

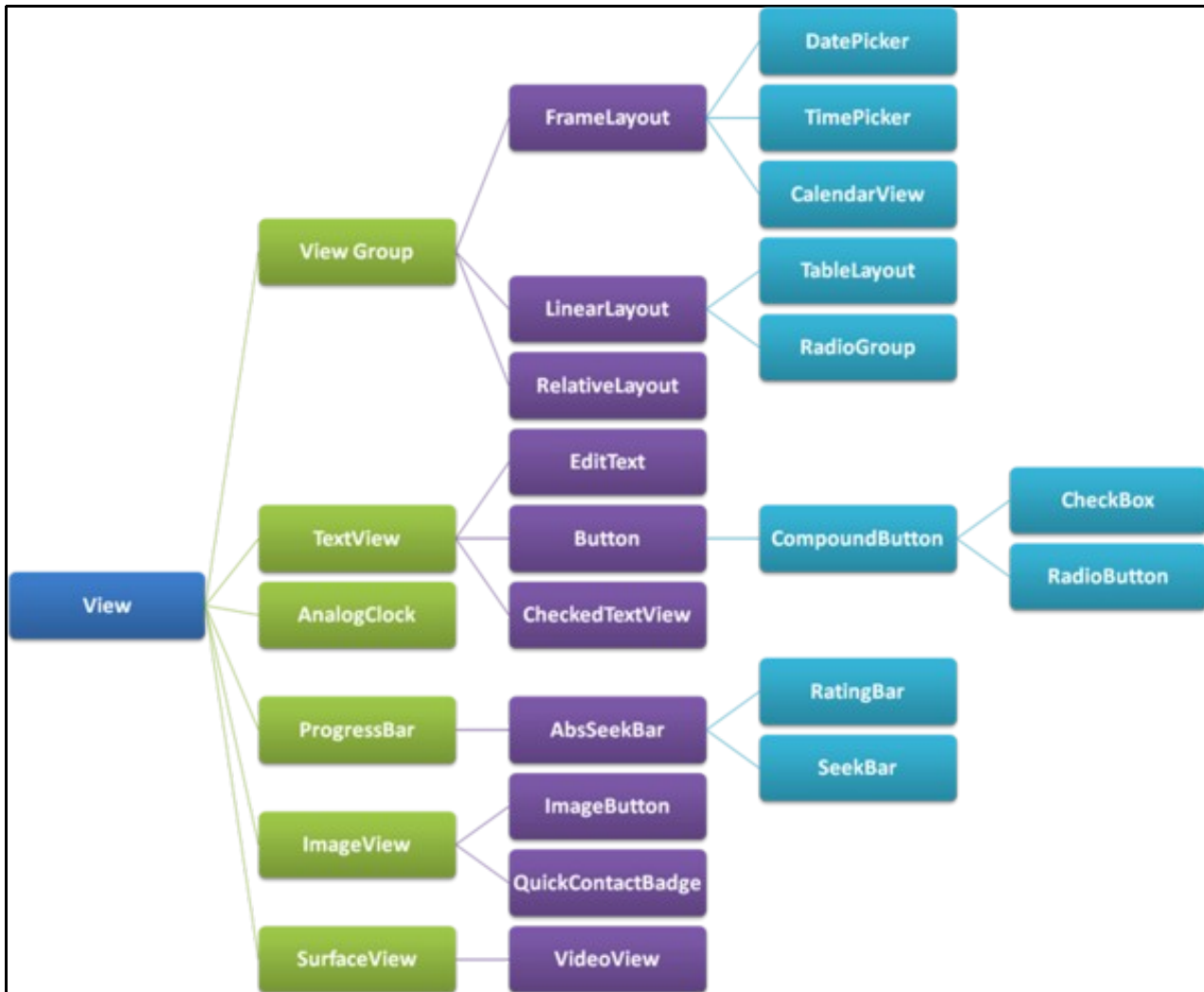
1 ImageView

1.1 Sumario

- 1 Introducción
- 2 Caso práctico
 - ♦ 2.1 Recursos de imaxes
 - ♦ 2.2 XML do layout
 - ♦ 2.3 Código Java

1.2 Introducción

- Un control **ImageView** mostra en pantalla unha imaxe calquera.
- Permite ser escalada (**android:scaleType**) e tinguida (**android:tint**)
- A clase **ImageView** herda directamente da clase **View**



- Imaxe obtida de: <http://www.itcsolutions.eu/2011/08/27/android-tutorial-4-procedural-vs-declarative-design-of-user-interfaces>

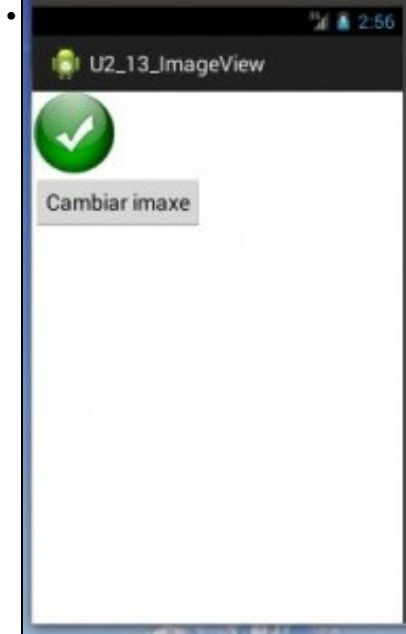
- **Referencias:**

- ♦ <http://developer.android.com/reference/android/widget/ImageView.html>

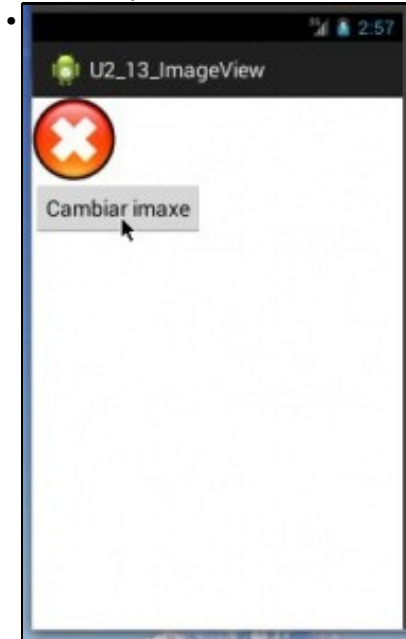
1.3 Caso práctico

- Comezamos creando o proxecto **U2_13_ImageView**
- Creamos unha aplicación que cargue por defecto unha imaxe de "OK" e un botón.
- Cando prememos no botón ou na imaxe cambia a imaxe por outra de "NON-OK" e así continuamente.

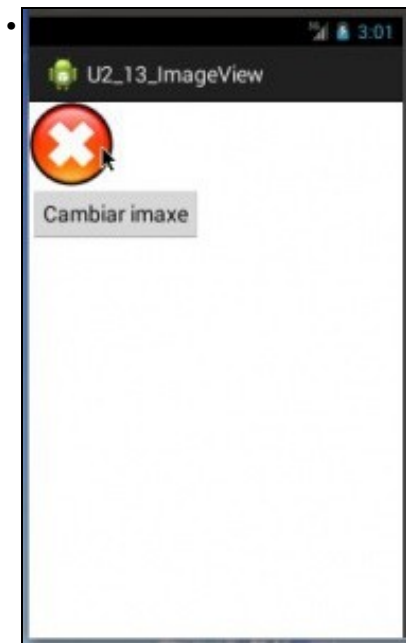
- **ImageView**



Iniciada a aplicación



Premer no botón cambia imaxe.



Premer na imaxe tamén a cambia.

1.3.1 Recursos de imaxes

- Crear o cartafol **/res/values/drawable**

- Recursos de imaxes



Descargar o boto ok.png



Descargar o boto no.png



Arrastralos ao cartafol anterior dentro do IDE. Lembrar que os nomes deben estar en minúsculas.

1.3.2 XML do layout

- Observar como se referencia á imaxe que vai cargar o control (líña12).
- Os dous controles chaman ao mesmo método no caso de facer click neles.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/img_imaxe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="Imaxe"
        android:src="@drawable/ok"
        android:onClick="onCambiarImaxeClick"
        android:tag="OK" />

    <Button
```

```

        android:id="@+id/btn_cambiar_imaxe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onCambiarImaxeClick"
        android:text="Cambiar imaxe" />

</LinearLayout>

```

- **Liña 14:** Pónselle unha etiqueta á imaxe para poder controlar por código cal é a imaxe cargada.

1.3.3 Código Java

```

package com.example.u2_13_imageview;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.ImageView;

public class U2_13_ImageView extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u2_13__image_view);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.u2_13__image_view, menu);
        return true;
    }

    public void onCambiarImaxeClick(View view) {
        // ImageView img = (ImageView) view;

        ImageView img = (ImageView) findViewById(R.id.img_imaxe);

        if (img.getTag() == "OK") {
            img.setTag("NO");
            img.setImageResource(R.drawable.no);

        } else {
            img.setTag("OK");
            img.setImageResource(R.drawable.ok);
        }
    }
}

```

- **Liña 29:** comprobamos cal é o Tag da imaxe cargada.
- **Liña 30, 31:** se a imaxe é ok.png entón cargamos a de no.png e cambiamos a etiqueta. O mesmo se fai nas liñas 34 e 35.
- A través da clase Java R accedemos ás imaxes.
- **Liña 25:** se o método se chama so dende a imaxe e non dende o botón poderíamos facer uso do parámetro view do método e comentar a liña 27.