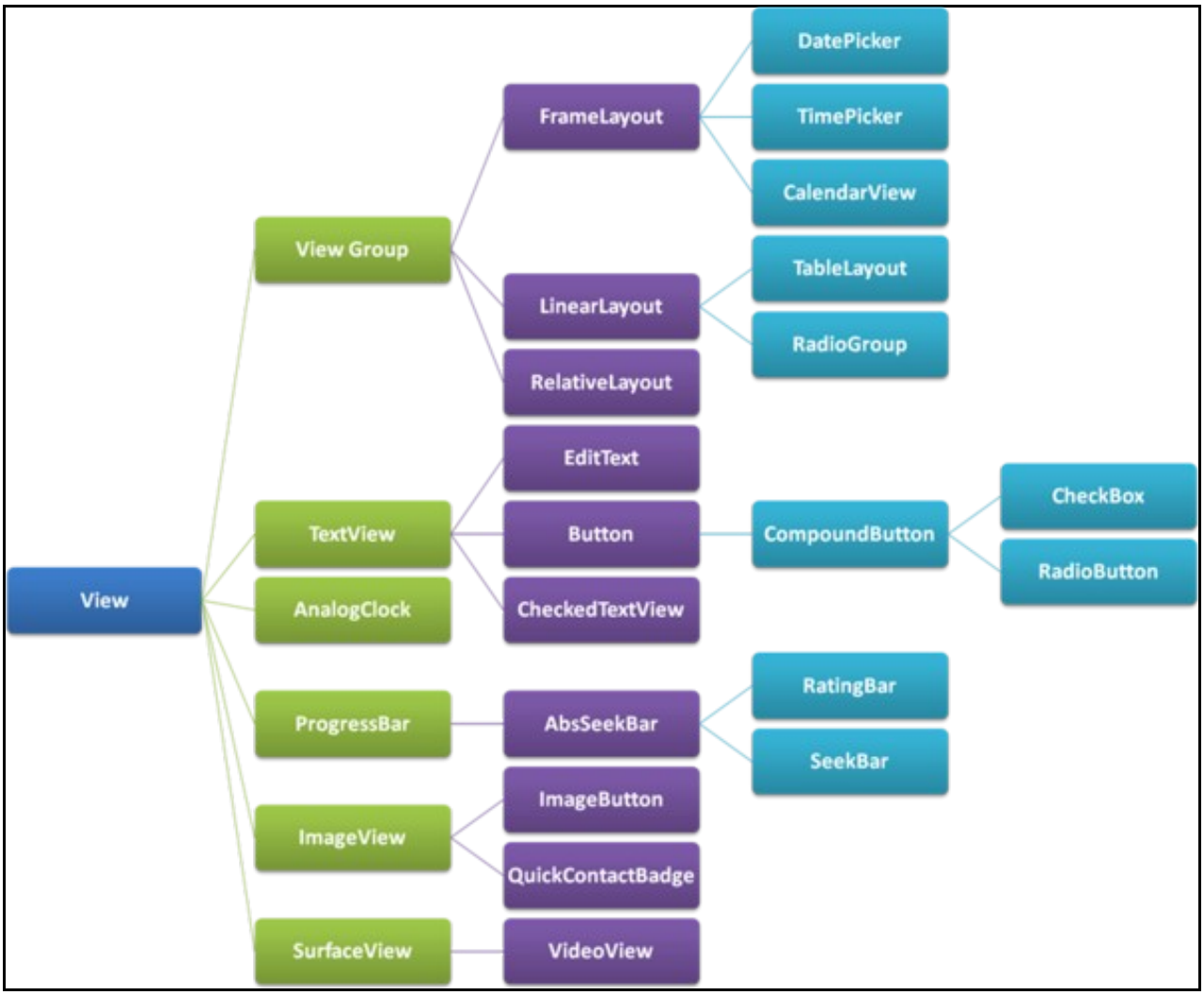
1 Button. ToggleButton. Control de eventos II

1.1 Sumario

1.2 Introdución

- Os **botóns (button)** permiten ao usuario indicar á aplicación que realice unha acción.
- Os botóns poden ter Texto, unha imaxe ou as dúas cousas:
- Ben o texto ou a imaxe comunican claramente ao usuario cal é a función do botón.

- Estes controles son subclases de:
  - Button de TextView
  - ToogleButton de CompoundButton
  - ImageButton de ImageView (que se verá proximamente)



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

- Observar como se definen os tres tipos de botóns anteriores:

- **Botón con texto**:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    ... />
```

- **Botón con imaxe**

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/button_icon"
    ... />
```

- **Botón con texto e imaxe**. A propiedade **android:drawableLeft** indica onde se sitúa a imaxe.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    android:drawableLeft="@drawable/button_icon"
    ... />
```

- Os botóns teñen unha propiedade **android:onClick="oMeuMetodo"** que nos permite chamar ao método indicado cando o pulsamos.
    - Ese método hai que declaralo en Java como **puclic void oMeuMetodo (View v) { ... }**
    - Recibe o obxecto View de quen o chamou.

- Os botóns (Button) son subclases de **TextView**.

- **Referencias**:
    - O Control botón: http://developer.android.com/reference/android/widget/Button.html
    - Introdución aos botóns: http://developer.android.com/design/building-blocks/buttons.html
    - Para programadores: http://developer.android.com/guide/topics/ui/controls/button.html
    - Para descargar iconas: http://www.iconarchive.com
    - Para descargar imaxes: http://www.openclipart.org

1.2.1 Botóns de 2 estados: ToggleButton/Switch

- Existe outro tipo de botón de 2 estados (ON/OFF).
- Un máis básico: **<ToggleButton>**

- E outro máis avanzado (versión Android 4.0 ou superior): **<Switch/>**

-
    - Permiten cambiar o seu estado desprazando co dun estado a outro. O funcionamento é semellante ao control ToogleButton.

- Un obxecto ToogleButton/Switch herda da clase **CompoundButton**, quen, á súa vez, herda da clase **Button**.
- Por tanto funcionan da mesma maneira, pero ademais o este tipo de botóns:
    - ten 2 estados (True/False), que pode podemos comprobar co método **isChecked ()**
    - Para cada estado podemos amosar un texto distinto no botón: **android:TextOn** e **android:TextOFF**.

- **Referencias**:
    - O control ToggleButton: http://developer.android.com/reference/android/widget/ToggleButton.html
    - O control switch: http://developer.android.com/reference/android/widget/Switch.html
    - Introdución as botóns de 2 estados: http://developer.android.com/guide/topics/ui/controls/togglebutton.html

1.3 Casos prácticos

- Como sempre creamos un novo proxecto:**U2_09_Buttons**

- Imos crear 3 botóns:

- 1 Botón con texto
- 1 ToogleButton
- 1 botón con imaxe

- Tres botóns
- 

Cando entremos na aplicación este será o seu estado.
- 

Se prememos o primeiro botón (Texto) amosarase un texto en pantalla.

- 

Se prememos o segundo botón (ToogleButton) amosarase outro texto en pantalla e o botón cambiará o seu estado e o texto do mesmo.
- 

Se o volvemos pulsar amosa outro texto e volve cambiar o estado.

- 

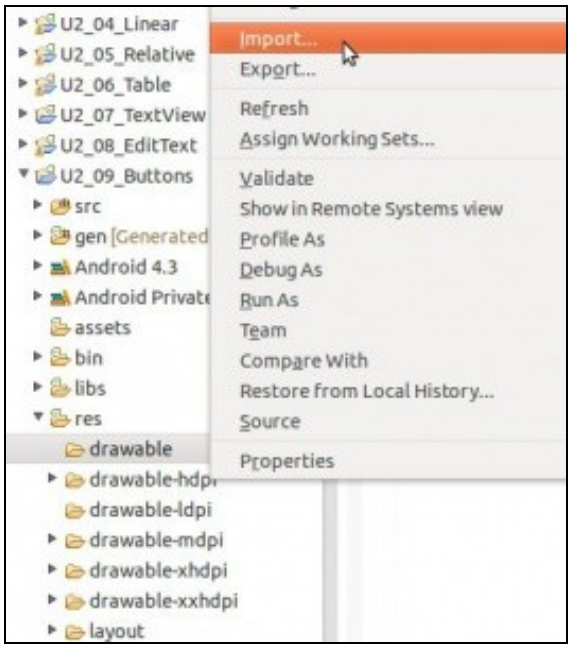
Se prememos o terceiro botón (Imaxe) amosarase de novo un texto diferente.

1.3.1 Importar unha imaxe como recurso

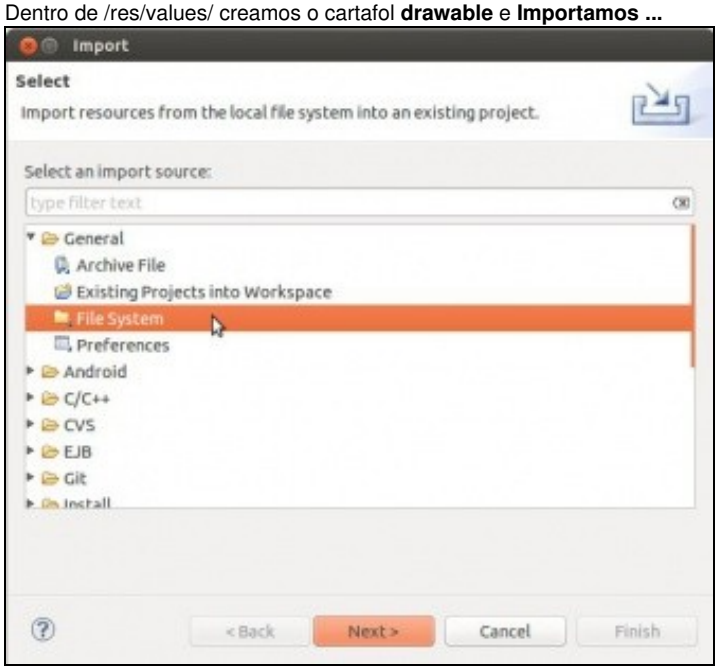
- Para poder asignar unha imaxe a un botón esta debe estar no cartafol /**res**/**drawable**

- Importar unha imaxe
- 

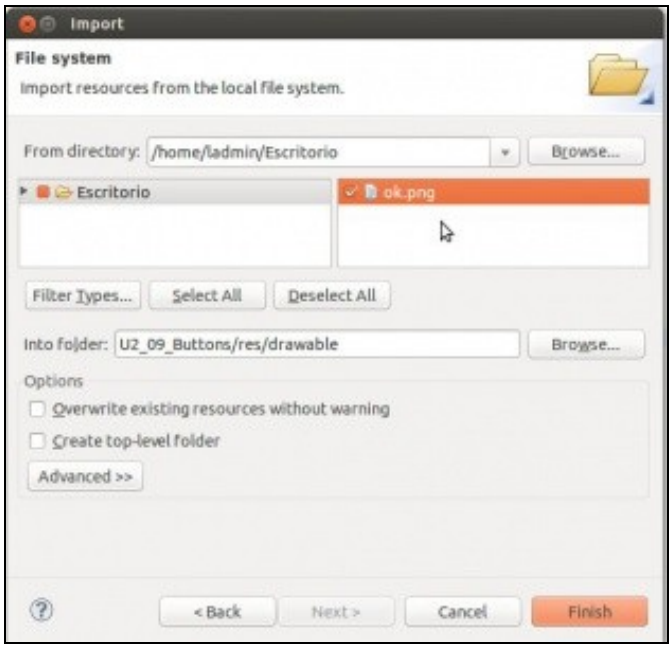
Descargamos a imaxe superior ou collemos outra icona calquera. O nome da imaxe só pode ter letras e números e nada máis, nin guións. O nome debe estar en minúscula. Ollo que cando se baixa este icono ten o nome comezando por maiúsculas.

- 

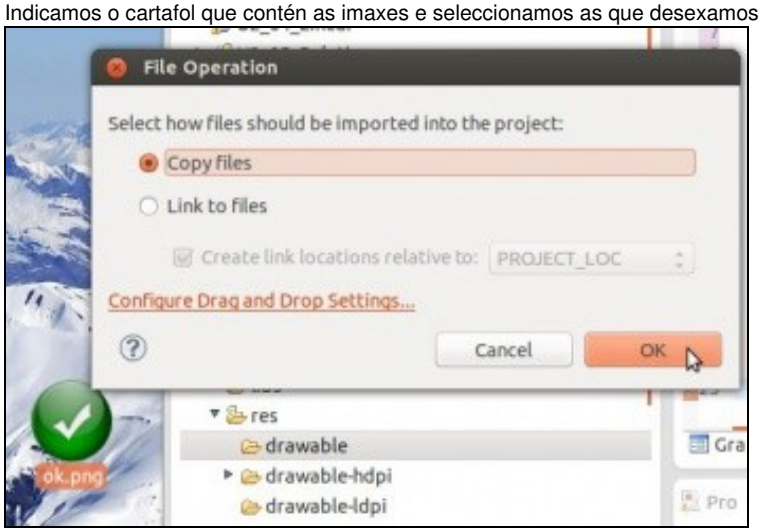
Dentro de /res/values/ creamos o cartafol **drawable** e **Importamos ...**

- 

Dende o sistema de ficheiros.

- 

Indicamos o cartafol que contén as imaxes e seleccionamos as que desexamos importar para o proxecto.

- 

Neste caso cópiase ao proxecto. Lembrar cambiar o nome a minúsculas, se este contén maiúsculas.

- Todo o anterior pode realizarse arrastrando a imaxe dende a súa localización ao IDE, ao cartafol desexado do proxecto dentro do IDE.
- En Android Studio realñizase copiando a imaxe e pegándoa no cartafol drawable no IDE

1.3.2 Creación do layout

- O layout XML ten 3 botóns e 1 TextView:
- Observar como hai un layout dentro doutro: un dispón os elementos en vertical e o outro en horizontal.

```
<LinearLayout 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:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/btn_boton"
            android:layout_width="150sp"
            android:layout_height="wrap_content"
            android:text="Preme aquí" >
        </Button>
```

```
        <ToggleButton
            android:id="@+id/tbtn_boton_2estados"
            android:layout_width="100dp"
            android:layout_height="match_parent"
            android:textOff="Apagado"
            android:textOn="Aceso" >
        </ToggleButton>

        <ImageButton
            android:id="@+id/ibtn_boton_imaxe"
            android:layout_width="55sp"
            android:layout_height="50sp"
            android:scaleType="fitXY"
            android:src="@drawable/ok" >
        </ImageButton>
    </LinearLayout>

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

</LinearLayout>
```

- Se cargamos este layout e non realizamos ningunha codificación en Java, non vai pasar nada cos botóns:

- Botóns inoperativos
- 

Prememos no primeiro e non fai nada.

- 

Se prememos no segundo só cambia o seu estado, pero tampouco fai nada.

- 

O mesmo pasa co terceiro botón.

1.4 Eventos

- Nesta ocasión imos ver dun xeito sinxelo como capturar os eventos dos botóns. Máis adiante afondaremos no control de eventos.
- Un **evento** é algo que acontece nun control e que nos interesa capturar no sistema para desencadenar (ou non) unha serie de accións.
- Xa vimos no caso anterior (EditText) que podíamos controlar eventos que acontecen nos controis.
- Nesta ocasión para os Botóns imos facelo de dúas formas:
    - ♦ Control de eventos dende o layout
    - ♦ Control de eventos a través dunha clase anónima.

1.4.1 Control de eventos dende o layout

- É a forma máis sinxela de desencadear unha acción.
- Imos facelo de dúas formas:
    - ♦ Creando un método para cada Botón.
    - ♦ Creando un único método para tódolos botóns. Hai que controlar que botón foi o que se premeu.

1.4.1.1 Crear un método para cada botón

- Observar a propiedade: **android:onClick** nos controis dos botóns.

```
<LinearLayout 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:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/btn_boton"
            android:layout_width="150sp"
            android:layout_height="wrap_content"
            android:onClick="onBotonClick"
            android:text="Preme aquí" >
        </Button>

        <ToggleButton
            android:id="@+id/tbtn_boton_2estados"
            android:layout_width="100dp"
            android:layout_height="match_parent"
            android:onClick="onBoton2EstadosClick"
            android:textOff="Apagado"
            android:textOn="Aceso" >
        </ToggleButton>

        <ImageButton
            android:id="@+id/ibtn_boton_imaxe"
            android:layout_width="55sp"
            android:layout_height="50sp"
            android:contentDescription="Botón imaxe"
            android:onClick="onBotonImaxeClick"
            android:scaleType="fitXY"
            android:src="@drawable/ok" >
        </ImageButton>
    </LinearLayout>

    <TextView
        android:id="@+id/tv_accion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/texto_tv" />

</LinearLayout>
```

- Para cada botón defínese un método que xestione o evento onClick.
- **Deixamos para o participante no curso a definición no ficheiro /res/values/strings.xml da constante "@string/texto_tv" do último TextView.**

- Agora só queda definir os métodos en Java.

```
package com.example.u2_09_buttons;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.ToggleButton;

public class U2_09_Buttons extends Activity {
private Button btnBoton;
private ToggleButton tbtnBoton2Estados;
```

```
private ImageButton ibtnBotonImaxe;
private TextView tvAccions;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_u2_09__buttons);

btnBoton = (Button) findViewById(R.id.btn_boton);
tbtnBoton2Estados = (ToggleButton) findViewById(R.id.tbtn_boton_2estados);
ibtnBotonImaxe = (ImageButton) findViewById(R.id.ibtn_boton_imaxe);
tvAccions = (TextView) findViewById(R.id.tv_accion);
}

public void onBotonClick(View v) {
tvAccions.setText("Premeches o primeiro botón\n");
tvAccions.append("O texto do botón é: " + btnBoton.getText());
}

public void onBoton2EstadosClick(View v) {
tvAccions.setText("Premeches o segundo botón\n");
if (tbtnBoton2Estados.isChecked())
tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOn());
else
tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOff());
}

public void onBotonImaxeClick(View v) {
tvAccions.setText("Premeches o terceiro botón\n");
tvAccions.append("O ancho é: " + ibtnBotonImaxe.getWidth());
}

@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_09__buttons, menu);
return true;
}

}
```

- **Liñas 13-16**: declaración de atributos.
- **Liñas 23-26**: asignación de valores aos atributos mediante a clase R.
- **Liñas 29-32**: Define o método asociado ao evento android:onClick do primeiro botón. "\n" introduce un salto de liña.
- **Liñas 34-40**: Define o método asociado ao evento android:onClick do botón ToggleButton. Comprobamos se está activado o non.
- **Liñas 42-45**: Define o método asociado ao evento android:onClick do botón con Imaxe.

- Lanzar a aplicación e comprobar que sucede ao premer os botóns.

1.4.1.2 Crear un único método para tódolos botóns

- Hai que modificar o XML para que as propiedades **android:onClick** de tódolos botóns chamen ao mesmo método.

```
<LinearLayout 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:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/btn_boton"
            android:layout_width="150sp"
            android:layout_height="wrap_content"
            android:onClick="onBotonClick"
            android:text="Preme aquí" >
        </Button>
```

```
        <ToggleButton
            android:id="@+id/tbtn_boton_2estados"
            android:layout_width="100dp"
            android:layout_height="match_parent"
            android:onClick="onBotonClick"
            android:textOff="Apagado"
            android:textOn="Aceso" >
        </ToggleButton>

        <ImageButton
            android:id="@+id/ibtn_boton_imaxe"
            android:layout_width="55sp"
            android:layout_height="50sp"
            android:contentDescription="Botón imaxe"
            android:onClick="onBotonClick"
            android:scaleType="fitXY"
            android:src="@drawable/ok" >
        </ImageButton>
    </LinearLayout>

    <TextView
        android:id="@+id/tv_accion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/texto_tv" />

</LinearLayout>
```

- A definición do método:

```
package com.example.u2_09_buttons;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.ToggleButton;

public class U2_09_Buttons extends Activity {
// private Button btnBoton;
// private ToggleButton tbtnBoton2Estados;
// private ImageButton ibtnBotonImaxe;
private TextView tvAccions;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_u2_09__buttons);

tvAccions = (TextView) findViewById(R.id.tv_accion);

}

public void onBotonClick(View vista) {

switch (vista.getId()) {
case R.id.btn_boton:
tvAccions.setText("Premeches o primeiro botón\n");
tvAccions.append("O texto do botón é: "
+ ((Button) vista).getText());
break;

case R.id.tbtn_boton_2estados:
```

```
tvAccions.setText("Premeches o segundo botón\n");
if (((ToggleButton) vista).isChecked())
tvAccions.append("O estado é: "
+ ((ToggleButton) vista).getTextOn());
else
tvAccions.append("O estado é: "
+ ((ToggleButton) vista).getTextOff());
break;

case R.id.ibtn_boton_imaxe:
tvAccions.setText("Premeches o terceiro botón\n");
        tvAccions.append("O ancho é: " + ((ImageButton) vista).getWidth());
}
}

/*public void onBotonClick(View v) {
                tvAccions.setText("Premeches o primeiro botón\n");
                tvAccions.append("O texto do botón é: " + btnBoton.getText());
        }*/


/*
           public void onBoton2EstadosClick(View v) {
           tvAccions.setText("Premeches o segundo botón\n"); if
           (tbtnBoton2Estados.isChecked()) tvAccions.append("O estado é: " +
           tbtnBoton2Estados.getTextOn()); else tvAccions.append("O estado é: " +
           tbtnBoton2Estados.getTextOff()); }
         */


/*
           public void onBotonImaxeClick(View v) {
           tvAccions.setText("Premeches o terceiro botón\n");
           tvAccions.append("O ancho é: " + ibtnBotonImaxe.getWidth()); }
         */

@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_09__buttons, menu);
return true;
}

}
```

- **Liñas 28-51**:
    - ♦ Na chamada ao método recibimos como parámetro unha View (vista) que apunta ao obxecto que o chamou.
    - ♦ Collendo o ID da vista, podemos saber que botón foi o que iniciou o evento.
    - ♦ Con **switch - case**, en función do botón que lanzou o evento executamos o código correspondente.
- **Liñas 39, 41 e 43**:
    - ♦ Observar como se fai casting do obxecto recibido. Recíbese un obxecto de tipo View (vista) e precisamos convertelo á ToogleButton para acceder aos seus métodos específicos.

1.4.2 Control de eventos usando un listener. Clase anónima

- Nesta ocasión imos crear un Listener para o primeiro botón. Co cal, hai que eliminar a propiedade **android:onClick** do XML do primeiro botón

- O ficheiro XML:
- Observar que no primeiro botón non se xestiona o evento onClick e que os outros dous botóns seguen mantendo a propiedade **android:onClick**.

```
<LinearLayout 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:orientation="vertical" >
```

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/btn_boton"
            android:layout_width="150sp"
            android:layout_height="wrap_content"
            android:text="Preme aquí" >
        </Button>

        <ToggleButton
            android:id="@+id/tbtn_boton_2estados"
            android:layout_width="100dp"
            android:layout_height="match_parent"
            android:onClick="onBoton2EstadosClick"
            android:textOff="Apagado"
            android:textOn="Aceso" >
        </ToggleButton>

        <ImageButton
            android:id="@+id/ibtn_boton_imaxe"
            android:layout_width="55sp"
            android:layout_height="50sp"
            android:contentDescription="Botón imaxe"
            android:onClick="onBotonImaxeClick"
            android:scaleType="fitXY"
            android:src="@drawable/ok" >
        </ImageButton>
    </LinearLayout>

    <TextView
        android:id="@+id/tv_accion"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/texto_tv" />

</LinearLayout>
```

- Imos partir do código da primeira versión da ampliación.
- Comentaremos aquel código que afectaba ao primeiro botón e usaremos un Listener asociado ao primeiro botón.

```
package com.example.u2_09_buttons;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.ToggleButton;

public class U2_09_Buttons extends Activity {
private Button btnBoton;
private ToggleButton tbtnBoton2Estados;
private ImageButton ibtnBotonImaxe;
private TextView tvAccions;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_u2_09__buttons);

btnBoton = (Button) findViewById(R.id.btn_boton);
```

```
tbtnBoton2Estados = (ToggleButton) findViewById(R.id.tbtn_boton_2estados);
ibtnBotonImaxe = (ImageButton) findViewById(R.id.ibtn_boton_imaxe);
tvAccions = (TextView) findViewById(R.id.tv_accion);

btnBoton.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// TODO Auto-generated method stub
tvAccions.setText("Premeches o primeiro botón\n");
tvAccions.append("O texto do botón é: " + btnBoton.getText());
}

});

}

/*      public void onBotonClick(View v) {
                tvAccions.setText("Premeches o primeiro botón\n");
                tvAccions.append("O texto do botón é: " + btnBoton.getText());
        }
*/

public void onBoton2EstadosClick(View v) {
tvAccions.setText("Premeches o segundo botón\n");
if (tbtnBoton2Estados.isChecked())
tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOn());
else
tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOff());
}

public void onBotonImaxeClick(View v) {
tvAccions.setText("Premeches o terceiro botón\n");
tvAccions.append("O ancho é: " + ibtnBotonImaxe.getWidth());
}

@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_09__buttons, menu);
return true;
}

}
```

- **Liñas 29-38**: chamamos ao método **setOnClickListener()** do primeiro botón que se activará cando se fai click no botón.
    - ♦ Creamos unha clase anónima asociada a interface OnClickListener e sobreescribimos o único método que ten: **onClick**.

- Como xa se dixo no anterior control (EditText), para coñecer o método asociado ao botón escribimos **obxecto.setOn**, prememos **CTRL+Barra espaciadora** e xa nos auto completa.
- O mesmo facemos para a creación da clase anónima.

- Premer **CTRL + Barra espaciadora**

- Clase anónima co seu método.

```
    btnBoton.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
        }

    });

}
```

- Quedan presentados os **listeners** máis adiante afondaremos sobre eles.

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).