

# Alvarez\_Valencia\_Gianfranco\_PMDM02 \_Tarea

# Índice

- 1. Ciclo de vida ..... 3
  - a) Apartado..... 3
  - b) Apartado..... 4
  - c) Apartado..... 4
- 2. Depuración ..... 4
  - a) Apartado..... 4
  - b) Apartado..... 5
  - c) Apartado..... 5
  - d) Apartado..... 6
  - e) Apartado..... 6
  - f) Apartado..... 7

## 1. Ciclo de vida

### a) Apartado

Mediante Log.i se consigue mostrar en el logcat los estados por los que pasa las activities.

- Cuando la aplicación se abre por primera vez.

Al abrir la aplicación:

I/Activity A: onCreate

I/Activity A: onStart

I/Activity A: onResume

La aplicación inicia en activity A ya que:

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />

  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

- Cuando pasamos de la activity A a la C.

I/Activity A: onPause

I/Activity C: onCreate

I/Activity C: onStart

I/Activity C: onResume

I/Activity A: onStop

- Cuando se abre el diálogo estando en cualquier activity y después lo cerramos.

Las activities no sufren ningún cambio de estado.

- Cuando volvemos atrás utilizando el botón back (C→A).

I/Activity C: onPause

I/Activity A: onRestart

I/Activity A: onStart

I/Activity A: onResume

I/Activity C: onStop

I/Activity C: onDestroy

- Cuando rotamos el dispositivo. ¿Qué sucede en este caso?

Al rotar se vuelve a generar:

I/Activity A: onPause

I/Activity A: onStop

I/Activity A: onDestroy

I/Activity A: onCreate

I/Activity A: onStart

I/Activity A: onResume

### b) Apartado

Modificar el main en AndroidManifest:

```
<activity android:name=".ActivityB">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER"
    />
    </intent-filter>
</activity>
<activity android:name=".ActivityA" />
<activity android:name=".ActivityC"/>
</application>
```

### c) Apartado

Es un objeto de mensajería que permite intercambiar datos entre aplicaciones o componentes de aplicaciones.

Los hay de dos tipos intents explícitas: se especifica de manera explícita el nombre correspondiente a la clase de la actividad que se quiere iniciar, e intents implícitas:

Será el propio sistema en tiempo de ejecución, decidirá qué actividad es la más adecuada para realizar la acción solicitada sobre los datos suministrados.

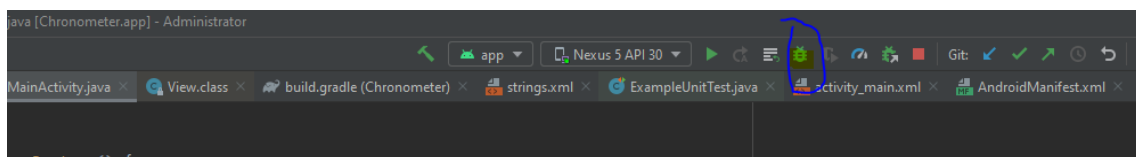
El tipo que se utiliza en esta aplicación es explícito:

```
Intent intent = new Intent(this, ActivityB.class);
startActivity(intent);
```

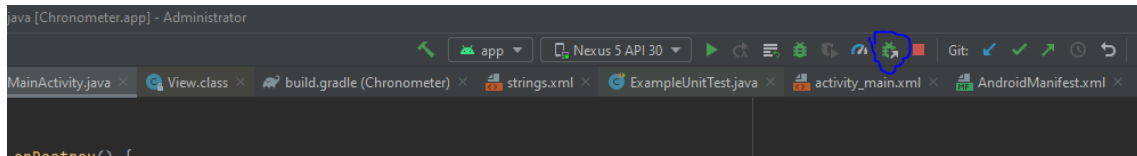
## 2. Depuración

### a) Apartado

Mediante el debug habiendo creado previamente interrupciones:



Mediante Attach debugger:



### b) Apartado

Son interrupciones en la ejecución del programa, sirven para establecer “paradas” de la ejecución del programa para el control del debug.

### c) Apartado



El primer botón sirve para continuar la ejecución del programa. Cmdo Teclado: F9.

El segundo botón sirve para detener la ejecución de la aplicación. Cmdo Teclado CTRL+F12.

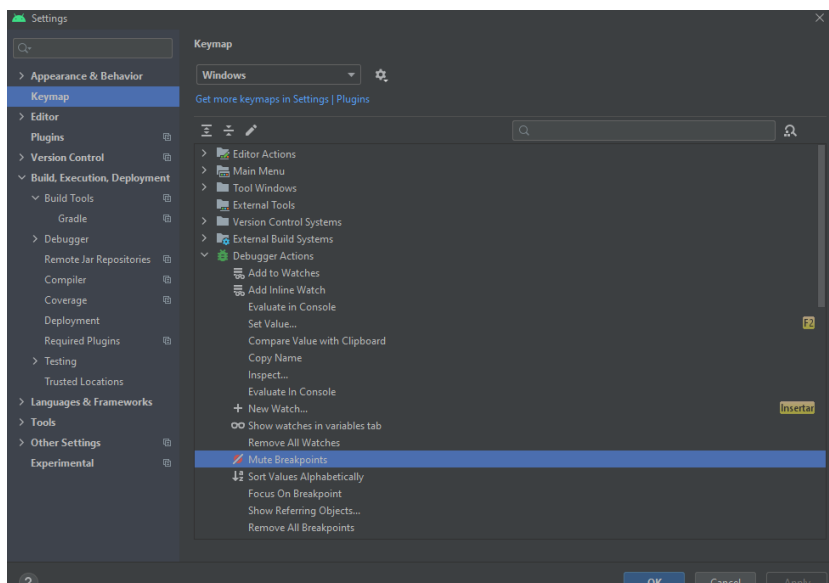
El tercer botón sirve para avanzar a la primera línea dentro de una llamada a un método Cmdo Teclado F7.

El Cuarto botón sirve para avanzar a la siguiente línea del código (sin ingresar a un método). Cmdo Teclado F8.

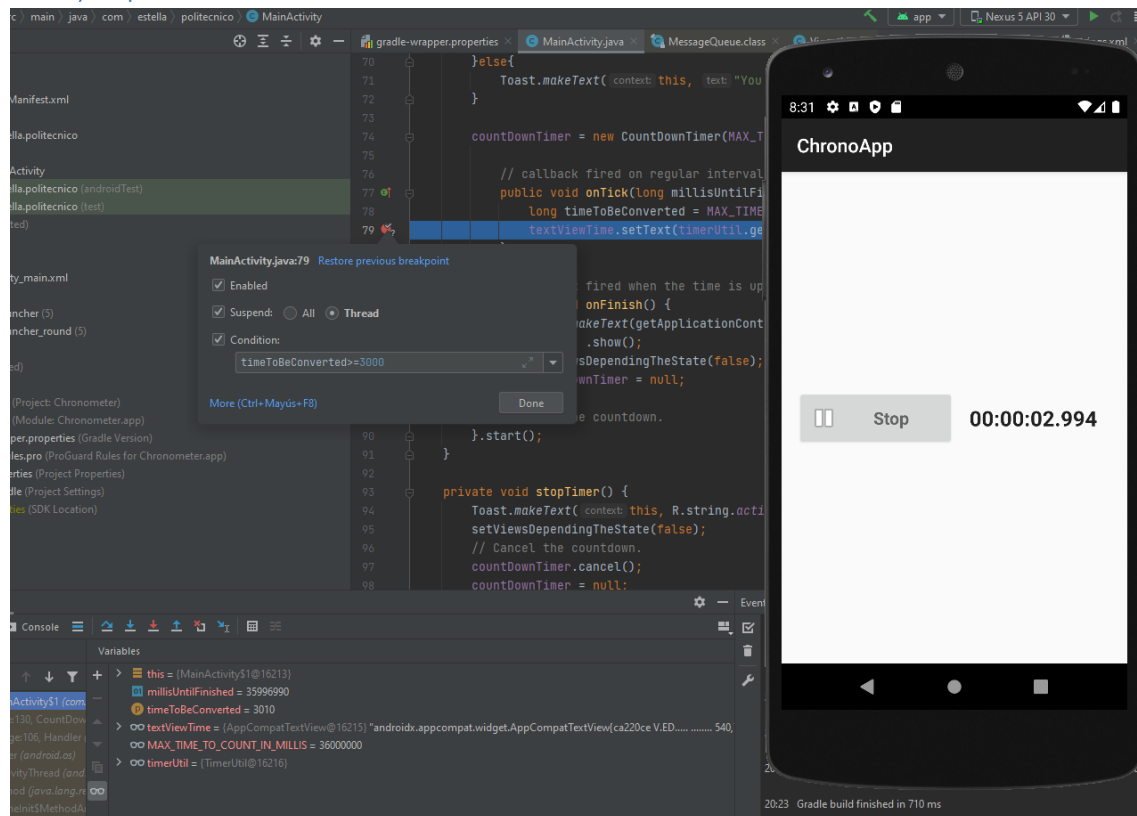
El quinto botón sirve para ver y configurar las interrupciones. Cmdo Teclado CTRL+MASYUS+F8.

El sexto botón sirve para deshabilitar todos los breakpoints.

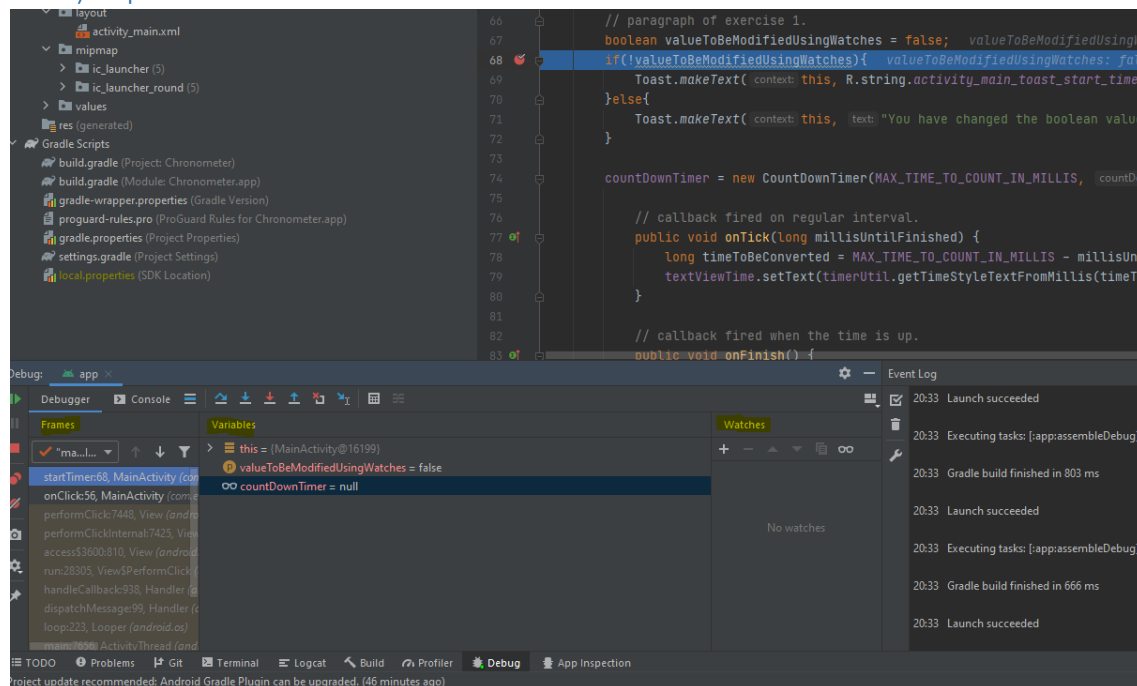
No tiene asignado un atajo de teclado se podría asignar desde settings; keymap:



## d) Apartado



## e) Apartado



## f) Apartado

