|  |
| --- |
| http://www.becas.sep.gob.mx/images/logo.png  TECNOLÓGICO NACIONAL DE MÉXICO  Instituto Tecnológico de Chihuahua II |
| Práctica A |
| Resumen de práctica[A]. |
|  |
|  |
| 17/04/2016  Unidad 2  Carmen Jazmin Garcia Medina  12550465  Programación de Dispositivos Móviles Plataforma II |

Contenido

[Introducción 2](#_Toc448691575)

[Desarrollo 2](#_Toc448691576)

[Conclusión. 5](#_Toc448691577)

[Link Github 5](#_Toc448691578)

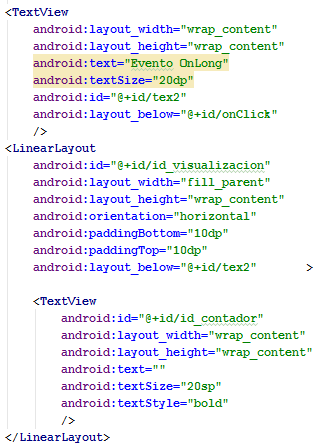
# Introducción

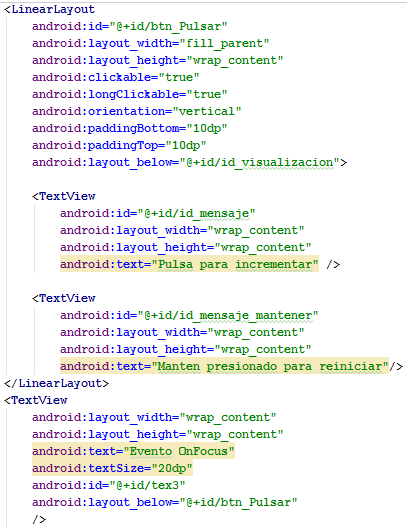
Construya una nueva aplicación llamada CuentaClicks. La aplicación mostrará un botón por cada evento de usuario(View.OnClickListener, View.OnLongClickListener, View.OnFocusChangeListener, View.OnKeyListener, View.OnTouchListener) y un contador que inicialmente tendrá el valor 0, cada vez que se pulse el botón se incrementará el contador, cada botón debe de tener un evento de usuario y un contador por botón que debe estar inicializado, cada vez que se presione un botón con el evento completo el contador debe de aumentar en uno, estos eventos deben de estar creados desde la parte lógica de la aplicación.

# Desarrollo

Paso 1.-

Agregamos los componentes del contenido visual que usaremos en nuestra aplicación.

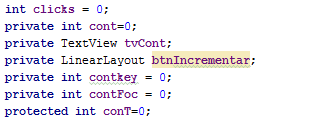






Paso 2.-

En nuestra actividad declaramos las variables que usaremos en nuestra aplicación.



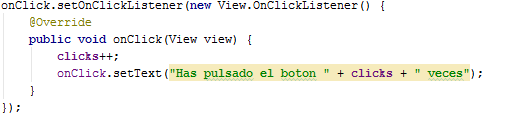
Paso 3.-

El primer evento que implementaremos será el onClick por lo que en el método onCreate declararemos el botón al que le podremos el evento, el texto que tendrá el botón cuando inicie la aplicación y crearemos el método setOnClickListener que accionara cuando presionemos nuestro botón.



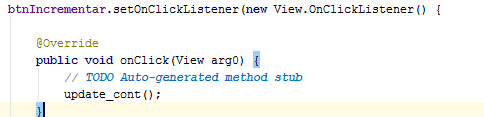
Paso 4.-

En nuestro método setOnClickListener incrementamos el contador cada vez que se precone nuestro botón, y cambiaremos el texto de nuestro botón.



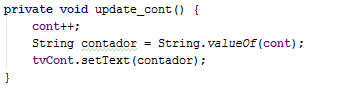
Paso 5.-

Creamos un método setOnClickListener que nos ayudara a implementar el evento onLongClickListener y creamos los listener para darle la funcionalidad de botón al LinearLayout.

.

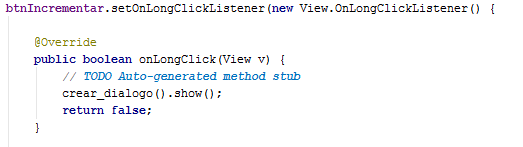
Paso 6.-

Cuando pulsemos con un click corto el LinearLayout, se llama al método update\_cont



Paso 7.-

Lo siguiente es el evento que surgirá cuando mantengamos pulsado un tiempo el botón y llamamos al método crear dialogo.



Paso 8.-

Este método crea una clase AlertDialog.Builder para crear un AlertDialog, un pequeño mensaje que nos ofrecerá un pequeño texto con 2 botones, uno afirmativo y otro de cancelar.

* builder.setCancelable(false)

Esto hace que cuando salga el AlertDialog no podamos salir de el pulsando el botón de retroceso en el terminal.

* builder.setMessage()

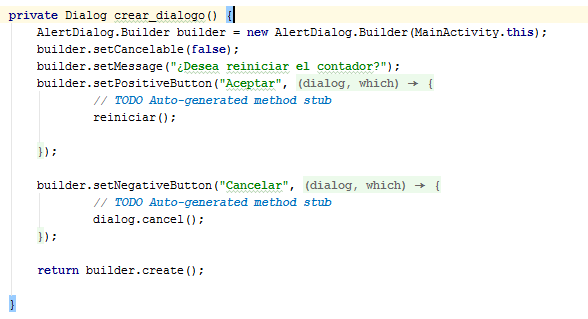
Sirve para establecer el mensaje que aparecerá cuando aparezca el AlertDialog.

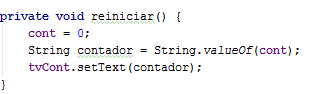
* builder.setPositiveButton()

Le pasamos como argumentos el texto del botón que genera, y un OnClickListener, el cual será encargado de realizar la función que le asignemos. En este caso llama al método reiniciar().

* builder.setNegativeButton()

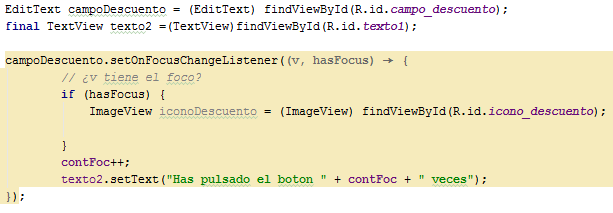
Algo parecido al anterior, le pasamos el texto que contendrá el botón, y el evento OnClickListener para que realice al función especificada, que en este caso, es cancelar el AlertDialog.





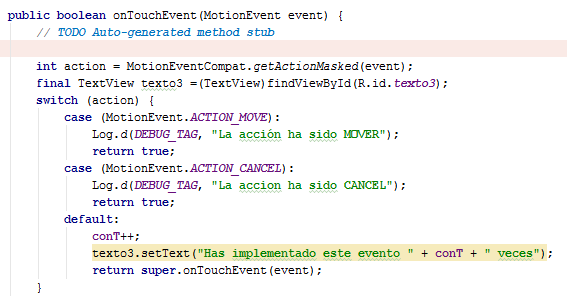
Paso 9.-

Para nuestro evento onFocus ChangeListener añadiremos una escucha anónima OnFocusChangeListener a campo\_descuento. Para ello usa setOnFocusChangeListener(), luego en caso de que ocurra hasFocus incrementamos el contador que usamos para este medo y pasamos el valor junto con una oración a nuestra variable texto2 .



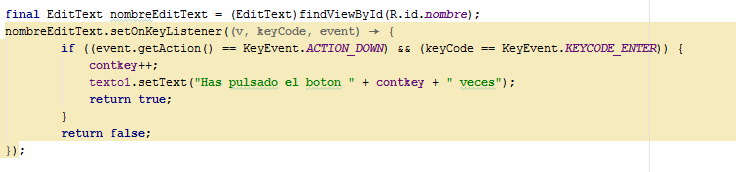
Paso 10.-

Para el evento onTouch creremos un evento, mediante case evaluaremos la acción que se está implementado en nuestra aplicación y cada vez que suceda alguna al igual que en las anteriores incrementamos nuestro contador especial y pasamos el valor a la variable texto3.



Paso 11.-

Por último el evento onKey haremos algo similar declaramos el método, evaluamos si la acción implementada fue dar entre y cada vez que esto suceda incrementaremos el contador y mandaremos el nuevo valor a la variable correspondiente.

Conclusión.

Gracias a esta práctica pudimos ver y probar el funcionamiento de estos métodos, el cual es muy importante e interesante ya que los usaremos bastante el las aplicaciones que desarrollemos.

# Link Github