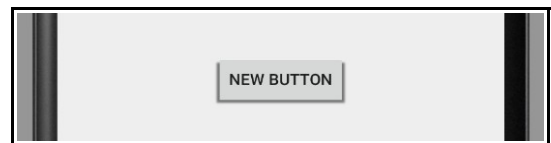


Button

La clase "Button" como todas las clases que representan un objeto que va en pantalla, hereda de la clase View. Este tipo de objeto nos permitirá colocar en pantalla un botón que contiene un texto y que el usuario puede presionar, el programador podrá capturar el evento de "botón presionado" para realizar alguna acción.

Para su definición en un archivo xml de layout, se utiliza el tag "Button", con los atributos que ya conocemos del TextView : id,layout_width,layout_height y text.

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



Para obtener en nuestra clase Activity una referencia de este objeto, lo haremos del mismo modo que se explicó anteriormente, mediante el método "findViewById()"

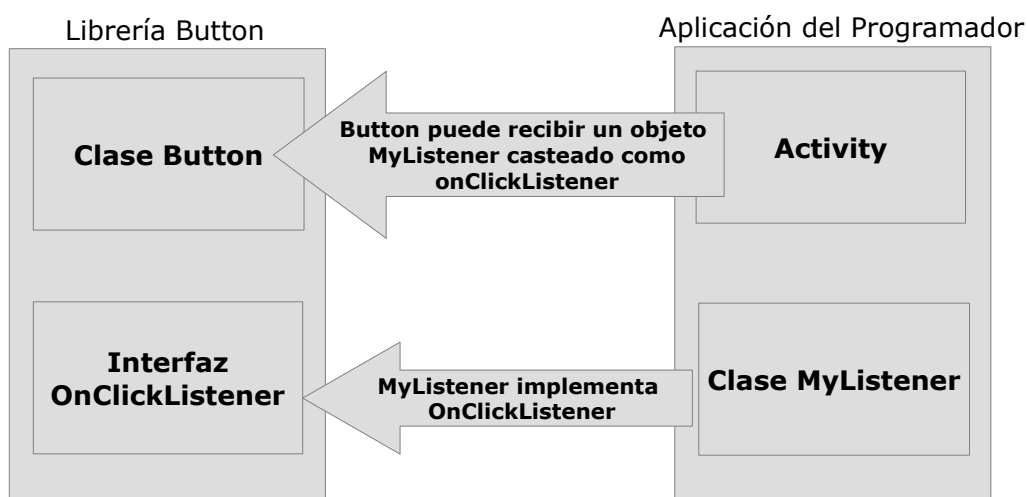
```
Button boton = (Button) findViewById(R.id.button1);
```

A continuación veremos como asociar una clase que "escuche" los eventos de este botón.

Listeners

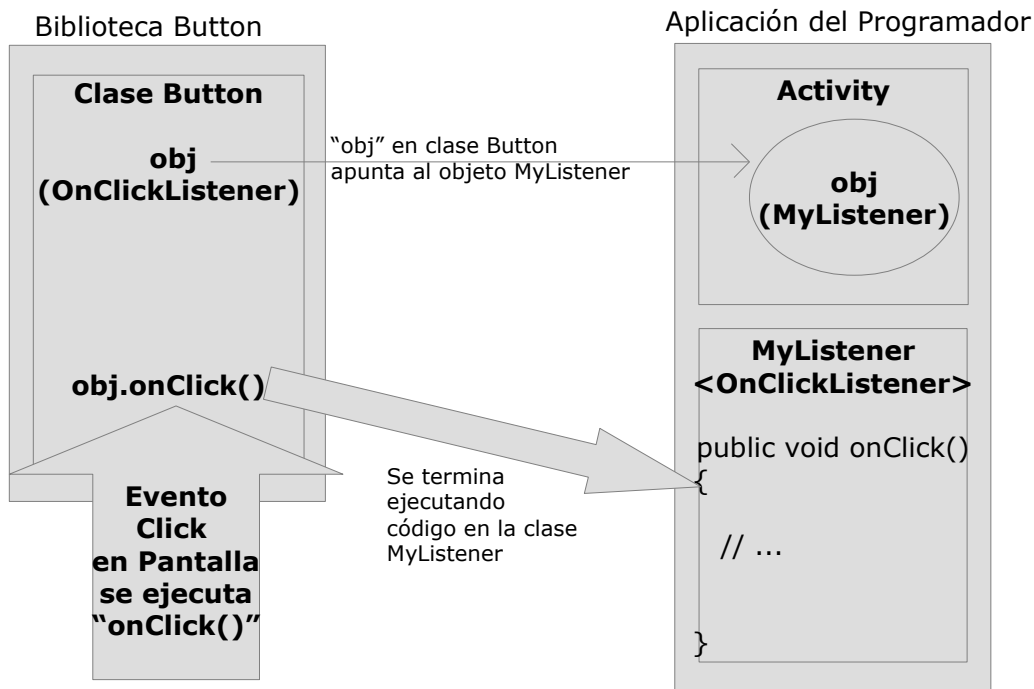
Un "Listener" es un objeto que "escucha" los eventos que otro objeto produce. En este caso, ese otro objeto será nuestro botón.

El Listener es un objeto de una clase que implementa una interfaz que el objeto Button conoce; esto es así debido a que de este modo podremos pasar una referencia de nuestro objeto Listener (que es de una clase que la clase Button no conoce) a el objeto Button, ya que esta clase sí conoce la interfaz que el Listener implementa:



De este modo, la Clase Button, que internamente controla los eventos de Touch que se producen sobre la pantalla, y detecta que se hizo "click" sobre el objeto botón, ejecutará un método del objeto Listener del cual recibió una referencia.

¿Cómo sabe la clase Button qué métodos tiene nuestra clase MyListener? la respuesta es que no lo sabe; pero como la clase MyListener implementa una interfaz que la clase Button sí conoce: `OnClickListener`, y esta interfaz define el método "`onClick()`", cuando se produzca un evento sobre el botón, la clase Button ejecutará el método "`onClick()`" del objeto MyListener pasado como referencia tratado como del tipo `OnClickListener`:



Como puede verse en el diagrama, la clase Button tiene una referencia de nuestro objeto del tipo `MyListener`, pero casteado al tipo "`OnClickListener`" por lo que el único método que la clase Button le puede ejecutar a este objeto es "`onClick()`" y lo hará cuando se produzca un click sobre el botón en pantalla.

Como la clase `MyListener` implementa la interfaz `OnClickListener`, esta obligada a definir el método "`onClick()`" que la clase Button llamará. De modo que el código que queremos que se ejecute cuando se produce un click sobre el botón en pantalla, deberá escribirse dentro del método "`onClick()`" de nuestra clase `MyListener`.

Veamos ahora el código Java que representa el diagrama:

La clase "`MyListener`", implementando la interfaz de la biblioteca del boton "`OnClickListener`":

```

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

```

En la activity, creamos un objeto de nuestra clase MyListener, y obtenemos una referencia del boton:

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

Luego pasamos una referencia de nuestro objeto "listener" al objeto "boton" para que éste le pueda ejecutar el método "onClick()" cuando se produce un evento sobre el mismo:

```
boton.setOnClickListener((OnClickListener) listener);
```

Como puede observarse esto se hace mediante el método "setOnClickListener()" de la clase Button, el cual recibe nuestro objeto "listener" casteado al tipo "OnClickListener" (que es el único tipo de dato que la clase Button conoce).

A partir de ese momento, cuando se produzca un click sobre el objeto Button que está en pantalla, se ejecutará el método "onClick()" de nuestra clase "MyListener".

NOTA: Es muy común implementar la interfaz "OnClickListener" directamente en la Activity, de modo que cuando setemos el listener al boton, le pasamos "this":

```
boton.setOnClickListener(this);
```

De este modo deberemos implementar en la clase que hereda de activity, el método onClick.