

Introdução ao Android

SECAP – 2014

Prof. Rone Ilídio - UFSJ

O que é o Android?

- Sistema operacional móvel, baseado numa versão modificada do Linux.
- Código aberto: qualquer pessoa pode baixar e modificar
- Máquina virtual: Dalvik

Instalação

- Ferramentas necessárias
 - Java JDK
 - Eclipse
 - Android SDK
- Guia de instalação

<http://www.programarandroid.com.br/2013/11/como-instalar-o-android-studio-no.html>
- Android Studio
 - <http://android-developers.blogspot.in/2013/05/android-studio-ide-built-for-android.html>

Instalação

- Existem vários tutoriais na Internet, veja alguns:
 - <http://www.androidbrasilprojetos.org/tutoriais/instalando-e-configurando-o-android-sdkeclipse/>
 - <http://www.forumweb.com.br/artigo/38/instalando-o-plugin-adb-no-eclipse-e-criando-um-projeto-android>
 - <http://www.universowap.com.br/antenado/sites-wap/como-instalar-e-configurar-o-sdk-do-android/>

Android Virtual Device - AVD

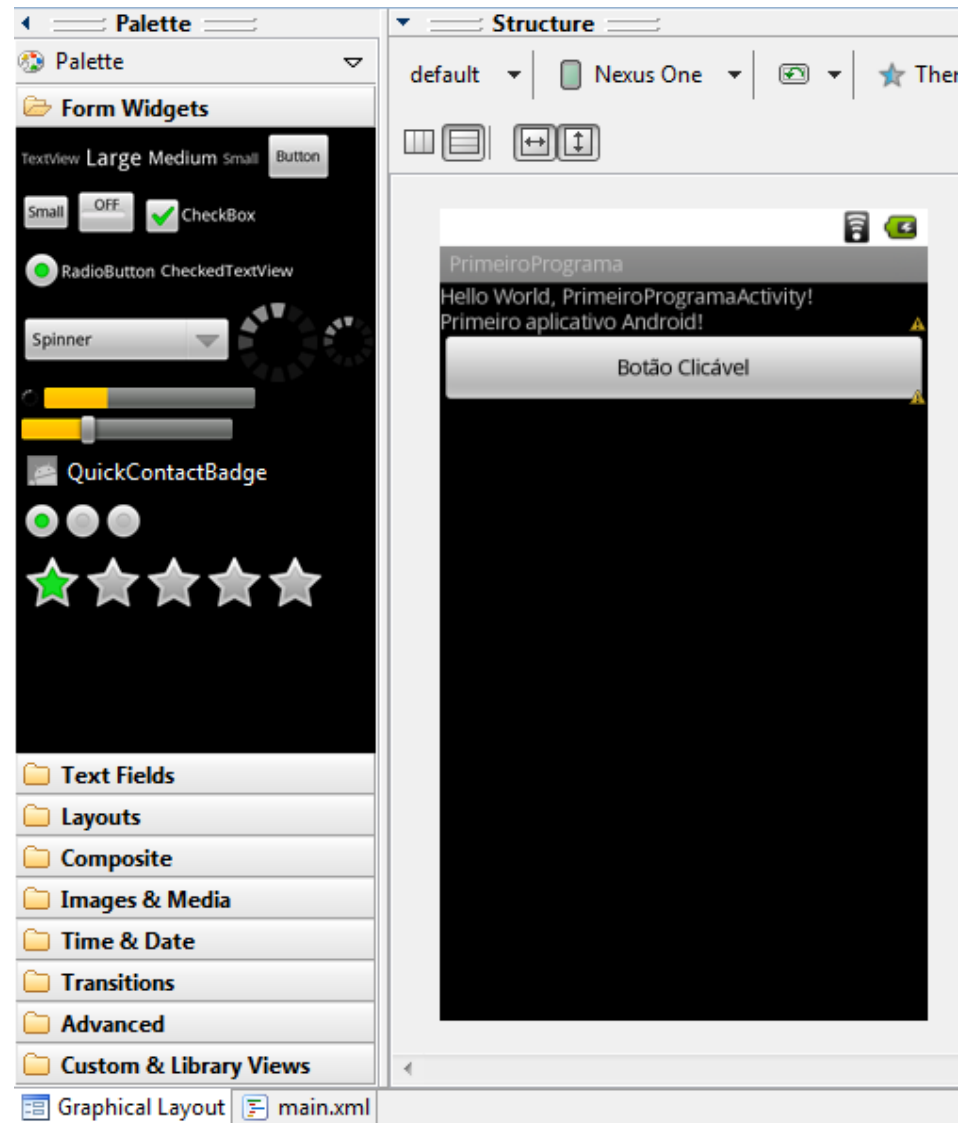
- AVD é uma instancia de simulador
 - Modela um dispositivo real
 - No número de AVDs é limitado à necessidade do programador
 - Importante para verificar o comportamento do programa em ambientes diferentes
- Criação no Eclipse:
 - Window->AVD Manager, clique em New ...
 - Forneça um nome, escolha uma versão do Android (target)
 - Cada versão possui um nível, cada nível duas plataformas: uma com e outra sem a API Google Maps

Aplicações Para Android

- Cada projeto se torna um aplicação
- Toda aplicação é formada por pelo menos uma Atividade (extends Activity)
- Cada tela é uma **Atividade** associada a um **arquivo xml** contendo a definição da interface gráfica, dentro do pacote res/layout
- Arquivo **AndroidManifest.xml**: possui a declaração das activities do projeto
- Arquivo **R.java**: faz a ligação do conteúdo XML com o Java. **IMPORTANTE**: nunca altere esse arquivo

Interface Gráfica

- Criada em um arquivo XML
- Utilização de mouse para criação das telas
- Componentes já prontos



Interface Gráfica

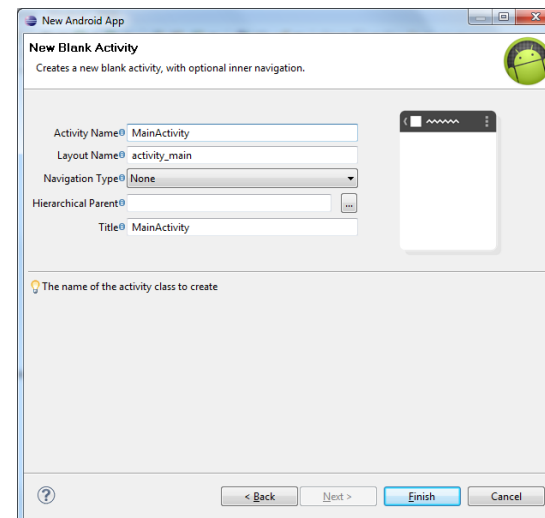
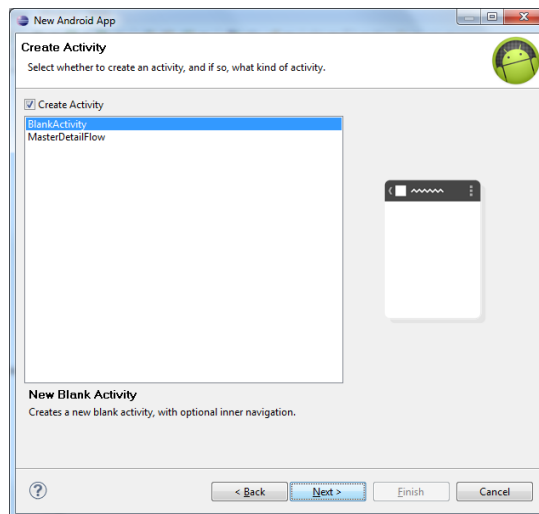
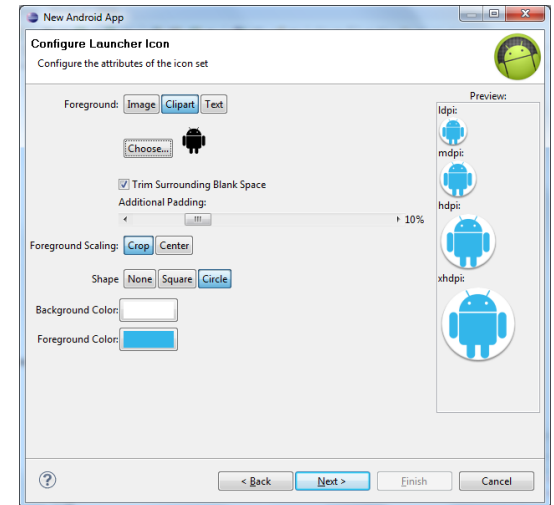
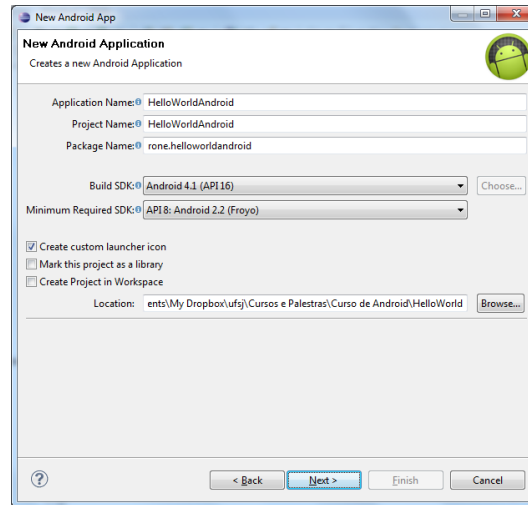
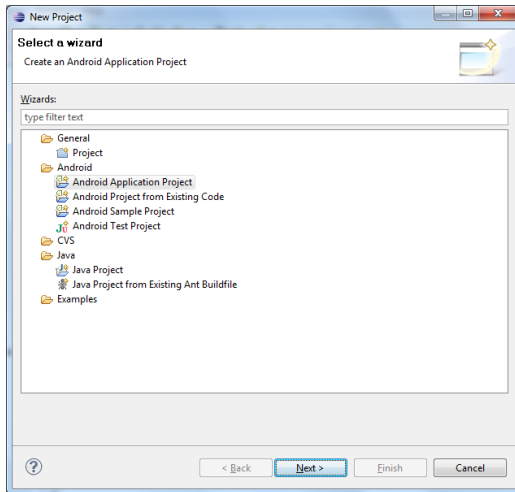
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello" />
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Primeiro aplicativo Android!"/>
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Botão Clicável"/>
</LinearLayout>
```

Arquivo XML criado

Primeiro Programa no Eclipse

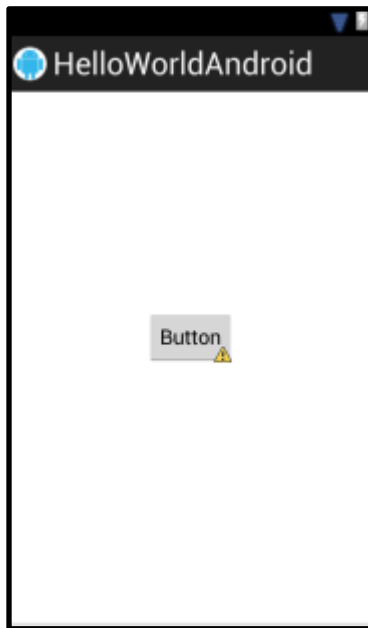
- File->New->Project->Android->Android Project e clique em Next
- Forneça:
 - um nome para a aplicação (Ex: HelloWorld)
 - o nome do Projeto
 - o nome do pacote
 - Escolha a plataforma → Clique em Next.
- Importante:
 - o nome do pacote precisa de pelo menos um ponto, ex: rone.primeiroprograma → Clique next
 - a plataforma define o tipo de dispositivo emulado
- Configure o ícone da aplicação (opcional) → Clique next
- Crie uma atividade (BlackActivity) → Finish

Primeiro Programa



Primeiro Programa no Eclipse

- Clique duas vezes sobre o arquivo `res/layout/activity_main.xml`
- Clique na aba `activity_main.xml`



Clique com o botão direito sobre o botão

```
<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" >
```

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="190dp"
    android:text="Button" />
```

```
</RelativeLayout>
```

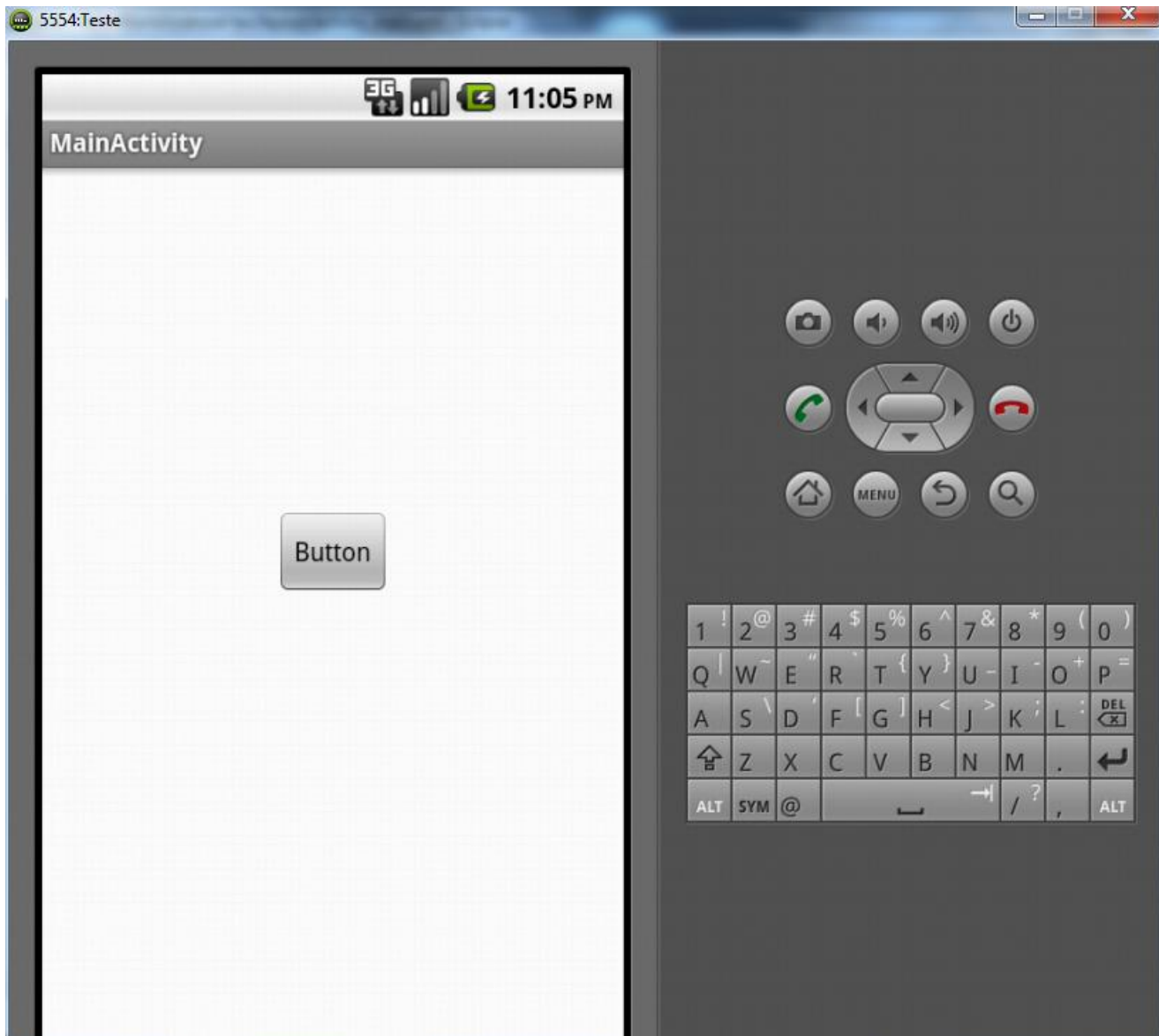
Primeiro Programa no Eclipse

- Clique duas vezes sobre o arquivo
src\rone.helloworld\MainActivity.java

```
public class MainActivity extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
    public boolean onCreateOptionsMenu(Menu menu) {  
        getMenuInflater().inflate(R.menu.activity_main, menu);  
        return true;  
    }  
}
```

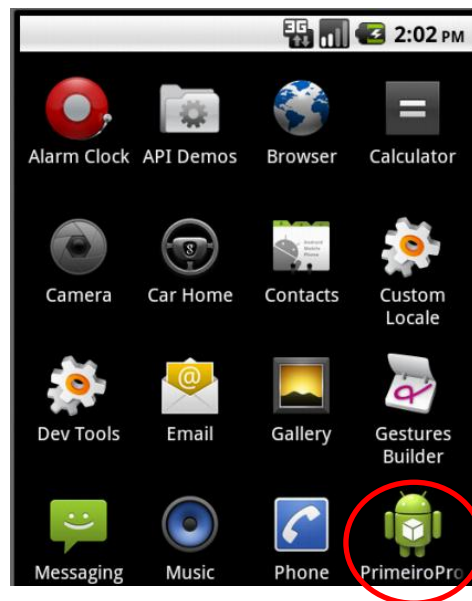
Primeiro Programa no Eclipse

- Salve
- Para executar, selecione o projeto no Eclipse e pressione F11
- Aparecerá o emulador, o qual iniciará como um dispositivo tradicional
- Sua aplicação será executada automaticamente. Ela será uma das aplicações instaladas no dispositivo



Primeiro Programa no Eclipse

- Clique no botão Home 
- Clique no Launcher (parte inferior da tela)
- Aparecerão todos os aplicativos instalados, inclusive o PrimeiroPrograma



Explicando o Projeto

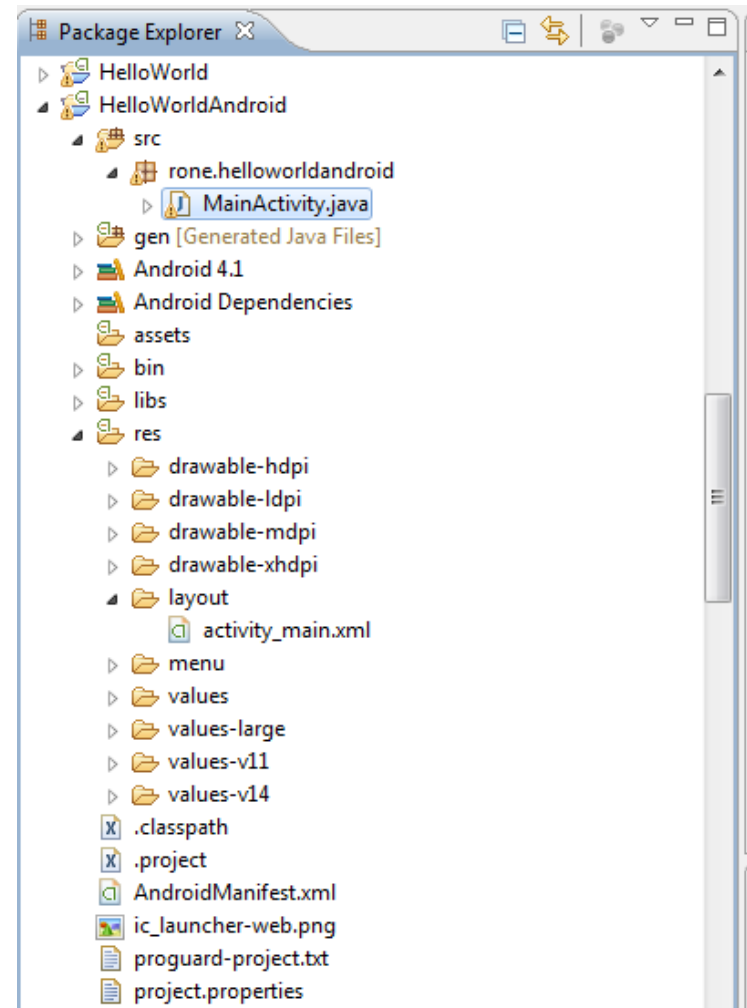
- O programador precisa fornecer para todos os projetos:
 - Project name: nome do projeto
 - Application name: nome do aplicativo
 - Package name: nome do pacote (com pelo menos um ponto)
 - Create Activity: nome da primeira atividade em seu projeto
 - Mini SDK version: versão mínima para execução

Explicando o Projeto

- Cada Activity é uma janela (interface gráfica com o usuário)
 - O aplicativo tem a MainActivity.java, a classe MainActivity que é filha de Activity
 - O arquivo activity_main.xml contém a descrição interface gráfica dessa Activity

Anatomia de um Aplicativo Android

- Vários arquivos compõem um projeto
- Vários são criados automaticamente e não devem ser alterados por programadores sem experiência em Android



Anatomia de um Aplicativo Android

- Pasta src: arquivos fonte (.java). No exemplo, temos MainActivity.java.
- Pasta gen: contém R.java, que gerencia os recursos de seu projeto (não deve ser alterado)
- Biblioteca Android 4.1: bibliotecas necessárias
- Pasta bin: resultado da compilação
- Pasta res: recursos utilizados pela aplicação
- AndroidManifest.xml: especifica as permissões e as funcionalidades da aplicação
- Activity_main.xml: interface gráfica

Anatomia de um Aplicativo Android

- Arquivo R.java
 - Faz a interface entre o código java e os arquivos XML
 - Criado e alterado automaticamente
 - Se apagado, outro é criado
 - Não deve ser alterado manualmente