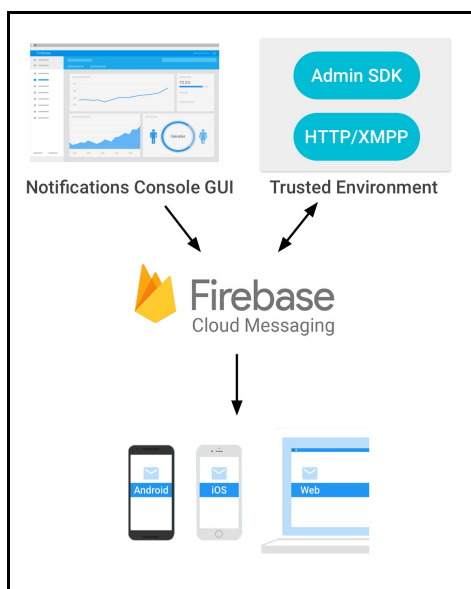


Firebase cloud messaging

Firebase Cloud Messaging (FCM) es una solución de mensajería multiplataforma que nos permite enviar mensajes de forma segura a nuestra app Android desde un servidor.

Esto nos evita tener un servicio corriendo en nuestra aplicación que este consultando al server de forma periódica si existen nuevas notificaciones.



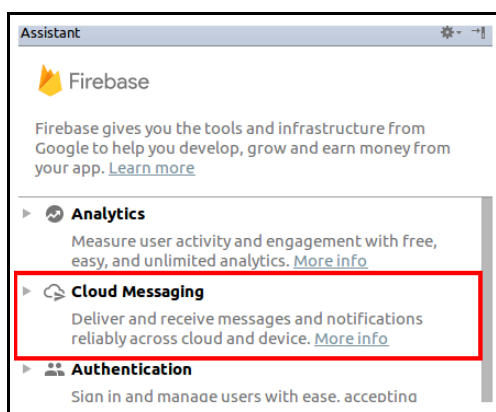
La arquitectura del sistema se observa en la figura, en donde es posible utilizar en nuestro server la AdminSDK o el protocolo HTTP/XMPP para enviar mensajes por medio de Firebase hacia los clientes (Aplicaciones Android, iOS o Web). También es posible utilizar la consola de Firebase que nos permite hacer envíos desde una interface web.

Los mensajes pueden ser dirigidos a un dispositivo en particular, a un grupo de dispositivos, o a un "topic" específico, al cual ciertos dispositivos se subscriben.

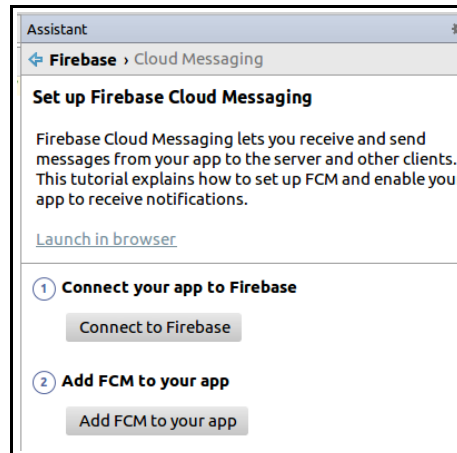
Configuración de la app Android con Firebase

Creamos un nuevo proyecto en Android Studio y luego seleccionamos en el menú "Tools" la opción "Firebase".

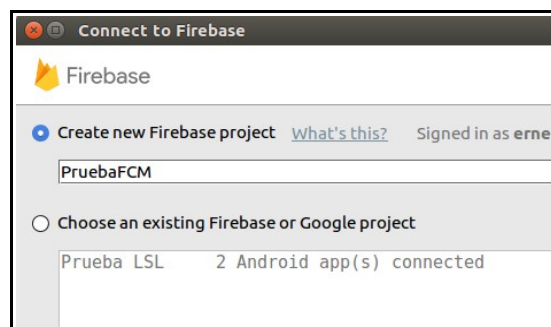
En el asistente que nos aparece a la derecha, seleccionamos "Cloud Messaging"



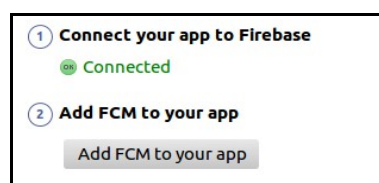
Presionamos el botón “Connect to Firebase”. Se abrirá un navegador, el cual nos pedirá que ingresemos con nuestra cuenta de Google. Esto permitirá a Android Studio dar de alta un nuevo proyecto y asignar la app actual al mismo, para poder funcionar con Firebase.



Luego aparecerá una ventana para elegir a qué proyecto se asigna la app, si no tenemos un proyecto creado, podemos asignar un nombre para el mismo, de lo contrario deberemos elegir el proyecto que ya existe, ya que no es posible tener más de un proyecto de forma gratuita.

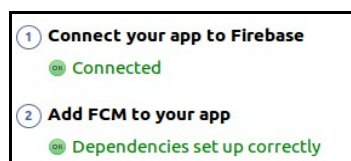


Seleccionamos el proyecto existente y presionamos “Connect to Firebase”. Android Studio se conectará y dará de alta la app en la consola de Firebase.



El estado de la opción 1 deberá ser “Connected”.

Presionamos el botón de la opción 2 “Add FCM to your app”. Una ventana nos indicará los cambios que se realizarán en nuestra app (Se agregará la biblioteca de firebase en el gradle script). Presionamos el botón “Accept Changes”



La indicación del punto 2 deberá ser “Dependencies set up correctly”

Escuchando mensajes desde Android

Para recibir mensajes desde el server, deberemos crear un servicio que se lanzará automáticamente cuando se reciba un mensaje, a pesar de que nuestra app no se esté ejecutando.

Hacemos click sobre el package de nuestra app y seleccionamos New > Service > Service. Le ponemos de nombre "MyFirebaseMessagingService" y presionamos el botón "Finish".

Deberá aparecer una clase Java la cual hereda de "Service". Cambiaremos esta herencia por la clase "FirebaseMessagingService" y eliminaremos el método "onBind" de dicha clase.

Para recibir los mensajes, reescribiremos el método "onMessageReceived" en la clase:

```
@Override
public void onMessageReceived(RemoteMessage remoteMessage) {
    Log.d("msg", "From: " + remoteMessage.getFrom());
    // Check if message contains a data payload.
    if (remoteMessage.getData().size() > 0) {
        Log.d("msg", "Message data payload: " + remoteMessage.getData());
    }
    // Check if message contains a notification payload.
    if (remoteMessage.getNotification() != null) {
        Log.d("msg", "Message Notification Body: " +
            remoteMessage.getNotification().getBody());
    }
}
```

Cuando se reciba un mensaje enviado por el server, se lanzará este servicio y se ejecutará este método si la app esta en foreground. Cuando la aplicación este en background, se creará una notificación automáticamente, al hacer click en ella, se lanzará la activity que se lanza por default de nuestra app. En el intent, viajarán los datos de la notificación y podrán leerse desde la activity.

Para que el servicio pueda ser lanzado, deberemos agregar en el Manifest de nuestra app, un intent filter indicando que este servicio es capaz de recibir mensajes. Para ello agregamos en el manifest, dentro del tag de "service" lo siguiente:

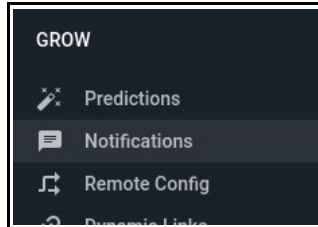
```
<service
    android:name=".MyFirebaseMessagingService"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT"/>
    </intent-filter>
</service>
```

Al recibir un mensaje, lo veremos impreso por el log del sistema.

Enviando mensajes con la consola de Firebase

Ingresamos a la consola de Firebase <https://console.firebase.google.com> y seleccionamos el proyecto.

En el menú de la izquierda, seleccionamos “Notifications”

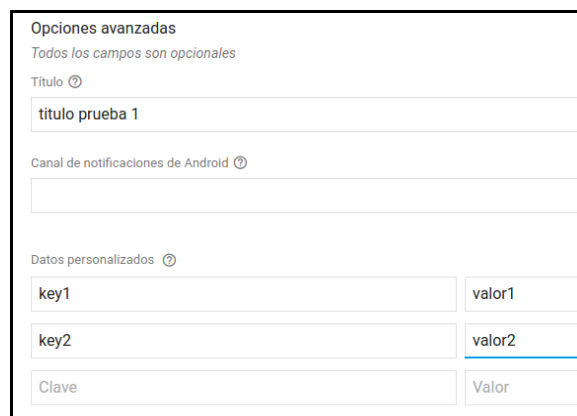


Aparecerán la lista de mensajes enviados, presionamos el botón “Mensaje nuevo” y aparecerá un formulario para enviar un mensaje:



Introducimos un texto y en el combo seleccionamos nuestra aplicación. Todos los dispositivos con la app instalada recibirán el mensaje.

Seleccionamos “Opciones avanzadas”, las cuales nos permitirán enviar datos en formato JSON de la forma clave-valor:



Al presionar “ENVIAR MENSAJE” el mismo llegará a la app de forma instantánea.

NOTA: Deberemos ejecutar la app en un dispositivo real, de lo contrario no funcionará.

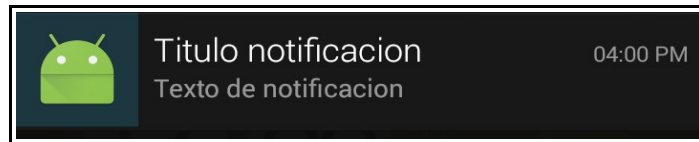
Con la app en **foreground**, al enviar el mensaje, deberemos ver en el log:

```

D/msg: Message data payload: {key1=valor1, key2=valor2}
D/msg: Message Notification Body: prueba 1 fcm

```

Con la app está en **background**, el enviar el mensaje, deberemos ver una notificación. Al hacer click sobre la misma, se lanzará la activity ppal.



Allí podremos leer los valores enviados, obteniendo el objeto Bundle a partir del intent, si colocamos el siguiente código en el método onCreate, podremos ver todos los valores recibidos:

```

if (getIntent().getExtras() != null) {
    for (String key : getIntent().getExtras().keySet()) {
        String value = getIntent().getExtras().getString(key);
        Log.d("actv", "Key: " + key + " Value: " + value);
    }
}

```

```

Key: google.sent_time Value: null
Key: clave1 Value: valor 1
Key: clave2 Value: valor 2
Key: clave3 Value: 67
Key: from Value: 300026467077
Key: google.message_id Value: 0:1510512714731315%a1540549a1540549
Key: collapse_key Value: com.lsl.pruebafcm

```

Como se observa, los valores enviados en formato clave-valor, se cargan directamente en el Bundle, por lo que si se envió en el mensaje la clave "clave1" con cierto valor, el mismo se leerá desde la activity con:

```
String value = getIntent().getExtras().getString("clave1");
```

Estos mensajes también pueden ser enviados desde un servidor propio utilizando la adminSDK o el protocolo HTTP/XMPP. Para más información:

<https://firebase.google.com/docs/cloud-messaging/http-server-ref>