

Button. ToggleButton. Control de eventos II

Sumario

Introdución

Botóns de 2 estados: ToggleButton/Switch

Casos prácticos

Importar unha imaxe como recurso

Creación do layout

Eventos

Control de eventos dende o layout

Crear un método para cada botón

Crear un único método para todos os botóns

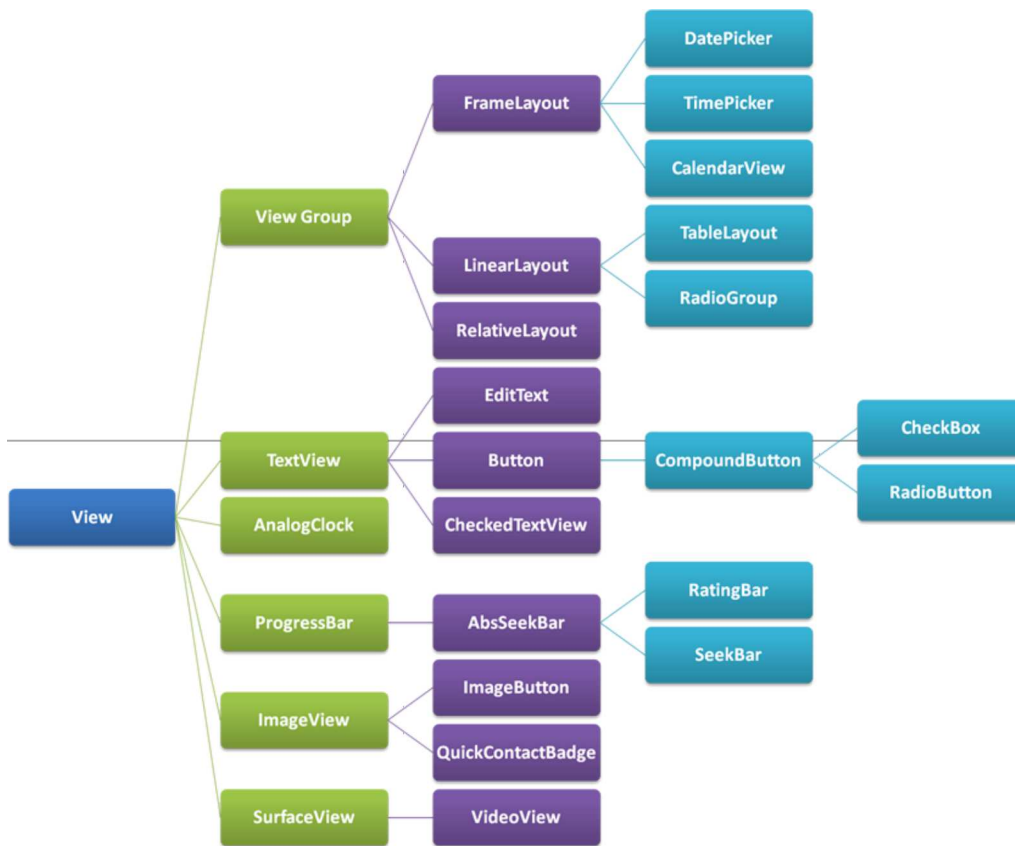
Control de eventos usando un listener. Clase anónima

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
 - ToggleButton 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:**

```

1 <Button
2   android:layout_width="wrap_content"
3   android:layout_height="wrap_content"
4   android:text="@string/button_text"
5   ... />

```

- **Botón con imaxe**

```

1 <ImageButton
2   android:layout_width="wrap_content"
3   android:layout_height="wrap_content"
4   android:src="@drawable/button_icon"
5   ... />

```

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

```

1 <Button
2   android:layout_width="wrap_content"
3   android:layout_height="wrap_content"
4   android:text="@string/button_text"
5   android:drawableLeft="@drawable/button_icon"
6   ... />

```

- 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 **public void onMeuMetodo (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>
 - Introducción 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>

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/>**



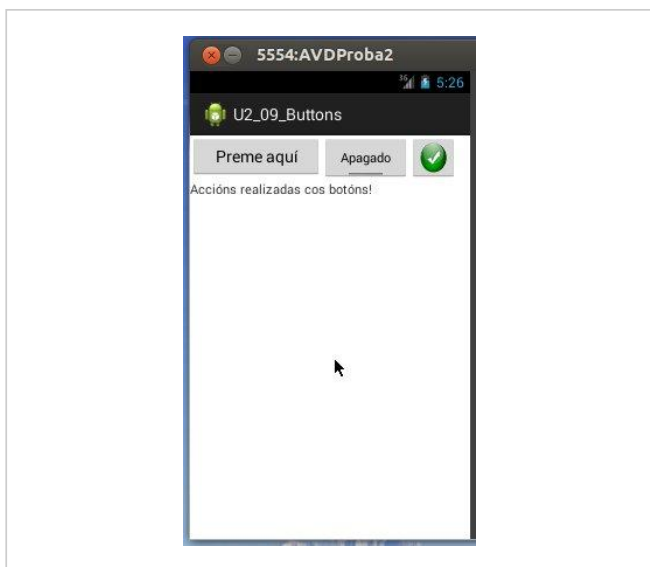
- Permiten cambiar o seu estado desprazando co dun estado a outro. O funcionamento é semellante ao control ToogleButton.
- Un obxecto ToogleButton/Switch herda da clase **CompoundButton**, quen, á súa vez, herda da clase **Button**.
- Por tanto funcionan da mesma maneira, pero ademais o este tipo de 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>
 - Introducción as botóns de 2 estados: <http://developer.android.com/guide/topics/ui/controls/togglebutton.html>

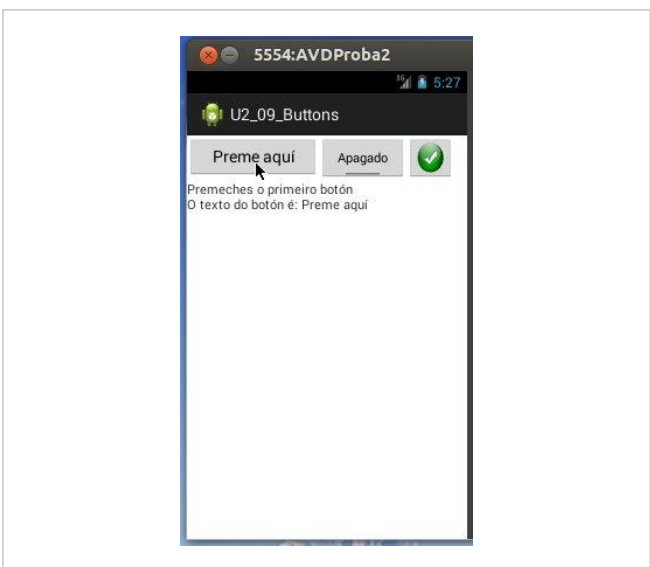
Casos prácticos

- Como sempre creamos un novo proxecto: **U2_09_Buttons**
- Imos crear 3 botóns:
 - 1 Botón con texto
 - 1 ToogleButton
 - 1 botón con imaxe

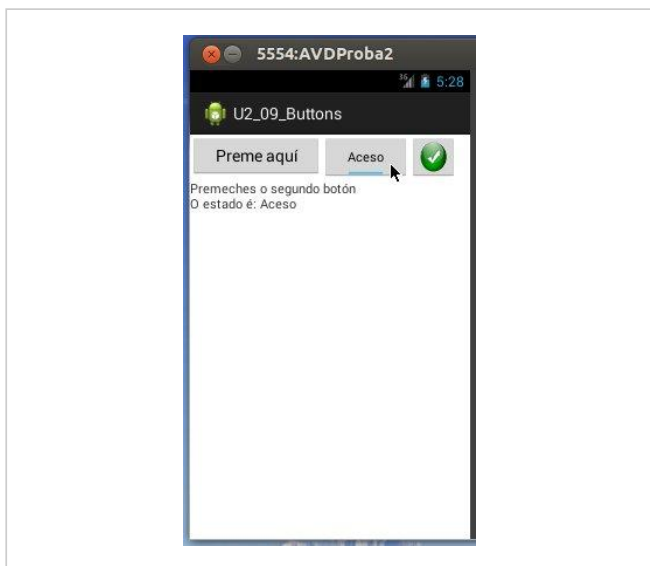
Tres botóns



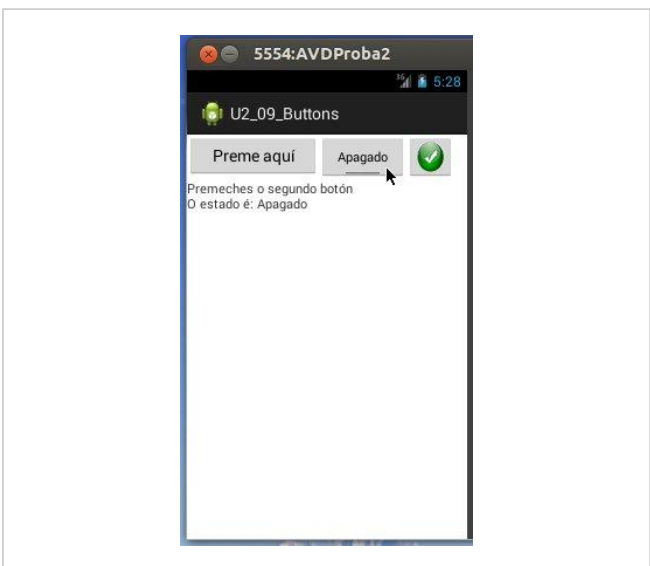
Cando entremos na aplicación este será o seu estado.



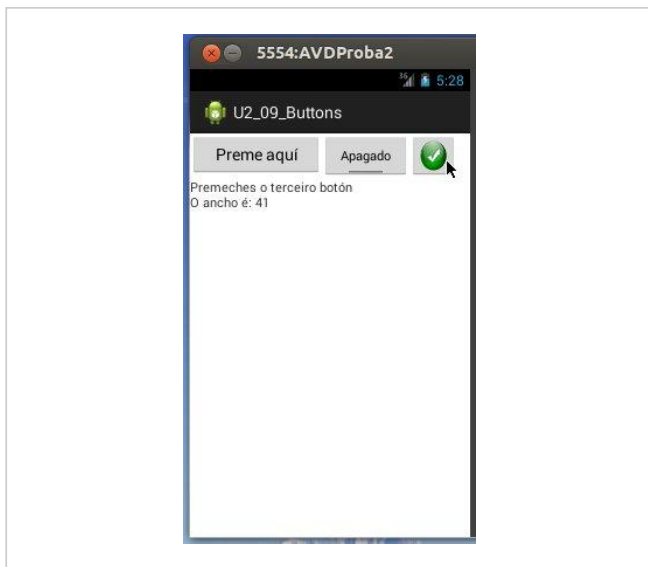
Se prememos o primeiro botón (Texto) amosarase un texto en pantalla.



Se prememos o segundo botón (ToggleButton) 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.

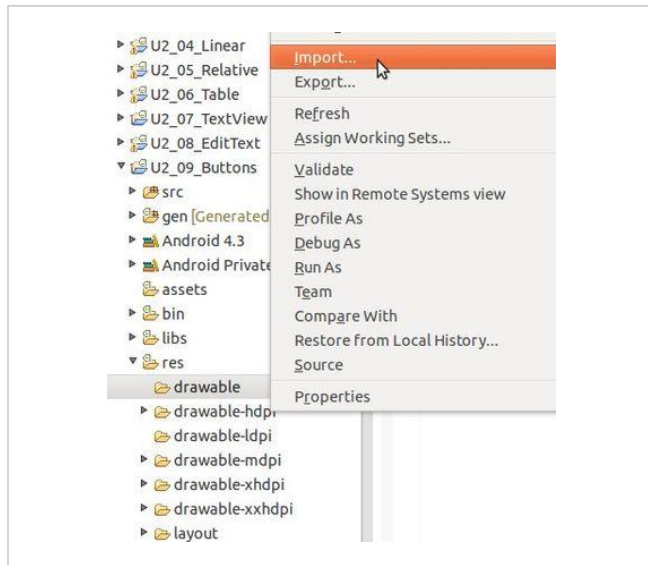


Se prememos o terceiro botón (Imaxe) amosarase de novo un texto diferente.

Importar unha imaxe como recurso

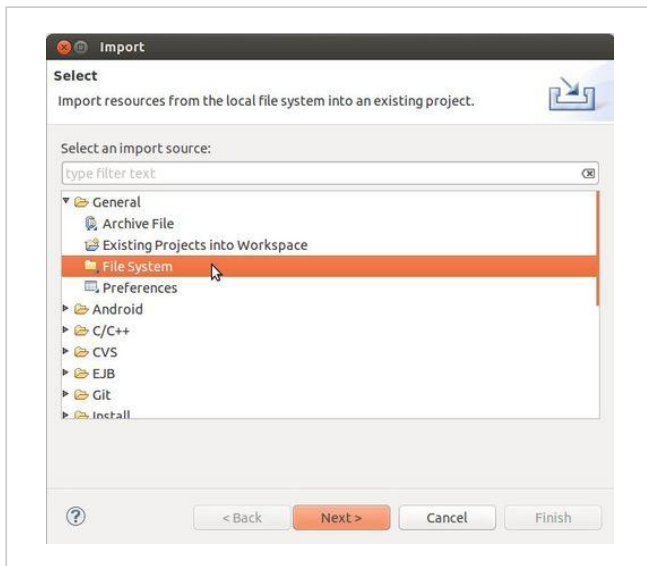
- Para poder asignar unha imaxe a un botón esta debe estar no cartafol **/res/drawable**

Importar unha imaxe

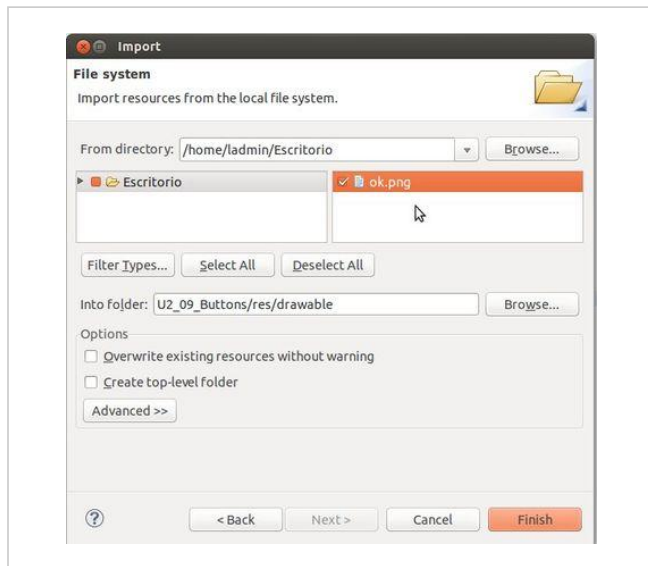


Descargamos a imaxe superior ou collemos outra icona calquera. O nome da imaxe só pode ter letras e números e nada máis, nin guións. O nome debe estar en minúscula. Olo que cando se baixa este icono ten o nome comezando por maiúsculas.

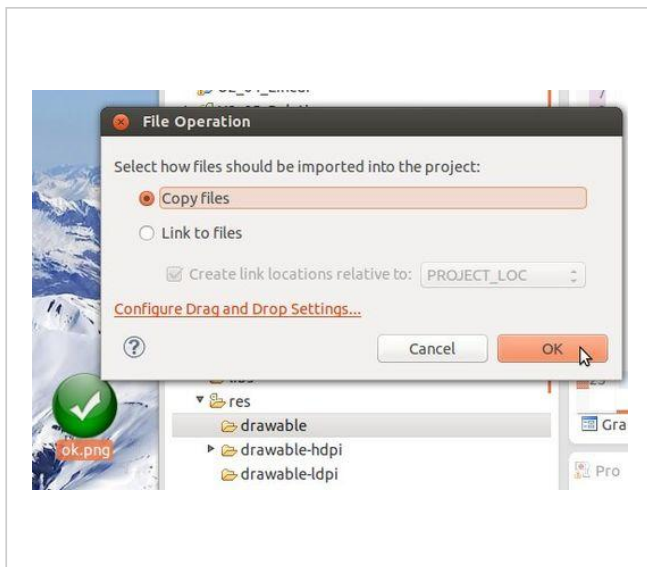
Dentro de /res/values/ creamos o cartafol **drawable** e **Importamos ...**



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 realízase copiando a imaxe e pegándoa no cartafol drawable no IDE

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.

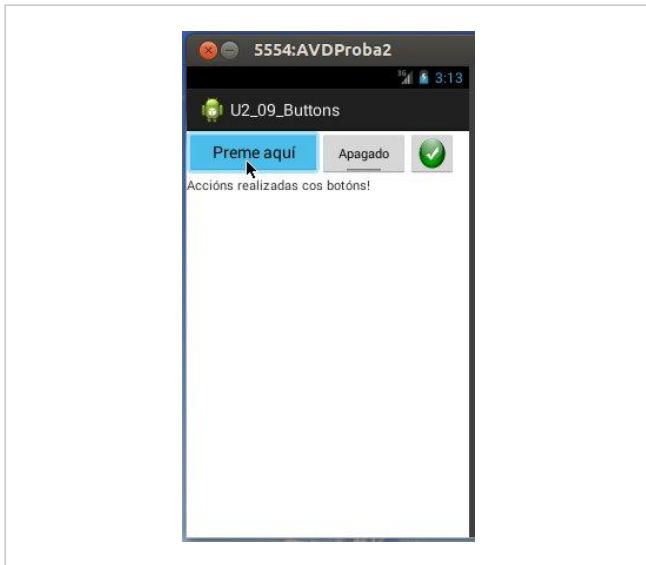
```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6
7   <LinearLayout
8     android:layout_width="match_parent"
9     android:layout_height="wrap_content"
10    android:orientation="horizontal" >
11
12     <Button
13       android:id="@+id/btn_boton"
14       android:layout_width="150sp"
15       android:layout_height="wrap_content"
16       android:text="Preme aqui" >
17     </Button>
18
19     <ToggleButton
20       android:id="@+id/tbtn_boton_2estados"
21       android:layout_width="100dp"
22       android:layout_height="match_parent"
23       android:textOff="Apagado"
24       android:textOn="Aceso" >
25     </ToggleButton>
26
27     <ImageButton
28       android:id="@+id/ibtn_boton_imaxe"
29       android:layout_width="55sp"
30       android:layout_height="50sp"
31       android:scaleType="fitXY"
32       android:src="@drawable/ok" >
33     </ImageButton>
34   </LinearLayout>
35
36   <TextView
37     android:layout_width="wrap_content"
38     android:layout_height="wrap_content"
39     android:text="@string/texto_tv" />
40
41 </LinearLayout>

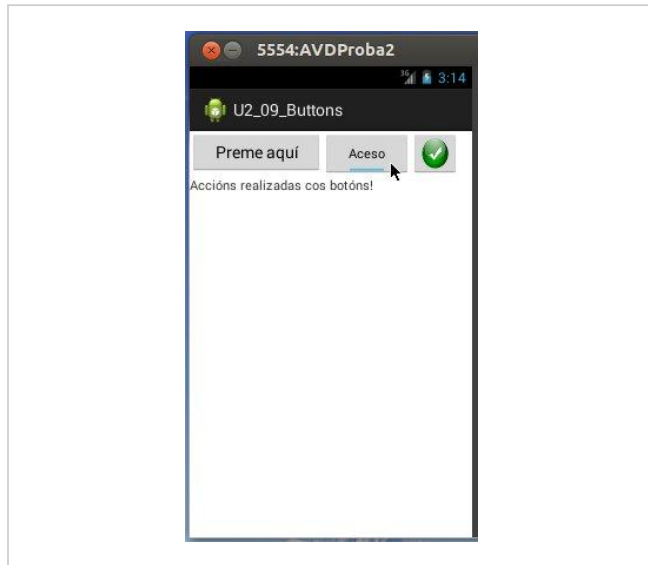
```

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

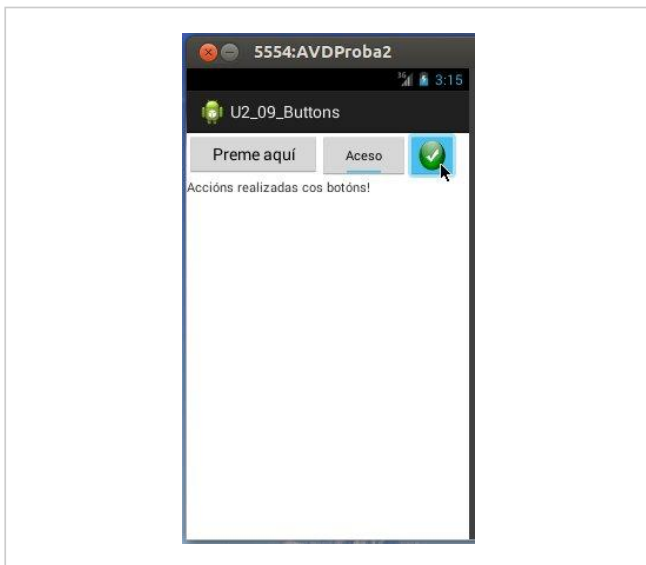
Botóns inoperativos



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.

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.

Control de eventos dende o layout

- É a forma máis sinxela de desencadenar 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.

Crear un método para cada botón

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

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6
7   <LinearLayout
8     android:layout_width="match_parent"
9     android:layout_height="wrap_content"
10    android:orientation="horizontal" >
11
12     <Button
13       android:id="@+id/btn_boton"
14       android:layout_width="150sp"
15       android:layout_height="wrap_content"
16       android:onClick="onBotonClick"
17       android:text="Preme aqui" >
18   </Button>
19
20   <ToggleButton
21     android:id="@+id/tbtn_boton_2estados"
22     android:layout_width="100dp"
23     android:layout_height="match_parent"
24     android:onClick="onBoton2EstadosClick"
25     android:textOff="Apagado"
26     android:textOn="Aceso" >
27 </ToggleButton>
28
29   <ImageButton
30     android:id="@+id/ibtn_boton_imaxe"
31     android:layout_width="55sp"
32     android:layout_height="50sp"
33     android:contentDescription="Botón imaxe"
34     android:onClick="onBotonImaxeClick"
35     android:scaleType="fitXY"
36     android:src="@drawable/ok" >
37 </ImageButton>
38 </LinearLayout>
39
40 <TextView
41   android:id="@+id/tv_accion"
42   android:layout_width="wrap_content"
43   android:layout_height="wrap_content"
44   android:text="@string/texto_tv" />
45
46 </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.

```

1 package com.example.u2_09_buttons;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Menu;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.ImageButton;
9 import android.widget.TextView;
10 import android.widget.ToggleButton;
11
12 public class U2_09_Buttons extends Activity {
13     private Button btnBoton;
14     private ToggleButton tbtnBoton2Estados;
15     private ImageButton ibtnBotonImaxe;
16     private TextView tvAccions;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_u2_09_buttons);

```

```

22
23     btnBoton = (Button) findViewById(R.id.btn_boton);
24     tbtnBoton2Estados = (ToggleButton) findViewById(R.id.tbtn_boton_2estados);
25     ibtnBotonImaxe = (ImageButton) findViewById(R.id.ibtn_boton_imaxe);
26     tvAccions = (TextView) findViewById(R.id.tv_accion);
27 }
28
29 public void onBotonClick(View v) {
30     tvAccions.setText("Premeches o primeiro botón\n");
31     tvAccions.append("O texto do botón é: " + btnBoton.getText());
32 }
33
34 public void onBoton2EstadosClick(View v) {
35     tvAccions.setText("Premeches o segundo botón\n");
36     if (tbtnBoton2Estados.isChecked())
37         tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOn());
38     else
39         tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOff());
40 }
41
42 public void onBotonImaxeClick(View v) {
43     tvAccions.setText("Premeches o terceiro botón\n");
44     tvAccions.append("O ancho é: " + ibtnBotonImaxe.getWidth());
45 }
46
47 @Override
48 public boolean onCreateOptionsMenu(Menu menu) {
49     // Inflate the menu; this adds items to the action bar if it is present.
50     getMenuInflater().inflate(R.menu.u2_09_buttons, menu);
51     return true;
52 }
53
54 }

```

- 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.

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.

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical" >
6
7   <LinearLayout
8     android:layout_width="match_parent"
9     android:layout_height="wrap_content"
10    android:orientation="horizontal" >
11
12     <Button
13       android:id="@+id/btn_boton"
14       android:layout_width="150sp"
15       android:layout_height="wrap_content"
16       android:onClick="onBotonClick"
17       android:text="Preme aqui" >
18
19     </Button>
20
21     <ToggleButton
22       android:id="@+id/tbtn_boton_2estados"
23       android:layout_width="100dp"
24       android:layout_height="match_parent"
25       android:onClick="onBotonClick"
26       android:textOff="Apagado"
27       android:textOn="Aceso" >
28
29     </ToggleButton>
30
31     <ImageButton
32       android:id="@+id/ibtn_boton_imaxe"
33       android:layout_width="55sp"
34       android:layout_height="50sp"
35       android:contentDescription="Botón imaxe"
36       android:onClick="onBotonClick"
37       android:scaleType="fitXY"
38       android:src="@drawable/ok" >

```

```

37     </ImageButton>
38 </LinearLayout>
39
40 <TextView
41     android:id="@+id/tv_accion"
42     android:layout_width="wrap_content"
43     android:layout_height="wrap_content"
44     android:text="@string/texto_tv" />
45
46 </LinearLayout>

```

■ A definición do método:

```

1 package com.example.u2_09_buttons;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Menu;
6 import android.view.View;
7 import android.view.View.OnClickListener;
8 import android.widget.Button;
9 import android.widget.ImageButton;
10 import android.widget.TextView;
11 import android.widget.ToggleButton;
12
13 public class U2_09_Buttons extends Activity {
14     // private Button btnBoton;
15     // private ToggleButton tbtnBoton2Estados;
16     // private ImageButton ibtnBotonImaxe;
17     private TextView tvAccions;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_u2_09_buttons);
23
24         tvAccions = (TextView) findViewById(R.id.tv_accion);
25
26     }
27
28     public void onBotonClick(View vista) {
29
30         switch (vista.getId()) {
31             case R.id.btn_boton:
32                 tvAccions.setText("Premeches o primeiro botón\n");
33                 tvAccions.append("O texto do botón é: "
34                     + ((Button) vista).getText());
35                 break;
36
37             case R.id.tbtn_boton_2estados:
38                 tvAccions.setText("Premeches o segundo botón\n");
39                 if (((ToggleButton) vista).isChecked())
40                     tvAccions.append("O estado é: "
41                         + ((ToggleButton) vista).getTextOn());
42                 else
43                     tvAccions.append("O estado é: "
44                         + ((ToggleButton) vista).getTextOff());
45                 break;
46
47             case R.id.ibtn_boton_imaxe:
48                 tvAccions.setText("Premeches o terceiro botón\n");
49                 tvAccions.append("O ancho é: " + ((ImageButton) vista).getWidth());
50         }
51     }
52
53     /*public void onBotonClick(View v) {
54         tvAccions.setText("Premeches o primeiro botón\n");
55         tvAccions.append("O texto do botón é: " + btnBoton.getText());
56     }*/
57
58
59     /*
60     public void onBoton2EstadosClick(View v) {
61         tvAccions.setText("Premeches o segundo botón\n"); if
62         (tbtnBoton2Estados.isChecked()) tvAccions.append("O estado é: " +
63         tbtnBoton2Estados.getTextOn()); else tvAccions.append("O estado é: " +
64         tbtnBoton2Estados.getTextOff()); }
65     */
66
67
68     /*

```

```

69     public void onBotonImaxeClick(View v) {
70         tvAccions.setText("Premeches o terceiro botón\n");
71         tvAccions.append("O ancho é: " + ibtnBotonImaxe.getWidth()); }
72     */
73
74     @Override
75     public boolean onCreateOptionsMenu(Menu menu) {
76         // Inflate the menu; this adds items to the action bar if it is present.
77         getMenuInflater().inflate(R.menu.u2_09__buttons, menu);
78         return true;
79     }
80
81 }

```

▪ 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.

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**.

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" >
6
7     <LinearLayout
8         android:layout_width="match_parent"
9         android:layout_height="wrap_content"
10        android:orientation="horizontal" >
11
12        <Button
13            android:id="@+id/btn_boton"
14            android:layout_width="150sp"
15            android:layout_height="wrap_content"
16            android:text="Preme aqui" >
17        </Button>
18
19        <ToggleButton
20            android:id="@+id/tbtn_boton_2estados"
21            android:layout_width="100dp"
22            android:layout_height="match_parent"
23            android:onClick="onBoton2EstadosClick"
24            android:textOff="Apagado"
25            android:textOn="Aceso" >
26        </ToggleButton>
27
28        <ImageButton
29            android:id="@+id/ibtn_boton_imaxe"
30            android:layout_width="55sp"
31            android:layout_height="50sp"
32            android:contentDescription="Botón imaxe"
33            android:onClick="onBotonImaxeClick"
34            android:scaleType="fitXY"
35            android:src="@drawable/ok" >
36        </ImageButton>
37    </LinearLayout>
38
39    <TextView
40        android:id="@+id/tv_accion"
41        android:layout_width="wrap_content"
42        android:layout_height="wrap_content"
43        android:text="@string/texto_tv" />
44
45 </LinearLayout>

```

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

```

1 package com.example.u2_09_buttons;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Menu;
6 import android.view.View;
7 import android.view.View.OnClickListener;
8 import android.widget.Button;
9 import android.widget.ImageButton;
10 import android.widget.TextView;
11 import android.widget.ToggleButton;
12
13 public class U2_09_Buttons extends Activity {
14     private Button btnBoton;
15     private ToggleButton tbtnBoton2Estados;
16     private ImageButton ibtnBotonImaxe;
17     private TextView tvAccions;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_u2_09_buttons);
23
24         btnBoton = (Button) findViewById(R.id.btn_boton);
25         tbtnBoton2Estados = (ToggleButton) findViewById(R.id.tbtn_boton_2estados);
26         ibtnBotonImaxe = (ImageButton) findViewById(R.id.ibtn_boton_imaxe);
27         tvAccions = (TextView) findViewById(R.id.tv_accion);
28
29         btnBoton.setOnClickListener(new OnClickListener() {
30
31             @Override
32             public void onClick(View v) {
33                 // TODO Auto-generated method stub
34                 tvAccions.setText("Premeches o primeiro botón\n");
35                 tvAccions.append("O texto do botón é: " + btnBoton.getText());
36             }
37
38         });
39     }
40
41     /* public void onBotonClick(View v) {
42         tvAccions.setText("Premeches o primeiro botón\n");
43         tvAccions.append("O texto do botón é: " + btnBoton.getText());
44     }
45 */
46
47     public void onBoton2EstadosClick(View v) {
48         tvAccions.setText("Premeches o segundo botón\n");
49         if (tbtnBoton2Estados.isChecked())
50             tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOn());
51         else
52             tvAccions.append("O estado é: " + tbtnBoton2Estados.getTextOff());
53     }
54
55     public void onBotonImaxeClick(View v) {
56         tvAccions.setText("Premeches o terceiro botón\n");
57         tvAccions.append("O ancho é: " + ibtnBotonImaxe.getWidth());
58     }
59
60     @Override
61     public boolean onCreateOptionsMenu(Menu menu) {
62         // Inflate the menu; this adds items to the action bar if it is present.
63         getMenuInflater().inflate(R.menu.u2_09_buttons, menu);
64         return true;
65     }
66
67 }
68

```

- **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 **objecto.setOn**, prememos **CTRL+Barra espaciadora** e xa nos auto completa.
- O mesmo facemos para a creación da clase anónima.
- Premer **CTRL + Barra espaciadora**

```

20 protected void onCreate(Bundle savedInstanceState) {
21     super.onCreate(savedInstanceState);
22     setContentView(R.layout.activity_u2_09_buttons);
23
24     btnBoton = (Button) findViewById(R.id.btn_boton);
25     btnBoton2Estados = (ToggleButton) findViewById(R.id.tbtn_boton_2estados);
26     btnBotonImaxe = (ImageButton) findViewById(R.id.ibtn_boton_imaxe);
27     tvAccions = (TextView) findViewById(R.id.tv_accion);
28
29     btnBoton.setOnClickListener(new OnClickListener() {
30
31     });
32
33 }
34
35
36

```



- 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).

Obtenido de «https://manuais.iessanclemente.net/index.php?title=Button._ToggleButton._Control_de_eventos_II&oldid=60676»

Esta página se editó por última vez el 3 oct 2016 a las 08:14.

El contenido está disponible bajo la licencia [Creative Commons: CC-BY-NC-SA](#), a menos que se indique lo contrario.