Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc448254544)

[1.1 Autores 2](#_Toc448254545)

[1.2 Planificación 2](#_Toc448254546)

[1.3 Entrega 2](#_Toc448254547)

[2. Requisitos del prototipo a implementar 3](#_Toc448254548)

[2.1 Requisitos funcionales 3](#_Toc448254549)

[2.2 Otros requisitos 3](#_Toc448254550)

[3. Criterios de comparación en la implementación 4](#_Toc448254551)

[3.1 Criterio 1: Nombre del criterio 4](#_Toc448254552)

[3.2 Criterio 2: Nombre del criterio 4](#_Toc448254553)

[3.N Criterio N: Nombre del criterio 4](#_Toc448254554)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A 5](#_Toc448254555)

[4.1 Documentación de diseño 5](#_Toc448254556)

[4.2 Documentación de construcción 5](#_Toc448254557)

[4.3 Documentación de pruebas 5](#_Toc448254558)

[4.4 Documentación de instalación 5](#_Toc448254559)

[4.5 Manual de usuario 5](#_Toc448254560)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 6](#_Toc448254561)

[5.1 Documentación de diseño 6](#_Toc448254562)

[5.2 Documentación de construcción 6](#_Toc448254563)

[5.3 Documentación de pruebas 6](#_Toc448254564)

[5.4 Documentación de instalación 6](#_Toc448254565)

[5.5 Manual de usuario 6](#_Toc448254566)

[6. Comparación de las dos implementaciones 7](#_Toc448254567)

[6.1 Evaluación de los criterios en la implementación usando la tecnología A 7](#_Toc448254568)

[6.2 Evaluación de los criterios en la implementación usando la tecnología B 7](#_Toc448254569)

[7. Comparación de la implementación de las tecnologías 8](#_Toc448254570)

[8. Conclusiones 10](#_Toc448254571)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

En este apartado se debe indicar el número de grupo y los nombres de los autores, poniendo en primer lugar al coordinador del grupo.

## 1.2 Planificación

En este apartado se debe incluir un enlace (URL) compartido a la planificación del trabajo utilizando una herramienta online de diagramación Gantt (por ejemplo, GanttPro, versión gratuita).

Hay que tener en cuenta que cada participante del grupo debe tener asignadas tareas que sumen al menos 45 horas. El peso de este trabajo en la calificación total de la asignatura es de un 30%, por tanto requiere de una dedicación de 45 horas del total de 150 horas de la asignatura.

## 1.3 Entrega

En este apartado debe incluirse un enlace (URL) a un repositorio en GitHub o en BitBucket creado para el trabajo.

En dicho repositorio debe encontrarse, al menos los siguientes archivos en la rama máster:

* Informe del trabajo: con el nombre TG3\_final.docx
* Presentación del trabajo: TG3\_final.pptx
* Prototipos obtenidos implementando cada una de las tecnologías (deben incluir el código fuente y todos los archivos necesarios para la instalación y uso de cada prototipo):
  + PrototipoTecnologiaA\_final.zip (o .rar)
  + PrototipoTecnologiaB\_final.zip (o .rar).

Dichos archivos serán los que se tendrán en cuenta para la calificación del trabajo.

# 2. Requisitos del prototipo a implementar

El objetivo del proyecto es comparar la implementación de un mismo prototipo de sistema utilizando dos tecnologías diferentes (A y B).

Es importante cumplimentar este apartado antes de empezar a implementar el prototipo de cada tecnología, porque ambos prototipos deben cumplir los requisitos que se establezcan en este apartado. Si se van a crear dos equipos de trabajo, uno para cada prototipo, el contenido de este apartado es lo que han de compartir ambos equipos como punto de partida.

Cuanto más detallados sean los requisitos, mayor será la precisión en la comparación que se realizará al final del trabajo. Se trata de conseguir dos prototipos con igual funcionalidad, pero utilizando diferentes tecnologías.

Se puede dar libertad a los equipos de desarrollo en cuanto al diseño, pero la funcionalidad debe ser lo más parecida posible. Por ejemplo, no es necesario que los colores utilizados en las pantallas sean exactamente los mismos en ambos prototipos, a no ser que los miembros del grupo lo hayan decidido así, en cuyo caso, esos detalles de colores deben incluirse en el catálogo de requisitos, para que ambos equipos los cumplan.

## 2.1 Requisitos funcionales

Los requisitos funcionales deben ser los mismos para las dos implementaciones.

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema.

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01 | …. |
| RF02 | …. |
|  |  |

## 2.2 Otros requisitos

Se pueden incluir aquí otros requisitos para el prototipo que no puedan considerarse como funcionales. Por ejemplo, requisitos de datos, de seguridad, de interfaz de usuario, de rendimientos, etc.

Se puede dejar libertad

En la siguiente tabla se indicará el catálogo de requisitos no funcionales del sistema.

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| R01 | …. |
| R02 | …. |
|  |  |

# 3. Criterios de comparación en la implementación

En el trabajo TG2 se definieron criterios de comparación de las dos tecnologías a nivel teórico.

En este trabajo hay que definir criterios para la comparación de la implementación de las tecnologías en la construcción del prototipo de sistema de ejemplo, cuyos requisitos son los establecidos en el apartado 2.

Se trata de criterios del tipo” “horas empleadas en el desarrollo del sistema”, “velocidad de funcionamiento del sistema”, “recursos necesarios”, etc.

## 3.1 Criterio 1: Nombre del criterio

Por cada criterio hay que indicar el nombre, una breve descripción, y el tipo de valor a asignar al criterio.

Por ejemplo, si se comparan dos herramientas CASE realizar el diseño UML de un mismo sistema, un criterio podría ser:

*Nombre del criterio: Tiempo de creación del diagrama de clases del sistema.*

*Descripción: Horas invertidas en la creación del diagrama de clases utilizando el editor de la herramienta.*

*Tipo de valor: Numérico (horas).*

## 3.2 Criterio 2: Nombre del criterio

## 3.N Criterio N: Nombre del criterio

# 4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología A, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 4.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 4.2 Documentación de construcción

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 4.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los caso de prueba.

## 4.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 4.5 Manual de usuario

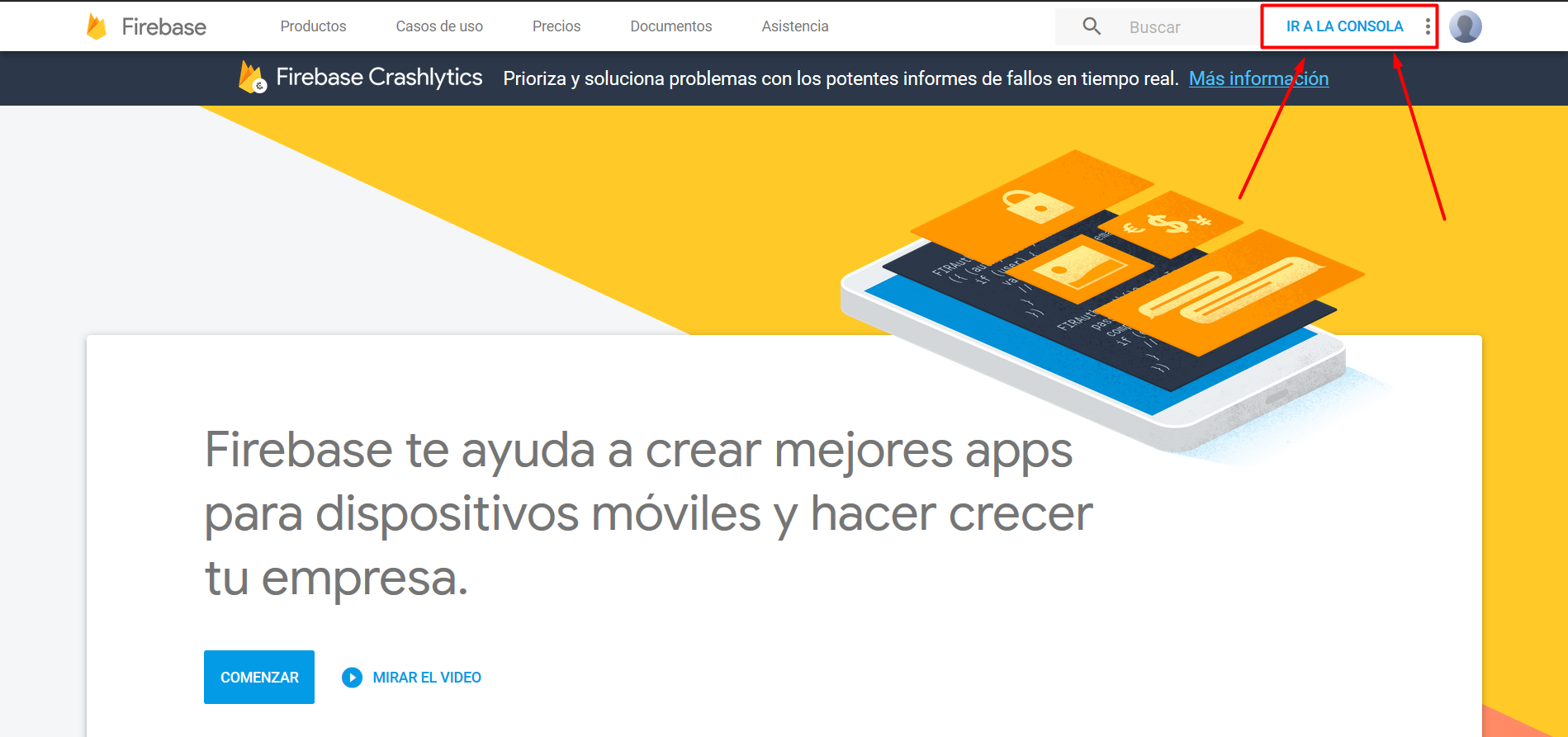
OneSignal es un SDK que debemos agregar a un proyecto que tengamos desarrollado en Android. Esto nos permitirá enlazar nuestra aplicación Android con el cliente de OneSignal y así poder enviar notificaciones a los usuarios que tengan la aplicación instalada en sus dispositivos móviles con sistemas operativos Android.

Se ha seguido el tutorial indicado en esta dirección de la documentación oficial disponible de OneSignal: <https://documentation.onesignal.com/docs/android-sdk-setup>

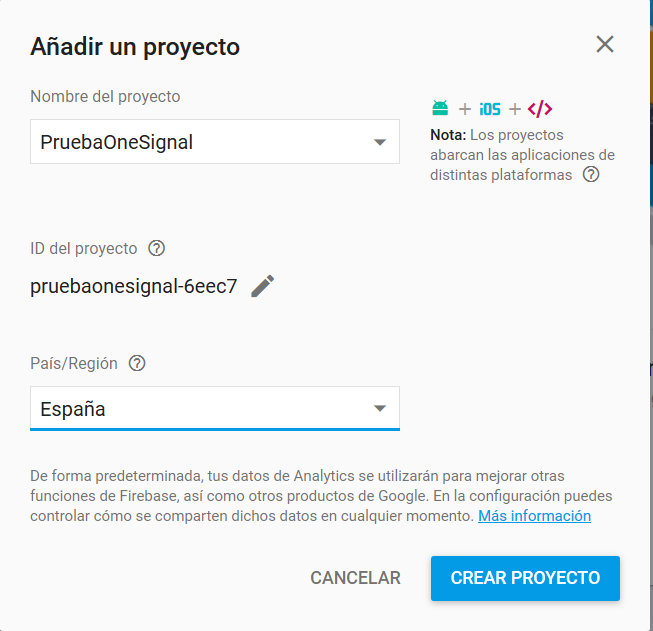
## 4.5.1 Crear un proyecto en FireBase.

FireBase es una página web de Google donde creamos proyectos para poder obtener claves API de nuestro proyecto. Lo necesitaremos para poder enviar notificaciones a los dispositivos móviles que tengan nuestra aplicación mediante OneSignal.

Para ello, entramos en [www.firebase.com](http://www.firebase.com), iniciamos una cuenta gmail y damos click a “Ir a consola”:



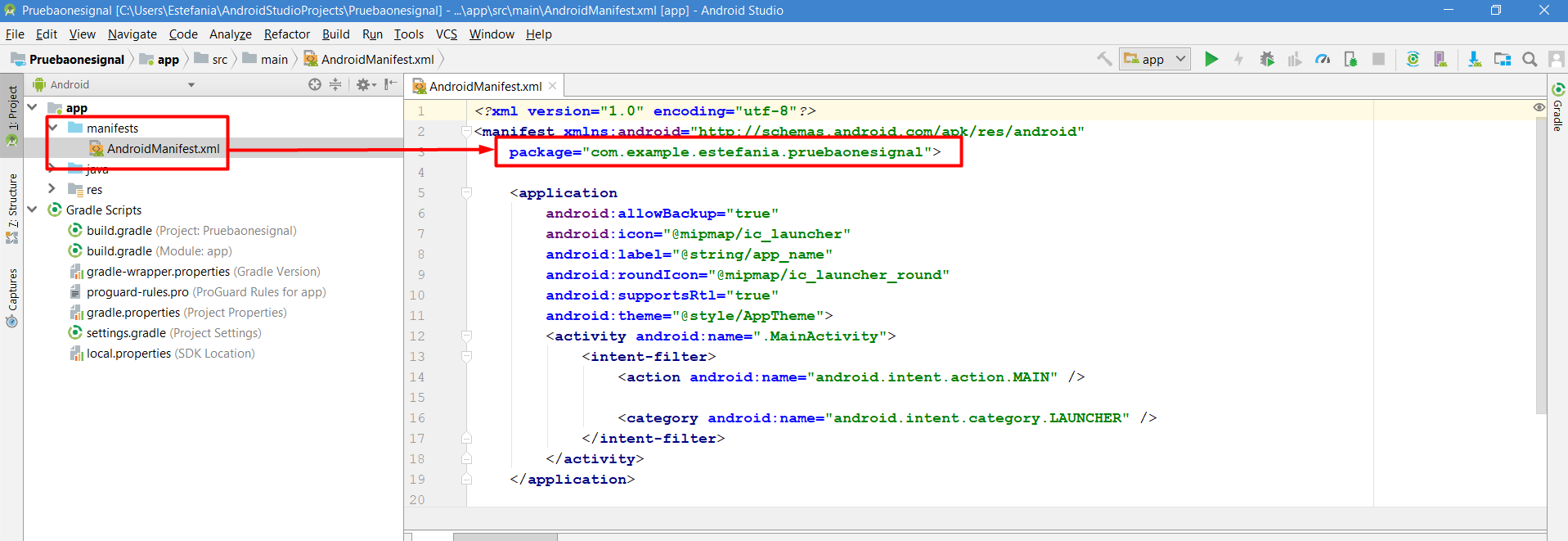
A continuación, damos click a “Añadir proyecto” y escribimos el nombre que le daremos a este proyecto (no es necesario que se llame igual que nuestra aplicación) y el país:



Una vez creado, deberemos esperar hasta que se haya creado nuestro proyecto. No llevara mucho tiempo.

Una vez dentro, damos click a “Añade Firebase a tu aplicación Android”:



El dato que necesitaremos a continuación será el nombre del paquete (package) de nuestro proyecto. Abrimos nuestro IDE donde tengamos el proyecto, en mi caso Android Studio, y entramos en el manifest del proyecto. Aquí encontraremos el nombre del paquete de nuestro proyecto.

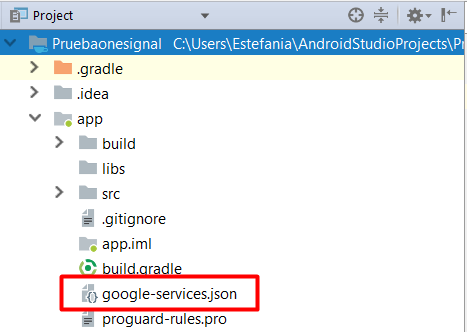
Lo copiamos y lo introducimos en el formulario de Firebase que corresponde con el nombre del paquete de Android y damos click a “Registrar la aplicación”.



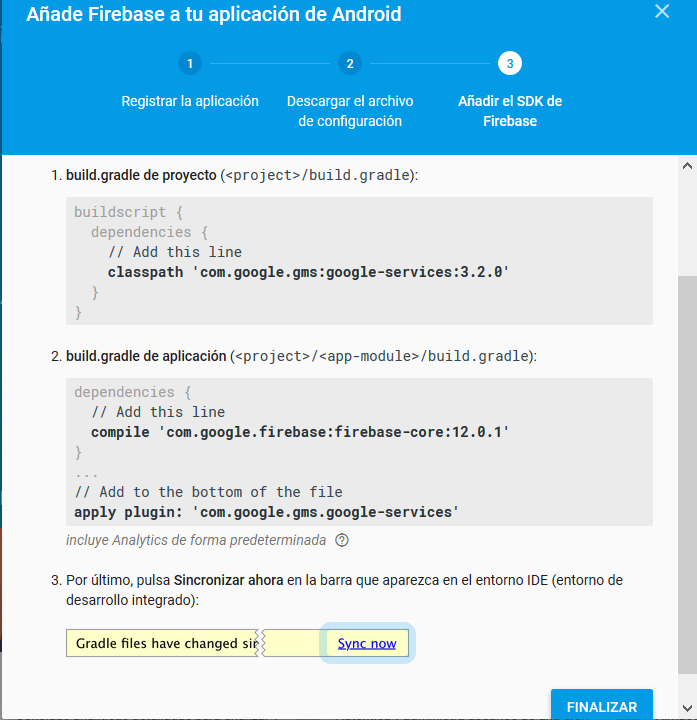
Como nos indica en el siguiente paso, debemos descargar el archivo que indica y guardarlo en la carpeta llamada “app” de nuestro proyecto:



Finalmente debería quedar así en nuestro proyecto:



Cuando ya hemos clicado en el botón “continuar”, nos aparecerá otro tutorial en Firebase con unas instrucciones que deberemos hacer en Gradle de nuestro proyecto:



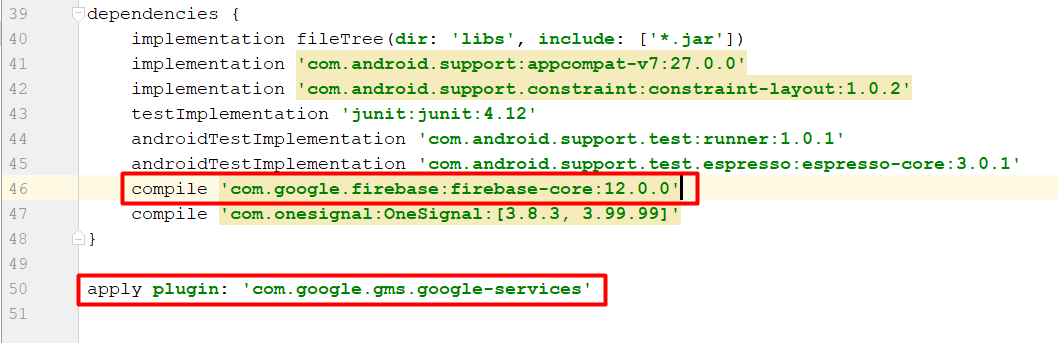
Deberemos abrir abrir nuestro apartado de Gradle de nuestro proyecto, que se situa en la parte izquierda de la pantalla de Android Studio:



En este caso, debemos abrir primero el documento llamado “build.gradle (Project: nombredelproyecto)”. Debería quedar así una vez copiemos las líneas del primer paso indicado en Firebase (línea 11):

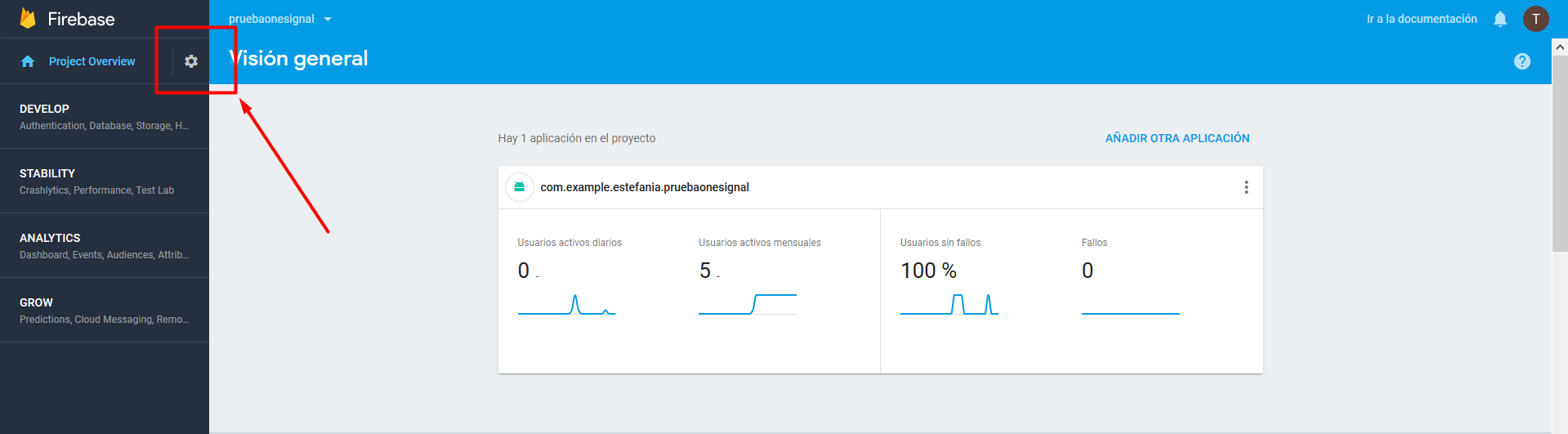


Una vez realizado esto, ahora tenemos que abrir el archivo llamado “bluild.gradle (Module: app)” y copiar las líneas que se indican en el paso 2 de Firebase. Cuando termine, debe quedar así (ignore el resto de líneas de código que no estén dibujadas con cuadrados rojos):

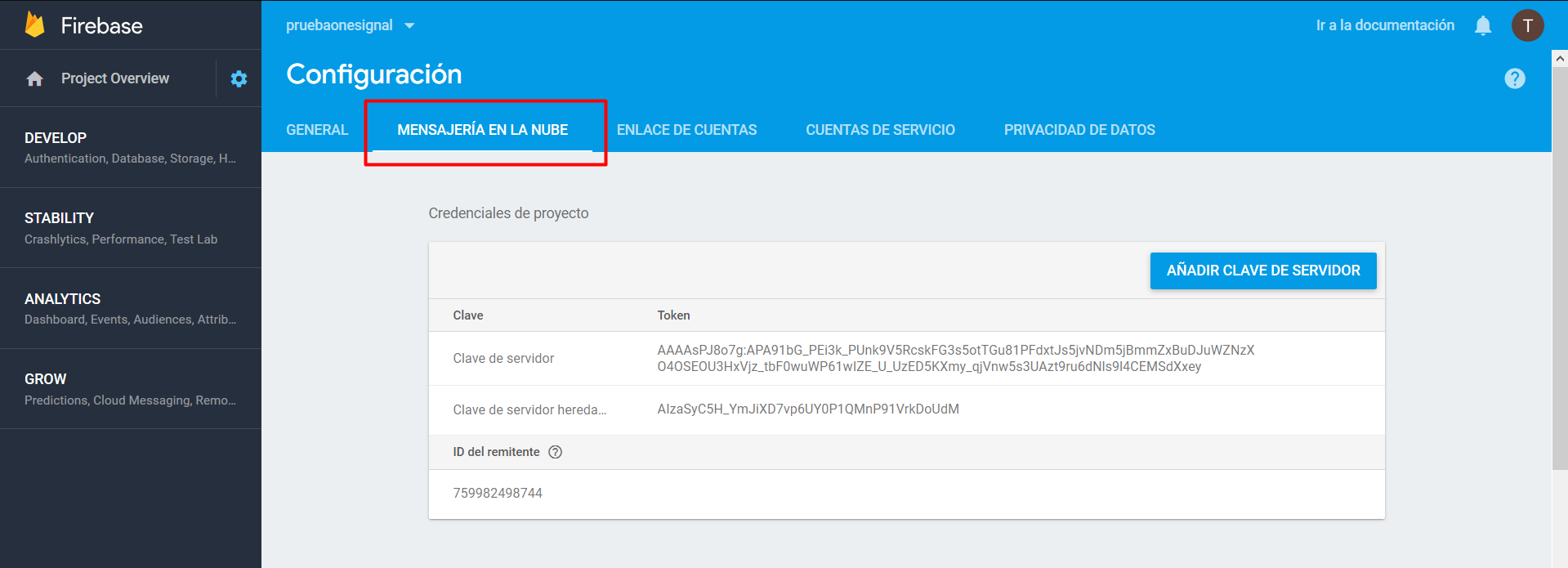


Cuando esté todo correctamente, encima de cualquiera de los dos archivos aparece una barra que indicará que Gradle debe sincronizarse. Damos a “Sync now” y esperamos a que termine.

Si no ha habido problemas, podemos finalizar en Firebase y ahora la página será así(si es la primera vez que lo hacemos, las gráficas estarán a cero):



Clicamos en la tuerca que se indica en la imagen y abrimos la configuración del proyecto, abrimos la pestaña de mensajería en la nube y la dejamos abierta para más tarde:

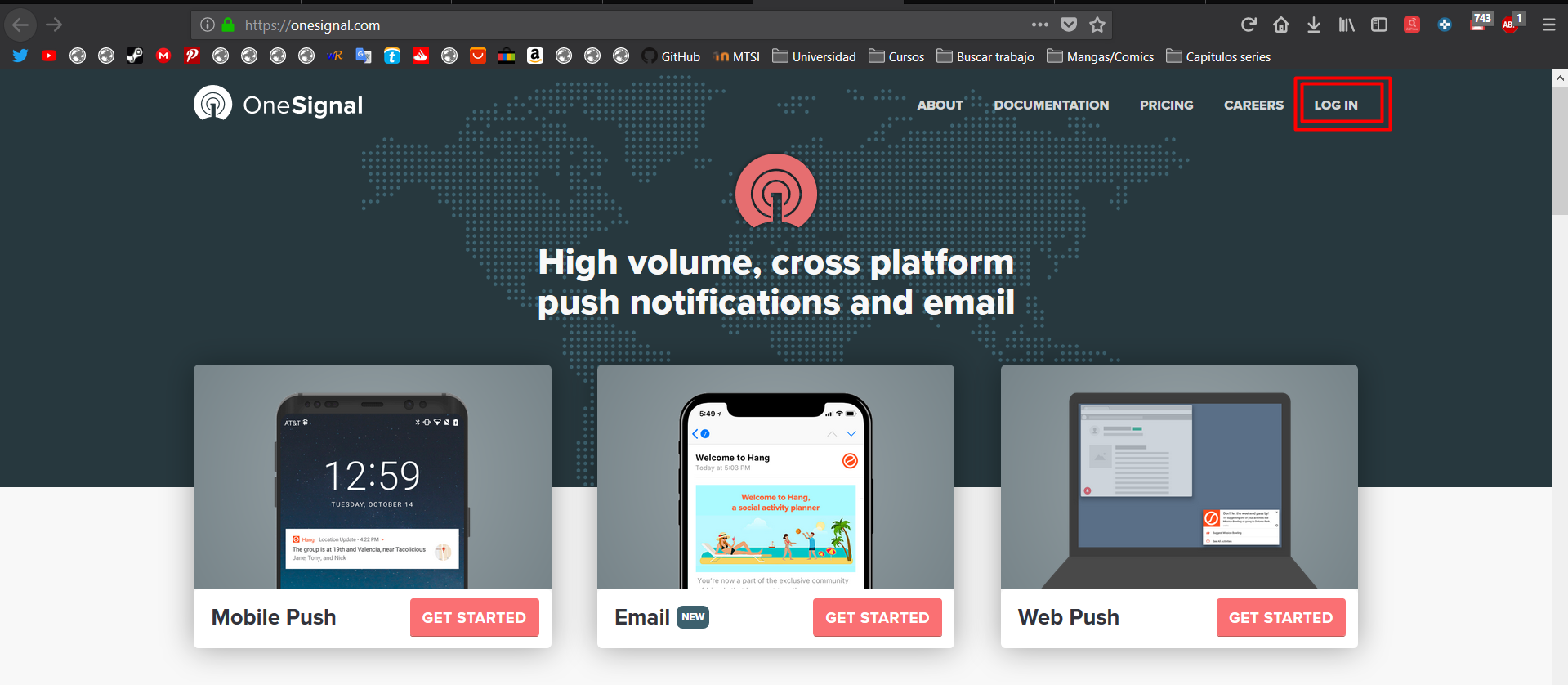


Si hemos realizado todos los pasos correctamente, ya tendríamos nuestro proyecto en Firebase.

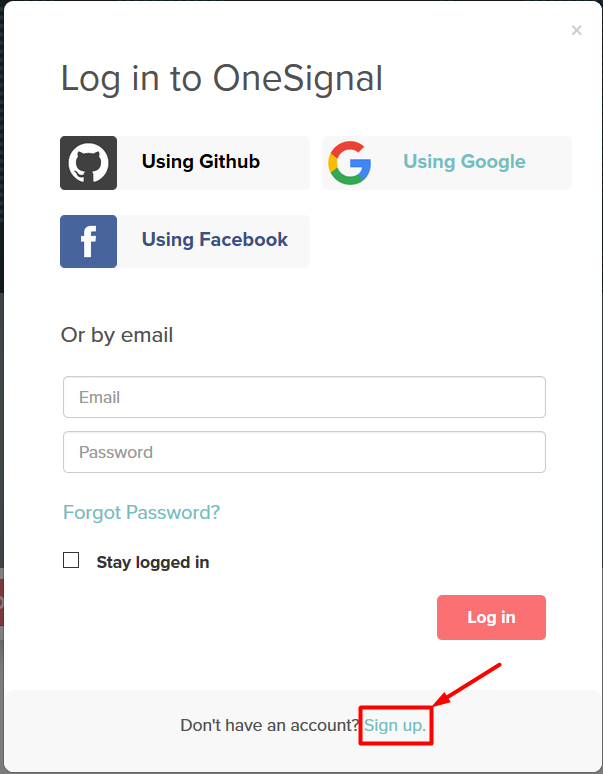
## 4.5.2 Crear una cuenta en OneSignal.

Necesitaremos crearnos una cuenta en la página de OneSignal si aún no la tenemos, ya que necesitaremos sincronizar nuestro proyecto con OneSignal con la API ofrecida por Firebase.

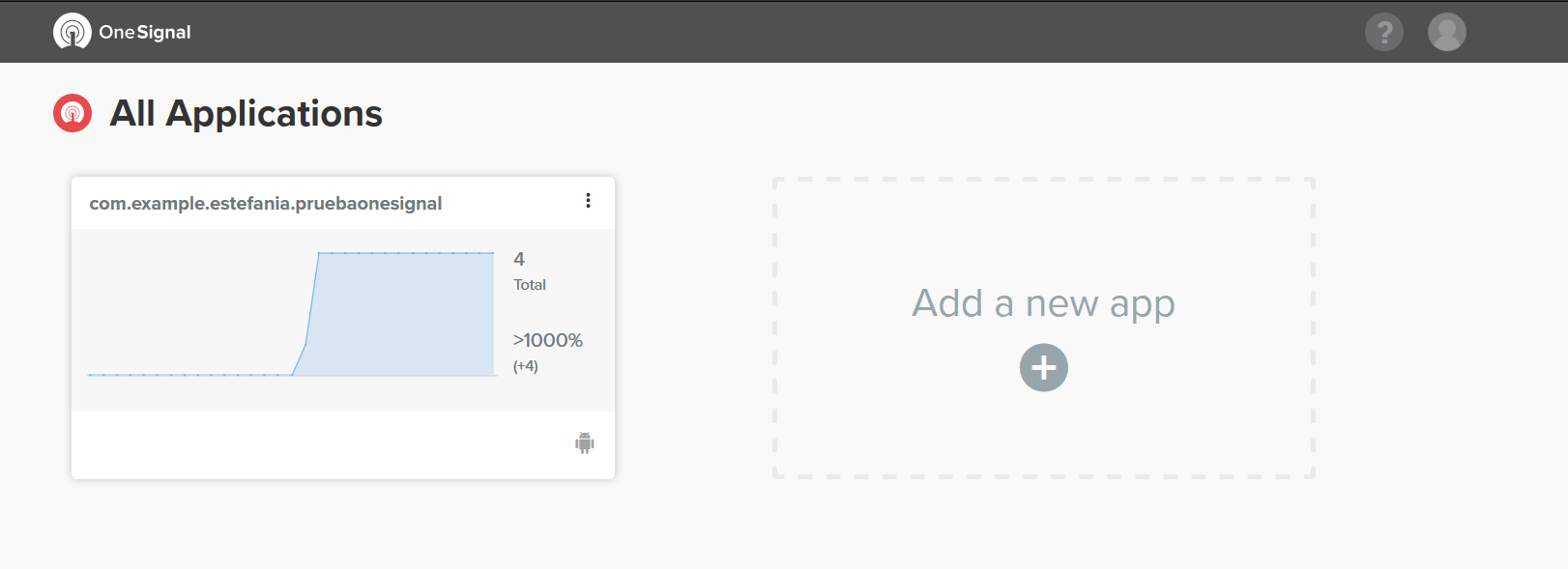
Entraremos en la página web de OneSignal (https://onesignal.com/) y haremos click en “Log in”:



Nos ofrecerá múltiples maneras de iniciar sesión. Si queremos registrarnos con una dirección de email, clicamos en el botón donde pone “Sign up” como indicamos con una flecha en esta imagen:



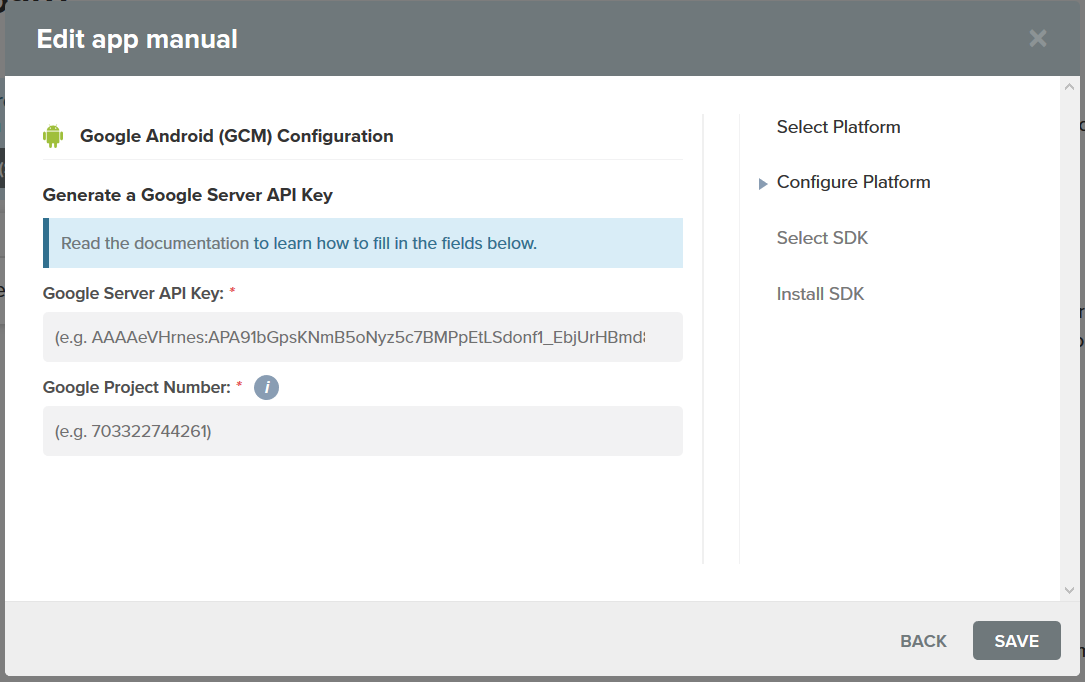
Cuando hayamos podido iniciar sesión, nos encontraremos con la dashboard de OneSignal donde podremos crear proyectos y poder enviar notificaciones:



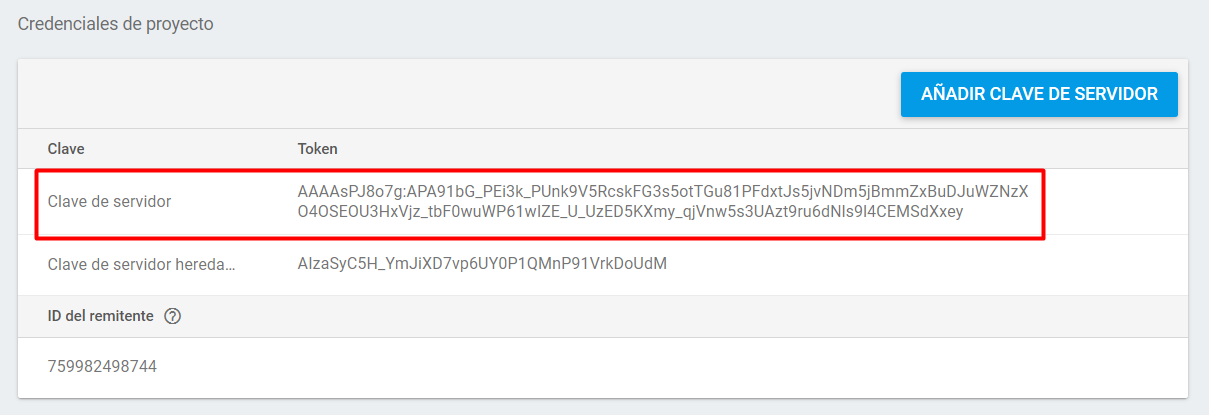
Pimero, damos click en “Add a new app”. Elegimos un nombre a la app, el que queramos. Nos aparecerá una ventana donde nos preguntará en qué plataforma vamos a desarrollar la aplicación, en nuestro caso, Android y damos a “Next”:



Ahora nos pregunta por las claves API de nuestro proyecto:

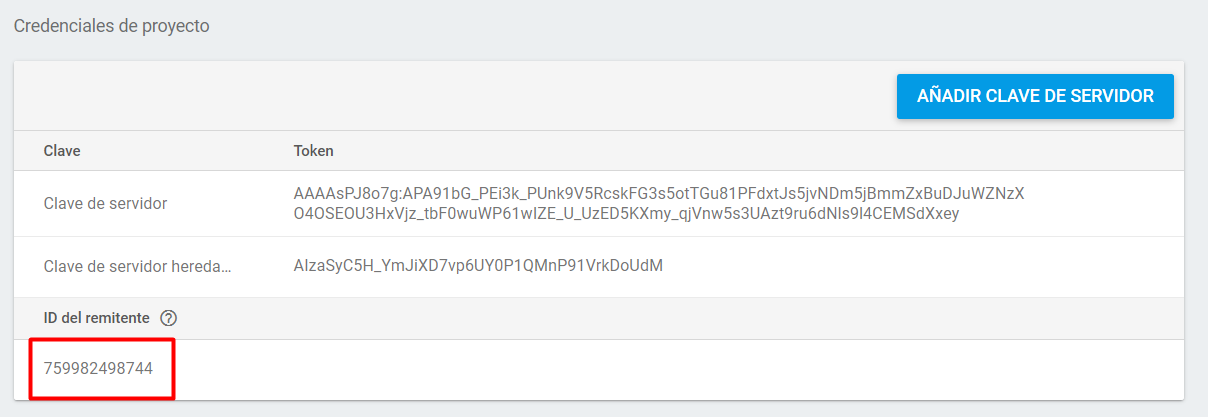


Abrimos Firebase como lo dejamos en el manual anterior y copiamos la clave que se llama “Clave del servidor”:



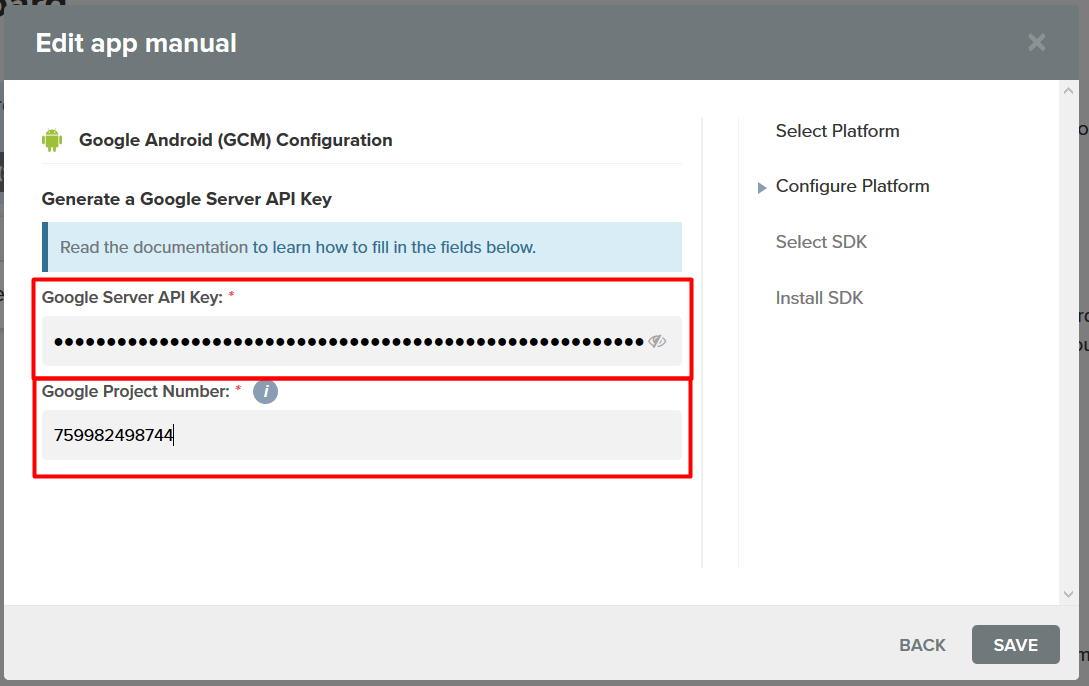
Y la pegamos en el formulario de One Signal que se llama “Google Server API Key”.

Ahora volvemos a entrar a Firebase y copiamos el ID del remitente:

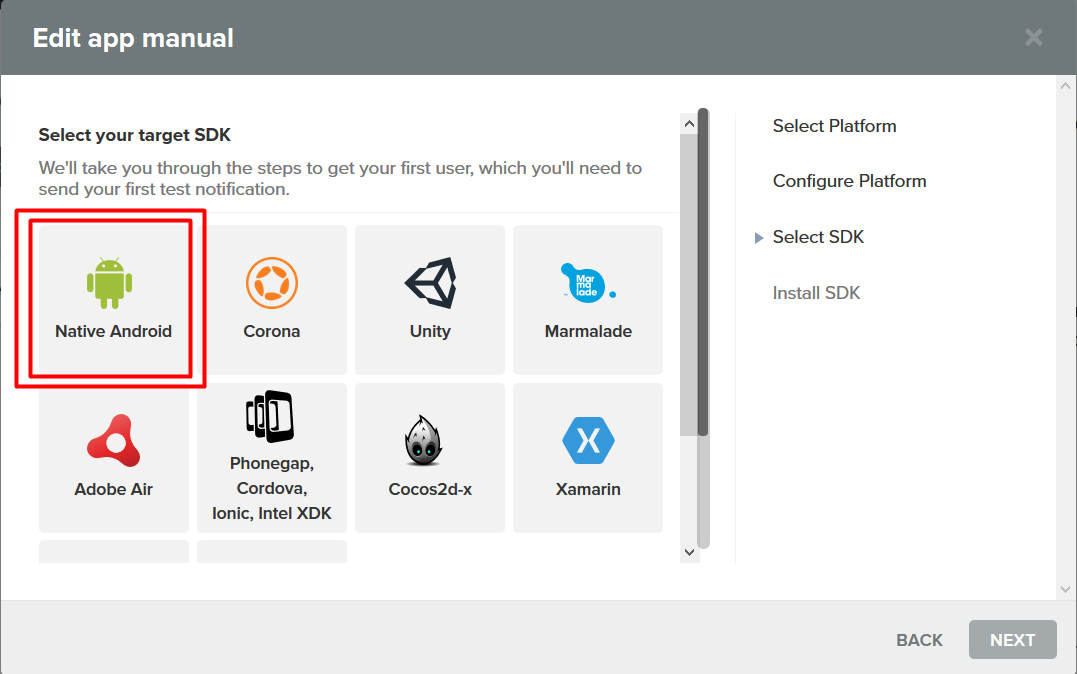


Y la pegamos en el formulario que se llama “Google Project number”.

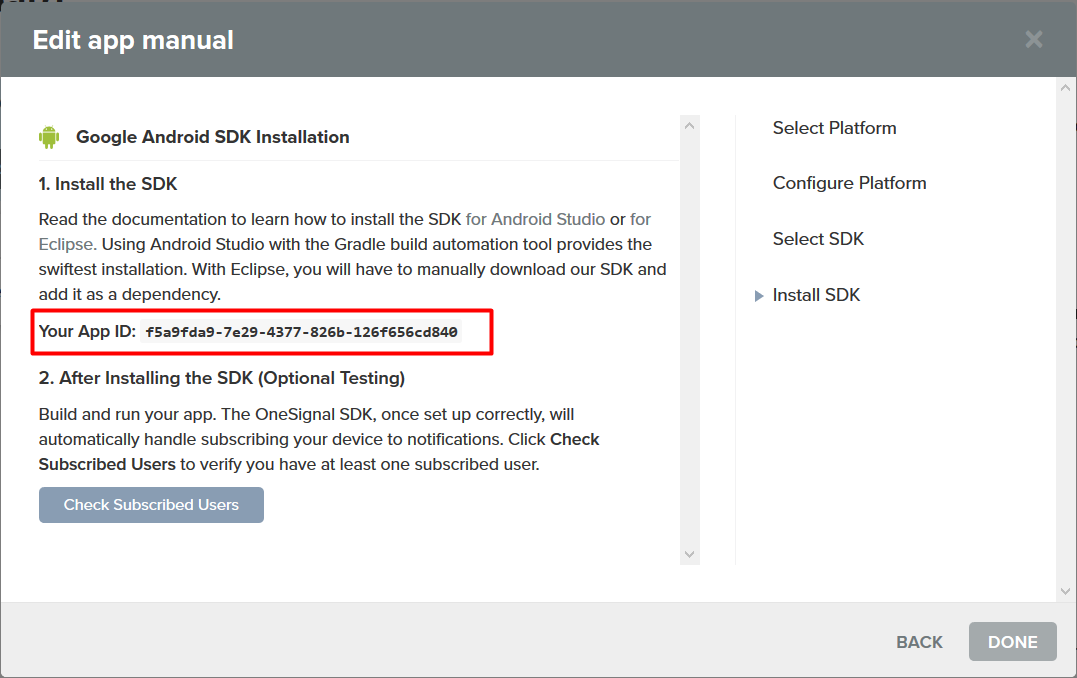
Finalmente quedaría de esta manera:



Cuando hayamos dado a “Save”, nos pedirá el tipo de SDK que hemos instalado en nuestro proyecto de Android Studio (que será el siguiente punto). Elegimos “Native Android” y le damos a “Next”:



Finalmente, OneSignal nos ofrecerá la “App ID” para poder utilizarla en el siguiente manual. **No cierre esta ventana.**



## 4.5.3 Instalar el SDK en OneSignal con Android Studio.

Para empezar, abrimos nuestro proyecto en Android Studio y abrimos el archivo “grandle.build (Module: app)” y pegamos estas líneas de código en la línea 1 del archivo:

buildscript {

repositories {

maven { url 'https://plugins.gradle.org/m2/'}

}

dependencies {

classpath 'gradle.plugin.com.onesignal:onesignal-gradle-plugin:0.8.1'

}

}

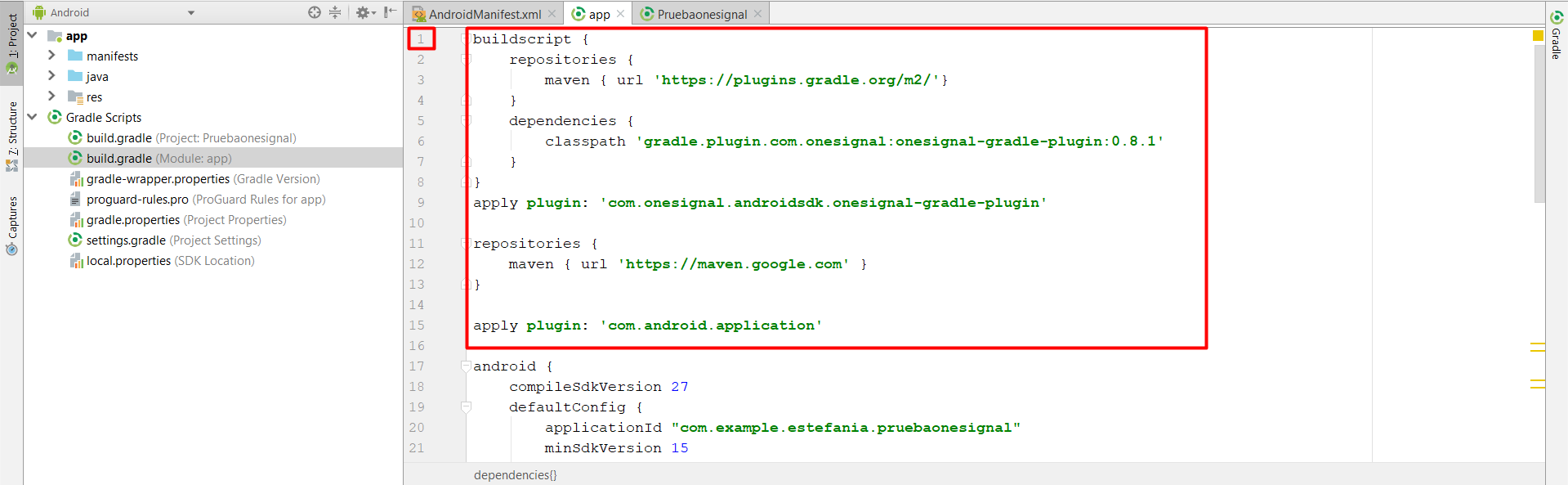
apply plugin: 'com.onesignal.androidsdk.onesignal-gradle-plugin'

repositories {

maven { url 'https://maven.google.com' }

}

Quedando de esta forma:

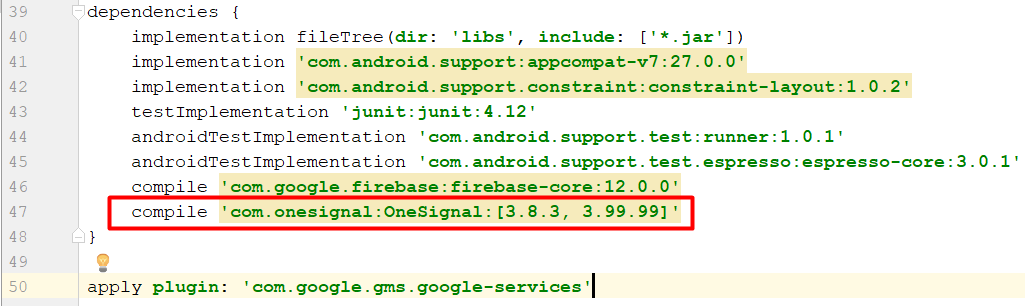
Ahora en la sección “dependencies” del mismo archivo, pegamos estas líneas de código:

dependencies {

compile 'com.onesignal:OneSignal:[3.8.3, 3.99.99]'

}

Quedando de esta manera:



Ahora tenemos que pegar las siguientes líneas de código en la sección Android > defaultConfig de nuestro proyecto:

android {

defaultConfig {

manifestPlaceholders = [

onesignal\_app\_id: 'PUT YOUR ONESIGNAL APP ID HERE',

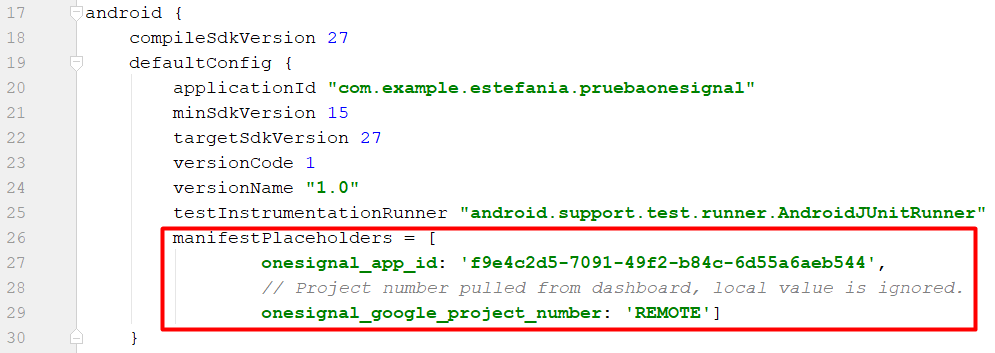
**// Project number pulled from dashboard, local value is ignored.**

onesignal\_google\_project\_number: 'REMOTE'

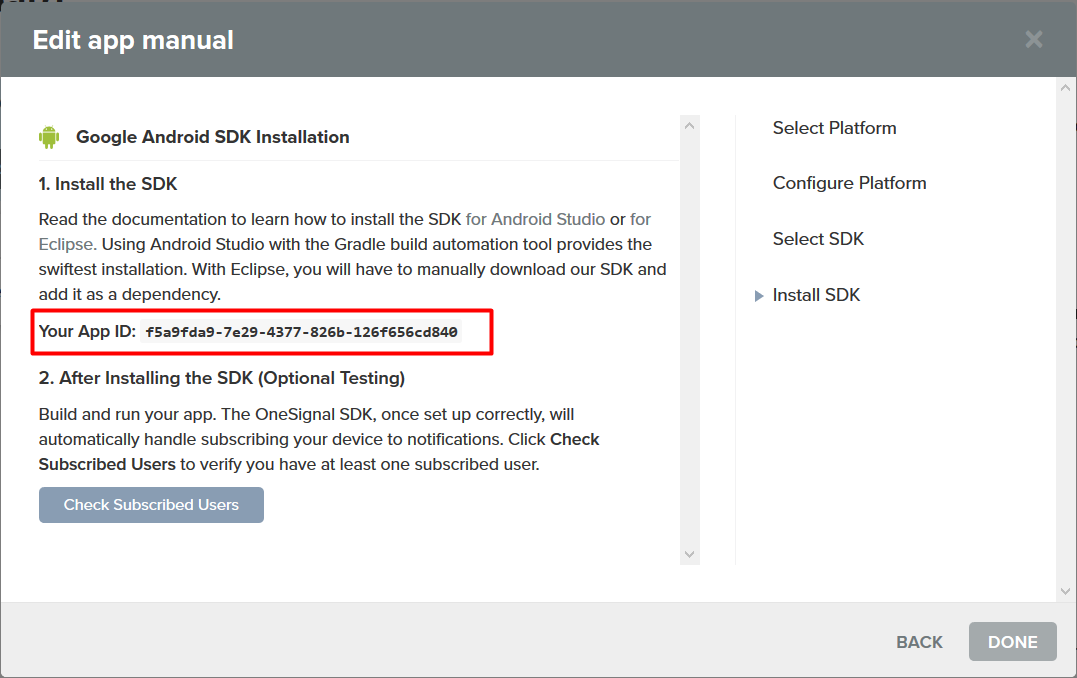
]

}

}

Quedando de la siguiente manera:

En la línea 27 del código de la imagen corresponde con el código que nos ha ofrecido OneSignal en el último paso del punto anterior:

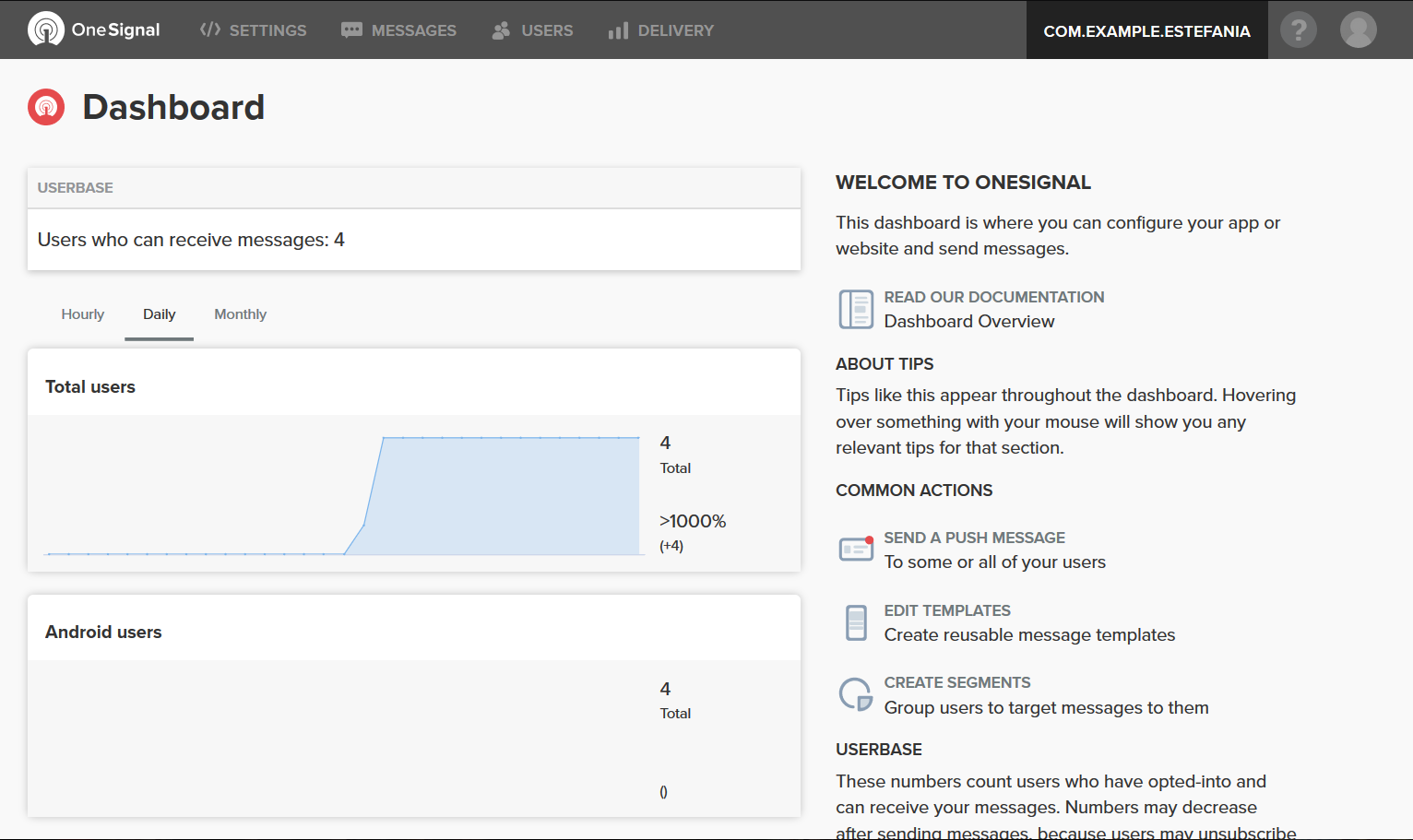


Tenemos que reemplazar PUT YOUR ONESIGNAL APP ID HERE por eso código entre comillas simples.

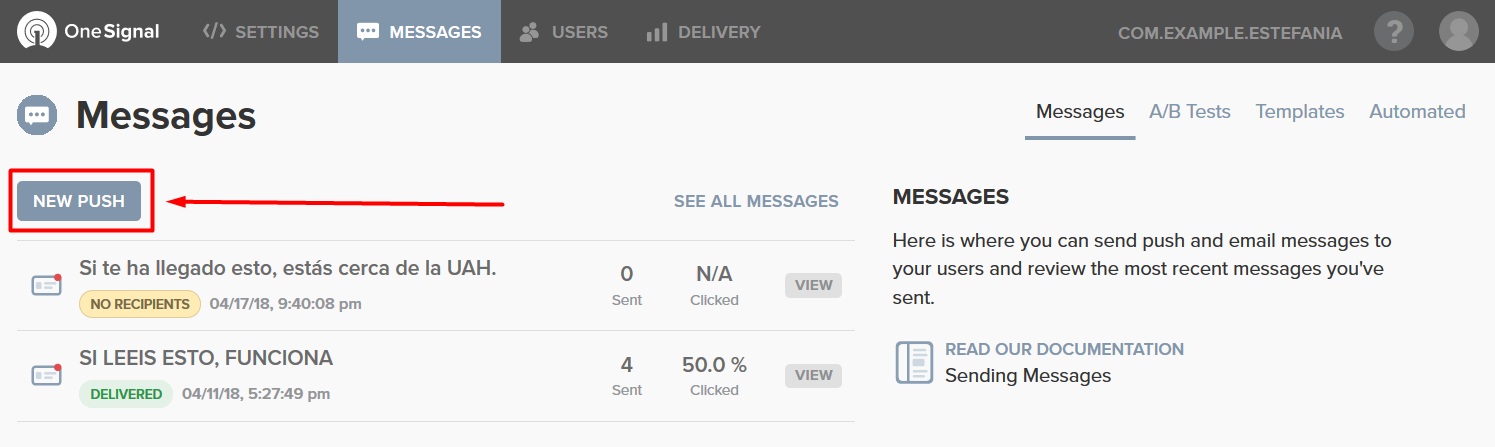
Una vez finalizado esto, debemos dar click en “Sync Now” en el archivo gradle.build (Module:app), arriba.

## 4.5.4. Enviar una notificación con OneSignal.

Si no hemos tenido problemas hasta ahora, podemos proceder a instalar el proyecto de Android Studio en nuestro móvil. Cuando ya lo tengamos, abrimos OneSignal y la página web será parecida a esto:



Para enviar una notificación, damos click en la pestaña superior llamada “MESSAGES”. Si hemos realizado alguna vez una notificación, ésta puede guardarse para poder repetirla en alguna ocasión. Para poder enviar una notificación nueva, damos click a “NEW PUSH”:

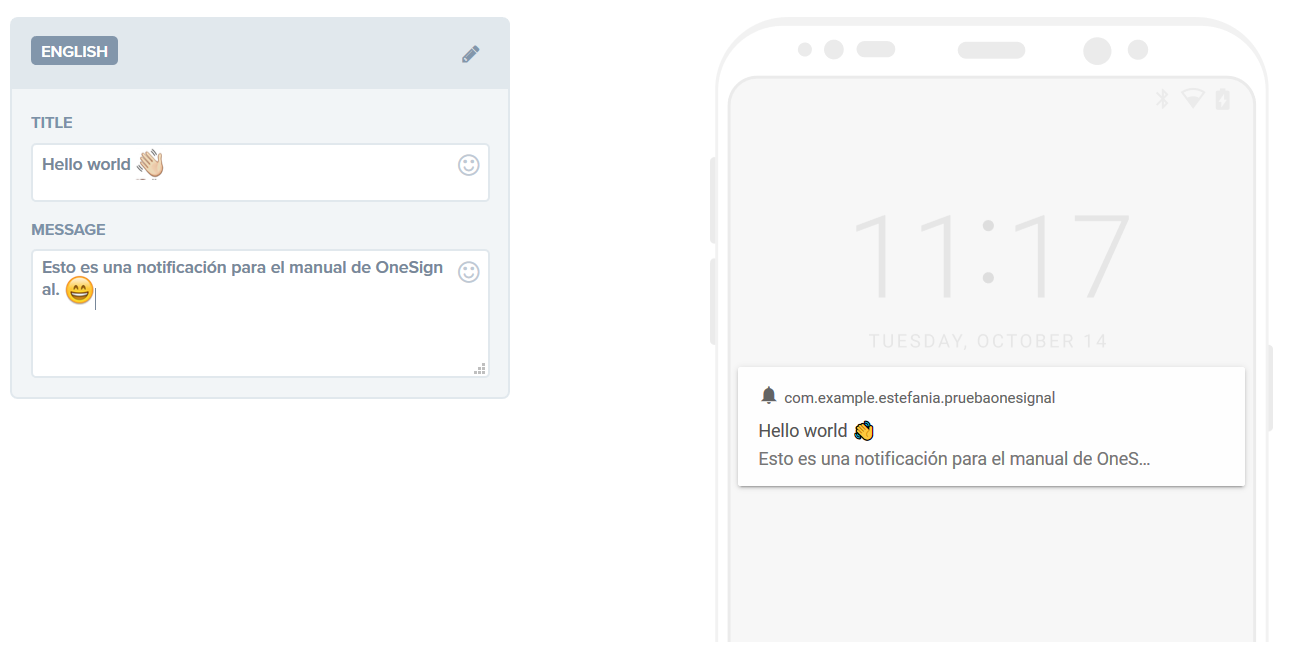


La configuración se dividirá en cuatro partes:

1. Audiencia: One Signal nos da tres opciones:
   1. **Send to everyone:** podremos enviar la notificación a todos los usuarios que tengan nuestra aplicación instalada en el móvil.
   2. **Send to Particular Segment(s):** podremos diseñar segmentos de usuarios a quienes queremos mandarles la notificación. Podremos elegir entre varios tipos de segmentos como la localización del usuario, la versión de su dispositivo, la última vez que usó la aplicación… etc.
   3. **Send to Test Device(s):** podremos mandar notificaciones solo a dispositivos de prueba.

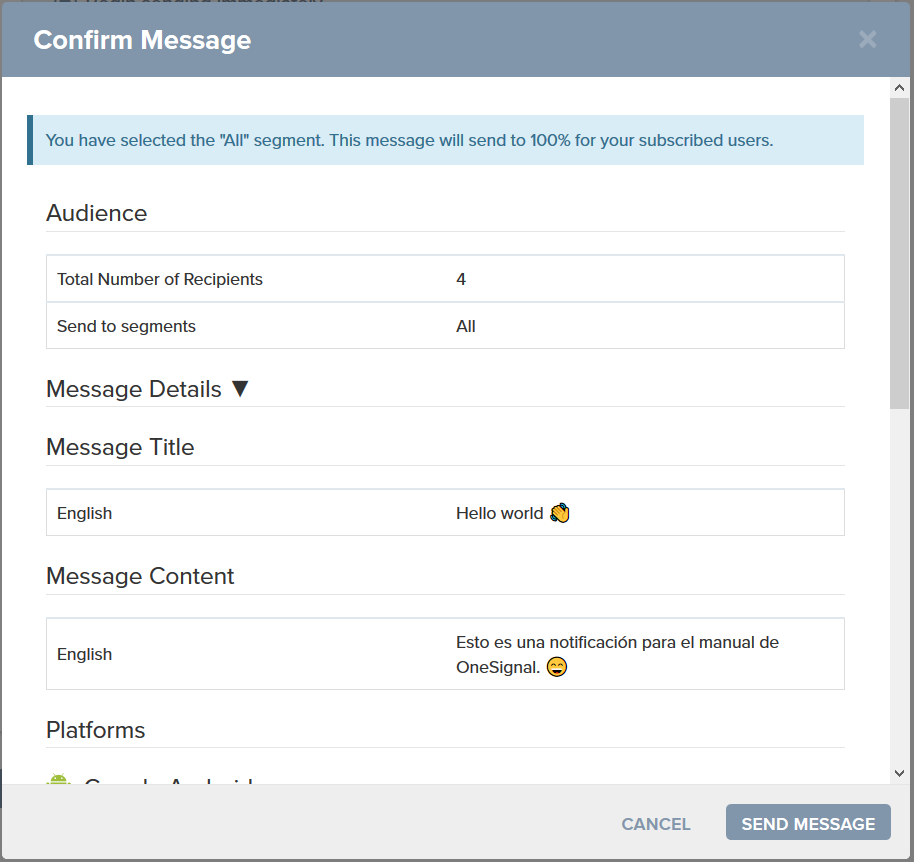
En este manual elegimos “Send to everyone”.

1. Message: Aquí podremos editar nuestra notificación y nos mostrará una vista previa de cómo quedaría.

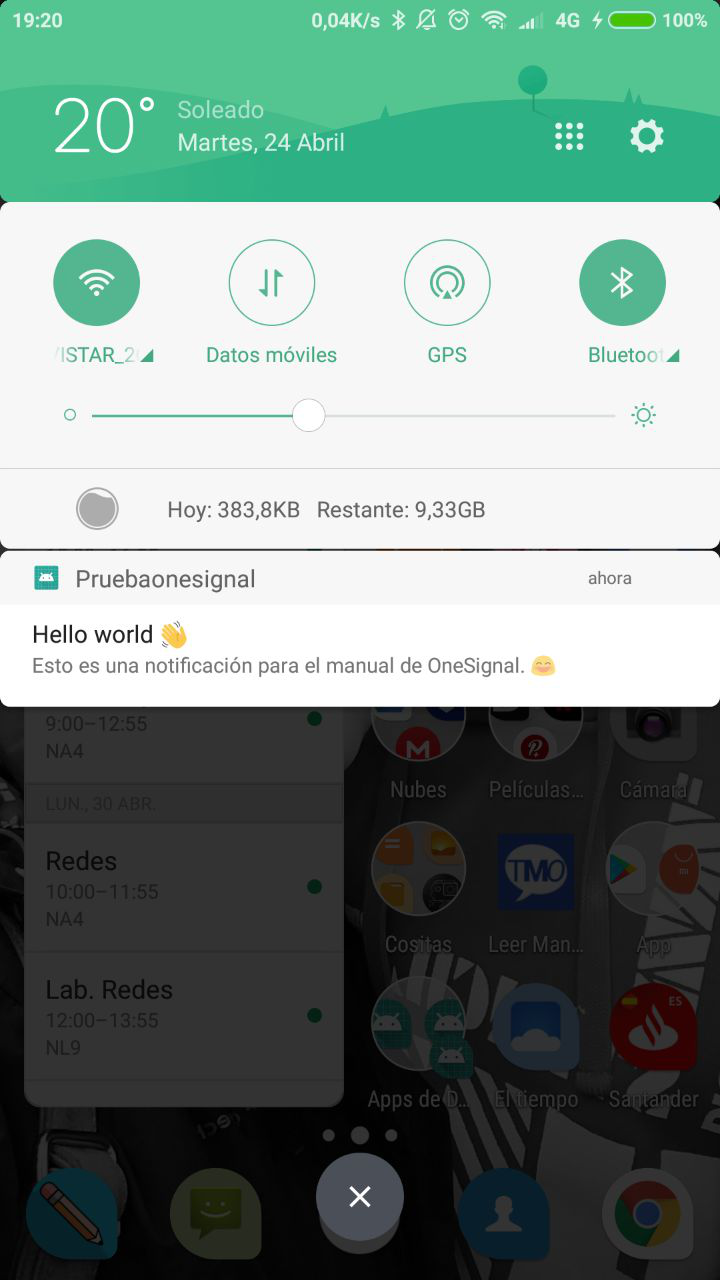


1. Options: Aquí podremos editar las opciones disponibles de la notificación como la categoría si la hubiera, el sonido, el color del led del dispositivo (si lo tuviera), el icono… En este manual no vamos a modificar nada.
2. Schedule: Aquí podremos editar cuándo enviar la notificación.

Una vez hayamos acabado de editar nuestra notificación, damos click a “CONFIRM”. A continuación, se nos mostrará un resumen de nuestra notificación antes de enviarla:



Si todo está correcto, damos a “SEND MESSAGE” y podremos observar a tiempo real cómo los usuarios abren la notificación en sus dispositivos móviles. La notificación tendrá esta pinta en los dispositivos que la hayan recibido:



# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 5.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 5.2 Documentación de construcción

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 5.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los caso de prueba.

## 5.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 5.5 Manual de usuario

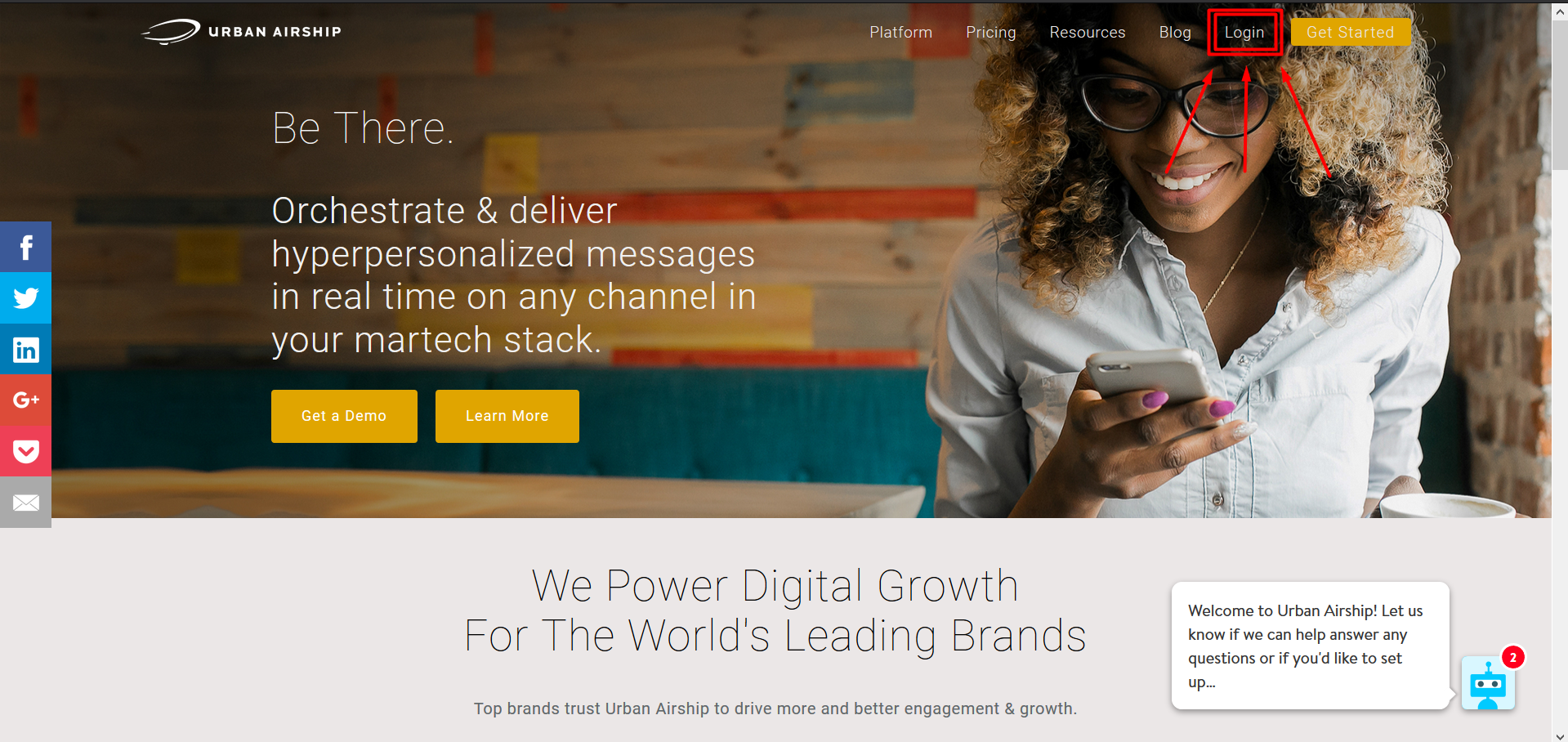
Al igual que OneSignal, Urban Airship se trata de un SDK que deberemos instalar en nuestro proyecto Android. La instalación es similar a OneSignal y para cualquier duda o problema, podemos visitar la documentación oficial de Urban Airship: <https://docs.urbanairship.com/platform/android/>.

## 5.5.1 Creación del proyecto en Firebase.

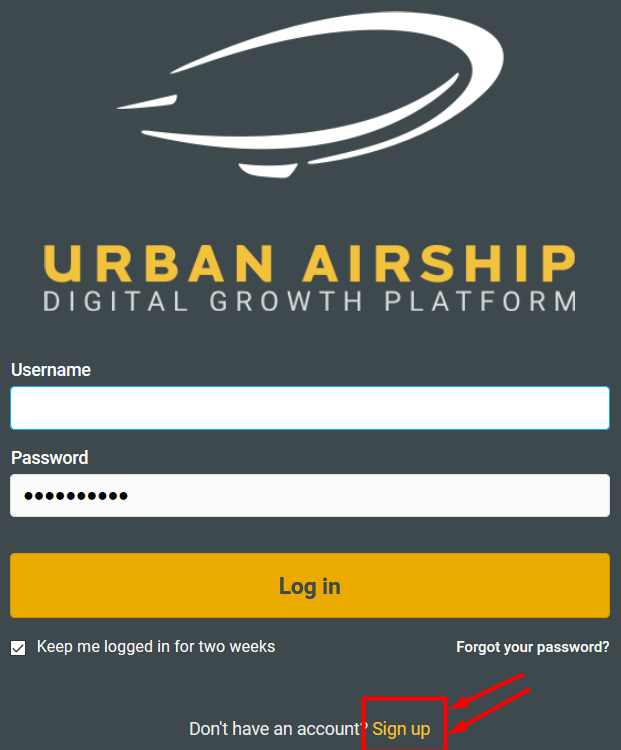
Este paso es idéntico al realizado en el punto 4.5.1 de este documento. [Ir allí](#_4.5.1_Crear_un).

## 5.5.2 Crear cuenta en Urban Airship.

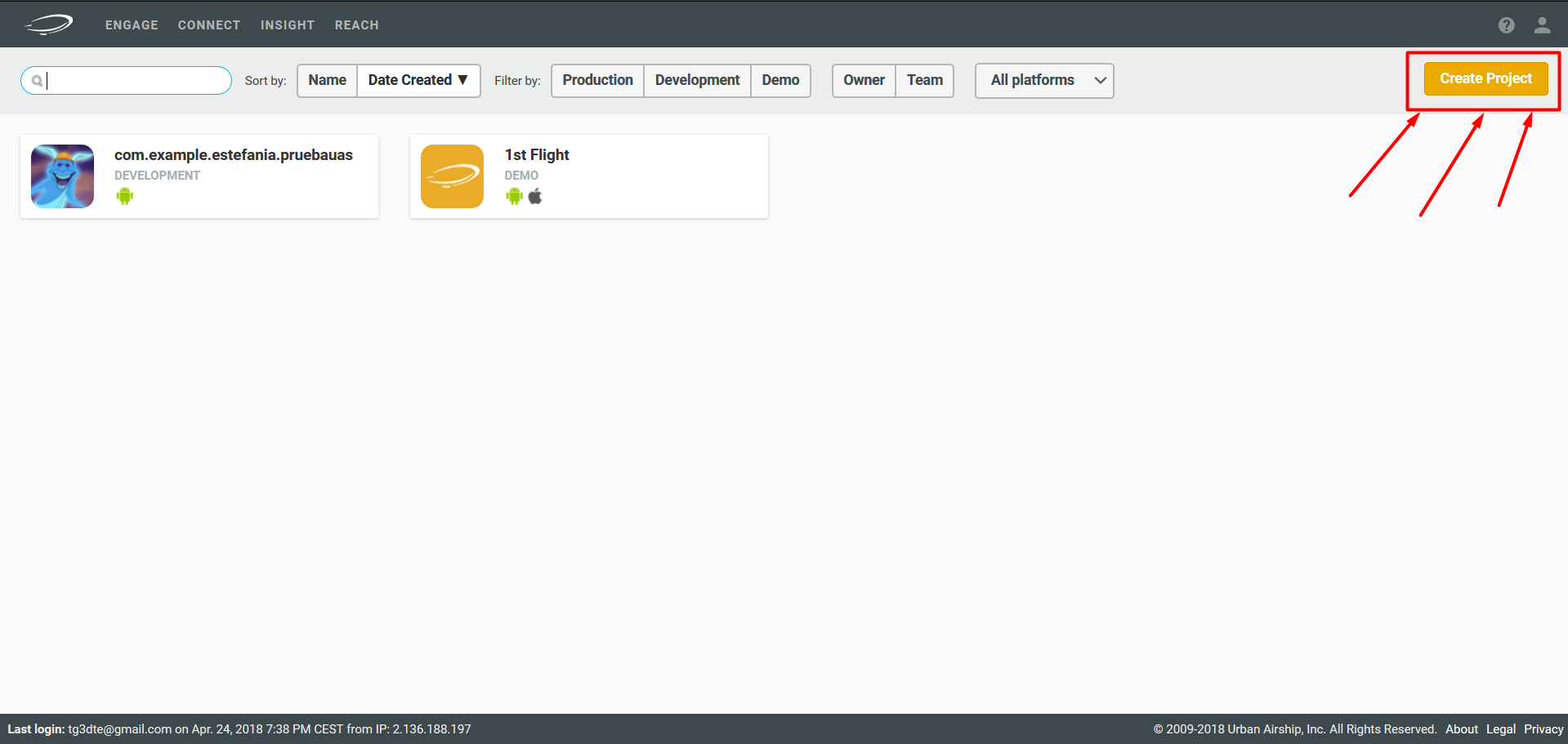
1. Entramos en la página web oficial de Urban Airship (<https://www.urbanairship.com/>) y damos click en “Log in” en la parte superior derecha de la pantalla.



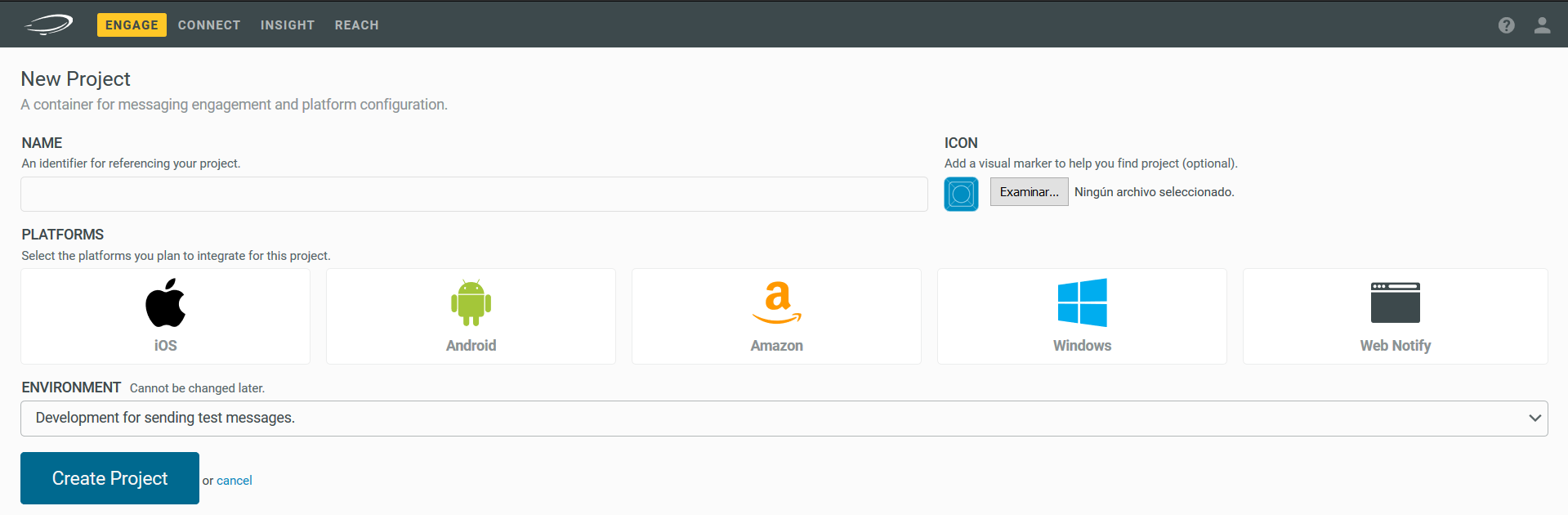
Se nos abrirá una página web nueva, donde deberemos ingresar las credenciales de nuestro usuario. Si es la primera vez que entramos, debemos hacer click en el botón inferior que dice “Sign up”, como mostramos en esta imagen:



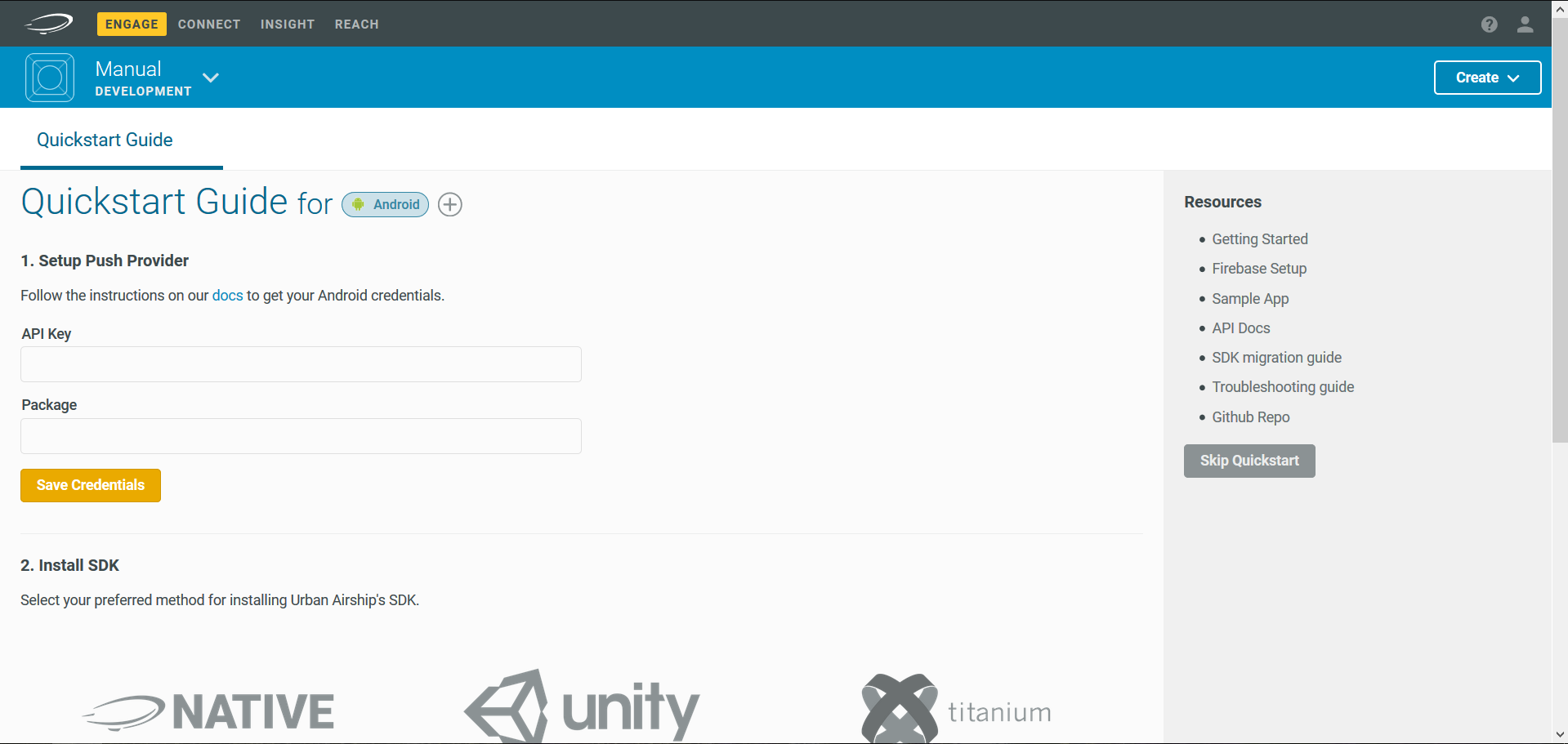
2. Una vez hayamos creado un usuario y podamos entrar, encontraremos la siguiente pantalla:



3. Hacemos click en “Create Project” como indicamos en la imagen. Nos mostrará una pantalla donde deberíamos elegir un nombre para el proyecto, un icono (opcional) para el proyecto, la plataforma del proyecto (elegimos Android) y el entorno en el que trabajaremos, que será de desarrollo (Development for sending test messages)



1. Una vez editado el proyecto, damos a “Create Project”. Cuando se haya creado, veremos la siguiente pantalla:

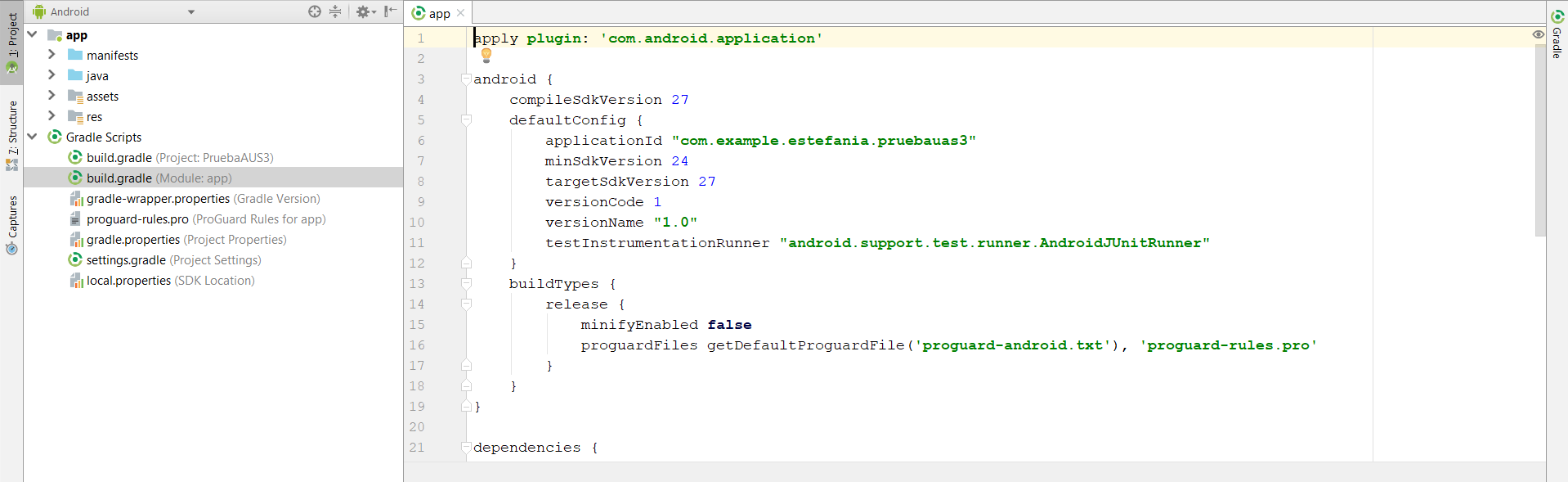


En la primera parte, nos pedirá las claves de nuestro proyecto en Firebase. “API Key” corresponderá con la clave de Firebase llamada “Clave de servidor” y “Package” corresponderá con la clave de Firebase llamada “ID del remitente”. Cuando las hayamos pegados correctamente, damos a “Save Credentials”.

En el segundo apartado, si elegimos “NATIVE” con el símbolo de Urban Airship nos llevará a la documentación del tutorial que deberemos seguir para instalar el SDK en un proyecto Android.

## 5.5.3 Instalación del SDK en Android Studio.

1. Abrimos nuestro proyecto Android en el IDE Android Studio. Una vez se cargue correctamente, abrimos el archivo “gradle.build (Module:app)”.



Deberemos pegar las siguientes líneas de código en la sección “dependencies”:

dependencies {

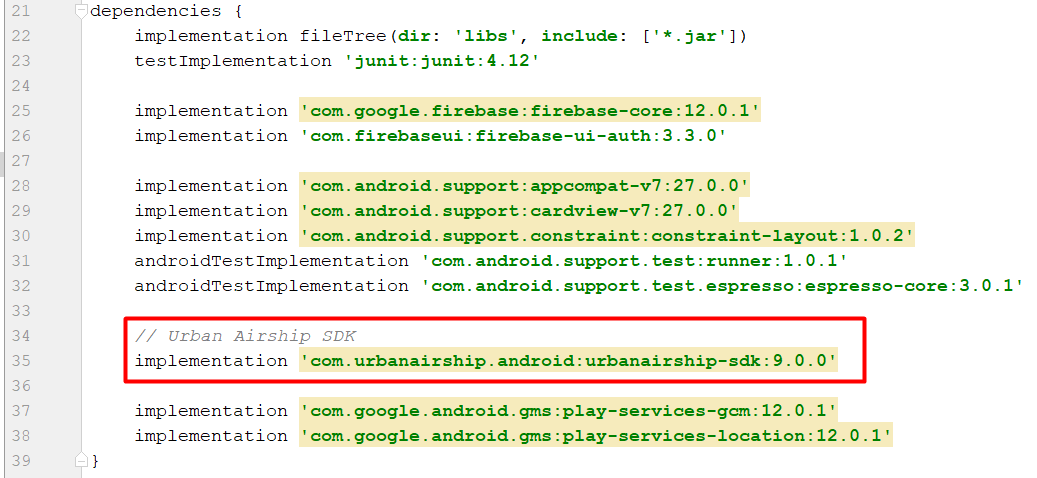
...

// Urban Airship - FCM

compile 'com.urbanairship.android:urbanairship-fcm:9.1.0'

}

Quedando de la siguiente manera:



2. Ahora debemos ir a la carpeta donde hemos creado el proyecto e ir a la carpeta C:\PROYECTO\_ANDROID\_STUDIO\app\src\main\assets y guardar como un archivo de nota con el nombre de “airshipconfig.properties”.

Nos vamos al proyecto en Android Studio y con la vista en “Project” abrimos el archivo que hemos creado y pegamos estas líneas de código a ese archivo:

#Si hemos elegido el modo DEVELOPMENT

developmentAppKey = Your Development App Key

developmentAppSecret = Your Development App Secret

#Si hemos elegido el modo PRODUCTION

productionAppKey = Your Production App Key

productionAppSecret = Your Production Secret

# Toggles between the development and production app credentials

# Before submitting your application to an app store set to true

inProduction = false

# LogLevel is "VERBOSE", "DEBUG", "INFO", "WARN", "ERROR" or "ASSERT"

developmentLogLevel = DEBUG

productionLogLevel = ERROR

# FCM/GCM Sender ID

fcmSenderId = Your Google API Project Number

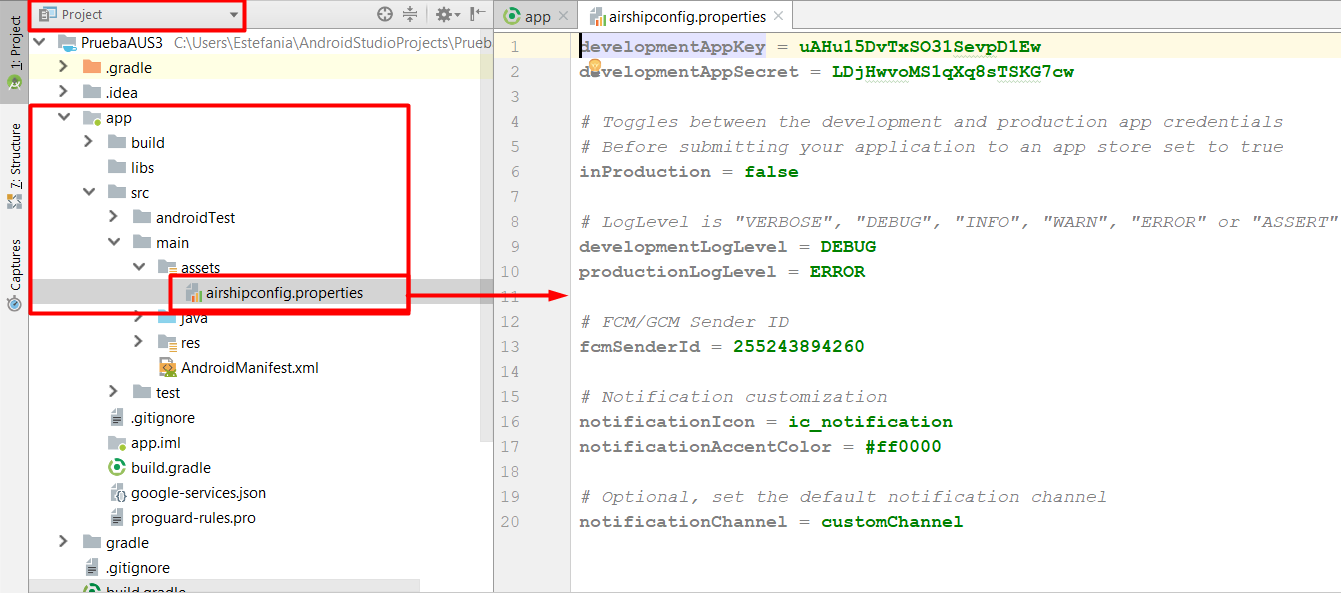
# Notification customization

notificationIcon = ic\_notification

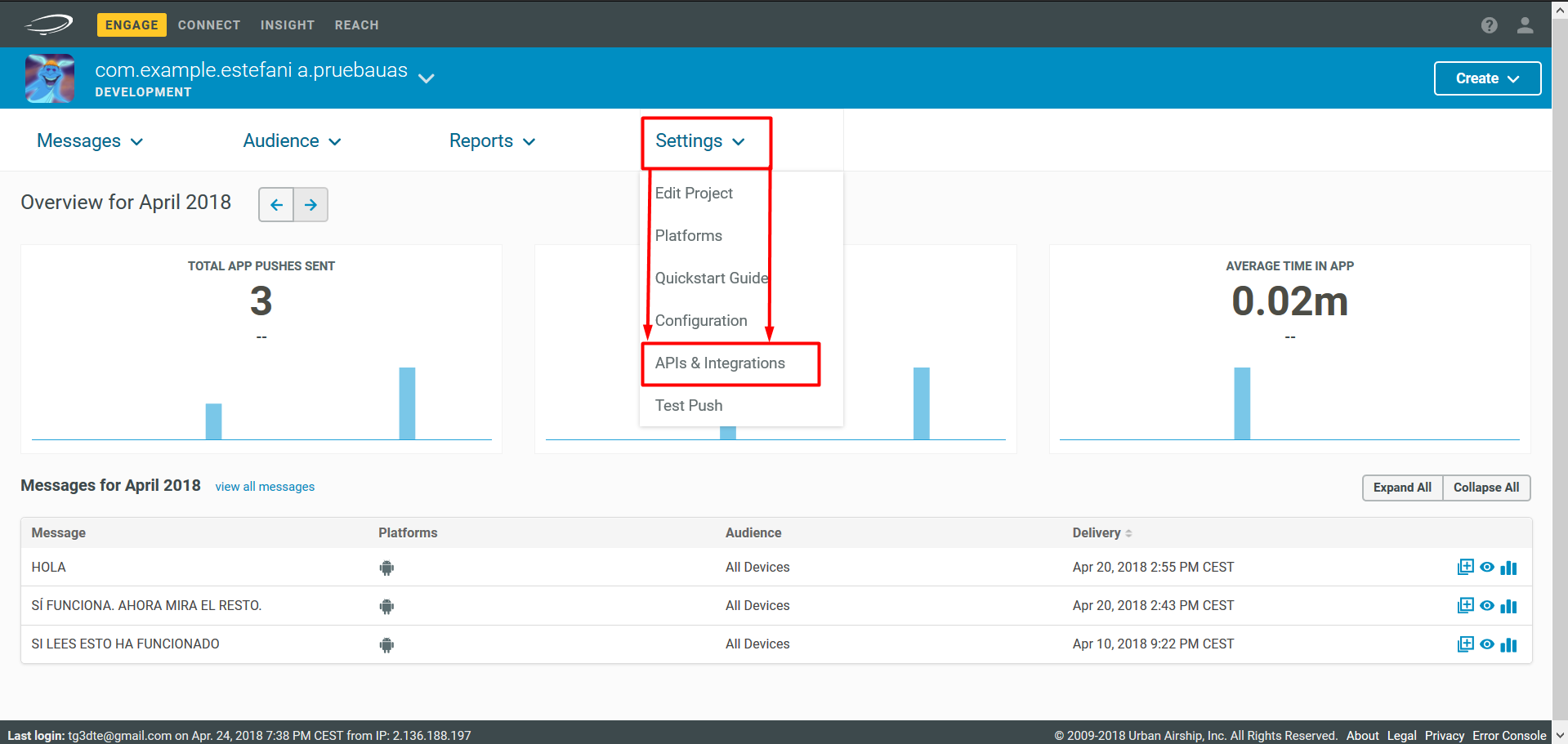
notificationAccentColor = #ff0000

# Optional, set the default notification channel

notificationChannel = customChannel

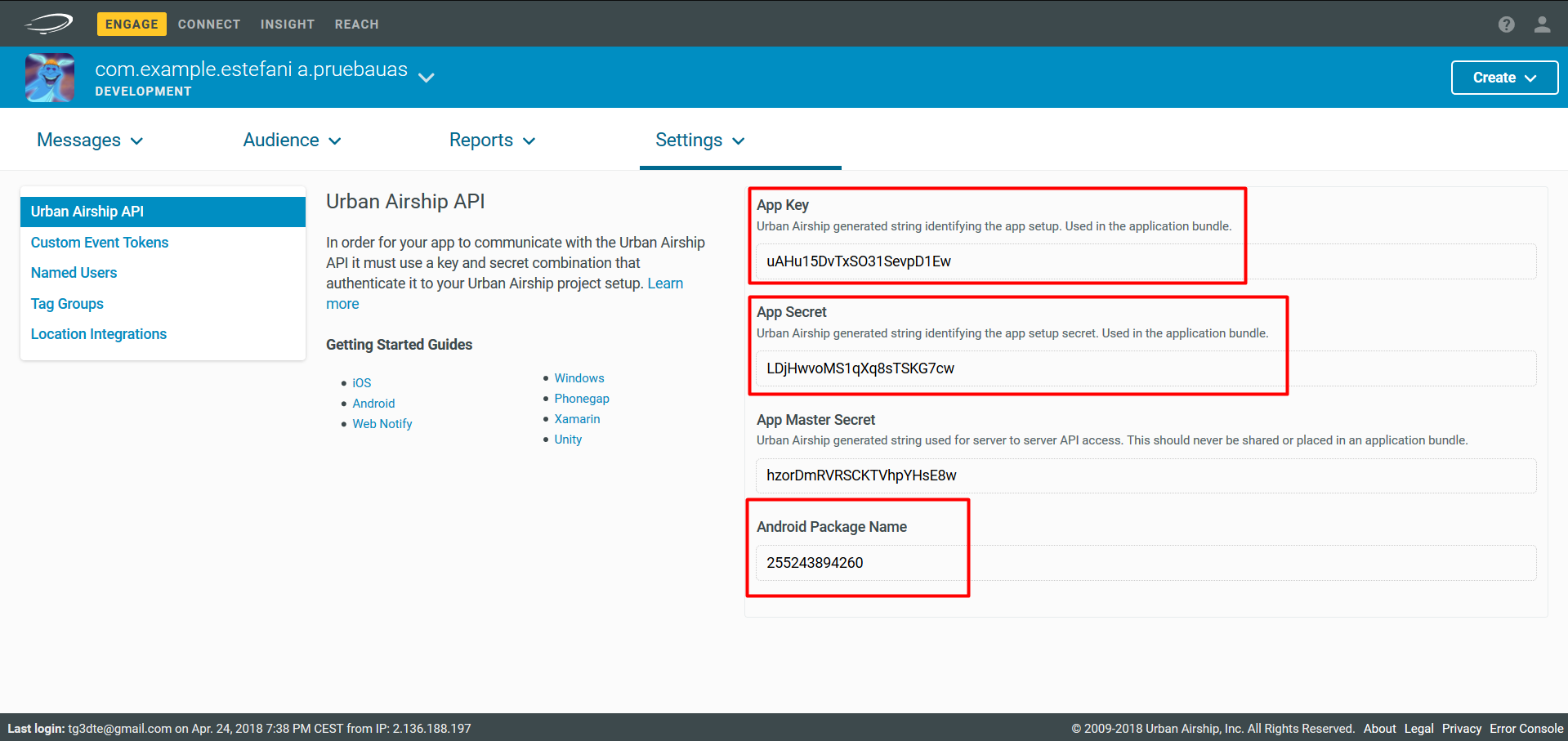


Cuando lo hayamos pegado, el archivo nos pedirá las claves que nos proporciona Urban Airship en su dashboard. Lo abrimos y pinchamos en la barra superior en “ENGAGE” y se nos abrirá esta ventana:



Pinchamos donde indicamos en la imagen: Settings > APIs & Integrations.

Podremos observar que en esta ventana es donde obtenemos las claves que necesitamos para el archivo de Android Studio que hemos creado:



Como hemos elegido el modo “Development” las líneas de Production las borramos. En estas líneas:

developmentAppKey = Your Development App Key

developmentAppSecret = Your Development App Secret

Pegaremos la clave de la dashboard de Urban Airship que coorresponda con la “App key” y la clave de “App Secret”.

En esta línea:

fcmSenderId = Your Google API Project Number

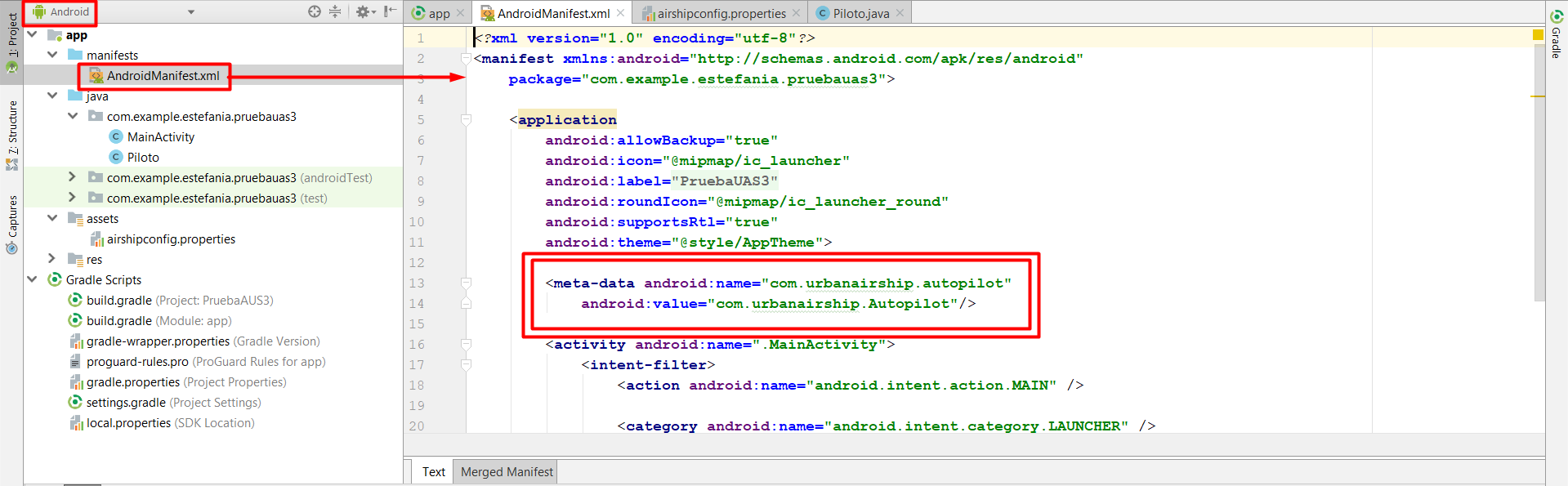
Debemos pegar la clave que corresponde a Android Package Name en Urban Airship.

Las demás líneas del archivo las dejamos como están.

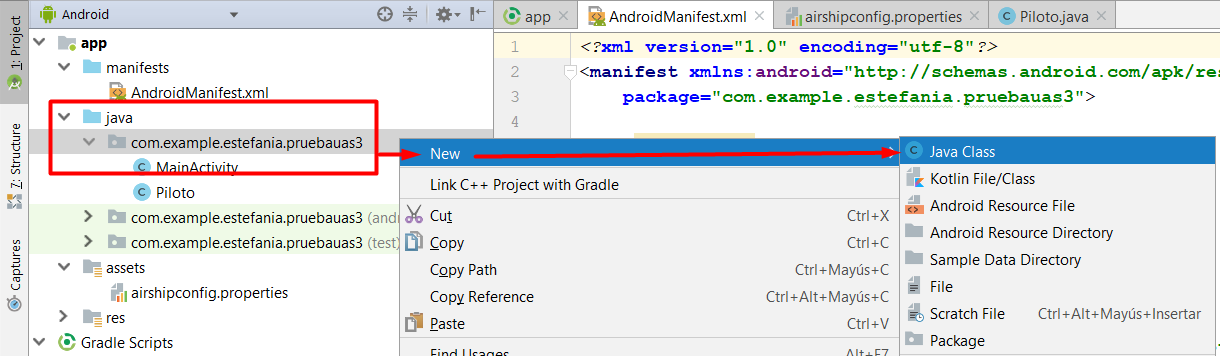
3. Una vez hayamos hecho esto correctamente, volvemos a poner la vista izquierda del proyecto en “Android” y abrimos el archivo “manifest”. Pegaremos en este archivo las siguientes líneas de código:

<meta-data android:name="com.urbanairship.autopilot"

android:value="com.urbanairship.Autopilot"/>

Se debería ver de este modo:

1. Ahora tenemos que crear una clase Java nueva en la carpeta que se llama igual que el paquete del proyecto. La llamaremos “Piloto”.



En esta clase deberemos pegar el siguiente código:

public class Piloto extends AutoPilot {

@Override

public void onAirshipReady(@NonNull UAirship airship) {

airship.getPushManager().setUserNotificationsEnabled(true);

// Android O

if (Build.VERSION.SDK\_INT >= 26) {

Context context = UAirship.getApplicationContext();

NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);

NotificationChannel channel = new NotificationChannel("customChannel",

context.getString(R.string.custom\_channel\_name),

NotificationManager.IMPORTANCE\_DEFAULT);

notificationManager.createNotificationChannel(channel);

}

}

@Override

public AirshipConfigOptions createAirshipConfigOptions(@NonNull Context context) {

AirshipConfigOptions options = new AirshipConfigOptions.Builder()

.setDevelopmentAppKey("Your Development App Key")

.setDevelopmentAppSecret("Your Development App Secret")

.setProductionAppKey("Your Production App Key")

.setProductionAppSecret("Your Production App Secret")

.setInProduction(!BuildConfig.DEBUG)

.setFcmSenderId("Your Google API Project Number") // FCM/GCM sender ID

.setNotificationIcon(R.drawable.ic\_notification)

.setNotificationAccentColor(ContextCompat(getContext(), R.color.accent))

.setNotificationChannel("customChannel")

.build();

return options;

}

}

**Se agregarán las librerías que sean necesarias.** Quedará de esta manera:

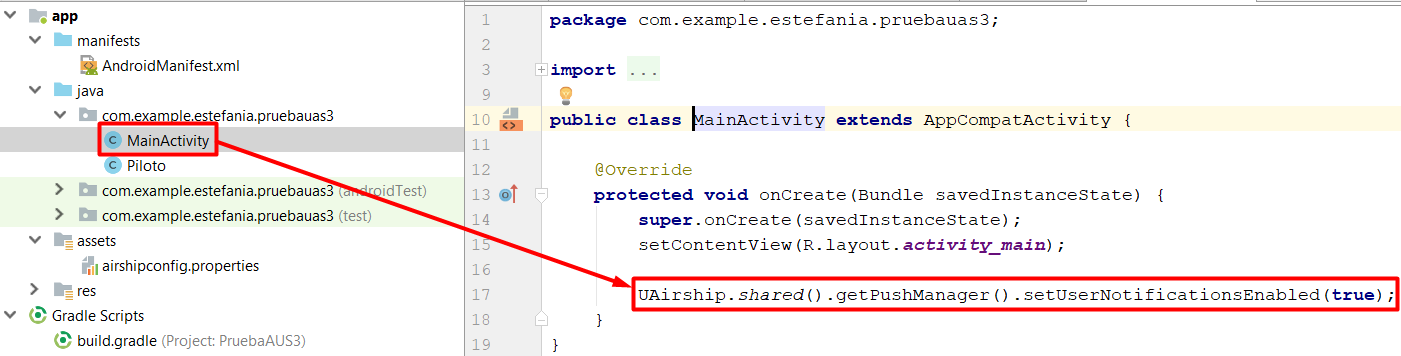


Tendremos que añadir las mismas claves que hemos añadido anteriormente en el archivo “airshipconfig.properties”.

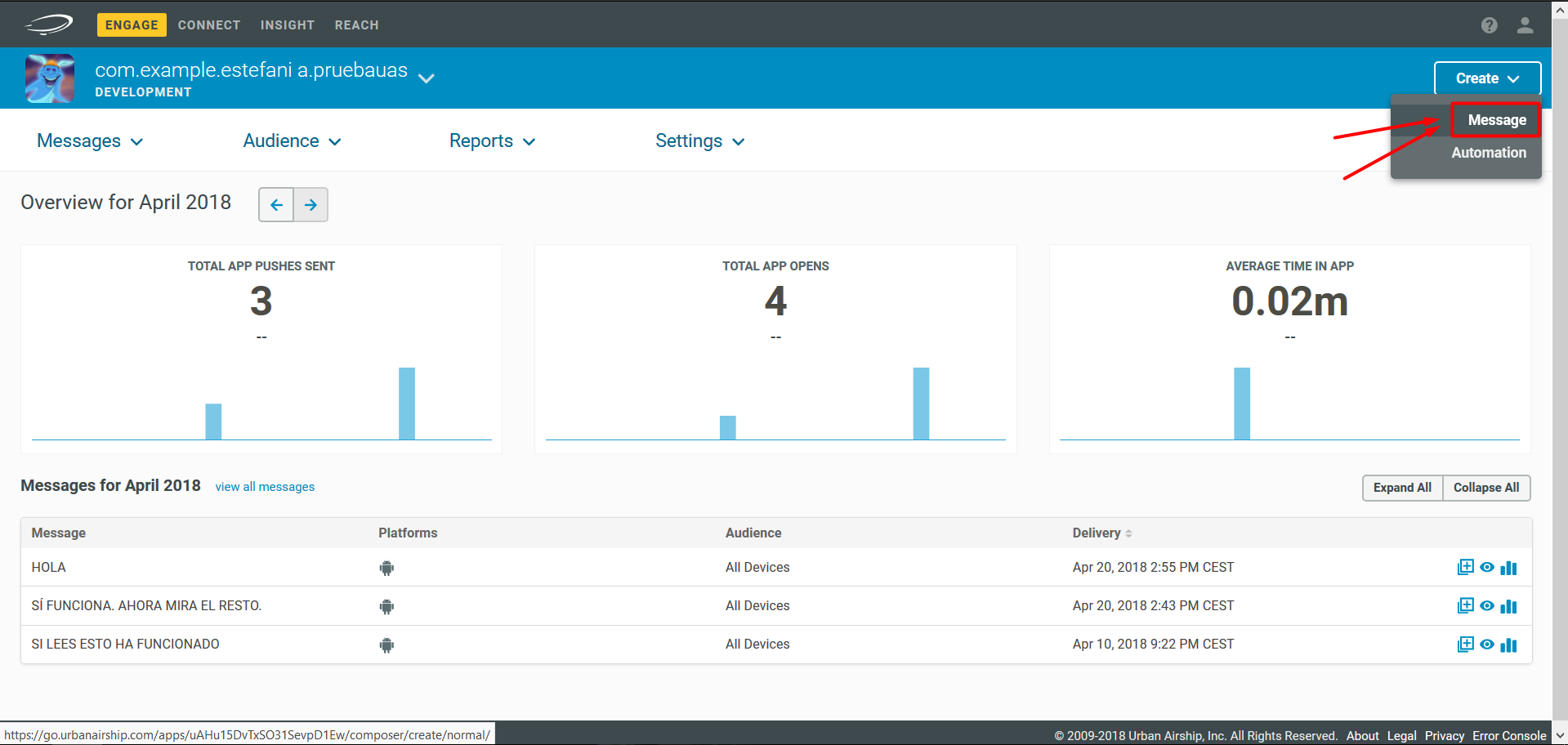
1. El siguiente paso que tenemos que hacer es ir a la clase MainActivity.class y pegar esta línea de código en la función onCreate():

UAirship.shared().getPushManager().setUserNotificationsEnabled(true);

Quedando de esta manera:



## 5.5.4 Enviar una notificación con Urban Airship.

1. Abrimos la dashboard en la pestaña “ENGAGE” y damos click al botón “Create” y después a “Message”:

2. Las opciones son similares a OneSignal. Primero nos ofrece tres maneras de enviar la notificación:

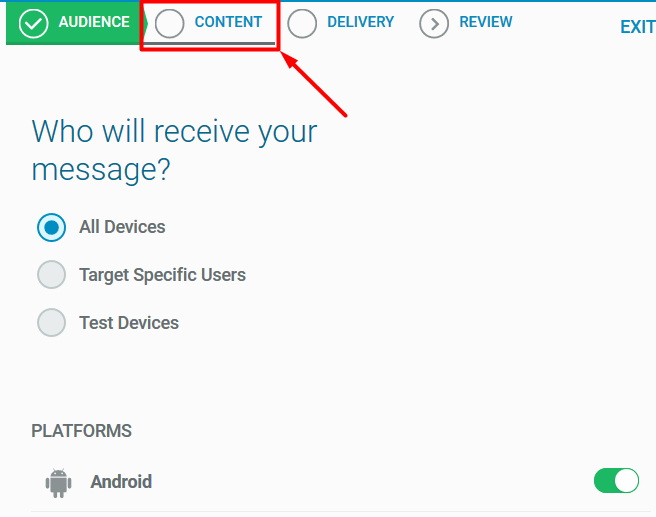
**a. All devices:** podremos enviar la notificación a todos los usuarios que tengan la aplicación instalada en sus dispositivos moviles.

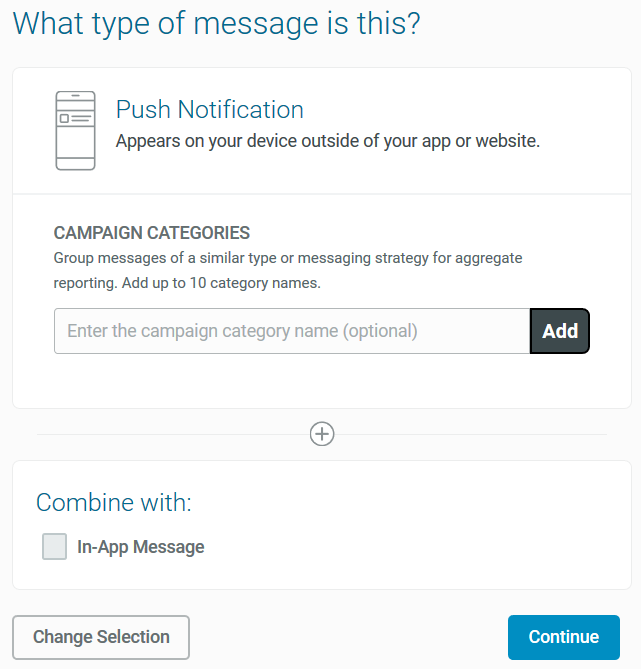
**b. Target Specific Users:** Podremos diseñar segmentos de usuarios dependiendo de su versión de Android, de su localización, de la versión de la aplicación… etc.

**c. Test Devices:** podremos enviar notificaciones a dispositivos de prueba.

En el manual elegiremos “All devices”.

3. Una vez elegido, clicamos en “Content”:



4. Ahora podemos elegir tres maneras en las que saldrán la notificación:

a. Push Notification: Aparecerá cuando no se esté usando la aplicación.

b. In-App Message: Aparecerá cuando el usuario abra la aplicación.

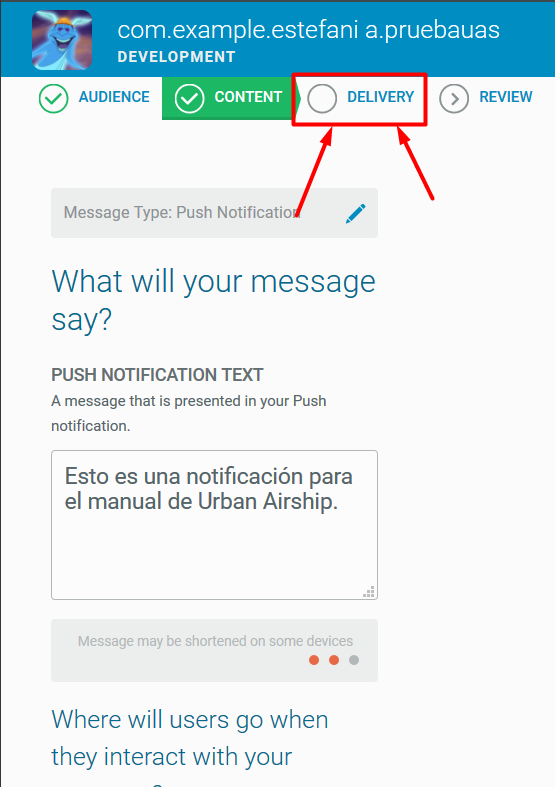
c. Silent Push Notification: Será una notificación texto.

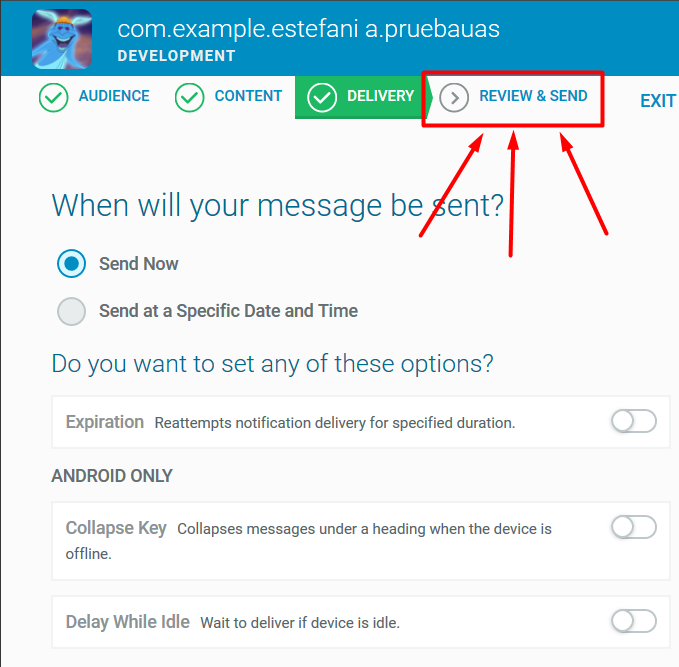
En el manual elegiremos “Push Notification”. Podremos observar que podemos crear una categoría, para futuras notificaciones de este tipo. Esto es opcional. También podremos combinarlo con una notificación del tipo “In-App Message”.

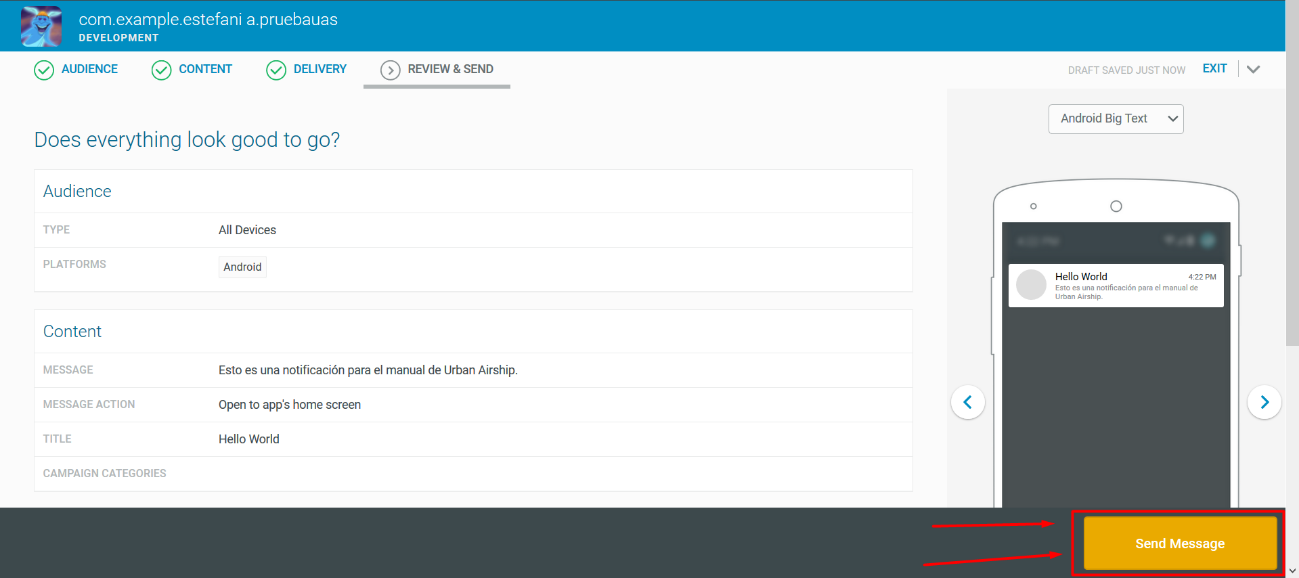
Cuando hayamos terminado, damos click a “Continue”.

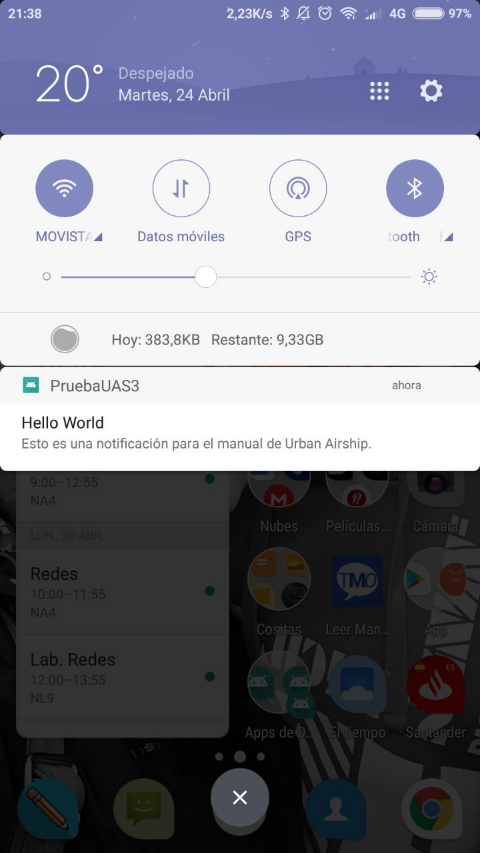
5. La siguiente ventana es similar a OneSignal. Aquí podremos editar la notificación que llegará a los dispositivos móviles de los usuarios. Al final de la página podremos ver la vista previa de la notificación que hemos diseñado. Podremos elegir qué acción hará la notificación cuando la cliquemos, como abrir la aplicación, una página web, compartir con un contacto… etc.

1. Cuando hayamos diseñado nuestra notificación a nuestro gusto. Damos click a “Delivery”.



En esta nueva pestaña, podremos elegir cuándo enviar la notificación, que desaparezca después de un tiempo u opciones adicionales solo para dispositivos Android. Nosotros elegiremos “Send Now”. Cuando hayamos finalizado damos click a “Review & Send”.

1. Ahora nos mostrará un resumen de las opciones que hemos elegido para esta notificación. Si vemos que está todo corecto, damos a “Send Message”.
2. Si hemos realizado todo correctamente, los usuarios que tengan la aplicación instalada tienen que haber recibido la notificación.



# 6. Comparación de las dos implementaciones

Se trata de dar valores a los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos.

## 6.1 Evaluación de los criterios en la implementación usando la tecnología A

Debe incluir al menos una tabla con la siguiente estructura.

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Criterio 1 |  |
| Criterio 2 |  |
| … |  |
| Criterio N |  |

Y algunos comentarios aclaratorios sobre aquellos criterios cuyo valor indicado en la tabla no sea suficiente para entenderlo.

## 6.2 Evaluación de los criterios en la implementación usando la tecnología B

# 7. Comparación de la implementación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

| **CRITERIOS** | **TECNOLOGÍA A** | **TECNOLOGÍA B** | **COMENTARIOS** |
| --- | --- | --- | --- |
| 1 |  |  |  |
| 2 |  |  |  |
| … |  |  |  |
| N |  |  |  |

# 8. Conclusiones

A partir de la información incluida en el apartado 7 y de la experiencia al realizar el trabajo, el grupo debe estar en condiciones de manifestar su opinión sobre la implementación del sistema utilizando ambas tecnologías, y debe plasmarla en este apartado, indicando las ventajas e inconvenientes más relevantes de utilizar una u otra tecnología para implementar el sistema.

---------------------------

(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)