

- Botones
- Campos de texto
- Listeners

- La clase Button, hereda de View.
- Proporciona un botón con un texto.
- Se le puede asociar un listener para el evento click.
- Puede definirse mediante un archivo xml de layout o por código.

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button" />
```

- Volvemos a utilizar el método “findViewById()”
- Casteamos a Button

```
Button boton = (Button) findViewById(R.id.button1);
```



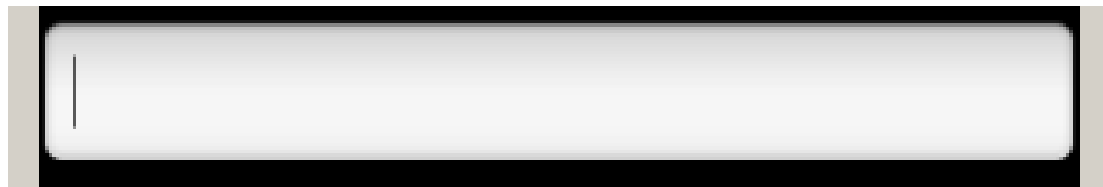


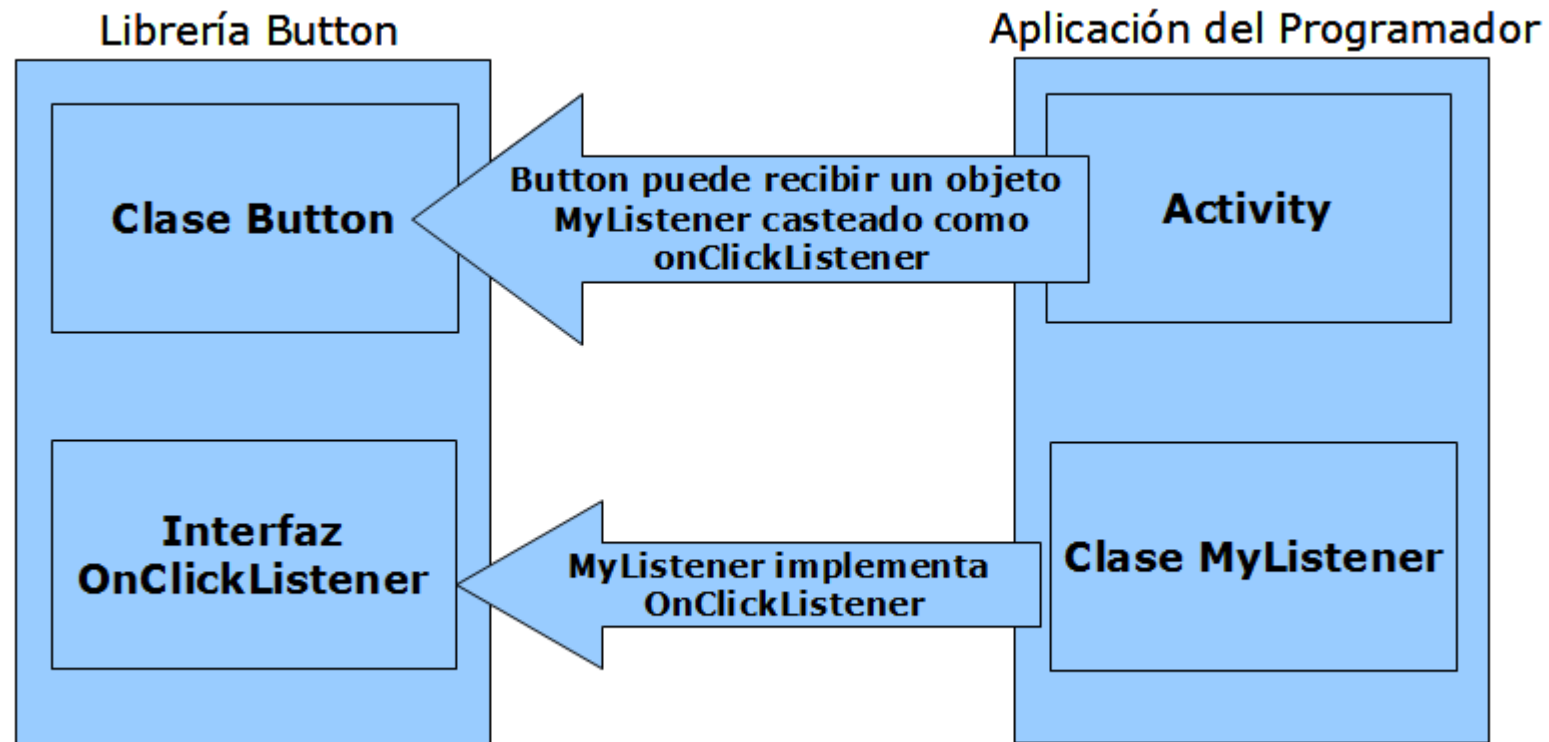
- La clase EditText hereda de View.
- Permite colocar un campo de texto en pantalla.
- Puede definirse mediante un archivo xml de layout o por código.

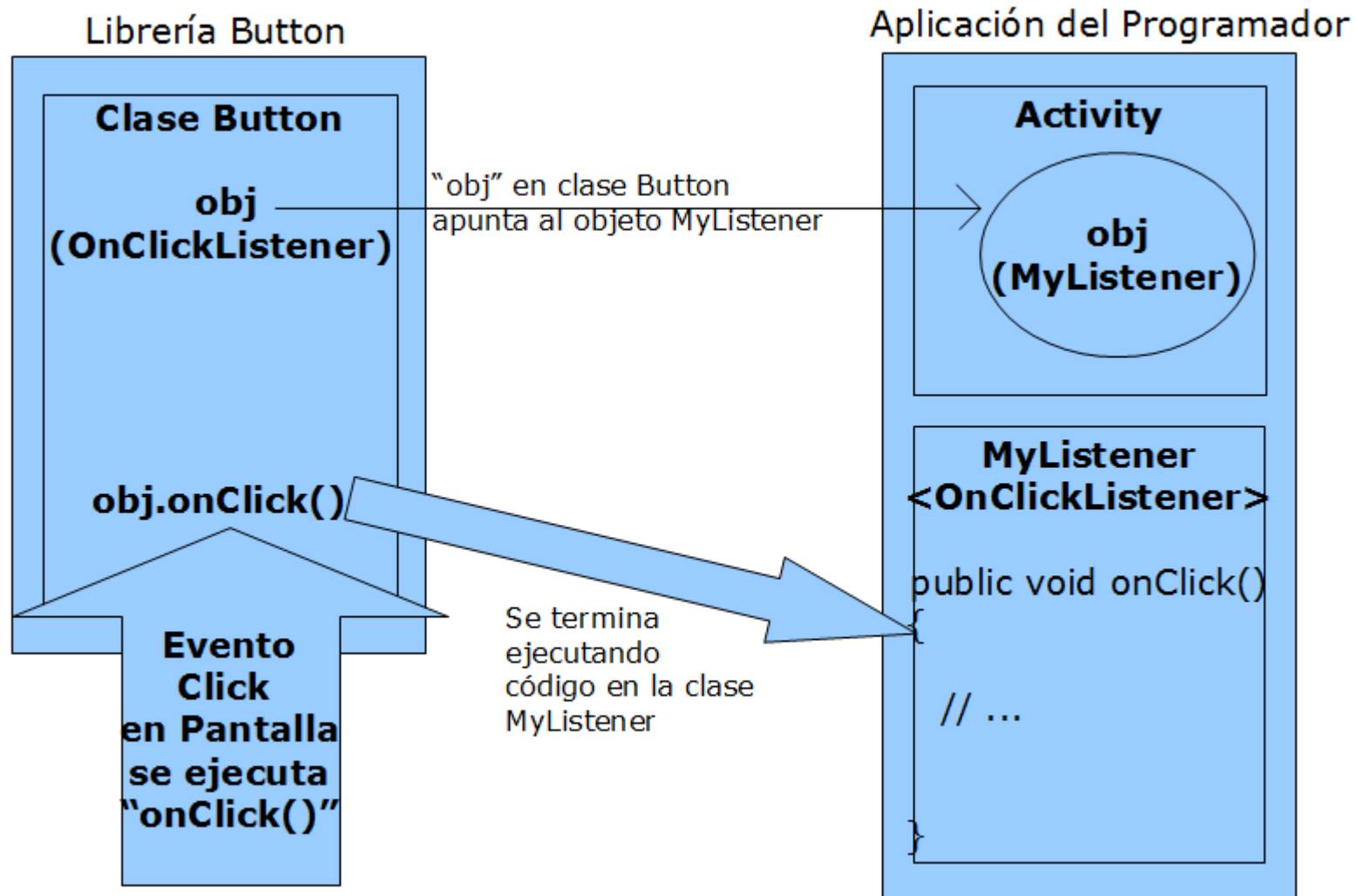
```
<EditText  
    android:id="@+id/editText1"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content" >
```

- Volvemos a utilizar el método "findViewById()"
- Casteamos a EditText
- Obtenemos el texto ingresado mediante "getText()"

```
EditText texto = (EditText) findViewById(R.id.editText1);  
String strTxt = texto.getText().toString();
```







- A los objetos que poseen evento de click, se les puede setear un Listener que implemente la interfaz "**onClickListener**"
- Se setea el listener mediante el método "**setOnClickListener()**"
- La interfaz posee un método llamado **onClick()**,
- **onClick()** se llama desde el objeto que posee el listener, cuando se hace click sobre éste.



```
MyListener listener = new MyListener();  
  
boton.setOnClickListener(listener);
```

```
public class MyListener implements OnClickListener{  
    @Override  
    public void onClick(View v) {  
        //...  
    }  
}
```

- También puede implementarse el listener como una clase anónima.

```
boton.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // ...
    }
});
```

- El método "**onClick()**" posee como parámetro el objeto al que se le hizo click, casteado a su clase padre (View).
- El mismo objeto listener puede ser asignado a más de un objeto View, (p. ej. varios botones).
- Para diferenciar cual objeto produjo el evento, se utiliza el id del objeto.

```
if (v.getId() == R.id.button1) {
    // ...
}
```