

Añadir comportamiento a los widgets: *EVENTOS*

Formas distintas de Implementar Eventos sobre un Control:
Proyecto:07_ui_one

Eventos:

Diferentes Formas de implementación

- Cada control puede responder a los eventos que el usuario realiza sobre él. Para que la actividad sea informada sobre él se puede hacer de estas tres formas
 - 1.-Definir el event listener y registrarlo para el widget (cada clase que depende de View tiene definidas una interfaces para los eventos que puede tratar)
 - 2.- Incorporando una propiedad en IU que invoque al método que trata el evento.
 - 3.- Sobreescibir el evento de la clase View al extender sobre ella. Este caso se trata cuando creamos nuestra propias clases que implementan la Interfaz que nos interesa

Probar en AS el proyecto:07_ui_one

Diferentes formas de implementar los Eventos

1.-Definir el event listener y registrarlo para el widget.
Cada clase que depende de View tiene definidas una interfaces para los eventos que puede tratar.

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    ctx=this;  
    setContentView(R.layout.main);  
  
    Button btn=(Button)this.findViewById(R.id.cmdTres);  
  
    btn.setOnClickListener(new OnClickListener() {  
        public void onClick(View v) {  
            Toast.makeText(ctx, "Pulsado boton Tres",Toast.LENGTH_SHORT).show();  
        }  
    });
```

2.- Incorporando una propiedad en IU que invoque al método que trata el evento.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:orientation="vertical" >  
  
    <Button  
        android:id="@+id/cmdDos"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="Dos" android:onClick="cmdDos_click"/>
```

Diferente formas de implementar los Eventos

3.-Sobreescribir el evento de la clase View al extender sobre ella. Este caso se trata cuando creamos nuestra propias clases que implementan la Interfaz que nos interesa.

```
public class miButton extends Button implements View.OnClickListener {
    Context ctx=null;

    public miButton(Context context) {
        super(context);
        ctx=context;
        this.setOnClickListener(this); //recoger evento
    }

    //cuando se cree desde un recurso XML
    public miButton(Context context, AttributeSet attr){
        super(context,attr);
        ctx=context;
        this.setOnClickListener(this);
    }

    //cuando se cree desde un recurso XML
    public miButton(Context context, AttributeSet attr, int defaultStyles){
        super(context, attr, defaultStyles);
        ctx=context;
        this.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Toast.makeText(ctx, "Pulsado mi botón", Toast.LENGTH_SHORT).show();
    }
}
```

Ejemplo: diferentes formas de implementar un evento

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

    <com.csjp.curso.miButton
        android:id="@+id/cmdUno"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Uno" />

    <Button
        android:id="@+id/cmdDos"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Dos"
        android:onClick="cmdDos_click"/>

    <Button
        android:id="@+id/cmdTres"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tres" />

</LinearLayout>
```

```
public class MainActivity extends Activity {

    Context ctx=null;

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        ctx=this;

        setContentView(R.layout.main);

        Button btn=(Button)this.findViewById(R.id.cmdTres);

        btn.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {

                Toast.makeText(ctx, "Pulsado boton Tres",
                Toast.LENGTH_SHORT).show();

            }

        });

        public void cmdDos_click(View v){

            Toast.makeText(this, "Pulsado boton Dos",
            Toast.LENGTH_SHORT).show();

        }

    }

}
```

Ejemplo: diferentes formas de implementar un evento

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

    <com.csjp.curso.miButton
        android:id="@+id/cmdUno"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Uno" />

    <Button
        android:id="@+id/cmdDos"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Dos" android:onClick="cmdDos_click"/>

    <Button
        android:id="@+id/cmdTres"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tres" />

</LinearLayout>
```

```
public class miButton extends Button implements
View.OnClickListener {
    Context ctx=null;

    public miButton(Context context) {
        super(context);
        ctx=context;
        this.setOnClickListener(this);//recoger evento
    }

    //cuando se cree desde un recurso XML
    public miButton(Context context, AttributeSet attr){
        super(context,attr);
        ctx=context;
        this.setOnClickListener(this);
    }

    //cuando se cree desde un recurso XML
    public miButton(Context context, AttributeSet attr,
int defaultStyles){

        super(context, attr, defaultStyles);
        ctx=context;
        this.setOnClickListener(this);
    }

    @Override
    public void onClick(View v){
        Toast.makeText(ctx, "Pulsado mi botón",
Toast.LENGTH_SHORT).show();
    }
}
```

Tostadas y Diálogos

- **Tostadas**

```
protected void showToast() {  
    Context context = getApplicationContext();  
    CharSequence text = getResources().getString(R.string.noNameMsg);  
    int duration = Toast.LENGTH_SHORT;  
    Toast toast = Toast.makeText(context, text, duration);  
    toast.show();  
}
```

Version reducida: `Toast.makeText(this, "el texto", Toast.LENGTH_SHORT).show();`

-

- **Diálogos:**

```
protected void showAlert() {  
    CharSequence text = getResources().getString(R.string.noNameMsg);  
    AlertDialog.Builder alert = new AlertDialog.Builder(this);  
    alert.setMessage(text);  
    alert.setPositiveButton(android.R.string.ok, null);  
    alert.show();  
}
```

Version reducida:

```
new AlertDialog.Builder(this).setMessage("el  
texto").setPositiveButton(android.R.string.ok, null).show();
```