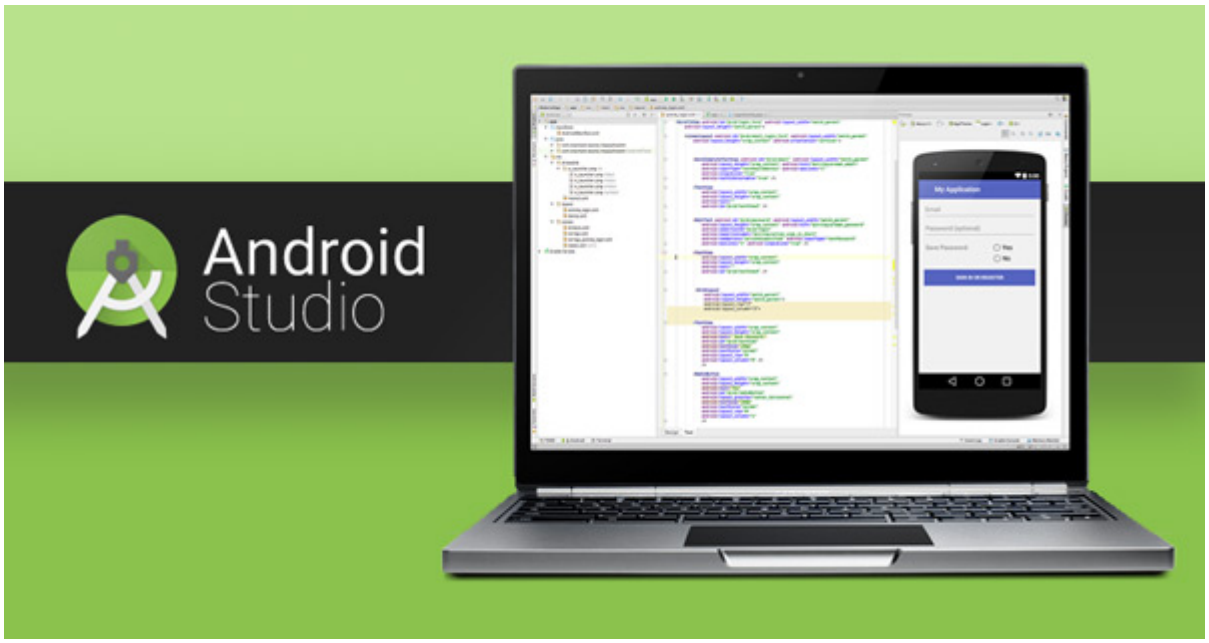
Configurando Android Studio



Introducción

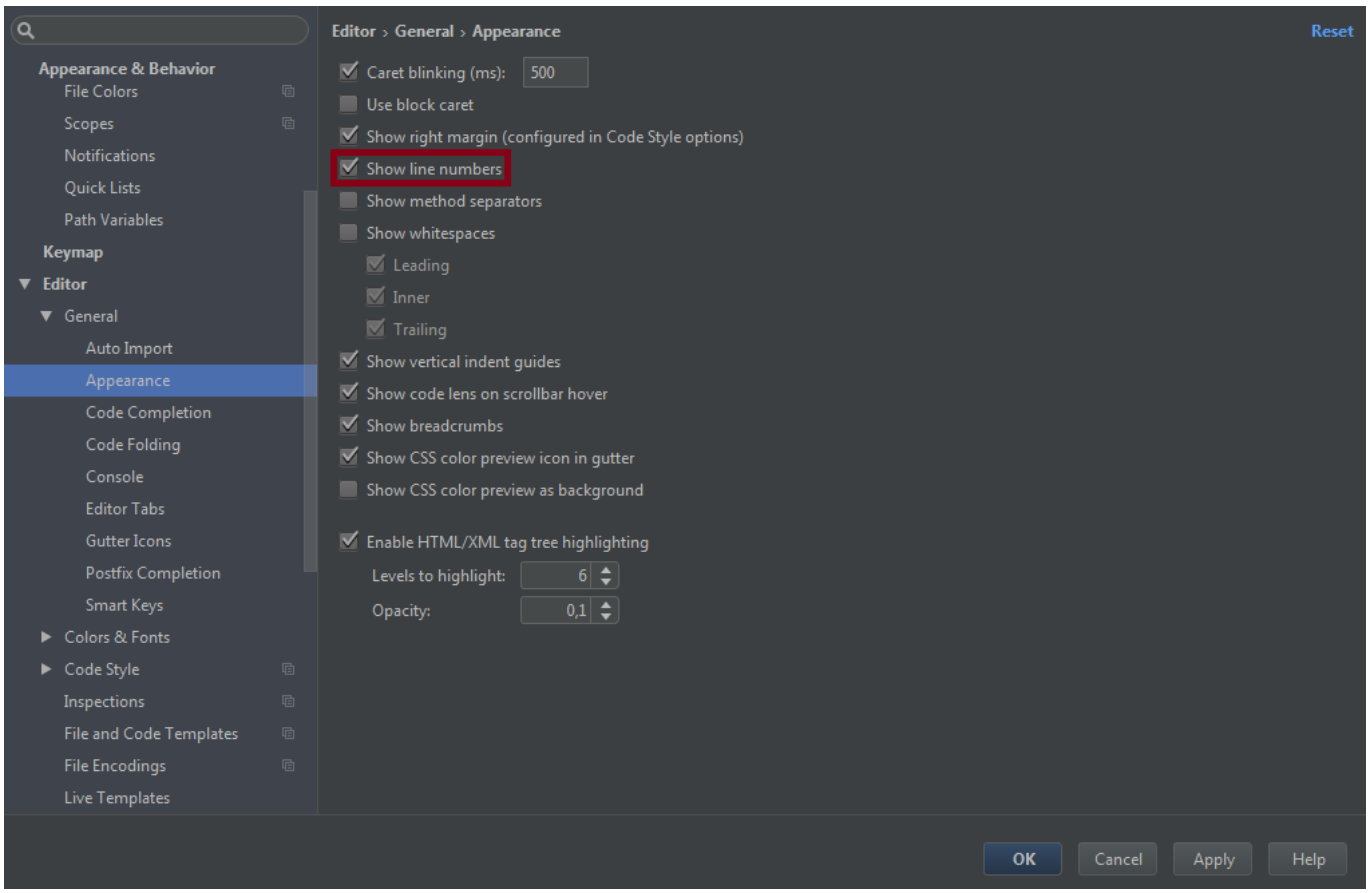
Este es el primer post de una pequeña serie sobre Android Studio. En ellos mostraremos alguna configuraciones,algún plugin, atajos de teclados y «truquillos».

- Número de linea
- Camel humps
- Imports automáticos
- Colores en el logview de Android Studio
- Plugins
    - Plugin ButterKnife Zelezny
    - Plugin Exynap
    - Markdown Navigator

Números de linea

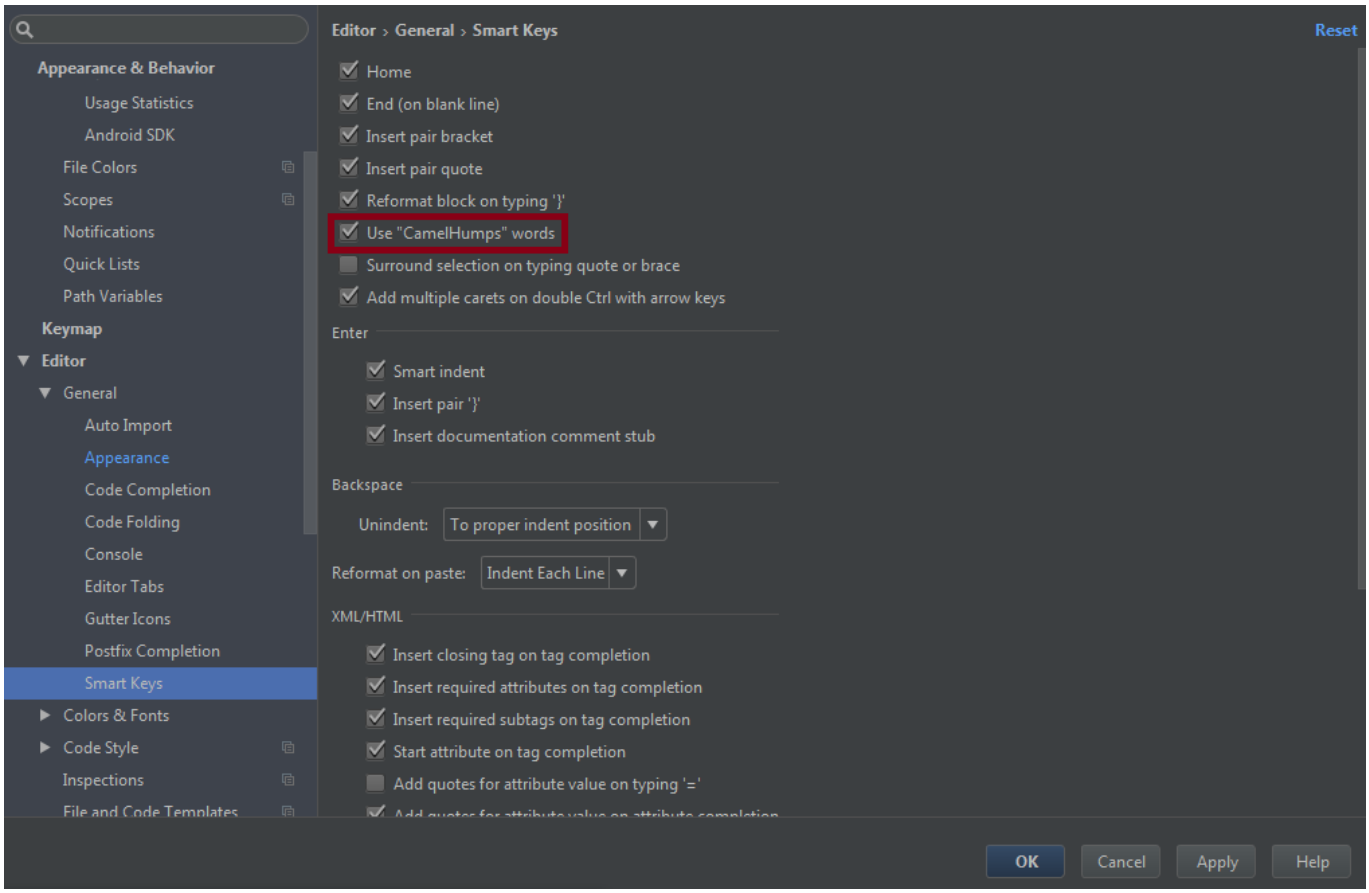
Por alguna razón, que no entiendo del todo, por defecto Android Studio no nos muestra la lineas dentro del editor.

Para habilitar esta funcionalidad deberemos acceder a ***File->Settings ->Editor->General->Appearance*** **y habilitar «*Show line numbers*».**

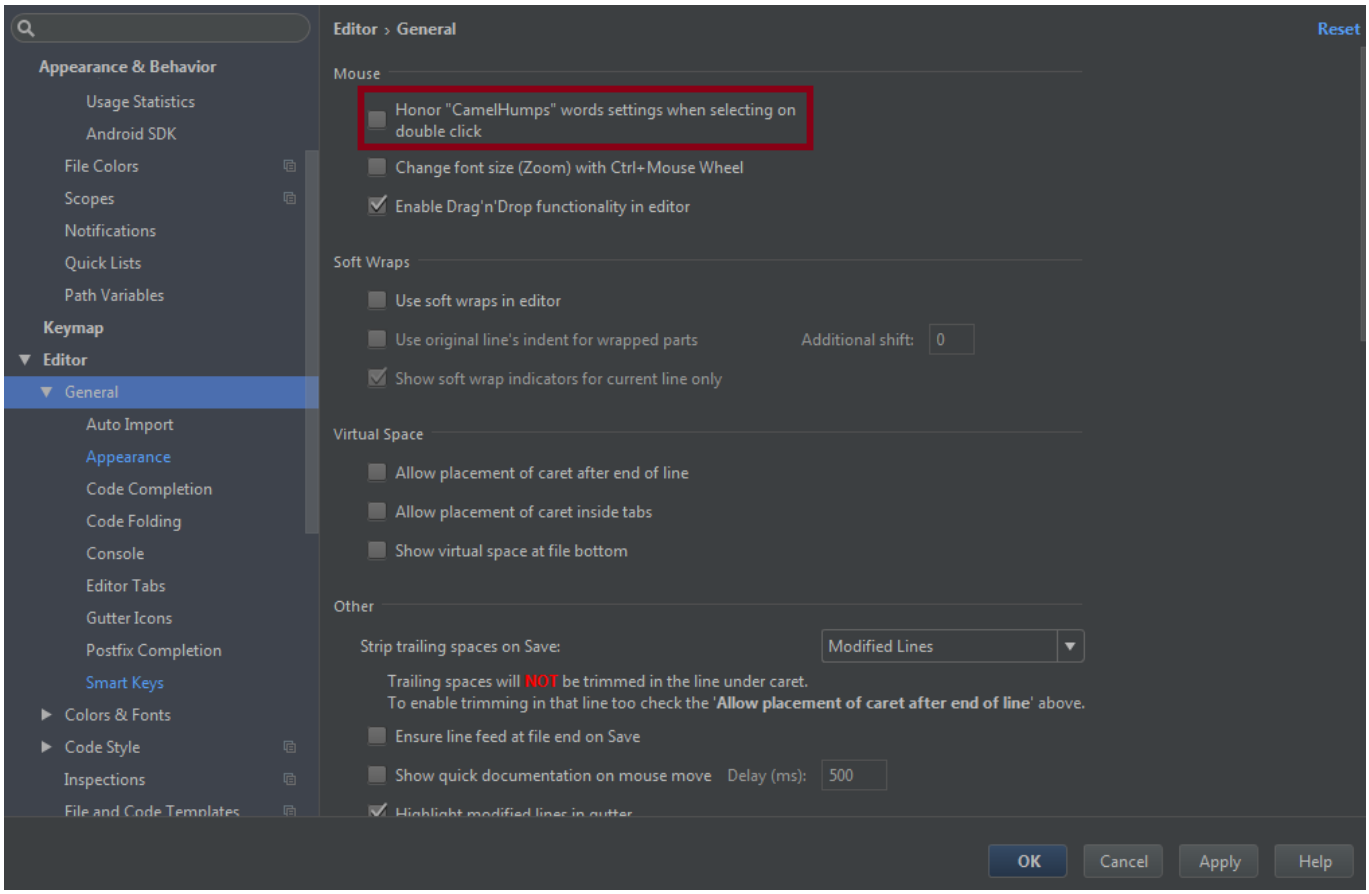


Camel humps

Android Studio no respeta las palabras ‘camel Humps’ cuando se navega a través del código con las teclas Ctrl + Izquierda / Derecha. Para solucionarlo accedemos a ***File->Settings->Editor->General->Smart Keys*** y finalmente *seleccionamos* ***Use ‘Camel Humps’ words***

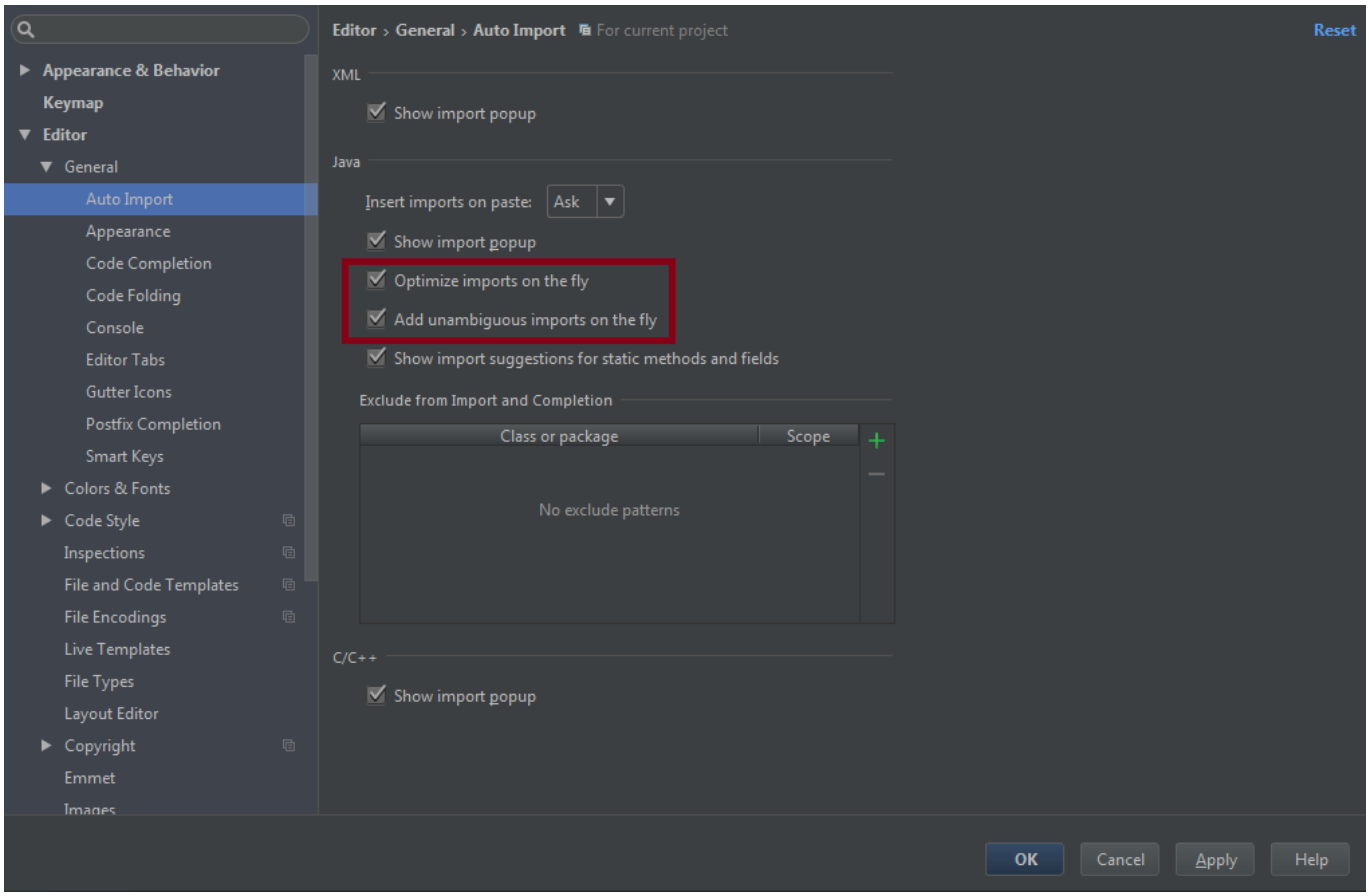


A continuación, lo que nos ocurrirá es que no podremos seleccionar con doble click una palabra. Para solucionarlo debemos acceder a ***File->Settings->Editor->General*** y deseleccionar ***Honor 'Camel Humps' words settings when selecting on double click***



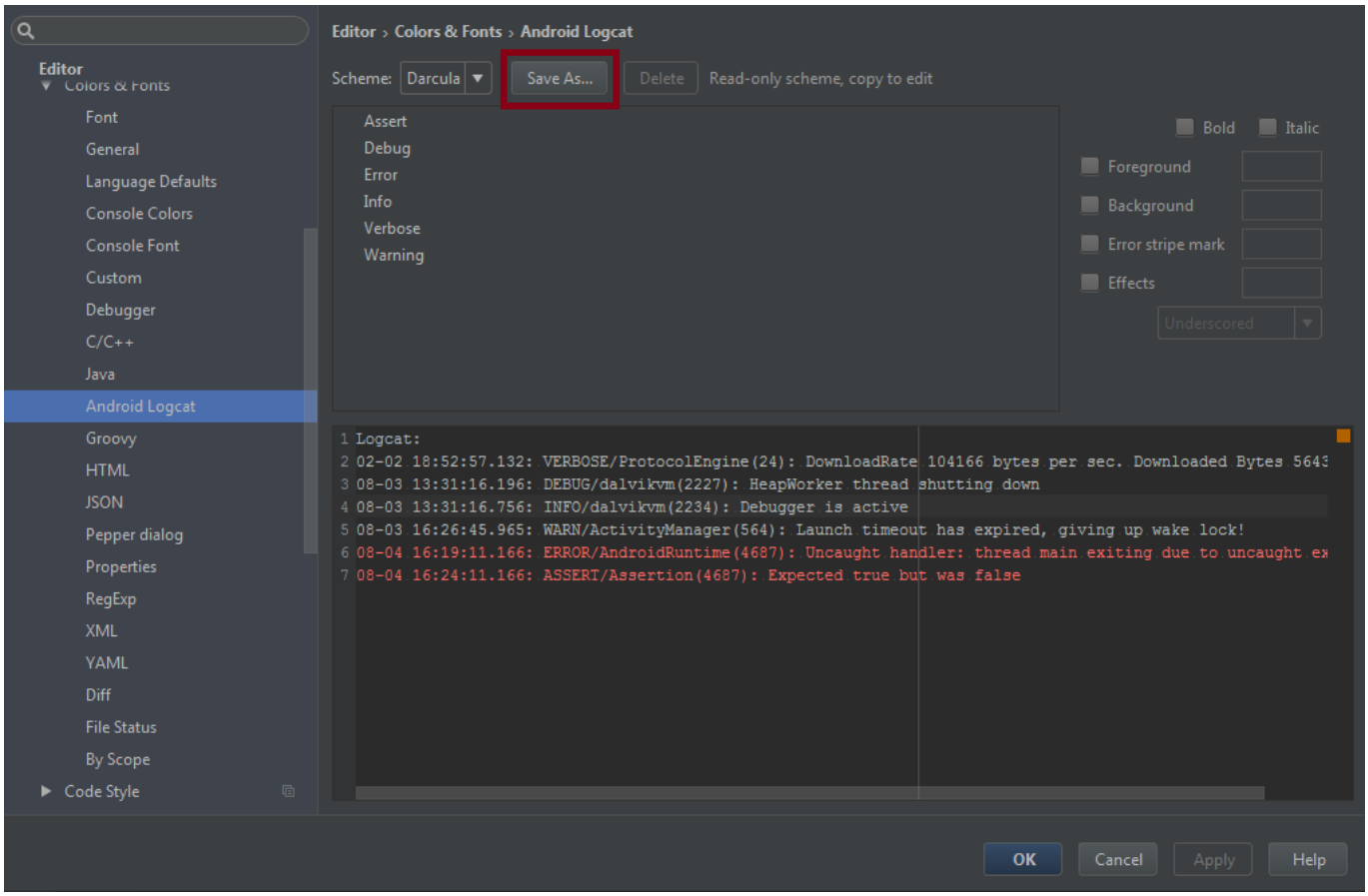
Imports Automáticos

Bien la verdad, es que hoy en día no es necesario saberse todos los packages que debemos usar, pues con una combinación de teclas podemos importarlos. Pero, ¿por que no hacerlo automáticamente? Para activarlo deberemos acceder a ***File->Settings->Editor->General->Auto Import***, seguidamente deberemos seleccionar ***Optimize imports on the fly*** y ***Add unambiguous imports on the fly***



Colores en el logview de Android Studio

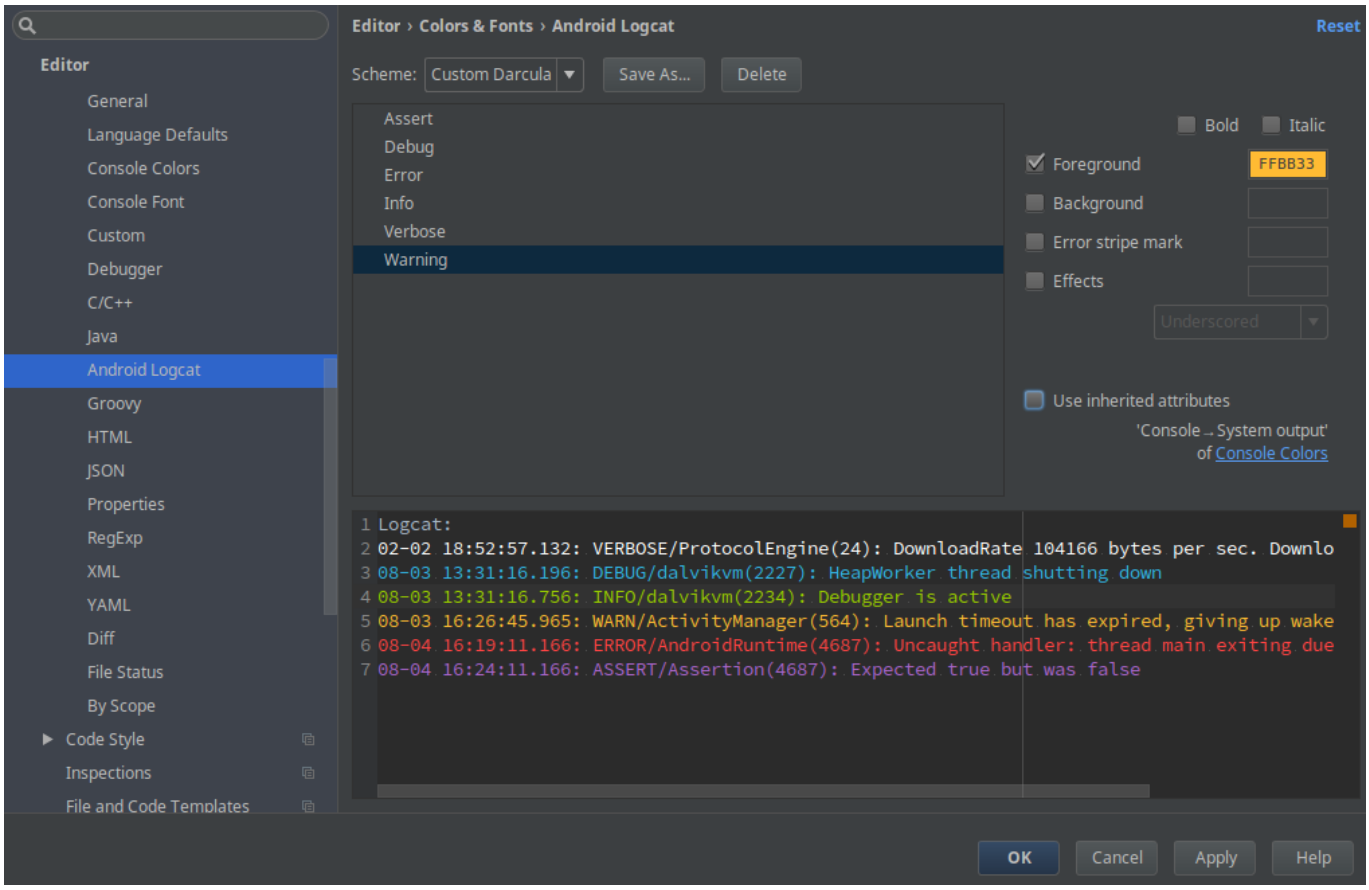
Bien esto es solo una preferencia. En logcat un buen esquema de colores nos ayudará a visualizar mejor la información. Deberemos acceder a ***File|Settings->Editor->Colors & Fonts->Android Logcat*** en este punto debemos hacer clic en ***Save As…*** y crear un nuevo esquema de colores.



Seguidamente por cada unos de los tipos de log (Assert, Debug, Error…) debemos deseleccionar ***‘Use inherited attributes’*** para cada color y aplicar el nuevo.

- Assert: #AA66CC
- Debug: #33B5E5
- Error: #FF4444
- Info: #99CC00
- Verbose: #FFFFFF
- Warning: #FFBB33

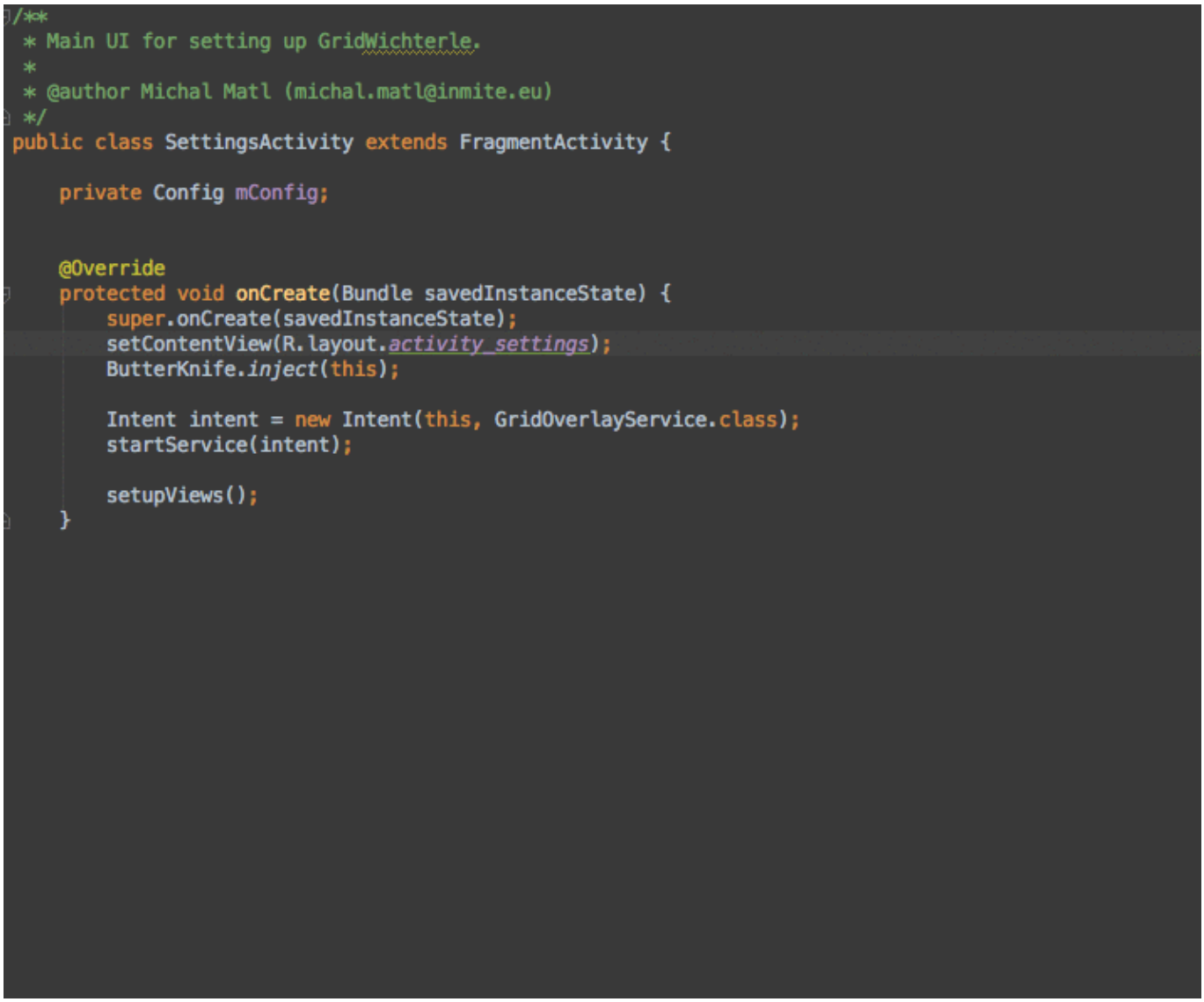
Este es el resultado



Plugins

Plugin ButterKnife Zelezny

De este plugin ya he hablado, la verdad es que nos facilita mucho el uso de Butterknife.Para instalarlo debemos acceder a ***File->settings->Plugin***, seguidamente clicar en ***Browse Repositories*** buscar ***ButterKnife Zelezny*** instalar y reiniciar. A continuación os dejo un gif de que muestra su utilización, sacado del proyecto de GitHub de este plugin. También os animo a que os paséis por la entrada de Butterknife



```
/**
 * Main UI for setting up GridWichterle.
 *
 * @author Michal Matl (michal.matl@inmite.eu)
 */
public class SettingsActivity extends FragmentActivity {

    private Config mConfig;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);
        ButterKnife.inject(this);

        Intent intent = new Intent(this, GridOverlayService.class);
        startService(intent);

        setupViews();
    }
```

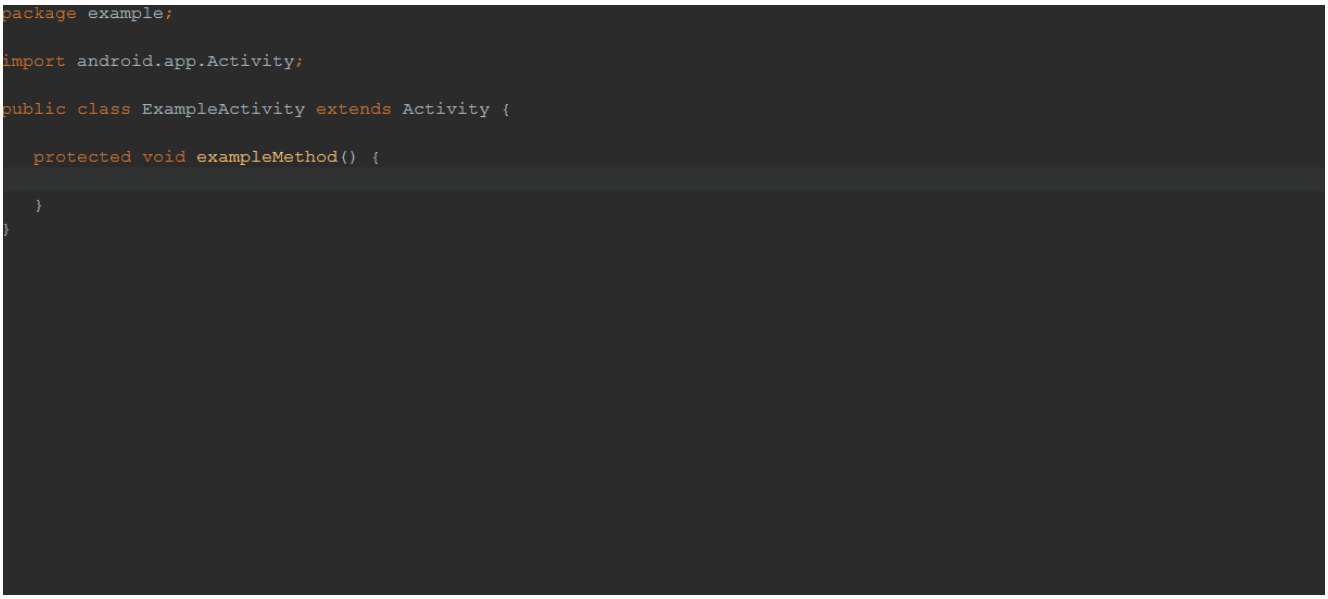
Plugin Exynap

Bien este plugin es espectacular. Nos permite encontrar e implementar el código que necesitemos en un instante. Para instalarlo debemos acceder a ***File->settings->Plugin***, seguidamente clicar en ***Browse Repositories*** buscar ***Exynap*** instalar y reiniciar.

Mediante la combinación de teclado ***Ctrl + Shift + D*** se abrirá una ventana contextual, des de la cual podremos buscar lo que necesitamos. Os dejo algunos gifs de su propia página.

```
package example;

import android.app.Activity;
import android.widget.TextView;

public class ExampleActivity extends Activity {

  protected void exampleMethod() {
    TextView textView = new TextView(this);
    textView
  }
}
```

```
package example;

import android.app.Activity;

public class ExampleActivity extends Activity {

  protected void exampleMethod() {
    int width
  }
}
```

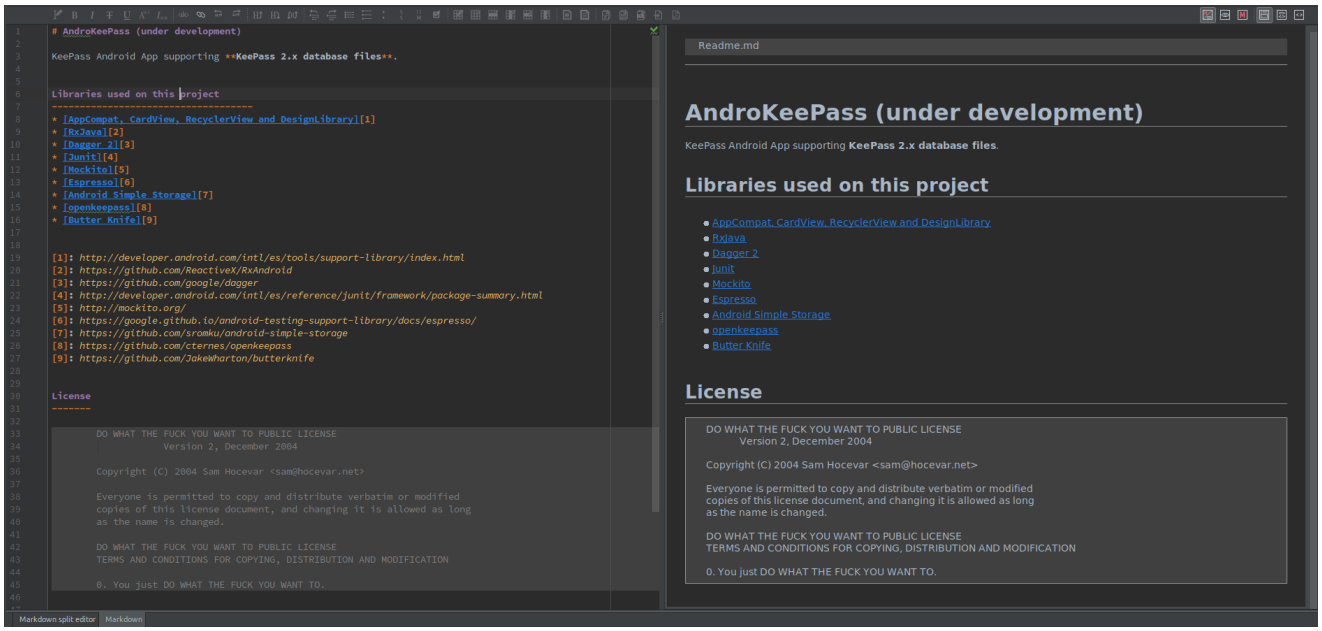


```
package example;

import android.app.Activity;

public class ExampleActivity extends Activity {

  protected void exampleMethod() {

  }
}
```

Markdown Navigator

Este plugin nos facilitara la edición de los archivos de markdown que tengamos en el proyecto. Por ejemplo ficheros de Changelog, Readme… Para instalarlo debemos acceder a ***File->settings->Plugin***, seguidamente clicar en ***Browse Repositories*** buscar ***Markdown Navigator*** instalar y reiniciar.

A continuación cuando abramos cualquier fichero de markdown se abrirá con el editor instalado.



Observaciones

Bien, este post no ha sido de programación propiamente dicha. Ha sido una preconfiguración de Android Studio. La verdad es que estas pequeñas configuraciones y plugins nos ayudan a programar de manera más optima y rápida. En el siguiente post mostraremos atajos de teclado de Android Studio (que son varios y muy útiles).