

# Aula prática

# Android



Tomaz Rocha da Silva  
[trdasilva@inf.ufrgs.br](mailto:trdasilva@inf.ufrgs.br)

Baseado no material de  
Daniele Fernandes e Silva  
[daniele.fernandes@inf.ufrgs.br](mailto:daniele.fernandes@inf.ufrgs.br)

# Introdução



ANDROID

# Introdução

Platform Version	API Level	VERSION_CODE	Notes
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD	
Android 2.2.x	8	FROYO	
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

# Recursos

- ✧ Máquina Virtual Dalvik
- ✧ OpenGL ES API
- ✧ SQLite
- ✧ Mídias áudio e vídeo (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- ✧ Touchscreen
- ✧ Bluetooth, 3G, WiFi
- ✧ Câmera, GPS, bússola, acelerômetro, giroscópio.

# Aplicação

- ☞ Activities
- ☞ Layout
- ☞ Integração entre Activity e Layout
- ☞ Intents
- ☞ Services
- ☞ Broadcast Receiver
- ☞ AndroidManifest
- ☞ Estrutura do projeto

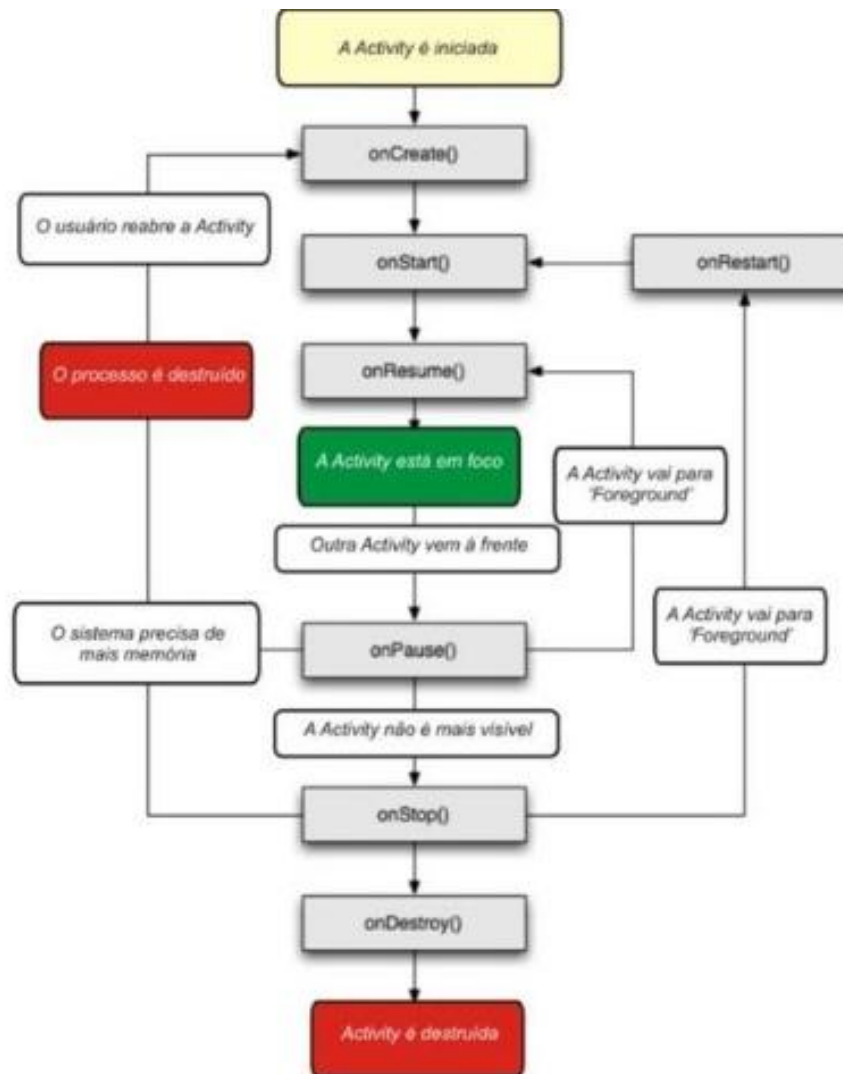
# Conceitos

## Activity

- Componente da aplicação que corresponde a uma tela
- Interação entre usuário e serviços/eventos
- Parte visível da aplicação
- Vinculado a um Layout

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // Inflate the menu; this adds items to the action bar if it is present.  
        getMenuInflater().inflate(R.menu.main, menu);  
        return true;  
    }  
}
```

# Conceitos



# Conceitos

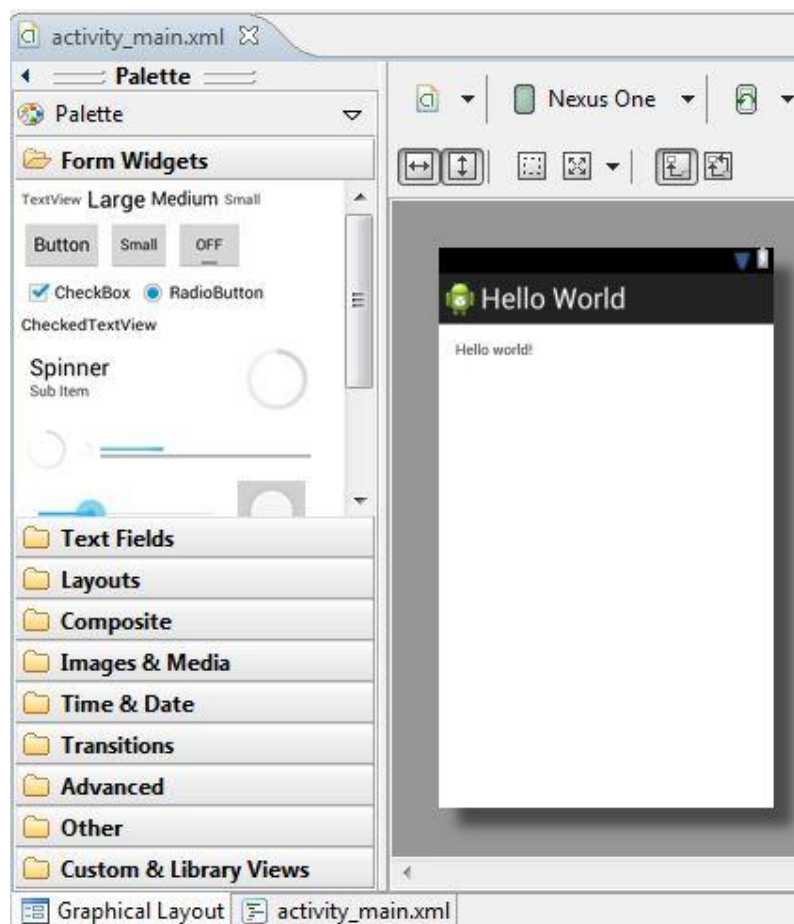
## Layout

- Arquivo xml nos quais são definidas as interfaces gráficas dos aplicativos
- Similar a criação de interfaces em java
- Se divide entre containers e widgets
- Containers: Elemento que agrupa outros elementos, containers ou widgets
- Widgets: São os elementos de interface propriamente dito, como botões, text view, etc.



# Conceitos

## Layout



```
<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=".MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

# Conceitos

## Integração entre Activity e Layout

- São vinculados no evento onCreate da Activity usando o método setContentView
- Para interagir com um elemento da interface utiliza-se a função findViewById, para obter uma referência para o elemento

```
<Button  
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/textView1"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="113dp"  
    android:text="Button" />
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    Button b1 = (Button) findViewById(R.id.button1);  
}
```

# Conceitos

## ∞ Integração entre Activity e Layout

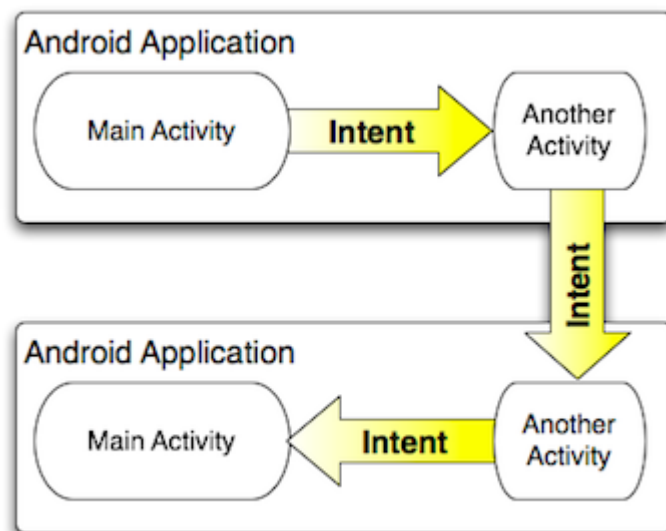
- Para capturarmos algum evento que ocorre em algum elemento da interface, como por exemplo um click, é necessário adicionarmos um callback neste elemento

```
b1 = (Button) findViewById(R.id.button1);
b1.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
    }
});
```

# Conceitos

## ☞ Intent

- Mensagens enviadas entre os blocos de instrução principais
- Mudar de activity ou executar um service



# Conceitos

## ☞ Intent

- Para ativar uma outra activity, criamos uma intent passando como parâmetro a classe da activity que queremos ativar e usamos o comando startActivity.

```
Intent t = new Intent(MainActivity.this, SecondActivity.class);  
startActivity(t);
```

- Para passarmos parâmetros para a outra activity utilizamos o comando putExtra

```
Intent t = new Intent(MainActivity.this, SecondActivity.class);  
t.putExtra("msg", ed1.getText().toString());
```

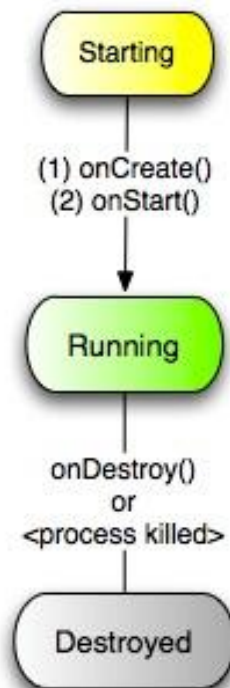
- Para obter a intent na activity que foi chamada, utilizamos o comando getIntent, para recuperar os valores passados usamos o comando getTipoDoValorExtra

```
Intent intent = getIntent();  
String text = intent.getStringExtra("msg");
```

# Conceitos

## Service

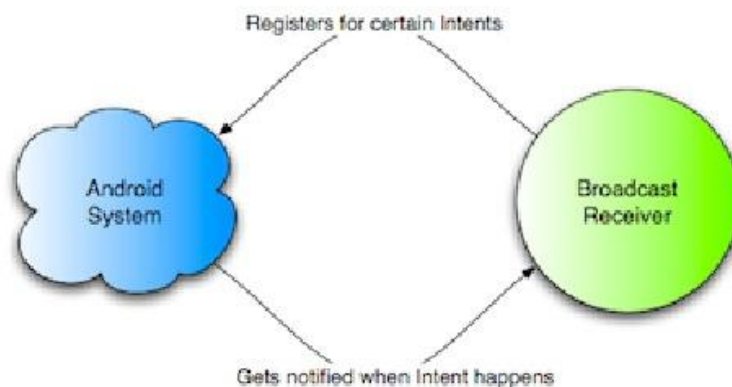
- Execução de recursos mesmo utilizando outras aplicações
- Não possui componentes de interface



# Conceitos

## ∞ Broadcast Receiver

- Trechos de código responsáveis pelo recebimento de mensagens do SO





# Conceitos

## Utilizando Recursos de Hardware

- Existem diversos hardwares e sensores que podemos utilizar no android e a forma de obtermos os dados destes sensores varia
- Para obtermos os dados dos sensores de movimento, é necessário somente fazer com que a nossa activity implemente a classe `SensorEventListener` e registrarmos os sensores que queremos obter os dados

```
public class SecondActivity extends Activity implements SensorEventListener{

    SensorManager sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

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

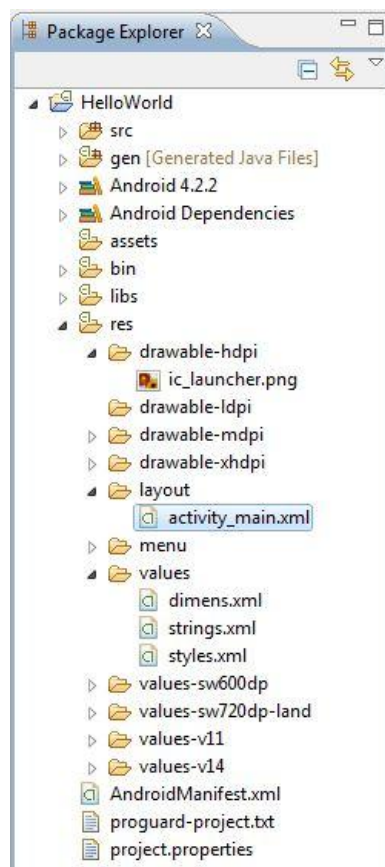
        sensorManager.registerListener(this, sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE),
            sensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        // TODO Auto-generated method stub
        float sensorX = event.values[0];
        float sensorY = event.values[1];
        float sensorZ = event.values[2];
    }
}
```



# Conceitos

## 🌀 Estrutura do projeto



# Conceitos

- ✎ *src* - é pasta onde fica o código fonte java da nossa aplicação.
- ✎ *gen* - pasta onde ficam arquivos gerados automaticamente pelo Eclipse.
- ✎ *assets* - pasta onde ficam arquivos de mídia que são usados na aplicação.
- ✎ *bin* - pasta onde ficam os arquivos compilados pelo Eclipse.

# Conceitos

- res - abreviação de resources, esta pasta é subdividida em várias pastas, com propósitos diferentes.
  - drawable:** são destinadas a armazenar imagens que são usadas na aplicação. Cada pasta corresponde a uma resolução diferente.
  - layout:** se destina a armazenar os arquivos XML que representam o layout das telas da aplicação
  - values:** também se destina a armazenar XMLs que serão usados na aplicação. Nesta pasta fica o arquivo strings.xml, que contém o nome do aplicativo.

# Conceitos

## AndroidManifest

- Arquivo Xml que descreve as características fundamentais do aplicativo
- Define compatibilidade com as diferentes versões do Android
- Declaração das activities e dos services

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.helloworld"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.helloworld.MainActivity"
            android:label="@string/app_name" >
```

# Android SDK

∞ Instalação e configuração: Tutorial no moodle

