**MÓDULO SMS**

**Esta guía de creación del módulo fue creada el 16 de marzo de 2018. Puede que con el tiempo algunos pasos sean innecesarios o que la consola de Firebase haya cambiado su aspecto visual debido a actualizaciones, pero en el fondo el proceso es el mismo.**

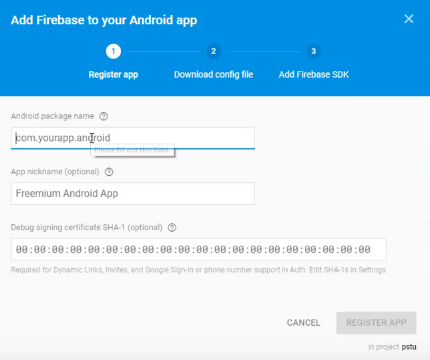
La interfaz del SMS consta de:

* Un campo para ingresar el número celular del usuario
* Un botón para confirmar el número ingresado y al cual se va a enviar el mensaje con un  codigo de verificacion
* Un botón de reenvío en caso de que el usuario no reciba ningun codigo
* Un campo para ingresar el código de verificación
* Un botón de verificar la cuenta con el número y el código enviado a este.

El módulo verificación de números por SMS ya se encuentra desarrollado, pero para implementarlo en cualquier otra aplicación, hay que seguir una serie de pasos que se muestran a continuación.

Una vez creada la aplicación, se procede a integrala a Firebase mediante una cuenta google. Firebase va a ser nuestro servidor y el encargado de recepcionar los números y mandarles un código único para luego verificar la cuenta. El proceso es el siguiente:

1. Abrir la consola de Firebase en un navegador de internet
2. Loguearse con el usuario que va a manejar la aplicación y la base de datos.
3. Crear un nuevo proyecto, darle un nombre y un país de ubicación. Hay que asegurarse que el país escogido sea en donde va a funcionar la aplicación, pues firebase cuenta con respaldo telefónico para muchos países y cada uno tiene un indicativo que será antepuesto al número que se enviará el código de verificación.
4. Aparecerá una ventana donde hay que escoger la plataforma para la cual funcionará la aplicación (iOs, Android, web). Escogemos Android y aparecerá otra ventana donde nos pide llenar varios campos, como se muestra a continuación:

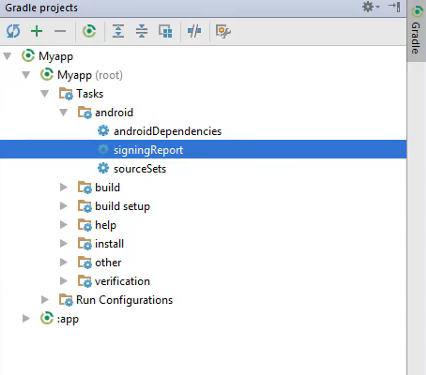


4.1  Abrimos el proyecto en Android Studio y en el archivo **AndroidManifest.xml** aparece un línea con el nombre del **paquete (package)**, el cual es básicamente el nombre que recibió nuestra aplicación al crearla, debemos copiar ese nombre que aparece entre comillas y pegarlo en el campo de **Android package name**

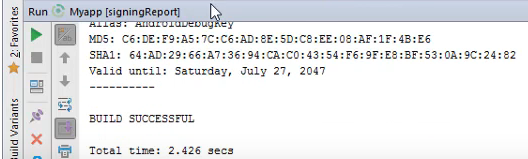
4.2 El campo de **nickname**  es opcional y se puede escoger cualquier nombre para la aplicación

4.3 Para completar el campo de **SHA-1** se sigue los siguientes pasos

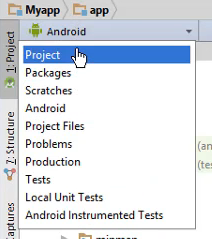
1. Abrimos nuevamente nuestra aplicación y en la parte derecha hay una pestaña llamada **Gradle**, la cual al darle clic se va a desplegar una ventana como se muestra



1. Al dar clic en **signingReport** aparecerá, en la parte inferior de la pantalla, el **SHA-1** que necesitamos. Copiar y pegar exactamente el código que aparezca. Cabe resaltar que es un código único para cada aplicación, el que se muestra en la imagen es solo un ejemplo.



4.4 Al darle clic en continuar, aparecerá una ventana en la cual vamos a descargar el archivo **google-services.json.** Vamos a la ubicación donde se guardó el archivo .json, le damos clic en copiar y Abrimos la aplicación nuevamente y desplegamos la pestaña que queda al lado de “**Project**” ubicado en la parte izquierda de la pantalla y escogemos precisamente **Project**.



4.5  Luego pegar en el apartado que aparece como “**app”** en Android Studio

4.6  Lo que sigue es agregar la SDK de Firebase a nuestra apĺicación. En pantalla aparecen instrucciones para hacerlo. Por último sincronizar el proyecto de nuevo. No deberían presentarse errores, si los hay, es por la versión de firebase que se está utilizando.

1. Volvemos a la consola de Firebase y vamos a **Authentication**, le damos clic en escoger el método de sign in. Va a aparecer todas las formas que tenemos de loguearnos con firebase, escogemos **Phone** y le damos en **Enable y guardar.** Escogemos también **Google** y la activamos

Con lo anterior logramos integrar completamente Firebase a nuestra app, ahora proseguimos con el código. Se abre la aplicación llamada **Firebase\_sms3** y de ella se copian varios fragmentos de código.

1. En **build.gradle(Module:app)** vamos a añadir las siguientes líneas:

compile 'com.google.firebase:firebase-messaging:11.8.0'

compile 'com.google.firebase:firebase-auth:11.8.0'

compile 'com.google.firebase:firebase-core:11.8.0'

implementation 'com.android.support:design:26.1.0'

Se debe tener especial cuidado en la última línea que aparece arriba, pues la versión debe concordar con la versión que aparece en:

implementation 'com.android.support:appcompat-v7:26.1.0'

1. Después que haya sincronizado el proyecto sin problemas, abrimos el apartado de diseño .xml del main\_activity, copiamos todo el contenido del .xml del proyecto base y pegamos en nuestro proyecto actual, excepto la siguiente línea que hace parte del propio proyecto según como lo hayamos nombrado:

tools:context="com.example.pdi.sms\_app.MainActivity"

Puede que haya muchas líneas en rojo (con problemas) pero es porque falta agregar ciertas librerías y dependencias, éstas se agregan de la siguiente manera:

* En la aplicación base (Firebase\_sms3) nos dirigimos a **values** al archivo **colors.xml** y copiamos todo el contenido que aparece en la misma ruta pero de nuestra aplicación
* Hacemos lo mismo con **strings.xml,** exceptuando una línea donde aparece el nombre de la aplicación
* Igualmente con **styles.xml.** Si salen errores es porque falta crear un .xml llamado dimens
* Para crear **dimens.xml,** damos clic derecho sobre **values,** vamos a **new, Values resource file** y le damos el nombre de “dimens”. Luego copiamos nuevamente el contenido de **dimens.xml** de la aplicación base
* Van a aparecer ciertos directorios en rojo, pues hay que incluir unas imágenes para la aplicación, pueden ser cualquiera o simplemente no colocar ninguna
* Luego vamos a **File, Project Structure, Dependencies.** En la parte derecha aparece una flecha verde al lado de scope, vamos a elegir **library dependency** y buscamos entre las librerías que nos aparecen una llamada **com.android.supportdesign,** por último damos OK
* Con lo anterior ya no debería aparecernos errores

1. Abrimos el **MainActivity** de la aplicación base y copiamos todo excepto la primera línea que indica el nombre de la aplicación.
2. Agregamos una nueva actividad, la cual va a ser la que accede después de verificar el número. En la aplicación base se llama **HalfFinder**, pero puede tomar cualquier nombre
3. Listo, con eso acabamos de implementar la actividad de verificacion de número.