

Xestion de eventos III. onLongClick

Sumario

- 1 Introducción
- 2 Caso práctico
 - ◆ 2.1 XML do Layout
 - ◆ 2.2 O Código Java
 - ◆ 2.3 Propiedade android:onClick
 - ◆ 2.4 Listener
 - ◇ 2.4.1 A través dunha clase anónima
 - ◇ 2.4.2 Implementar a interface a través dunha clase
 - ◇ 2.4.3 Crear un obxecto que implemente a interface
- 3 onLongClick()
 - ◆ 3.1 Caso Práctico
 - ◇ 3.1.1 XML do Layout
 - ◇ 3.1.2 Código Java

Introdución

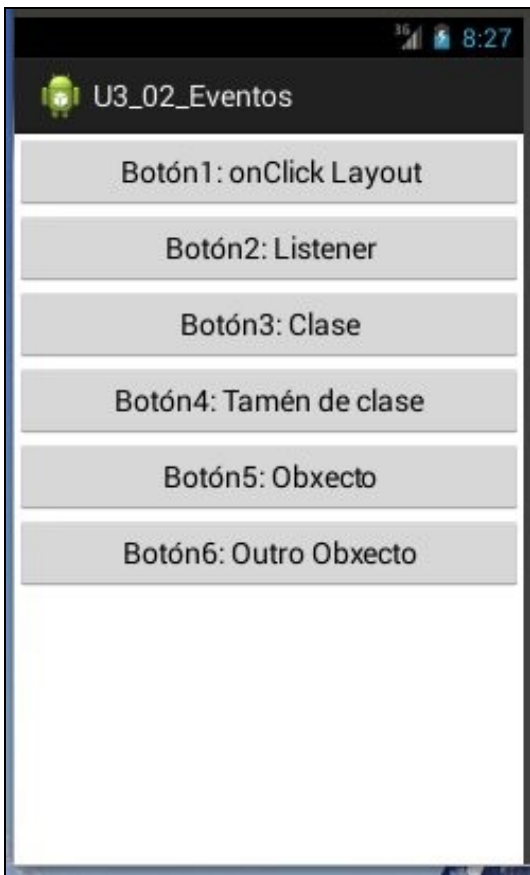
- En Android hai moitas formas de interceptar eventos do usuario.
- A clase View proporciona varias formas de xestionar eventos.
- Por exemplo cando un usuario toca un botón e chamado o método callback onTouchEvent() dese obxecto.
- Pero para interceptar ese evento debemos estender a clase e implementar o método.
- Pero non sería práctico para poder manexar o evento crear unha (sub)clase para cada obxecto Vista.
- Por iso a Clase Vista contén unha colección de interfaces anidadas.
- Estas interfaces chámanse **Event Listeners (Escoitadores de eventos)** e están listas para capturar a iteración do usuario coa UI.
- A continuación imos ver algunhas formas de capturar eventos:

- **Referencias**

- ◆ <http://developer.android.com/guide/topics/ui/ui-events.html>

Caso práctico

- Comezamos creando o proxecto: **U3_02_Eventos**
- Comezaremos capturado o evento **Click** e logo ao final veremos outros eventos.
- A aplicación coa que se traballará é a seguinte:



XML do Layout

- A liña 11 marca unha das formas máis sinxelas de capturar o evento Click sobre un compoñente.

```
<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">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Botón1: onClick Layout"
        android:onClick="onBotonClick"
        />

    <Button
        android:id="@+id/boton2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Botón2: Listener"
        />

    <Button
        android:id="@+id/boton3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Botón3: Clase"
        />

    <Button
        android:id="@+id/boton4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Botón4: Tamén de clase"
```

```

        />

        <Button
            android:id="@+id/boton5"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Botón5: Obxecto"
        />

        <Button
            android:id="@+id/boton6"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Botón6: Outro Obxecto"
        />
    </LinearLayout>

```

O Código Java

- A continuación preséntase o código Java, pero este vai ser debullado nos seguintes apartados.

```

package com.example.u3_02_eventos;

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.Toast;

public class U3_02_Eventos extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u3_02__eventos);

        Button boton2 = (Button) findViewById(R.id.boton2);
        boton2.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "Premeches o Botón2", Toast.LENGTH_SHORT).show();
            }
        });

        Button boton3 = (Button) findViewById(R.id.boton3);
        boton3.setOnClickListener(new XestionEventos());

        Button boton4 = (Button) findViewById(R.id.boton4);
        boton4.setOnClickListener(new XestionEventos());

        Button boton5 = (Button) findViewById(R.id.boton5);
        boton5.setOnClickListener(_OnClickListener);

        Button boton6 = (Button) findViewById(R.id.boton6);
        boton6.setOnClickListener(_OnClickListener);

    }

    private OnClickListener _OnClickListener = new OnClickListener() {

        @Override
        public void onClick(View v) {
            Button btn = (Button) v;
            Toast.makeText(v.getContext(), "Premeches" + btn.getText(), Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

    }
};

public void onBotonClick(View v) {
    Toast.makeText(this, "Premeches o Botón1", Toast.LENGTH_SHORT).show();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.u3_02__eventos, menu);
    return true;
}

}

```

Propiedade android:onClick

- Como xa se view é unha propiedade XML que teñen varios compoñentes (vistas).
- Hai que definir un método:
 - ◆ Que sexa public
 - ◆ Que non devolva nada: void
 - ◆ Que teña un parámetro da clase View, que será quen provoque o evento.
- No exemplo anterior son as liñas 50-52:

```

public void onBotonClick(View v) {
    Toast.makeText(this, "Premeches o Botón1", Toast.LENGTH_SHORT).show();
}

```

Listener

- Un Event Listener es una interface da clase Vista (View) que contén un único método de tipo callback que hai que implementar
- Android invoca estes métodos cando a Vista detecta que o usuario está provocando un tipo concreto de interacción con este elemento da interface de usuario.

Existen os seguintes métodos callback:

- - ◆ **onClick():** de **View.OnClickListener**. Este método invócase cando o usuario toca un elemento cun dedo (modo contacto), fai click coa bola de navegación (TrackBall) do dispositivo ou preme a tecla "Enter" estando nun compoeñente.
 - ◆ **onLongClick():** de **View.OnLongClickListener**. Este método chámase cando o usuario toca e mantén o dedo sobre un elemento (modo de contacto), fai click sen soltar coa bola de navegación (TrackBall) ou preme a tecla "Enter" perante un segundo estando nun elemento.
 - ◆ **onFocusChange():** de **View.OnFocusChangeListener**. Invócase cando o usuario move o cursor cara unha Vista ou se alonxa desta utilizando a bola de navegación ou usando as teclas de navegación.

onKey(): de **View.OnKeyListener**. Chámase cando o usuario se centra nun elemento e presiona ou libera una tecla do dispositivo.

- - ◆ ...
- **onLongClick()** verase nun exemplo posterior.

- Estes métodos son os únicos que se van implementar nas súas respectivas interfaces.
- Estas interfaces teñen o formato: **on...Listener()**
- Unha vez que temos implementada a interface témoslla que pasar como parámetro á vista (view) correspondente a través de **vista.set...Listener()**.

A través dunha clase anónima

- Exemplo: **Liñas 18-25** do código

```

Button boton2 = (Button) findViewById(R.id.boton2);
boton2.setOnClickListener(new OnClickListener() {

```

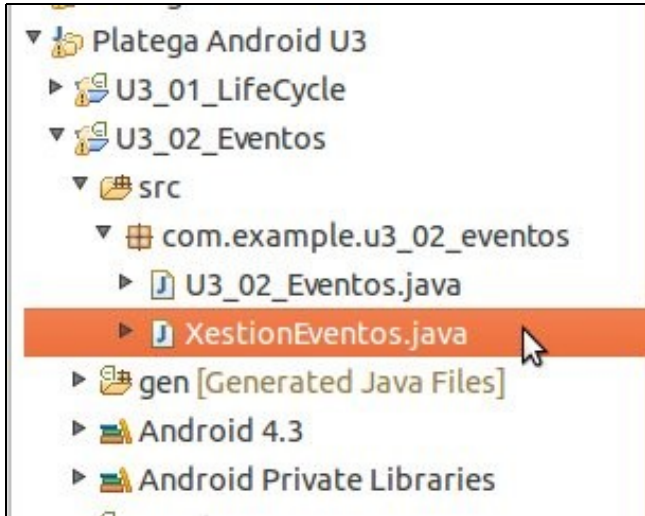
```

@Override
public void onClick(View v) {
    Toast.makeText(getApplicationContext(), "Premeches o Botón2", Toast.LENGTH_SHORT).show();
}
});

```

Implementar a interface a través dunha clase

- Creamos unha nova clase que implemente `onClickListener`



- Código Java da clase: **XestionEventos**

```

package com.example.u3_02_eventos;

import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class XestionEventos implements OnClickListener {

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

        Button btn = (Button) v;
        Toast.makeText(v.getContext(), "Premeches "+btn.getText(), Toast.LENGTH_SHORT).show();
    }
}

```

- Observar nas Liñas 28 e 31 (do código principal) como se crea unha nova clase `XestionEventos` para cada botón.
- Esa clase é a que vai implementar o método `onClick()`, como se ve enriba.

```

Button boton3 = (Button) findViewById(R.id.boton3);
boton3.setOnClickListener(new XestionEventos());

Button boton4 = (Button) findViewById(R.id.boton4);
boton4.setOnClickListener(new XestionEventos());

```

- Pero esta forma é un *engorro*, pois hai que estar creando clases.

Crear un obxecto que implemente a interface

- Para rematar imos crear un obxecto que implemente a interface e que poida ser pasado como parámetro as vistas que desexemos cada vez que se fai click nelas.
- **_OnClickListener** é un obxecto de tipo `OnClickListener`.

```
private OnClickListener _OnClickListener = new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        Button btn = (Button) v;  
        Toast.makeText(v.getContext(), "Premeches" + btn.getText(), Toast.LENGTH_SHORT).show();  
    }  
};
```

- Pasamos ese obxecto ás vistas que desexemos cando se faga Click nelas.

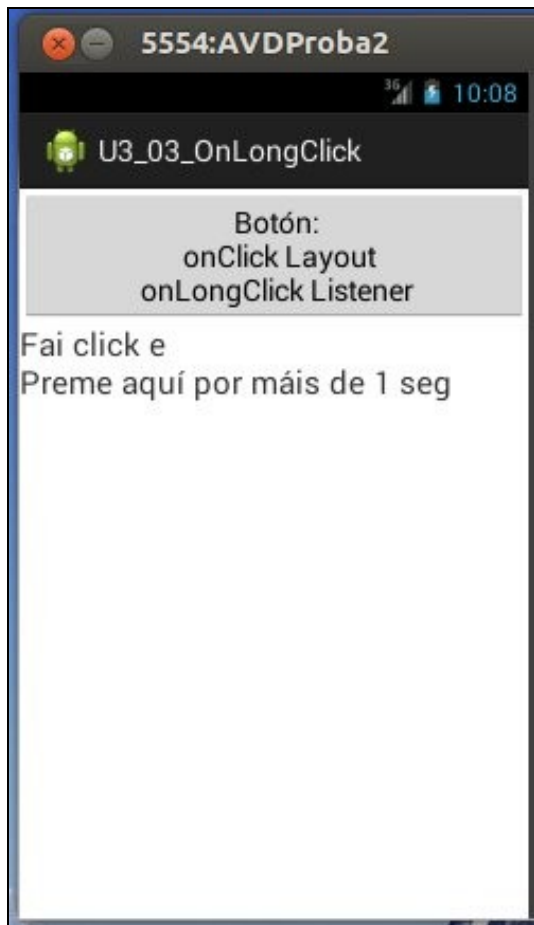
```
Button boton5 = (Button) findViewById(R.id.boton5);  
boton5.setOnClickListener(_OnClickListener);  
  
Button boton6 = (Button) findViewById(R.id.boton6);  
boton6.setOnClickListener(_OnClickListener);
```

onLongClick()

- Cando sobre unha vista se preme por duración de 1 segundo o máis lánzase o evento **LongClick** que é capturado polo método **onLongClick()** asociado ao correspondente interface.
- O método `onLongClick()` devolve un booleano para poder comprobar se o evento se consumou ou non. Se chegamos a premer por un segundo (ou máis) ou non.
 - ♦ Devolve **true** se puidemos capturar o evento e xa non fai nada máis.
 - ♦ Devolve **false** se non se puido capturar o evento e continua chamando a outros escoitadores tipo on-Click.

Caso Práctico

- Comezamos creando o proxecto **U3_03_OnLongClick**
- Esta actividade ten 2 vistas:
 - ♦ Un botón que se pode facer Click (usando a propiedade do XML) e LongClick (capturado a través dun Listener)
 - ♦ Unha etiqueta de texto (en 2 liñas) na que se pode facer Click (hai que habilitalo) e LongClick. Nos dos casos os eventos captúranse con cadanseu Listener.



XML do Layout

- Observar a **Liña 20** onde se habilita o TextView para sexa *clickable*.
- Agora poderíase engadir ao compoñente a propiedade: `android:onClick="onBotonClick"`

```
<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" >

    <Button
        android:id="@+id/boton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onBotonClick"
        android:text="Botón:\n
            onClick Layout\n
            onLongClick Listener" />

    <TextView
        android:id="@+id/text_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:clickable="true"
        android:text="Fai click e
            \nPreme aquí por máis de 1 seg"
        android:textSize="20sp" />

</LinearLayout>
```

Código Java

```
package com.example.u3_03_onlongclick;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnLongClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class U3_03_OnLongClick extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_u3_03__on_long_click);

        Button boton = (Button) findViewById(R.id.boton);
        boton.setOnLongClickListener(new OnLongClickListener() {
            @Override
            public boolean onLongClick(View v) {
                // TODO Auto-generated method stub
                Toast.makeText(v.getContext(), "Premeches 1 seg no botón", Toast.LENGTH_SHORT).show();
                return true;
            }
        });

        TextView tv = (TextView) findViewById(R.id.text_view);
        tv.setOnLongClickListener(new OnLongClickListener() {
            @Override
            public boolean onLongClick(View v) {
                // TODO Auto-generated method stub
                Toast.makeText(v.getContext(), "Premeches 1 seg na etiqueta de texto", Toast.LENGTH_SHORT).show();
                return true;
            }
        });

        tv.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Toast.makeText(v.getContext(), "Fixeches click na etiqueta de texto", Toast.LENGTH_SHORT).show();
            }
        });

        public void onBotonClick(View v) {
            Toast.makeText(this, "Fixeches Click no botón", Toast.LENGTH_SHORT).show();
        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is present.
            getMenuInflater().inflate(R.menu.u3_03__on_long_click, menu);
            return true;
        }

    }
}
```

• Liñas 21-28:

- ◆ Cando se faga un LongClick en *boton* este executará o método onLonClick() que devolverá true se se chegou a consumir o evento.
- ◆ Probar a cambiar a liña 26 a **return false** e executar a aplicación. Que pasa?

- **Liñas 33-40:**

- ♦ Acontece o mesmo que no caso anterior so que para a etiqueta de texto.

- **Liñas 42-49:**

- ♦ Se na etiqueta sucede o evento Click executase chámase ao Listener correspondente da etiqueta.

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2013).