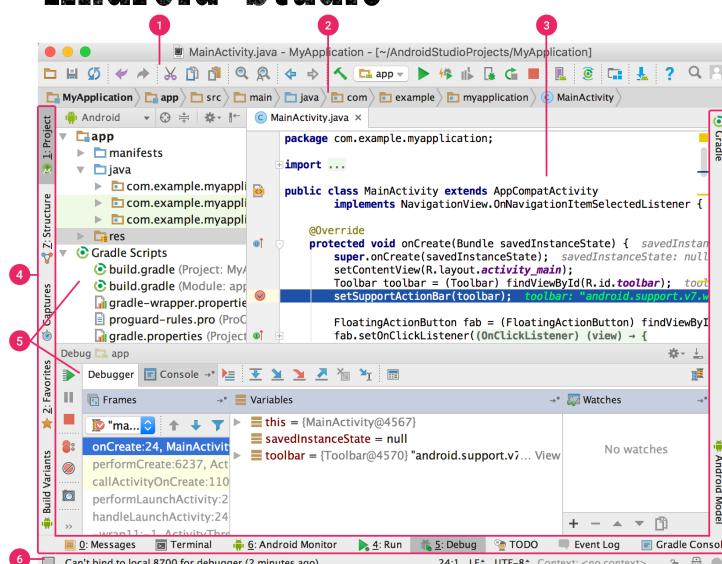
- Máquina virtual otimizada
 - Rodar com poucos recursos, como memória
- Ferramenta "dx" do SDK transforma *bytecode Java (.class)* em arquivo da Dalvik (.dex)
- Para desenvolver em Android:
 - Link: <https://developer.android.com/index.html>
- Ferramentas mínimas
 - Android Studio:
 - Baseado em IntelliJ IDEA
 - <https://developer.android.com/studio/index.html?hl=pt-br>
 - Java SDK 7 (mínimo)
 - Android SDK (vem com o Android Studio)

Prof. Razer A N R Montaño

2017



Android Studio



- 1) A barra de ferramentas
- 2) A barra de navegação
- 3) A janela do editor
- 4) A barra de janela de ferramentas
- 5) A janela das ferramentas
- 6) A barra de status

Prof. Razer A N R Montaño

2017



Sistema de Compilação Gradle

- Ferramenta para gerenciamento do projeto e controle de dependências
- Build de aplicativos Android
 - **Etapa 1:** Android Asset Packaging Tool (aapt) busca arquivos de recursos (ex, AndroidManifest), leiautes (xml), conteúdo da pasta res, etc, gerando o R.java
 - **Etapa 2:** Interfaces .aidl (Android Interface Definition Language) são transformadas em interfaces Java. AIDL é uma linguagem usada para que dois processos possam conversar entre si
 - **Etapa 3:** Todo o código é compilado, gerando os .class
 - **Etapa 4:** Os .class são convertidos para bytecode da Dalvik (DEX)
 - **Etapa 5:** Todos os recursos não compilados (imagens, etc), recursos compilados e arquivos DEX são enviados para serem empacotados em um .apk
 - **Etapa 6:** O .apk é construído e assinado (debug / release). Se for release o .apk deve ser alinhado para otimizá-lo e diminuir o consumo de memória

Sistema de Compilação Gradle

- Gradle automatiza o processo de construção (build) do aplicativo
- Configuração do Gradle: **build.gradle**
- Definem-se repositórios para resolução de dependências
 - JCenter, Maven Central, Ivy
- Arquivo build.gradle do projeto
 - Define repositórios e dependências do Gradle
- Arquivo build.gradle do módulo
 - Configurações específicas do módulo
 - Pode substituir informações do manifesto (opções customizadas)

build.gradle do Projeto

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.0.1'
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```

Prof. Razer A N R Montaño

2017

11

build.gradle do Projeto

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.0.1'
    }
}
```

```
allprojects {
    repositories {
        jcenter()
    }
}
```

Configura o plugin do Gradle para Android

- **repositories:** repositórios usados para buscar dependências do Gradle
- **dependencies:** dependências do plugin do Gradle

Prof. Razer A N R Montaño

2017

12

build.gradle do Projeto

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.0.1'
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```

Configurações que ficam disponíveis para todos os módulos do projeto

Prof. Razer A N R Montaño

2017

13

build.gradle do Módulo

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 19
    buildToolsVersion "19.0.0"

    defaultConfig {
        minSdkVersion 8
        targetSdkVersion 19
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled true
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
dependencies {
    compile project(":lib")
    compile 'com.android.support:appcompat-v7:19.0.1'
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

Usar o plugin do Android e configura os elementos dentro de android {...}

Prof. Razer A N R Montaño

2017

14

build.gradle do Módulo

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 19
    buildToolsVersion "19.0.0"

    defaultConfig {
        minSdkVersion 8
        targetSdkVersion 19
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled true
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile project(":lib")
    compile 'com.android.support:appcompat-v7:19.0.1'
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

Prof. Razer A N R Montaño

2017

15

Configurações específicas de compilação:

- **compileSdkVersion**: Versão da plataforma que será compilado o aplicativo.
- **buildToolsVersion**: Versão das ferramentas de compilação da plataforma.
- **defaultConfig**: Configurações adicionadas dinamicamente no `AndroidManifest.xml`. Sempre sobreescrivem o manifesto.
- **buildTypes**: Como a compilação será feita. Dois tipos: `debug` e `release`.

build.gradle do Módulo

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 19
    buildToolsVersion "19.0.0"

    defaultConfig {
        minSdkVersion 8
        targetSdkVersion 19
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled true
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile project(":lib")
    compile 'com.android.support:appcompat-v7:19.0.1'
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

Prof. Razer A N R Montaño

2017

16

Configura dependências e bibliotecas do módulo

build.gradle do Módulo

```
dependencies {
    compile project(":lib")
    compile 'com.android.support:appcompat-v7:19.0.1'
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

Dependências de módulo

Se este módulo depende de outros módulos, ao ser construído, os demais são incluídos

build.gradle do Módulo

```
dependencies {
    compile project(":lib")
    compile 'com.android.support:appcompat-v7:19.0.1'
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

Dependências externas

Se tem dependência disponível em um repositório externo, não há necessidade de baixar e adicionar ao projeto

Ex.: Versão 16.0.1 do Google Guava (utilidades) no mavenCentral()

`com.google.guava:guava:16.0.1`

build.gradle do Módulo

```
dependencies {  
    compile project(":lib")  
    compile 'com.android.support:appcompat-v7:19.0.1'  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
}
```

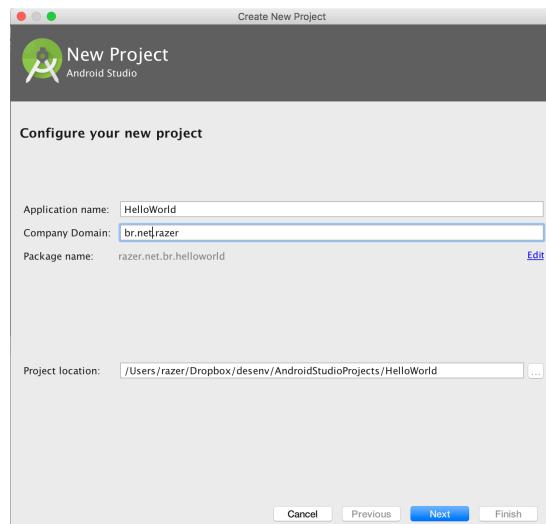
Dependências Locais

Se o projeto depende de bibliotecas como JARs, aqui eles são declarados

20

Primeiro Projeto

Hello World

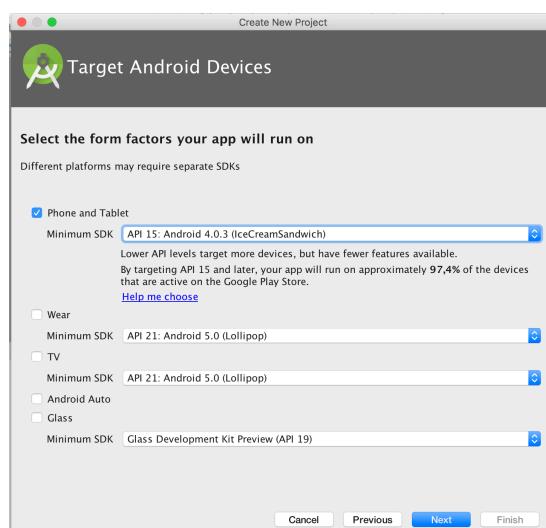


Prof. Razer A N R Montaño

2017

21

Hello World

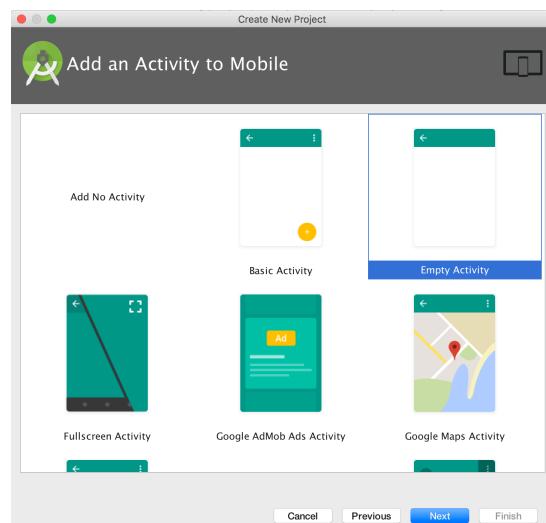


Prof. Razer A N R Montaño

2017

22

Hello World

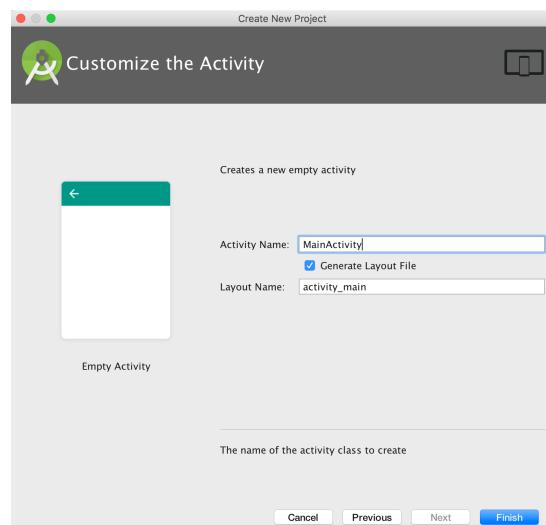


Prof. Razer A N R Montaño

2017

23

Hello World

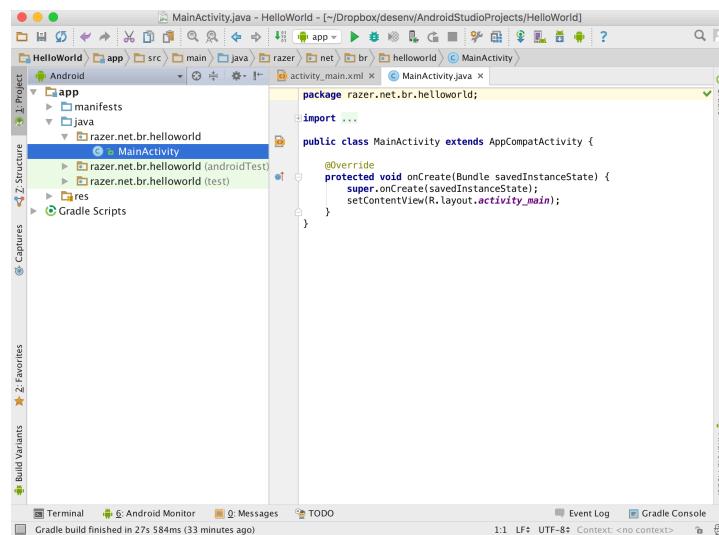


Prof. Razer A N R Montaño

2017

24

Hello World



The screenshot shows the Android Studio interface with the project 'HelloWorld' open. The code editor displays the MainActivity.java file, which contains the following Java code:

```

package razer.net.br.helloworld;
import ...;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

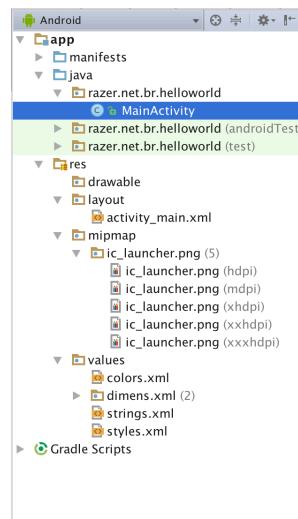
The 'app' folder in the Project Structure is expanded, showing the manifest, Java, and resources. The Java folder contains the MainActivity class. The resources folder contains layout, drawable, and mipmap folders. The mipmap folder contains several versions of an icon named 'ic_launcher'. The code editor has syntax highlighting and a code completion dropdown.

Prof. Razer A N R Montaño

2017

25

Hello World



Prof. Razer A N R Montaño

2017

26

Hello World



```

activity_main.xml x MainActivity.java x
package razer.net.br.helloworld;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

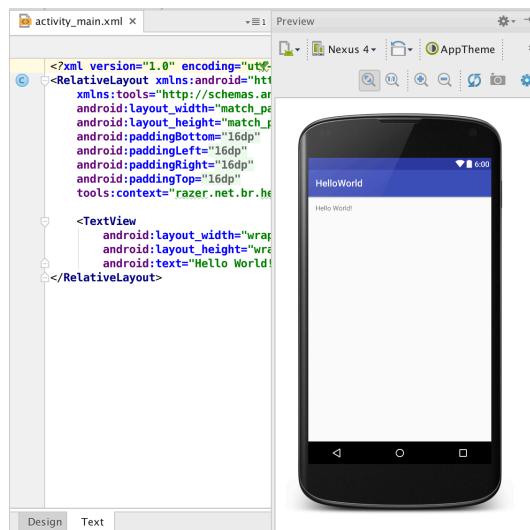
```

Prof. Razer A N R Montaño

2017

27

Hello World

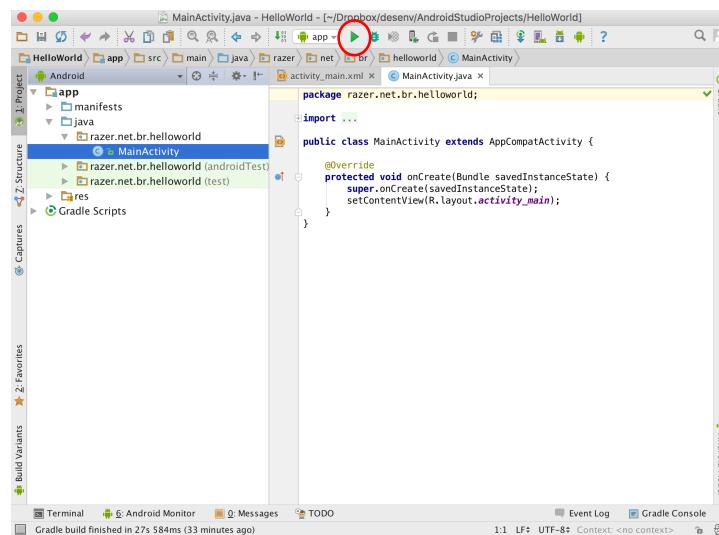


Prof. Razer A N R Montaño

2017

28

Hello World



```

package razer.net.br.helloworld;
import ...;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

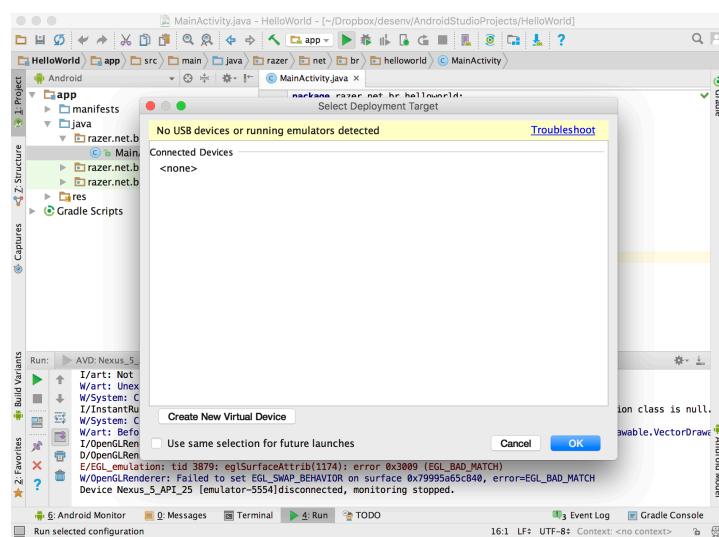
```

Prof. Razer A N R Montaño

2017

29

Hello World



No USB devices or running emulators detected

Select Deployment Target

Connected Devices <none>

Create New Virtual Device

Device Nexus_5 API_25

EGL_emulation: tid 3879: eglSurfaceAttrib(1174): error 0x3009 (EGL_BAD_MATCH)

W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0x79995a65c840, error=EGL_BAD_MATCH

Device Nexus_5 API_25 [emulator-5554] disconnected, monitoring stopped.

Prof. Razer A N R Montaño

2017

30

Hello World

```

package razer.net.br.helloworld;
import ...;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

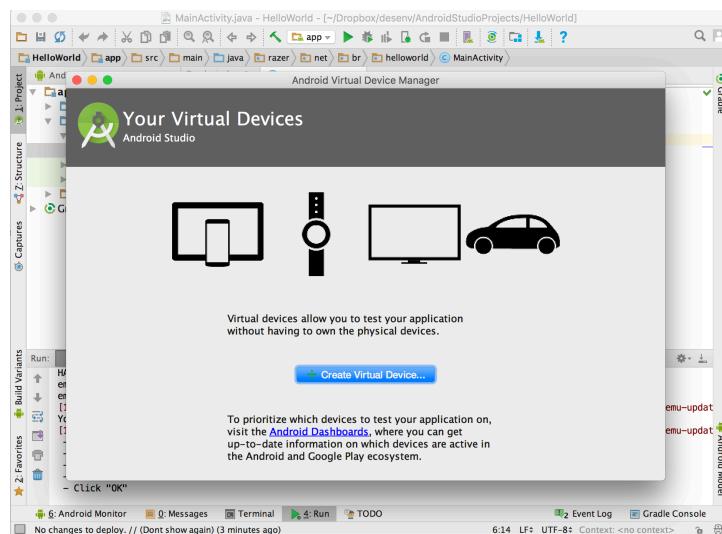
```

Prof. Razer A N R Montaño

2017

31

Hello World

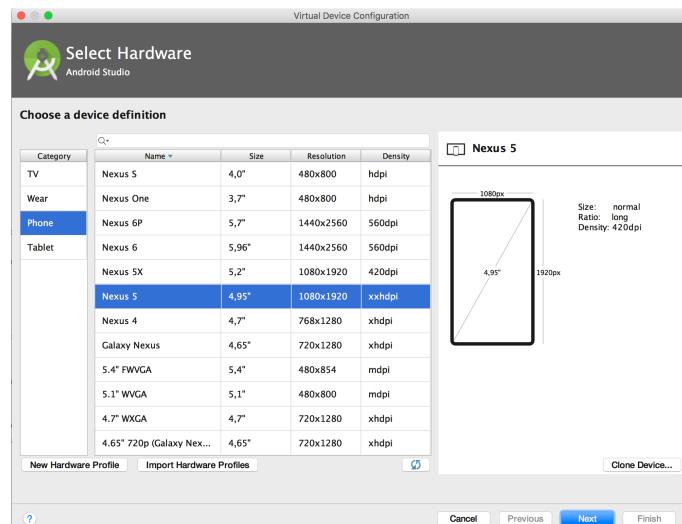


Prof. Razer A N R Montaño

2017

32

Hello World

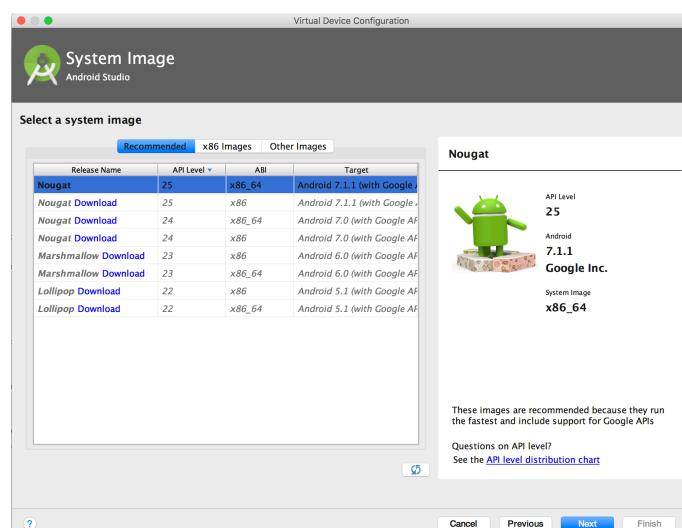


Prof. Razer A N R Montaño

2017

33

Hello World

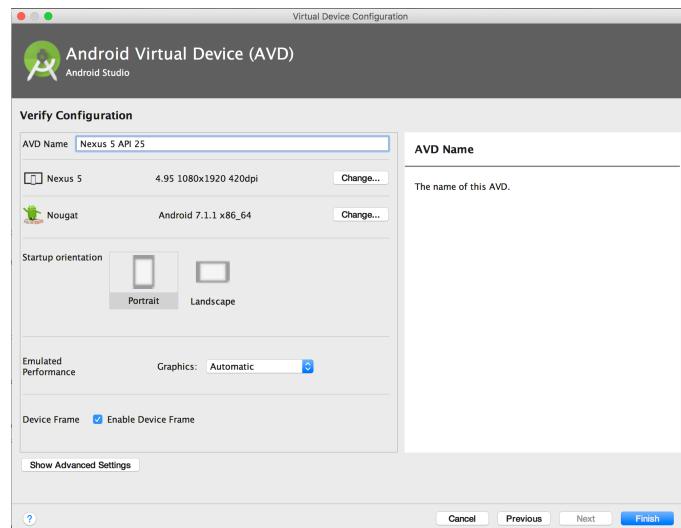


Prof. Razer A N R Montaño

2017

34

Hello World

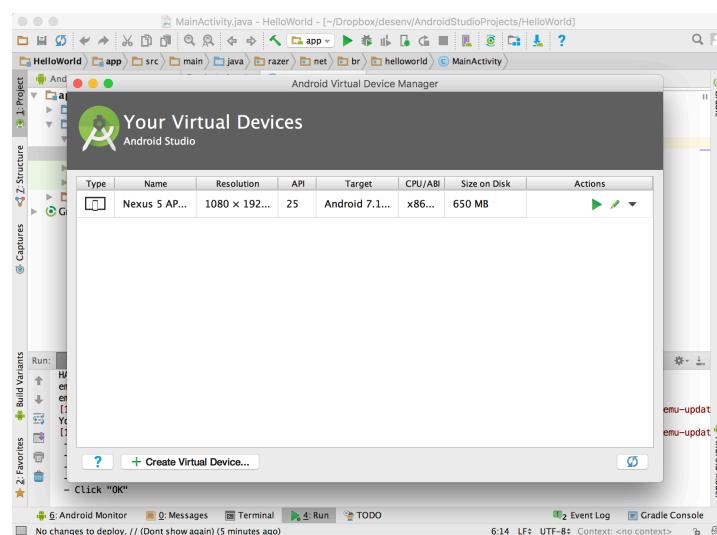


Prof. Razer A N R Montaño

2017

35

Hello World

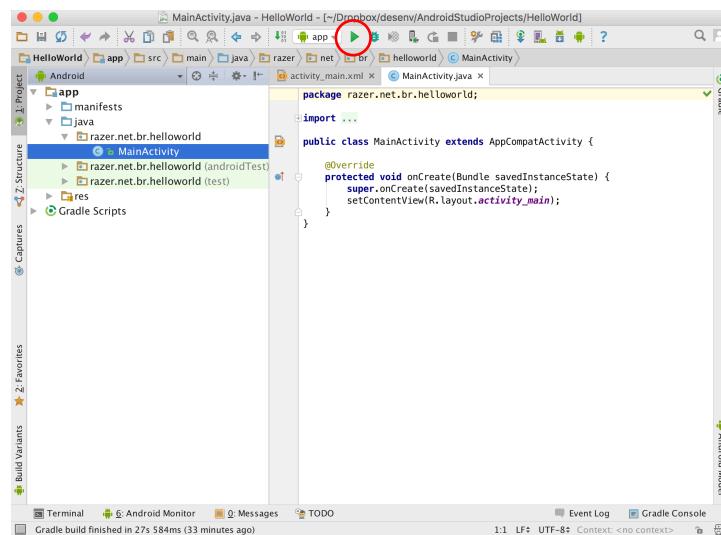


Prof. Razer A N R Montaño

2017

36

Hello World

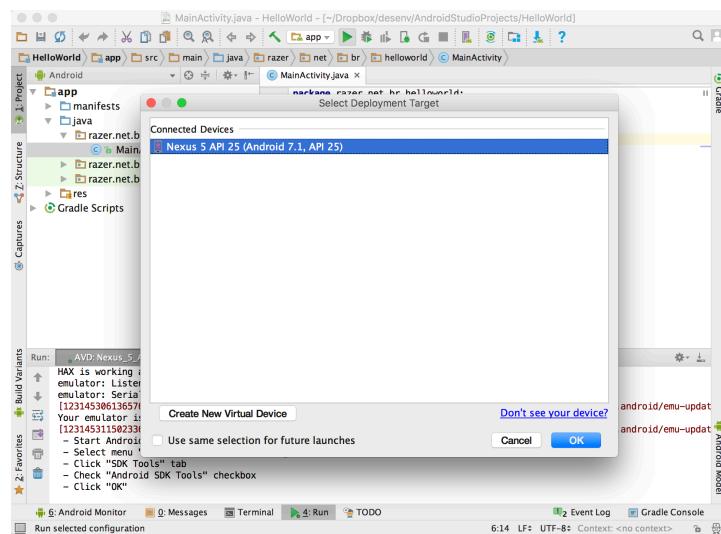


Prof. Razer A N R Montaño

2017

37

Hello World

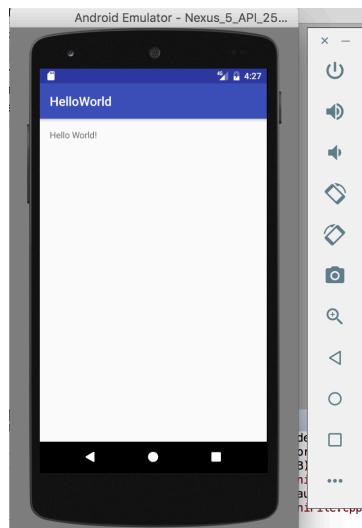


Prof. Razer A N R Montaño

2017

38

Hello World



Prof. Razer A N R Montaño

2017

39

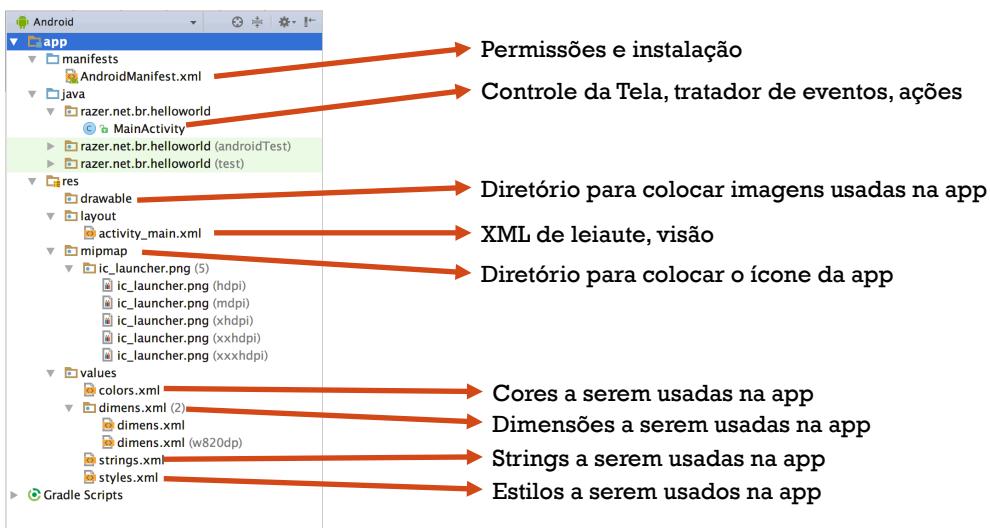
40

Elementos do Projeto

Prof. Razer A N R Montaño

2017

Hello World



Prof. Razer A N R Montaño

2017

41

Elementos do Projeto

- Diretório "app"
 - Contém todos os arquivos da aplicação
 - Pastas:
 - **manifests**
 - **AndroidManifest.xml**: contém permissões e configurações da aplicação.
 - **java**
 - Arquivos .java que fazem parte da aplicação, dentro dos pacotes definidos na criação do projeto
 - Contém a primeira Activity, que é a classe principal da aplicação
 - **res**
 - Contém todos os recursos da aplicação

Prof. Razer A N R Montaño

2017

42

Elementos do Projeto

- Dentro da pasta "res"

- Pastas:

- **drawable**

- Armazena imagens para serem usadas na aplicação

- **layout**

- Armazena todos os arquivos XML referentes às telas da aplicação

- **mindmap**

- Armazena imagens do ícone da aplicação, para se adequar às várias resoluções de tela

- **values**

- Armazena constantes para serem usadas na aplicação

- **menu**

- Armazena arquivos XML referentes aos menus da aplicação

Elementos do Projeto

- Pasta "**drawable**"

- Armazena imagens para serem usadas na aplicação

- Arquivo Bitmap (.png, .jpg, .gif)

- Arquivo Nine-Patch (png com áreas esticáveis, .9.png)

- Para usá-lo:

- No Java: `R.drawable.nome_do_arquivo`

- No XML: `@drawable/nome_do_arquivo`

Elementos do Projeto

- Pasta "values"

- Armazena constantes a serem usadas no projeto
- **colors.xml**: armazena constantes de cores, ex:

```
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

- **strings.xml**: armazena constantes strings para serem usadas
- **dimens.xml**: armazena constantes relacionadas a dimensões
- **styles.xml**: armazena constantes de estilo

Elementos do Projeto

- Pasta "values"

- Armazena constantes a serem usadas no projeto
- **colors.xml**: armazena constantes de cores, ex:
- **strings.xml**: armazena constantes strings para serem usadas

```
<resources>
    <string name="app_name">MinhaApp</string>
    <string name="teste">Meu teste</string>
    <string-array name="itens">
        <item>Item 1</item>
        <item>Sair</item>
    </string-array>
</resources>
```

- **dimens.xml**: armazena constantes relacionadas a dimensões
- **styles.xml**: armazena constantes de estilo

Elementos do Projeto

- Pasta "values"

- Armazena constantes a serem usadas no projeto
- **colors.xml**: armazena constantes de cores, ex:
- **strings.xml**: armazena constantes strings para serem usadas
- **dimens.xml**: armazena constantes relacionadas a dimensões

```
<resources>
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

- **styles.xml**: armazena constantes de estilo

Elementos do Projeto

- Pasta "values"

- Armazena constantes a serem usadas no projeto
- **colors.xml**: armazena constantes de cores, ex:
- **strings.xml**: armazena constantes strings para serem usadas
- **dimens.xml**: armazena constantes relacionadas a dimensões
- **styles.xml**: armazena constantes de estilo

```
<resources>
    <style name="AppTheme" parent="Theme.AppCompat.Light">
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

XML Layout

- Link: <https://developer.android.com/guide/topics/ui/declaring-layout.html?hl=pt-br>
- Serve para:
 - Declarar a interface da atividade
 - Layout da tela
 - Componentes da tela

Prof. Razer A N R Montaño

2017

49

Hello World – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="razer.net.br.helloworld.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
</RelativeLayout>
```

Prof. Razer A N R Montaño

2017

50

Atividade

- Link: <https://developer.android.com/reference/android/app/Activity.html>
- Serve para:
 - Carregar a visão da tela (XML)
 - Interceptar as ações do usuário
 - Efetuar o processamento necessário

Hello World – MainActivity.java

```
package razer.net.br.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

AndroidManifest.xml

- Link: <https://developer.android.com/guide/topics/manifest/manifest-intro.html?hl=pt-br>
- Serve para:
 - Nomear a aplicação
 - Indicar o ícone
 - Indicar permissões da aplicação
 - Descrever componentes da aplicação: Activities, Serviços, etc
 - Indicar o nível mínimo do android
 - Indicar o tema da aplicação
 - etc

Hello World – AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="razer.net.br.helloworld">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

55

Alterações - 1

Prof. Razer A N R Montaño

2017

Hello World Alterado - 1

- Adicionar mais uma caixa de texto
- Como está usando um RelativeLayout, deve-se:
 - Adicionar um id ao TextView existente
 - Adicionar outro TextView (copiar e colar o anterior)
 - Indicar a posição do segundo TextView relativa ao primeiro
- A classe MainActivity não é alterada

Prof. Razer A N R Montaño

2017

56

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="razer.net.br.helloworld.MainActivity">

    <TextView
        android:id="@+id/texto1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
    <TextView
        android:id="@+id/texto2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto1"
        android:text="Meu teste lindo!!!" />
</RelativeLayout>
```

Prof. Razer A N R Montaño

2017

57

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="razer.net.br.helloworld.MainActivity">

    <TextView
        android:id="@+id/texto1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
    <TextView
        android:id="@+id/texto2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto1"
        android:text="Meu teste lindo!!!" />
</RelativeLayout>
```

ID: "@+id/" indica que é um ID que será adicionado aos recursos. O ID propriamente dito vem após

Posição relativa: abaixo do elemento com o id "texto1"

Prof. Razer A N R Montaño

2017

58



Exercícios

1. Executar as duas versões de HelloWorld mostradas
2. Alterar a última versão (a que possui 2 TextViews) para adicionar mais um TextView, abaixo de todos, contendo seu nome completo

60

Alterações - 2

Hello World Alterado - 2

- Adicionar mais uma caixa de entrada de texto (EditText) e um botão (Button)
- O esquema de posicionamento RelativeLayout funciona para todos os componentes
 - Indicar a posição dos novos componentes, um abaixo do outro
- Tratar o clique do botão, para apresentar o que o usuário digitou na entrada de texto
 - Alteração em MainActivity

Prof. Razer A N R Montaño

2017

61

activity_main.xml

```
<RelativeLayout ... >
    <TextView
        android:id="@+id/texto1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
    <TextView
        android:id="@+id/texto2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto1"
        android:text="Meu teste lindo!!!" />
    <TextView
        android:id="@+id/texto3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto2"
        android:text="Razer Anthom Nizer Rojas Montaño" />
    <EditText
        android:id="@+id/edit1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto3" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edit1"
        android:text="Processar"
        android:onClick="processar" />
</RelativeLayout>
```

Prof. Razer A N R Montaño

2017

62

activity_main.xml

```

<RelativeLayout ... >
    <TextView
        android:id="@+id/texto1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
    <TextView
        android:id="@+id/texto2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto1"
        android:text="Meu teste lindo!!!" />
    <TextView
        android:id="@+id/texto3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto2"
        android:text="Razer Anthon Nizer Rojas Montaño" />
    <EditText
        android:id="@+id/edit1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texto3" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edit1"
        android:text="Processar"
        android:onClick="processar"/>
</RelativeLayout>

```

Componente para entrada de texto

Botão

Texto do Botão

O método da Activity a ser invocado
Quando o botão é pressionado

Prof. Razer A N R Montaño

2017

63

MainActivity.java

```

package razer.net.br.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void processar(View view) {
        EditText edit = (EditText)findViewById(R.id.edit1);
        String str = edit.getText().toString();
        Toast t = Toast.makeText(getApplicationContext(),
            "Digitado: " + str,
            Toast.LENGTH_SHORT);
        t.show();
    }
}

```

Prof. Razer A N R Montaño

2017

64

MainActivity.java

```

package razer.net.br.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void processar(View view) {
        EditText edit = (EditText)findViewById(R.id.edit1);
        String str = edit.getText().toString();
        Toast t = Toast.makeText(getApplicationContext(),
            "Digitado: " + str,
            Toast.LENGTH_SHORT);
        t.show();
    }
}

```

Método processar() precisa receber um View como parâmetro

Obtém um componente, dado seu ID

Obtém a string dentro do EDIT

Apresenta uma mensagem na tela



Exercícios

1. Executar a versão alterada do HelloWorld com botão
2. Adicionar mais um EditText e mais um Button nesta aplicação, de tal forma que ao ser pressionado o segundo botão deve ser mostrada a concatenação dos conteúdos dos dois EditTexts



Novo Projeto

Prof. Razer A N R Montaño

2017

Novo Projeto

- Criar um novo projeto vazio
- Criar um pequeno formulário contendo rótulos (TextViews) e caixas de entrada de texto (EditTexts), um ao lado do outro
 - Deve-se criar um LinearLayout para cada linha
- Um botão que, ao ser pressionado, apresenta os dados digitados
- Os dados são:
 - Nome
 - E-mail

Prof. Razer A N R Montaño

2017



Explicações

- RelativeLayout: organiza os elementos posicionando-os um relativo ao outro
- LinearLayout: organiza os elementos de forma linear, ou horizontalmente ou verticalmente
- Pode-se ter um layout dentro do outro (que é o caso aqui)

```
<RelativeLayout ... >
    <LinearLayout
        android:id="@+id/linha1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        ....COMPONENTES DO FORMULÁRIO.....
    </LinearLayout>
    <LinearLayout
        android:id="@+id/linha2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/linha1"
        android:orientation="horizontal">
        ....COMPONENTES DO FORMULÁRIO.....
    </LinearLayout>
</RelativeLayout>
```

Prof. Razer A N R Montaño

2017

69

Explicações

```
<RelativeLayout ... >
    <LinearLayout
        android:id="@+id/linha1" → ID dos LinearLayouts
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        ....COMPONENTES DO FORMULÁRIO.....
    </LinearLayout>

    <LinearLayout
        android:id="@+id/linha2" → Indica que o segundo LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/linha1"
        android:orientation="horizontal">
        ....COMPONENTES DO FORMULÁRIO.....
    </LinearLayout>
</RelativeLayout>
```

Prof. Razer A N R Montaño

2017

70

activity_main.xml

```
<RelativeLayout ... >

    <LinearLayout
        android:id="@+id/linha1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/lbl_nome"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nome:" />
        <EditText
            android:id="@+id edt_nome"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>

    <LinearLayout
        android:id="@+id/linha2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/linha1"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/lbl_email"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="E-mail:" />
        <EditText
            android:id="@+id edt_email"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/linha2"
        android:onClick="processar"
        android:text="Processar"/>
</RelativeLayout>
```

Prof. Razer A N R Montaño

2017

71

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void processar(View view) {
        EditText edtNome = (EditText) findViewById(R.id.edt_nome);
        EditText edtEmail = (EditText) findViewById(R.id.edt_email);

        String strNome = edtNome.getText().toString();
        String strEmail = edtEmail.getText().toString();

        String texto = "Nome: " + strNome + "\nE-Mail: " + strEmail;

        Toast t = Toast.makeText(getApplicationContext(), texto, Toast.LENGTH_SHORT);
        t.show();
    }
}
```

Prof. Razer A N R Montaño

2017

72

MainActivity.java

```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void processar(View view) {
        EditText edtNome = (EditText) findViewById(R.id.edt_nome);
        EditText edtEmail = (EditText) findViewById(R.id.edt_email);

        String strNome = edtNome.getText().toString();
        String strEmail = edtEmail.getText().toString();

        String texto = "Nome: " + strNome + "\nE-Mail:" + strEmail;

        Toast t = Toast.makeText(getApplicationContext(), texto, Toast.LENGTH_SHORT);
        t.show();
    }
}

```

Método que trata o clique do botão

Obtém os Edits da tela

Obtém os textos de dentro dos Edits recuperados

Apresenta a mensagem



Exercícios

1. Executar o projeto descrito
2. Adicionar os campos Endereço e Cidade ao formulário, e apresentá-los na mensagem



Aplicação no Celular

Prof. Razer A N R Montaño

2017

Rodar a Aplicação no Celular

- Habilitar "Modo Desenvolvedor" no celular
 - Acessar o menu de configuração do celular
 - Submenu "Sistema" | "Sobre o telefone"
 - Encontre o "Número da versão" ou "Número de compilação"
 - Toque 7 vezes sucessivamente até aparecer a mensagem "Você agora é desenvolvedor"
 - Voltar ao Menu "Sistema" e escolha "Opções do desenvolvedor"
 - Procure a opção "Depuração de USB" e ative-a

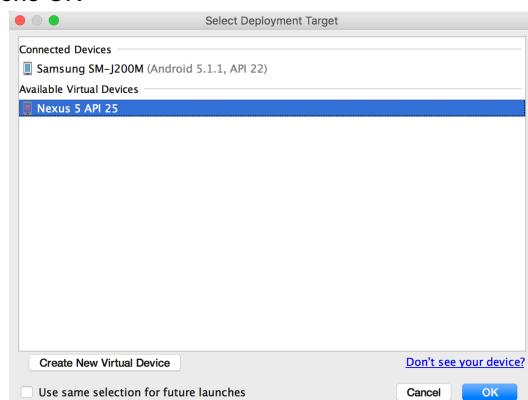
Prof. Razer A N R Montaño

2017



Rodar a Aplicação no Celular

- Instalar a aplicação no celular
 - Plugue o celular no computador, via USB
 - No Android Studio aparecerá a opção do seu celular
 - Escolha-a e pressione OK



Prof. Razer A N R Montaño

2017

77

Rodar a Aplicação no Celular

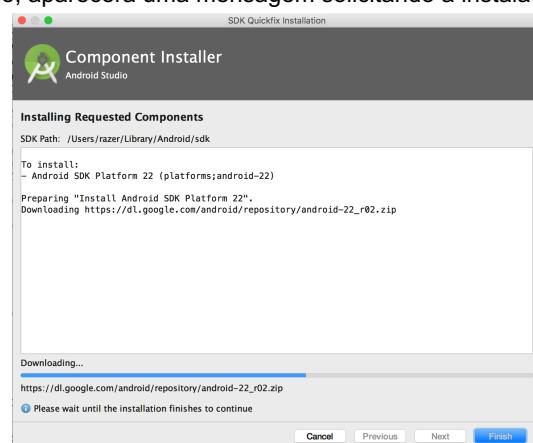
- Se a SDK da plataforma do celular já estiver instalada, logo a aplicação será executada
 - Caso contrário, aparecerá uma mensagem solicitando a instalação



Prof. Razer A N R Montaño

2017

78



Rodar a Aplicação no Celular

- Ao terminar a instalação
 - Pressionar "Finish"
 - A aplicação será instalada no celular
 - A aplicação será executada