HCI Summer Workshop

Android #1

hcim

Tiago Guerreiro

tjvg@di.fc.ul.pt

1 Introdução

Equipa

Luis Carriço
Tiago Guerreiro
Luis Duarte
Diogo Marques
Gonçalo Gomes
Eduardo Matos
Nikolay Stanchenko
Rafael Nunes
Filipe Fernandes
Ana Almeida

Temas

Android (3 e 4 de Julho)

- Basic Concepts
- Graphical User Interfaces
- Sensors
- External APIs

Kinect (5 de Julho)

- Introduction to C#
- Basic Concepts
- 3D User Interfaces
- Multimodal Interfaces (Speech, Gestures)

Arduino (8 de Julho)

- Electronics 101
- Sensors and actuators

Integration (9 de Julho)

- Communication Arduino – Mobile Device
- Communication Arduino – PC
- Middleware

Projectos

Lista de mini-projectos: 9 de Julho

Tutoria por um membro da equipa

Tópicos avançados

Investigação Aplicada

Possível candidatura conjunta a bolsa da Fundação Amadeu Dias

Logística

Android (~2 alunos por telemóvel)

Kinect (~5 alunos por Kinect)

Arduino (~5 alunos por kit)

1+ PC por grupo

2 Android 101

Pré-requisitos

Android Bundle

Eclipse and ADT plugin OK

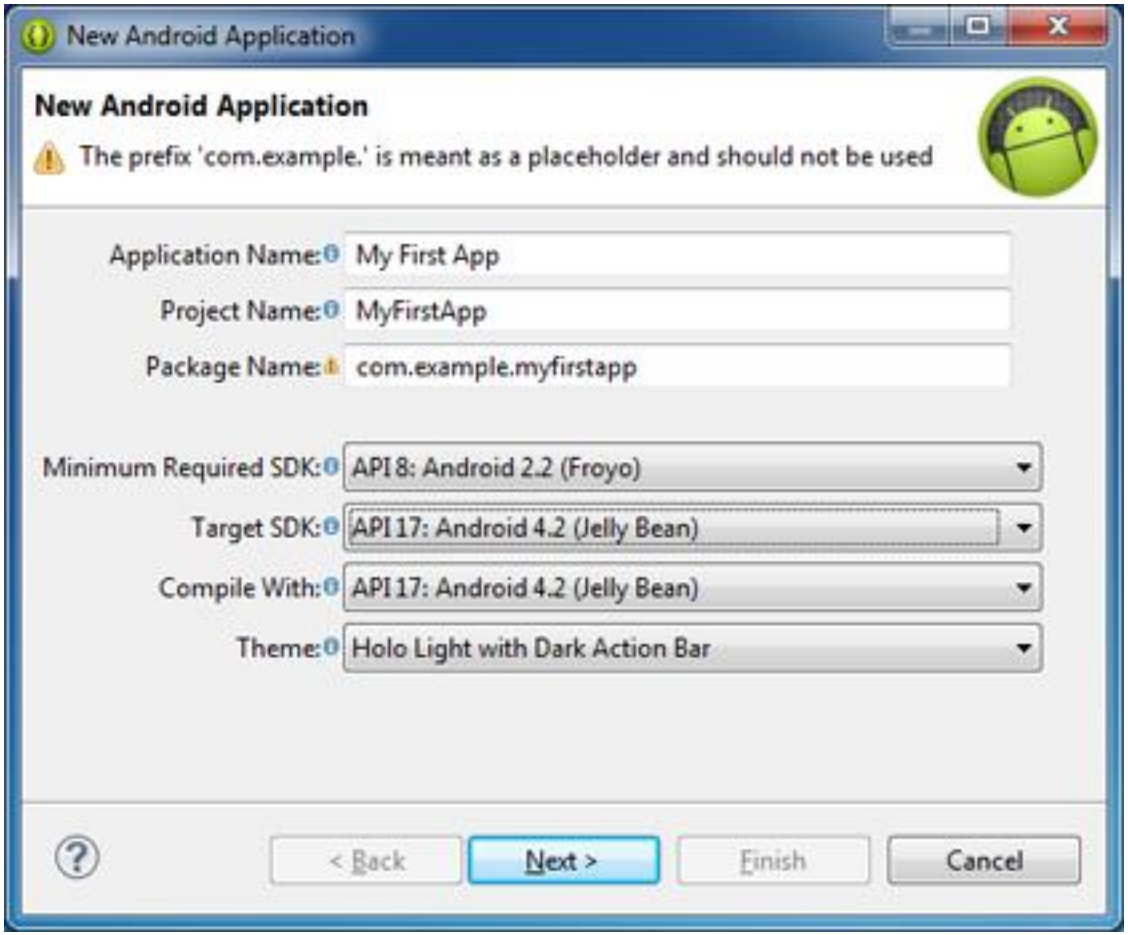
AVD

Device Driver

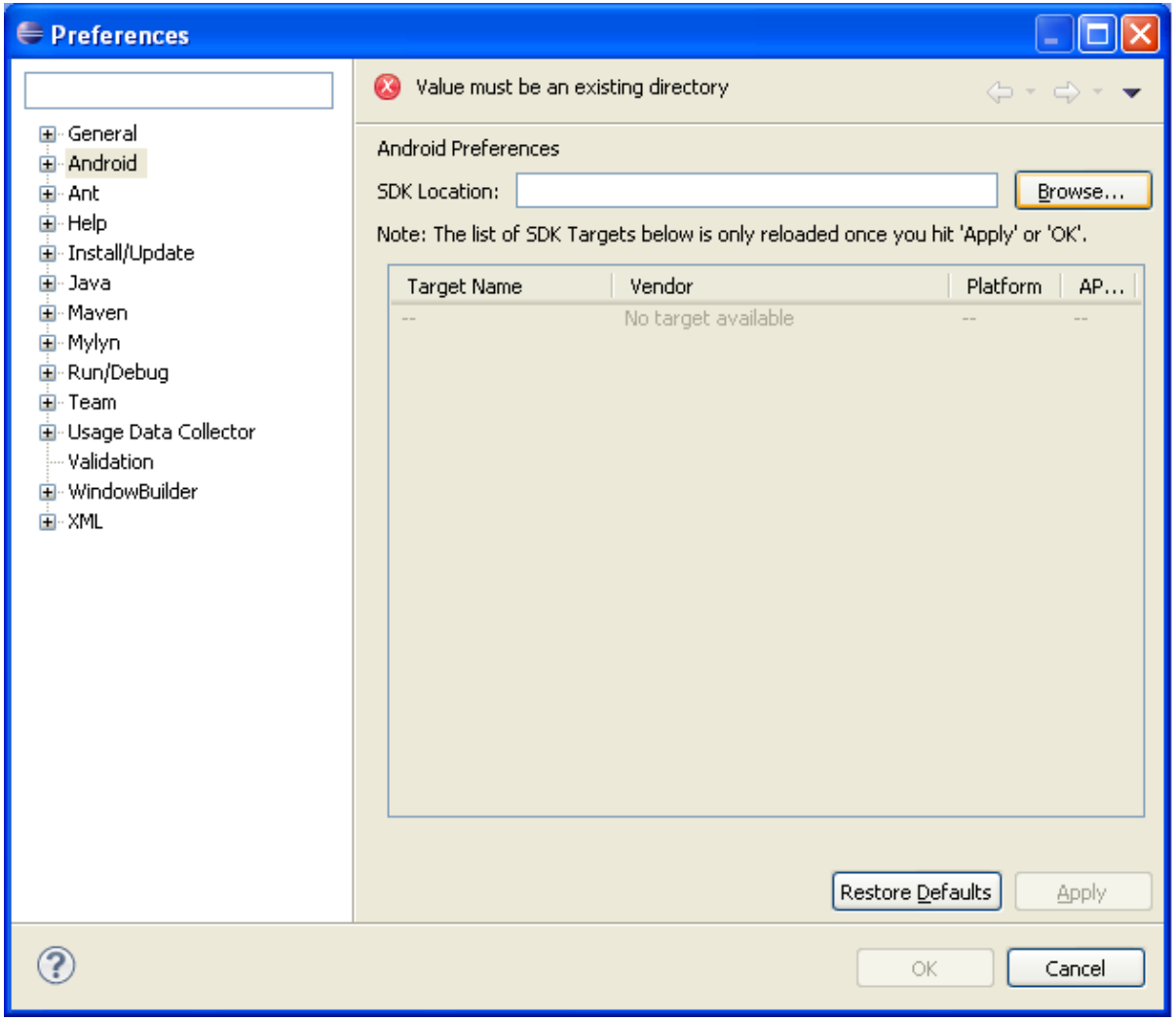
Vamos verificar….

Criar um novo projecto

New -> Android -> Android Application



Não encontra o SDK?



Emulador

AVD – Android Virtual Device

Abrir o AVD Manager

New

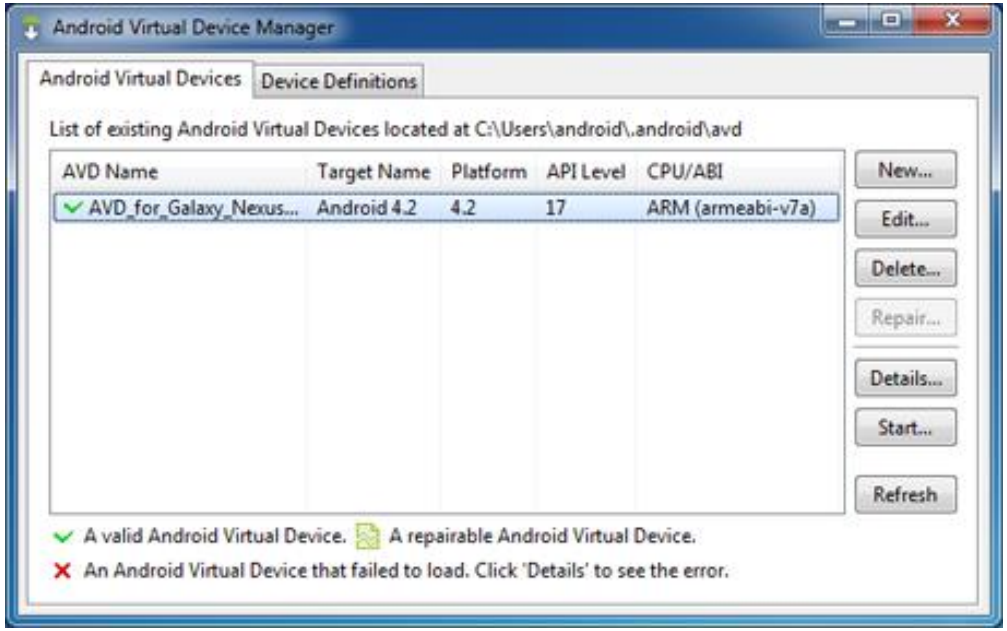
Nome, target, ….

Create AVD

Start

Desbloquear….

… depois de carregar



Correr app no emulador

Run

Se aparecer Run As…

… escolher Android Application

O Eclipse corre a app no emulador

Correr app no dispositivo

Ligar o cabo USB do dispositivo e PC

Ligar USB Debugging no dispositivo

Settings -> Developer Options

Run…

Android Basics

OS Open source

- Kernel Linux
- Optimizado para ambiente com poucos recursos

Geralmente apps desenvolvidas em Java

Apps correm numa máquina virtual Dalvik

- *Não é uma JVM, mas funciona de forma semelhante na perspectiva do programador*
- *Uma app por DVM*
- *Cada DVM corre como um utilizador Linux separado*

Componentes

Activity

Geralmente um único ecrã

Intent

Declarar intenções. Ex: passar de um ecrã para outro

Broadcast Receivers

Reacções a eventos externos (ex: bateria baixa, receber uma chamada)

Services

App corre em background (ex: music player)

Content Providers

Partilhar dados com outras apps

De volta ao projecto

src: pasta do código fonte, que contém um pacote com o código fonte gerado

gen: gerado automaticamente

res: contém todos os recursos que não são código (ex: imagens, strings e os ficheiros de layout). Está organizada em três pastas:

drawables: imagens e outros componentes gráficos

layout: definições de layout dos componentes da aplicação (XML)

values: XMLs respeitantes a constantes, como strings ou cores

Manifest file: caracterização, necessidades e capacidades da aplicação

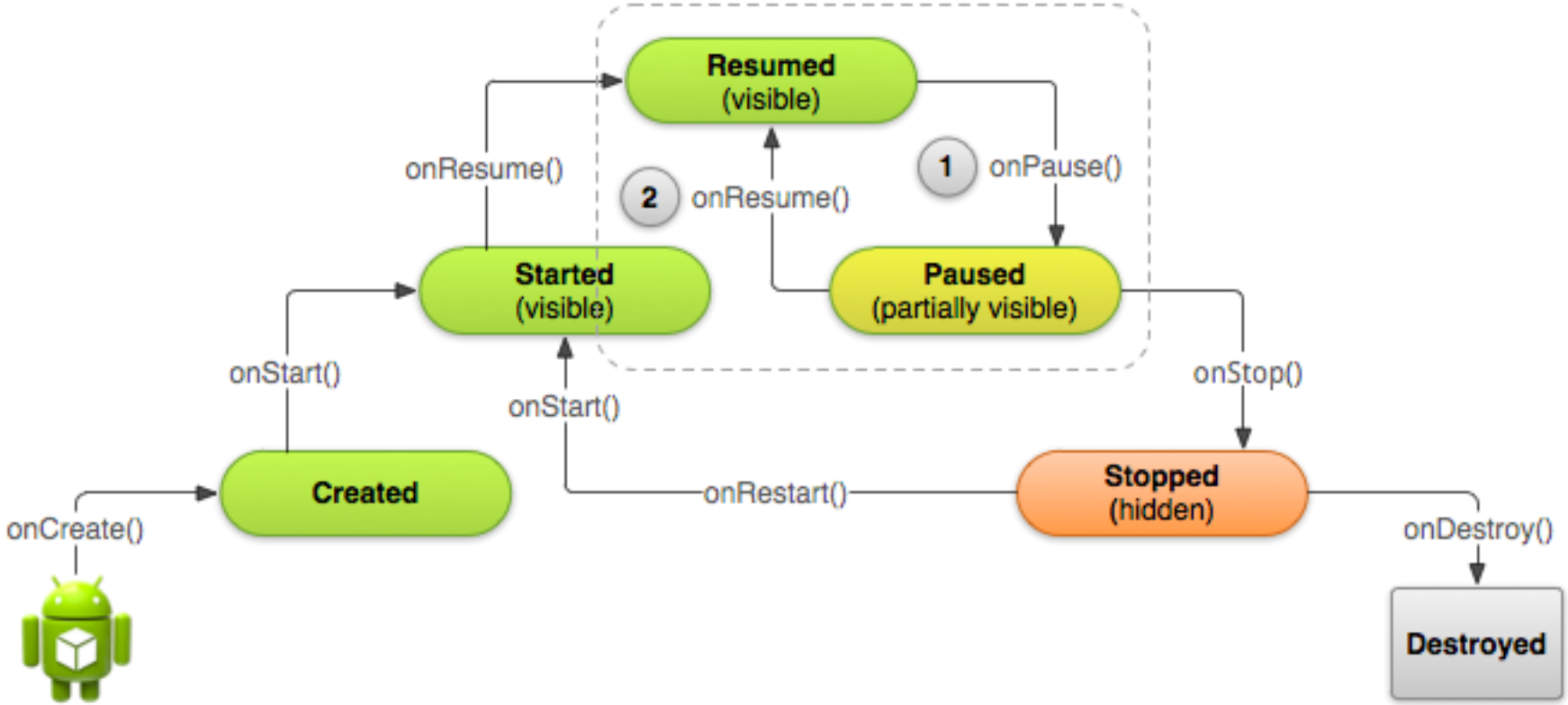
Actividades

Ficheiro *.java

```
package com.android.hello;
import android.app.Activity;
 import android.os.Bundle;
public class HelloAndroid extends Activity {

        /** Called when the activity is first created. */
        @Override public void onCreate(Bundle savedInstanceState)
        {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
        }
}
```

Actividades: Ciclo de vida



Actividades

Ficheiro *.java

```
package com.android.hello;
import android.app.Activity;
 import android.os.Bundle;
public class HelloAndroid extends Activity {

        /** Called when the activity is first created. */
        @Override public void onCreate(Bundle savedInstanceState)
        {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);
        }
}
```

Actividades: Layout

Layout separado da lógica da aplicação

main.xml

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

Strings.xml

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

Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string name="hello">Hello World, HelloAndroid!</string>
   <string name="app_name">Hello, Android</string>
</resources>
```

Actividades: Layout

Declarado em XML

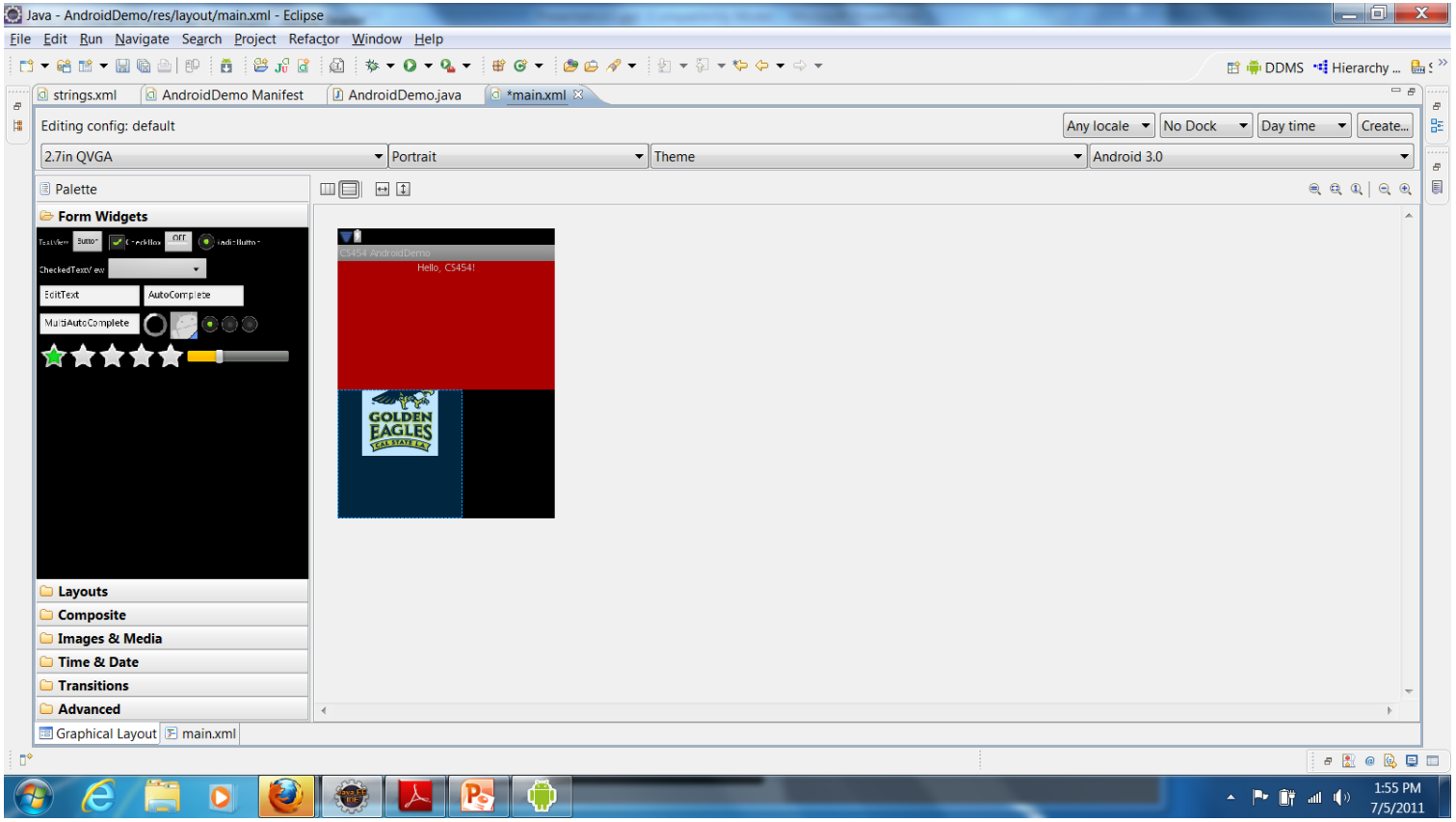
Declarado no código

```
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

// Activity is a subclass of context, so the TextView takes this as a parameter
        TextView tv = new TextView(this);
        tv.setText("Hello, CS454");
        setContentView(tv);
    }
```

GUI Builder

GUI Builder


Java - AndroidDemo/res/layout/main.xml - Eclipse
File Edit Run Navigate Search Project Refactor Window Help
DDMS
Hierarchy ...
strings.xml
AndroidDemo Manifest
AndroidDemo.java
*main.xml
Editing config: default
Any locale
No Dock
Day time
Create...
2.7in QVGA
Portrait
Theme
Android 3.0
Palette
Form Widgets
Button
CheckBox
RadioButton
CheckedTextView
EditText
AutoComplete
MultiAutoComplete
Layouts
Composite
Images & Media
Time & Date
Transitions
Advanced
Graphical Layout
main.xml
Hello, CS454!
GOLDEN EAGLES
1:55 PM
7/5/2011

Actividades: Tipos de Layout

Linear Layout

Frame Layout

Table Layout

Absolute Layout

Relative Layout

Alterar a actividade

Na aba Layout, arrastar um botão para a janela

Mude o valor do atributo Text para “Mudar Texto”

main.xml

acrescente o atributo **android:id="@+id/Text01”** ao elemento TextView.

Vai permitir referenciar a TextView a partir do código java

Eventos UI

Eventos da View

onClick() - é chamado quando o utilizador toca num objecto ou quando navega para ele através de teclas ou trackball e prime a tecla "enter".

onLongClick() - é chamado quando o utilizador fica a carregar sobre um objecto ou navega até ele e fica a carregar na tecla "enter" (durante um segundo).

onFocusChange() - é chamado quando o utilizador navega para ou de um objecto através das teclas de navegação ou da trackball.

onKey() - é chamado quando o objecto está focado e o utilizador prime ou liberta uma tecla do dispositivo.

onTouch() - é chamado quando o utilizador realiza uma acção que seja entendida como um toque, o que inclui premir, largar ou movimentos no ecrã.

onCreateContextMenu() - é chamado quando o menu de contexto está a ser construído.

Eventos UI

Definir um event listener

```
private OnClickListener mBotaoListener = new OnClickListener() {
                @Override
                public void onClick(View v) {
                        mText.setText("Novo texto");
                }
        };
```

Associar o listener ao botão (no OnCreate)

```
Button mBotao = (Button) findViewById(R.id.Button01);
mBotao.setOnClickListener(mBotaoListener);
```

Acrescentar ao código do listener

```
TextView mText;,
mText = (TextView) findViewById(R.id.Text01);
```

Run the app

Tirem dúvidas

Inspeccionem o código

Alterem a actividade para incluir uma caixa de texto (EditText) e que altera a TextView com o texto escrito nesta quando se carrega no botão.

Controlos UI

Botões - classe: android.widget.Button

Menu - classe: android.widget.Menu

Campo para edição de texto - classe: android.widget.EditText

Campo para apresentação de texto: classe: android.widget.TextView

RadioButton - classe: android.widget.RadioButton. Os RadioButton devem estar agrupados num grupo. Para isso utiliza-se a classe android.widget.Group

CheckBox - classe: android.widget.CheckBox

ToggleButton - classe: android.widget.ToggleButton (é um controlo com funcionalidade semelhante à CheckBox, mas tem a apresentação de um botão)

Spinner - classe: android.widget.Spinner

ProgressBar - classe: android.widget.ProgressBar (só permite apresentação)

SeekBar - classe: android.widget.SeekBar (também permite interacção)

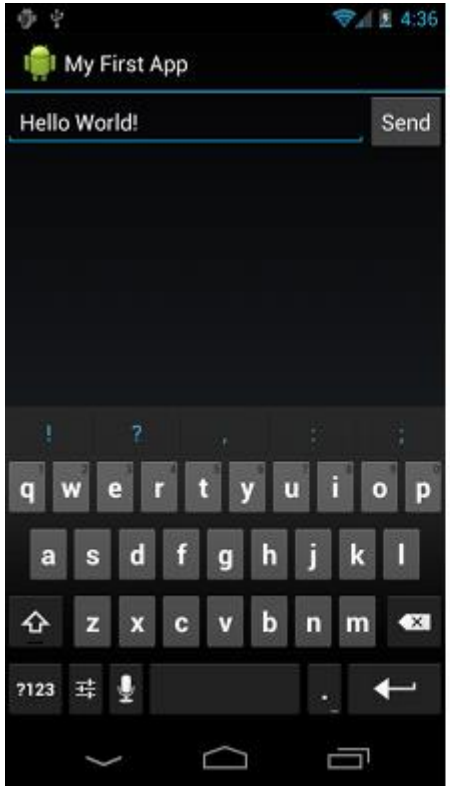
Adicionar uma actividade

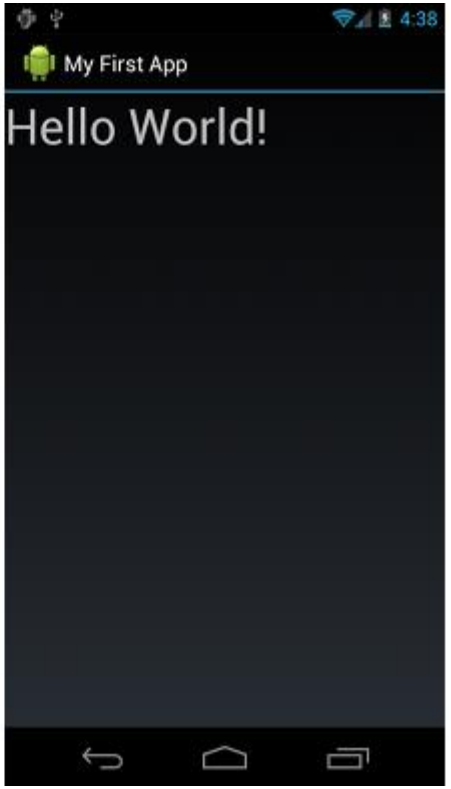
New->Android Activity

Add to Manifest

```
<application
android:icon="@drawable/icon“
android:label="@string/app_name" >
        <activity android:name=".ActivityOne" android:label="@st
        ……
        </activity>
        <activity
                android:name="ActivityTwo“
                android:label="ActivityTwo" >
        </activity>
</application>
```

Objectivo


4:36
My First App
Hello World!
Send

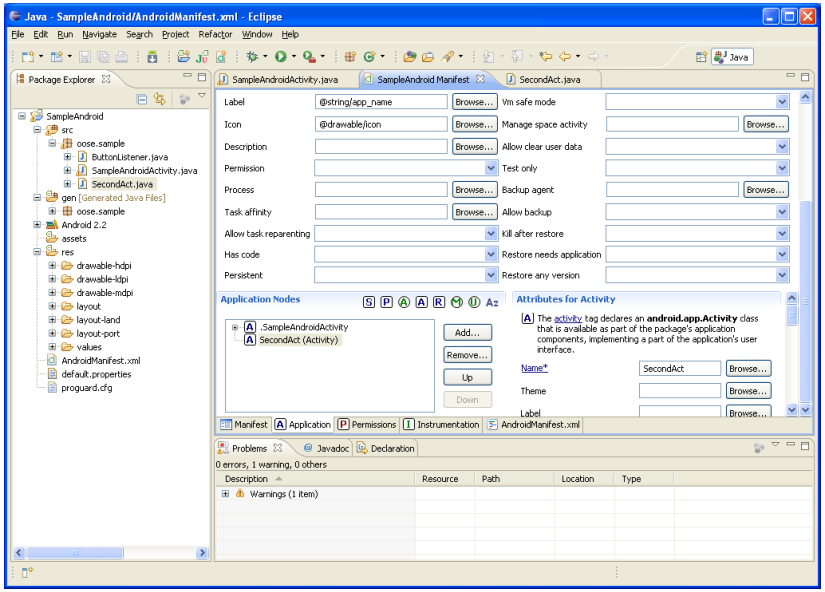

4:38
My First App
Hello World!

Passos

Editar ficheiro de layout

- Editar TextView
- Criar um id para a mesma

Adicionar nova actividade ao Manifest



Intents

Mensagens assíncronas

Permitem aos componentes Android pedir funcionalidade a outros componentes

```
# Start a new activity

Intent i = new Intent(this, ActivityTwo.class);
startActivity(i);
```

Intents: Dados

Intents podem ter dados

Para serem usados pelo componente *receptor*

```
String url = "http://www.google.com";
Intent i = new Intent(Intent.ACTION_VIEW);
i.setData(Uri.parse(url));
startActivity(i);
```

Intents types

Explicit Intents

```
Intent i = new Intent(this, ActivityTwo.class);
```

Implicit Intents

```
Intent i = new Intent(Intent.ACTION_VIEW,
                Uri.parse("http://www.vogella.com"));
```

Intents: Adding data

```
Intent i = new Intent(this, ActivityTwo.class);
i.putExtra("Value1", "This value one for ActivityTwo ");
i.putExtra("Value2", "This value two ActivityTwo");
startActivity(i);
```

Intents: Receiving Data

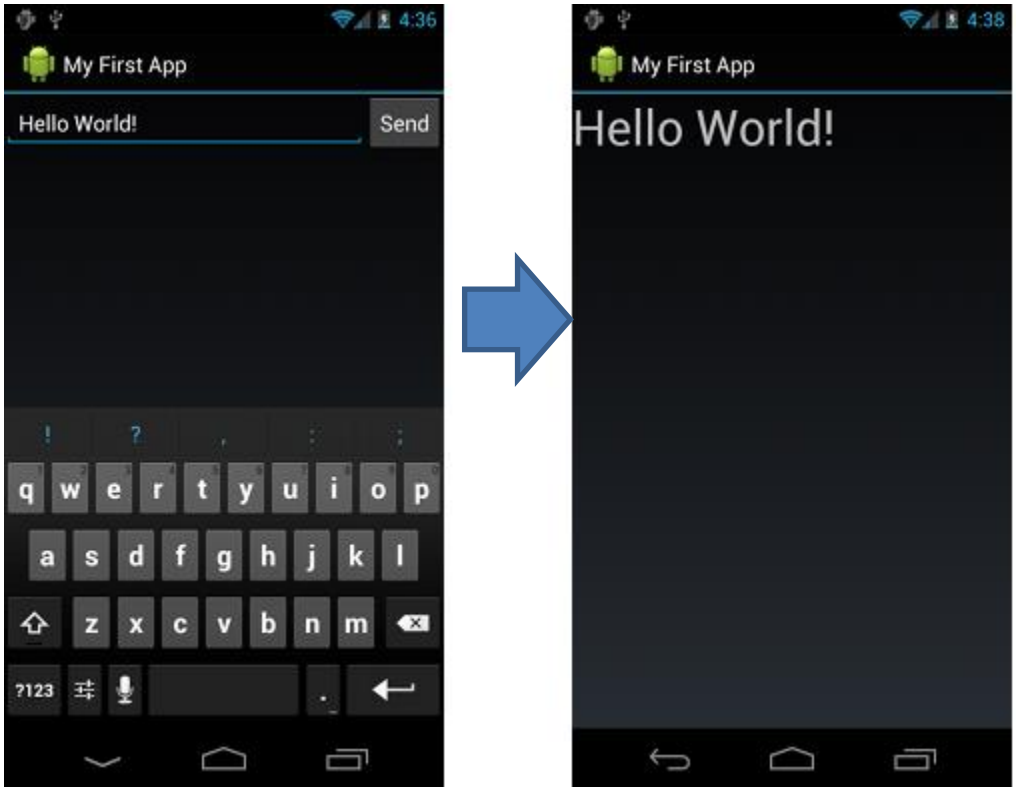
On ActivityOne:

```
Intent i = new Intent(this, ActivityTwo.class);
i.putExtra("Value1", "This value one for ActivityTwo ");
startActivity(i);
```

On ActivityTwo (onCreate):

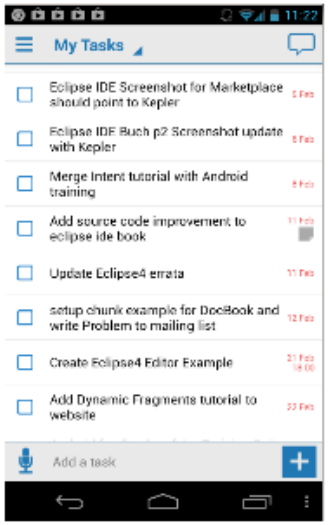
```
Bundle extras = getIntent().getExtras();
String value1 = extras.getString(“Value1”);
if (value1 != null)
{
// Do something with the data
}
```

Do it!


4:36
My First App
Hello World!
Send
4:38
My First App
Hello World!

Intents: return

Intent + resultCode provided by called activity



onActivityResult(requestCode, resultCode, intent)

RequestCode provided by Android to identify which activity type was started

Intents: startActivityfor Result

On ActivityOne

```
public void onClick(View view) {
        Intent i = new Intent(this, ActivityTwo.class);
        i.putExtra("Value1", "This value one for ActivityTwo ");
        i.putExtra("Value2", "This value two ActivityTwo");
        // Set the request code to any code you like, you can
        // identify the callback via this code
        startActivityForResult(i, REQUEST_CODE);
}
```

Intents: Responding

On ActivityTwo

```
@Override
public void finish()
{
    // Prepare data intent
     Intent data = new Intent();
     data.putExtra("returnKey1", "Swinging on a star. ");
     data.putExtra("returnKey2", "You could be better. ");
     // Activity finished ok, return the data
     setResult(RESULT_OK, data);
     super.finish();
}
```

Intents: Receiving Data

On ActivityOne

```
@Override
protected void onActivityResult(int requestCode, int resultCode,  Intent data)
{
if (resultCode == RESULT_OK && requestCode == REQUEST_CODE) {
            if (data.hasExtra("returnKey1")) {
                        Toast.makeText(this, data.getExtras().getString("returnKey1"),
                                                Toast.LENGTH_SHORT).show();
            }
}
}
```

Hands-on

Construa uma aplicação com duas actividades semelhantes Em cada uma delas devem existir três RadioButtons e um Button. Na primeira actividade os RadioButtons devem permitir escolher uma de três cores: verde, azul, vermelho. Na segunda actividade devem permitir escolher uma de outras três cores: amarelo, magenta, e ciano. Ao seleccionar o Botão em cada das actividades deve mudar a cor de fundo da outra actividade e visualizá-la. Por exemplo, na primeira actividade selecciono verde e carrego no botão. Salto para a outra actividade que deve ter a cor de fundo verde.