



Firestore Cloud Messaging

Mercedes Wyss (fb)
@itrjwyss (twitter, github)




¿Qué son Push Notifications?



- Mensaje que se muestra en un dispositivo móvil.
- Mecanismo que permite enviar información a un dispositivo móvil desde una aplicación servidor.



Historia

- 
- 2009 Apple lanzó “Apple Push Notification Server (APNs)”
 - 2010 Google lanzó “Google Cloud to Device Service (C2DM)”
 - 2013 Google introduce “Rich Notifications”
 - 2014 Apple los implementa y poco después extiende lo push a Apple Watch.

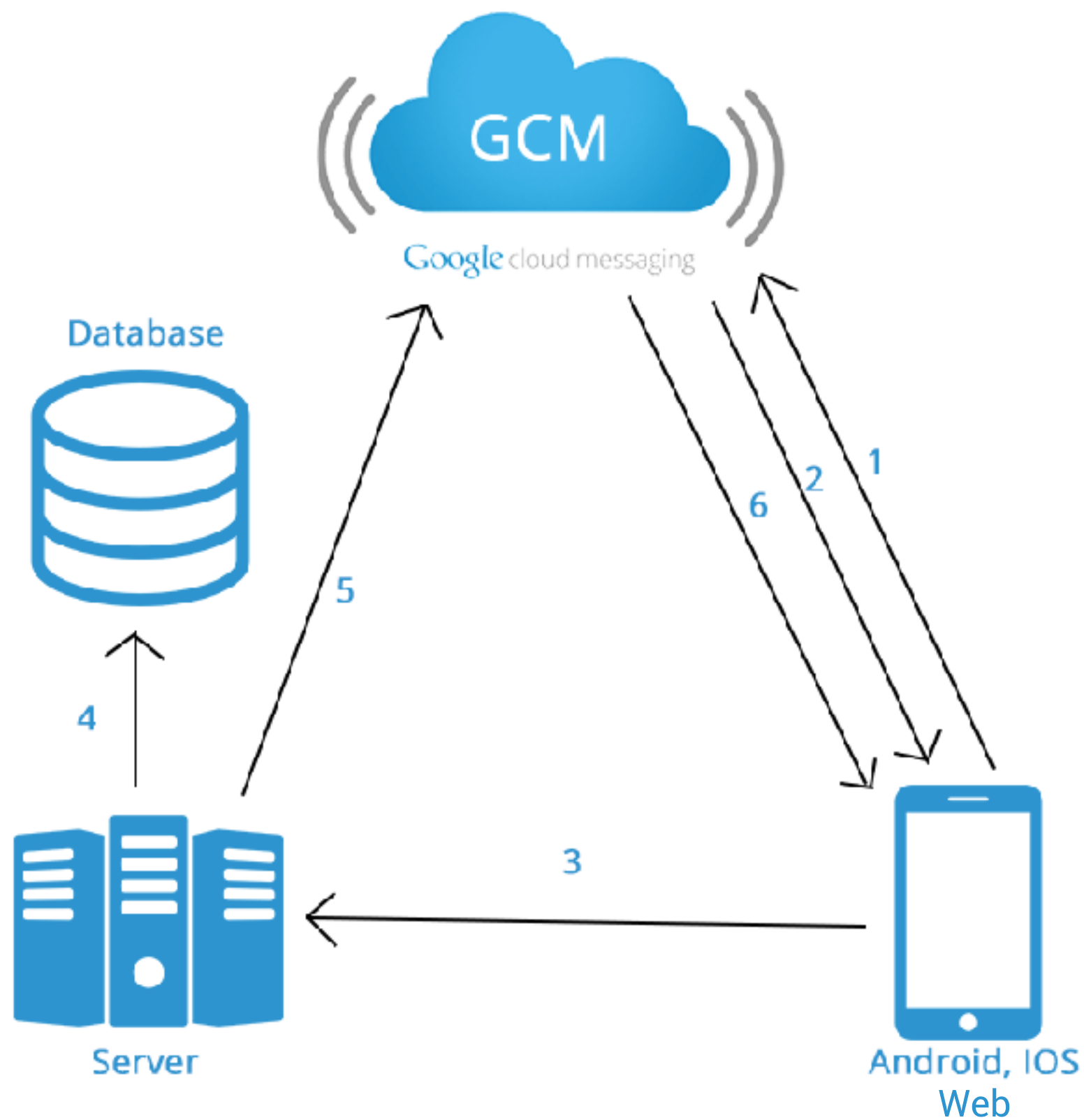


¿Cómo Funciona?



Proceso

1. Dispositivo de registra en FCM.
2. Dispositivos recibe Token de FCM que lo identificará.
3. Dispositivo envía dicho Token al Servidor.
4. Servidor lo almacena en la Base de Datos.
5. Servidor solicita a FCM envíe un Push Notification al dispositivo haciendo uso del Token.
6. FCM envía push notificacion al dispositivo.





Firebase

Cloud Messaging

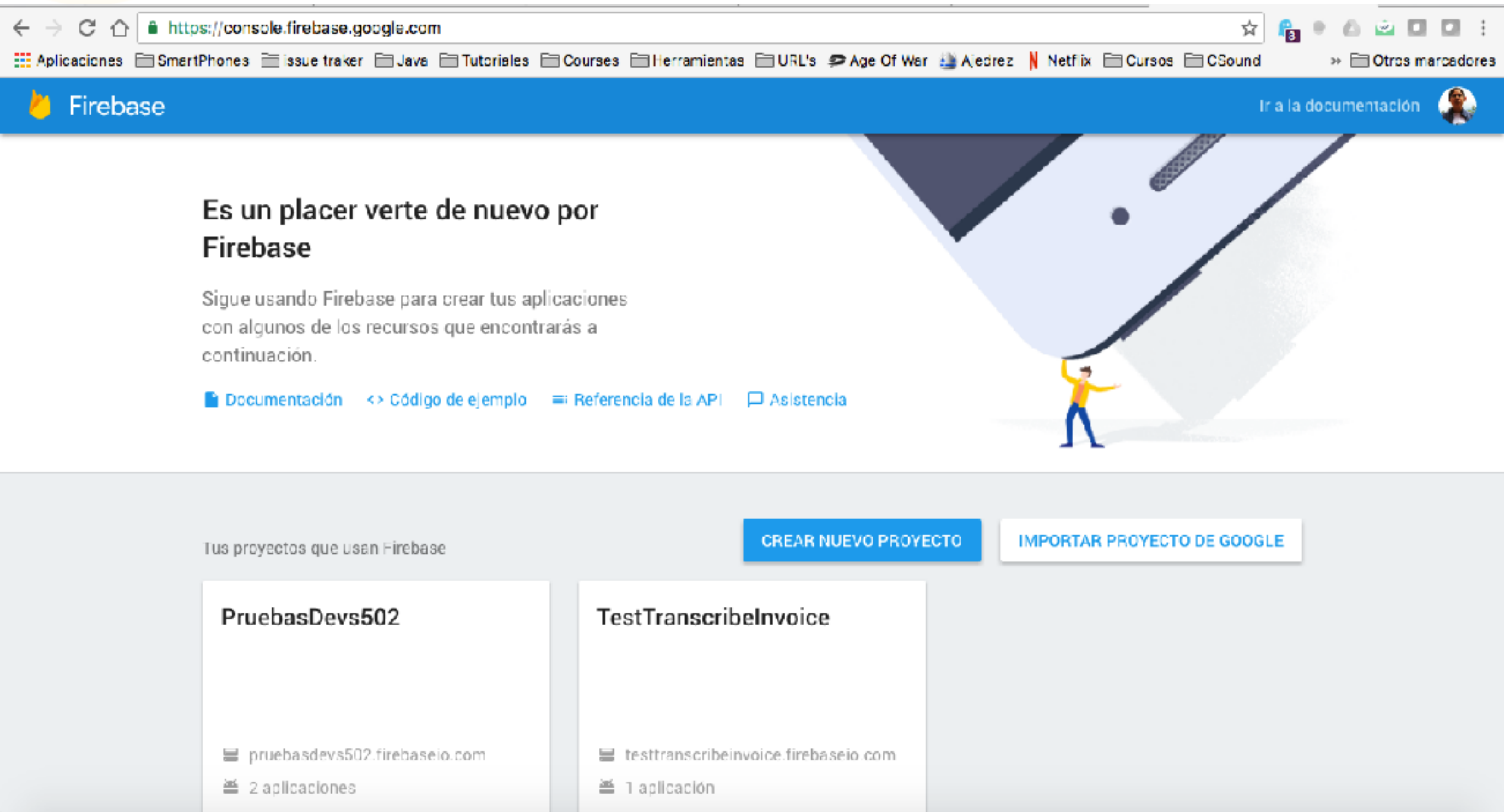


Comunidad Desarrolladores en Tecnologías
< Google > en Guatemala



¿Cómo Empezamos?

1. Ir a la consola de Firebase



The screenshot shows the Firebase console interface. At the top, there's a browser window with the URL `https://console.firebase.google.com`. Below the browser, a blue header bar contains the Firebase logo and a link to the documentation. The main content area features a large illustration of a person standing next to a giant pencil, with the text "Es un placer verte de nuevo por Firebase" and a message about using Firebase resources. Below this, there are links for "Documentación", "Código de ejemplo", "Referencia de la API", and "Asistencia". At the bottom, a section titled "Tus proyectos que usan Firebase" displays two project cards: "PruebasDevs502" and "TestTranscribeInvoice", each showing its domain and the number of applications.

Es un placer verte de nuevo por
Firebase

Sigue usando Firebase para crear tus aplicaciones con algunos de los recursos que encontrarás a continuación.

[Documentación](#) [Código de ejemplo](#) [Referencia de la API](#) [Asistencia](#)

Tus proyectos que usan Firebase

PruebasDevs502

`pruebasdevs502.firebaseio.com`

2 aplicaciones

TestTranscribeInvoice

`testtranscribeinvoice.firebaseio.com`

1 aplicación

[CREAR NUEVO PROYECTO](#) [IMPORTAR PROYECTO DE GOOGLE](#)

2. Creamos un proyecto

CREAR NUEVO PROYECTO

IMPORTAR PROYECTO DE GOOGLE

Crear proyecto

Nombre del proyecto

Mi fabuloso proyecto

País/Región ?

Estados Unidos

De forma predeterminada, los datos de Firebase Analytics mejoran otras funciones de Firebase y productos de Google. En la configuración puedes controlar cómo se comparten dichos datos en cualquier momento. [Más información](#)

CANCELAR

CREAR PROYECTO

Crear proyecto

Nombre del proyecto

FCMExample

País/Región ?

Guatemala

De forma predeterminada, los datos de Firebase Analytics mejoran otras funciones de Firebase y productos de Google. En la configuración puedes controlar cómo se comparten dichos datos en cualquier momento. [Más información](#)

CANCELAR

CREAR PROYECTO

3. Agregamos Firebase a nuestra app Android

The screenshot shows the Firebase console interface. At the top, the 'Firebase' logo is on the left, and 'FCMExample' with a dropdown arrow and a link to 'Ir a la documentación' are on the right. Below the header, a left sidebar contains navigation links: 'FCMExample' (active), 'Analytics', 'DEVELOP', 'Auth', 'Database', 'Storage', 'Hosting', 'Remote Config', 'Test Lab', 'Crash', 'GROW', and 'Notifications'. The main area is titled 'Overview' and features a welcome message: 'Te damos la bienvenida a Firebase. Empieza aquí.' Below this, three large colored circles with icons represent different platforms: a blue circle with 'iOS' for 'Añade Firebase a tu aplicación de iOS', a green circle with an Android robot for 'Añade Firebase a tu aplicación de Android', and a pink circle with code symbols for 'Añade Firebase a tu aplicación web'. At the bottom of the main area, a section titled 'Descubre Firebase' contains three horizontal image banners: a person with an upward arrow, a person with a medal, and server racks.

Añade Firebase a tu aplicación de Android

1

Introduce los detalles
de la aplicación

2

Copiar archivo de
configuración

3

Añadir a build.gradle

Nombre del paquete ⓘ

Certificado de firma de depuración SHA-1 (opcional) ⓘ

Obligatorio para enlaces dinámicos, invitaciones y compatibilidad con inicio de sesión de Google en Auth. Edita SHA-1 en la configuración.

CANCELAR

AÑADIR APLICACIÓN

*descarga
google-services.json para
tu aplicación*

Visualización de la huella digital del certificado de depuración

1. Busca tu archivo de depuración de depósito de claves. El nombre del archivo es `debug.keystore` y se crea la primera vez que compilas tu proyecto. De manera predeterminada, se guarda en el mismo directorio que tus archivos del Dispositivo virtual de Android (AVD):

- **OS X y Linux:** `~/.android/`
- **Windows Vista y Windows 7:** `C:\Users\your_user_name\.android\`

Si usas Eclipse con ADT y no estás seguro de dónde se encuentra tu archivo de depuración de depósito de claves, puedes seleccionar **Windows > Prefs > Android > Build** para verificar la ruta de acceso completa.

2. Indica la huella digital SHA-1:

- Para Linux u OS X, abre una ventana de la terminal y escribe lo siguiente:

```
keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey
```

- Para Windows Vista y Windows 7, ejecuta lo siguiente:

```
keytool -list -v -keystore "%USERPROFILE%.android\debug.keystore" -alias al
```

<http://goo.gl/9QPYkW>

Añade Firebase a tu aplicación de Android



Introduce los detalles
de la aplicación

2

Copiar archivo de
configuración

3

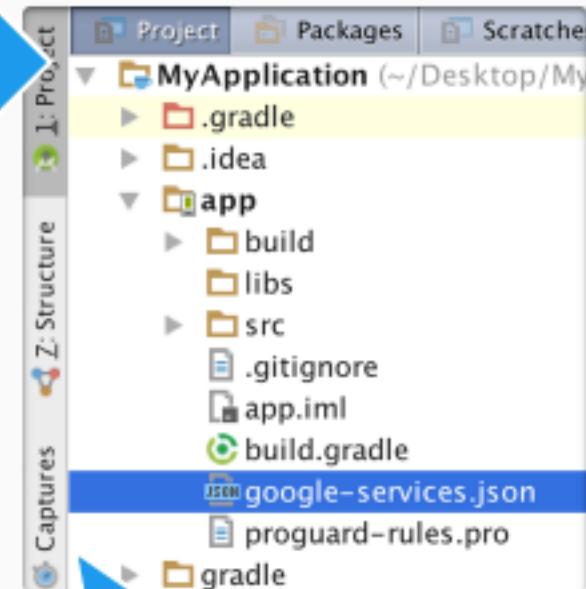
Añadir a build.gradle

Cambia a la vista **Proyecto** de Android Studio para ver el directorio "root" de tu proyecto.

Mueve el archivo **google-services.json** que acabas de descargar al directorio "root" del módulo de tu aplicación de Android.



google-services.json



¿Ya has añadido las dependencias?
[Saltar a la consola](#)

CONTINUAR

google-services.json - FirebaseCloudMessaging - [~/Documents/gitrepost/aKnowledgeJourney/Android/FirebaseCloudMessaging]

Project: FirebaseCloudMessaging (~/.Documents)

- gradle
- idea
- app
 - build
 - libs
 - src
 - .gitignore
 - app.iml
 - build.gradle
 - google-services.json
 - proguard-rules.pro
- build
- gradle
 - .gitignore
 - build.gradle
 - FirebaseCloudMessaging.iml
 - gradle.properties
 - gradlew
 - gradlew.bat
 - local.properties
 - settings.gradle
- External Libraries

```
{
  "project_info": {
    "project_number": "22189759802",
    "firebase_url": "https://fcmexample-792e5.firebaseio.com",
    "project_id": "fcmexample-792e5",
    "storage_bucket": "fcmexample-792e5.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:22189759802:android:68beb2c2668e686a",
        "android_client_info": {
          "package_name": "com.powerwtechnology.firebasecloudmessaging"
        }
      },
      "oauth_client": [
        {
          "client_id": "22189759802-8rsrrlnokfb9vh6thjpiq953pbavu40o.apps.googleusercontent.com",
          "client_type": 1,
          "android_info": {
            "package_name": "com.powerwtechnology.firebasecloudmessaging",
            "certificate_hash": "442552833A3EC3EB744296D256476E3B0D10CA8A"
          }
        },
        {
          "client_id": "22189759802-976rcgvpeprhnu63d0hgid772pfc744r.apps.googleusercontent.com",
          "client_type": 3
        }
      ],
      "api_key": [
        {
          "current_key": "AIzaSyAQgHKAi0SYI_wuNiATQ6ytAG_wv3Kt2jQ"
        }
      ],
      "services": {
        "analytics_service": {
          "status": 1
        }
      }
    }
  ]
}
```

Messages | Terminal | Android Monitor | TODO

Gradle build finished in 42s 573ms (today 2:12 AM)

1:1 LF UTF-8 Context: <no context>

Añade Firebase a tu aplicación de Android




Introduce los detalles
de la aplicación



Copiar archivo de
configuración

3

Añadir a build.gradle

El complemento de los servicios de Google para [Gradle](#)  carga el archivo `google-services.json` que acabas de descargar. Para poder usar el complemento, debes modificar los archivos `build.gradle`.

1. `build.gradle` de proyecto (`<project>/build.gradle`):

```
buildscript {  
    dependencies {  
        // Add this line  
        classpath 'com.google.gms:google-services:3.0.0'  
    }  
}
```

2. `build.gradle` de aplicación (`<project>/<app-module>/build.gradle`):

```
...  
// Add to the bottom of the file  
apply plugin: 'com.google.gms.google-services'
```

incluye Firebase Analytics de forma predeterminada 

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

[Sync Now](#)

```
// Top-level build file where you can add configuration options common to all sub-projects

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.1.3'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files

        classpath 'com.google.gms:google-services:3.0.0'
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

[Sync Now](#)

```
apply plugin: 'com.android.application'
```

```
android {  
    compileSdkVersion 24  
    buildToolsVersion "24.0.1"
```

```
    defaultConfig {  
        applicationId "com.powerwtechnology.firebasecloudmessaging"  
        minSdkVersion 14  
        targetSdkVersion 24  
        versionCode 1  
        versionName "1.0"
```

```
    }  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  
        }  
    }  
}
```

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:24.2.0'
```

```
    }  
  
apply plugin: 'com.google.gms.google-services'
```

4. Sincronizamos Gradle

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

[Sync Now](#)

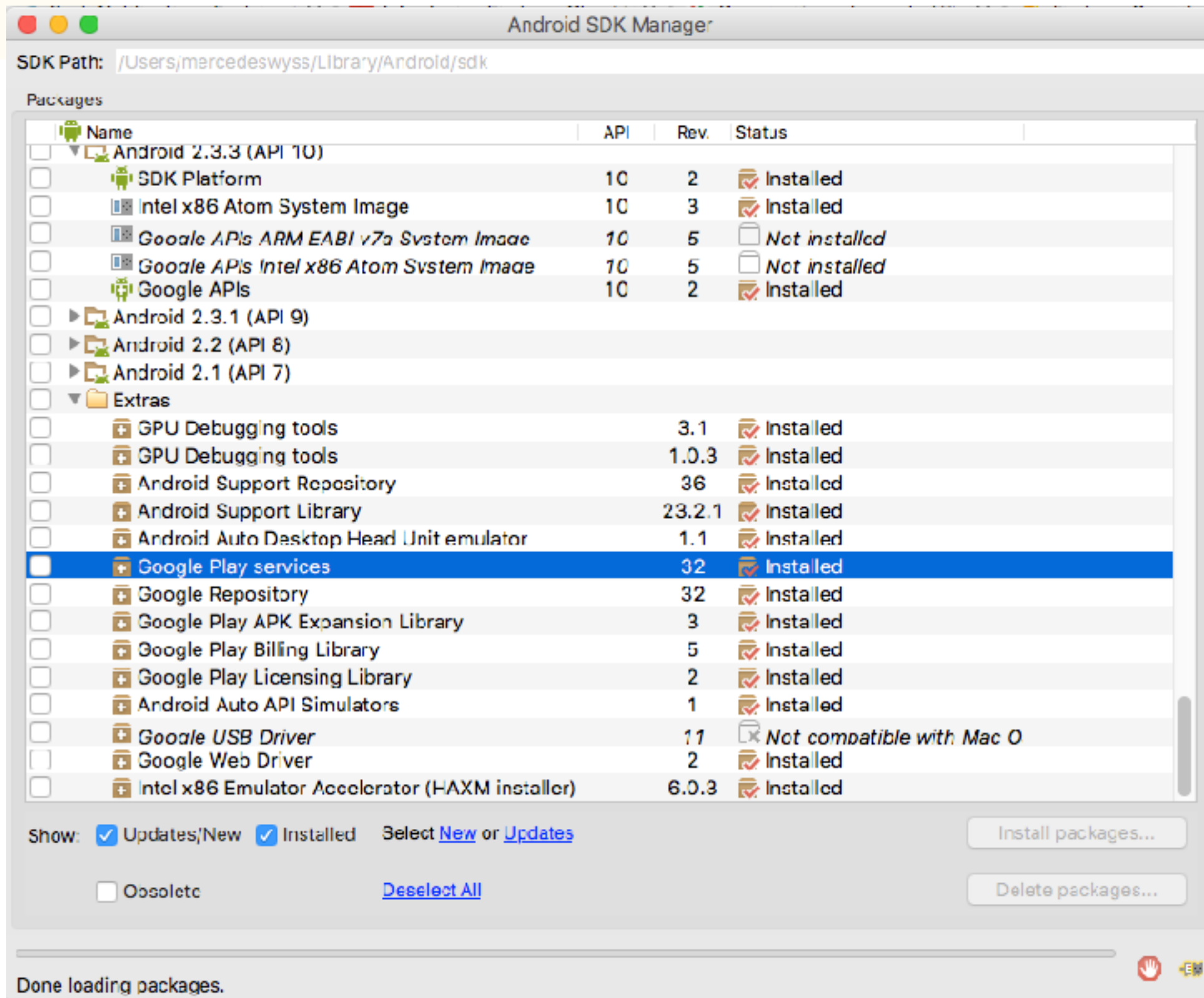


Comunidad Desarrolladores en Tecnologías
< Google > en Guatemala



Implementando el Cliente Android

1. Verificar la Instalación de Google Play Services



2. Importamos Firebase Messaging

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:24.2.0'  
  
    compile 'com.google.firebase:firebase-messaging:9.0.0'  
}
```


3. IntentService para el Registro

```
activity_splash_screen.xml x  SplashScreenActivity.java x  app x  RegistrationIntentService.java x
package com.powerwtechnology.firebasecloudmessaging;

import android.app.IntentService;
import android.content.Intent;

public class RegistrationIntentService extends IntentService {

    private static final String TAG = "RegIntentService";

    public RegistrationIntentService() {
        super(TAG);
    }

    @Override
    protected void onHandleIntent(Intent intent) {

    }

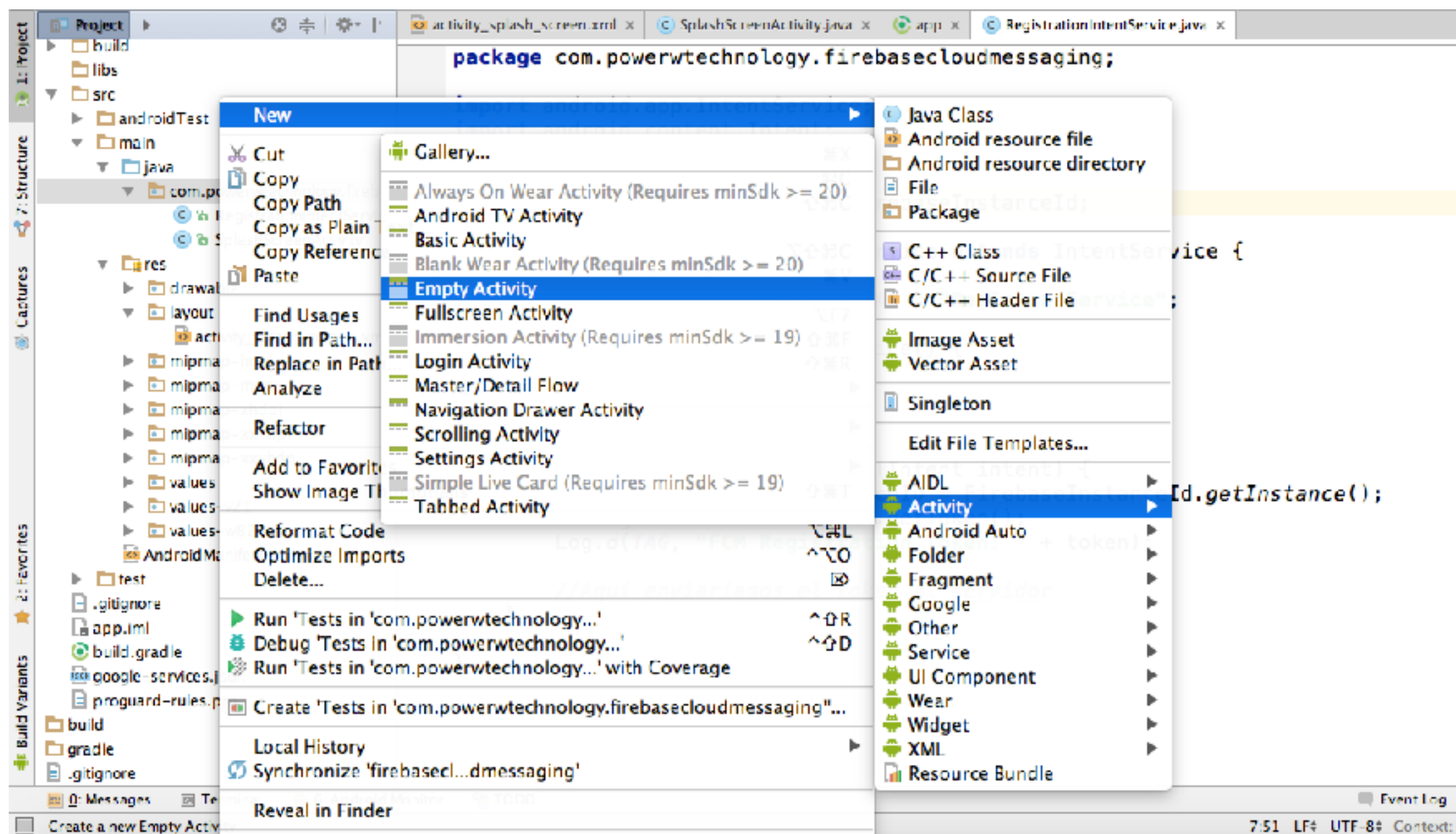
}
```

3. Implementamos el Registro

```
@Override
protected void onHandleIntent(Intent intent) {
    FirebaseInstanceId instanceID = FirebaseInstanceId.getInstance();
    String token = instanceID.getToken();
    Log.d(TAG, "FCM Registration Token: " + token);

    //Aquí enviaríamos el token al servidor
}
```

4. Creamos una Activity



```

public class MainActivity extends AppCompatActivity {

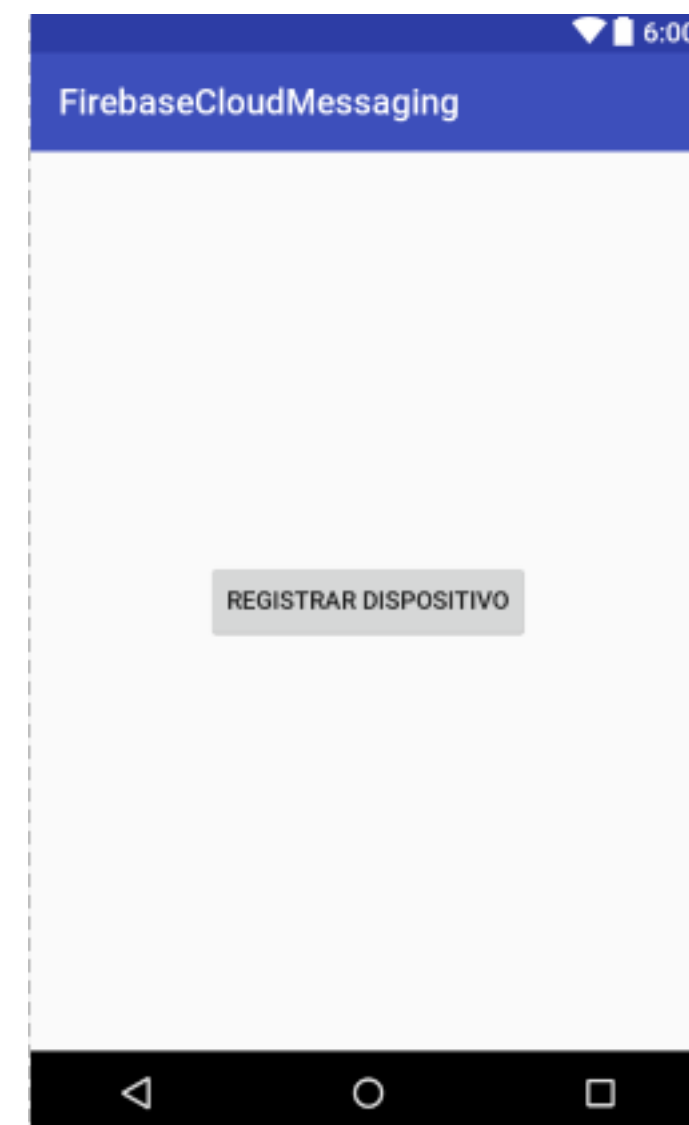
    private static final int PLAY_SERVICES_RESOLUTION_REQUEST = 1000;
    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    private boolean checkPlayServices() {
        GoogleApiAvailability apiAvailability = GoogleApiAvailability.getInstance();
        int resultCode = apiAvailability.isGooglePlayServicesAvailable(this);
        if (resultCode != ConnectionResult.SUCCESS) {
            if (apiAvailability.isUserResolvableError(resultCode)) {
                apiAvailability.getErrorDialog(this, resultCode, PLAY_SERVICES_RESOLUTION_REQUEST)
                    .show();
            } else {
                Log.i(TAG, "This device is not supported.");
                finish();
            }
            return false;
        }
        return true;
    }

    public void registerClick(View view){
        if(checkPlayServices()){
            startService(new Intent(this, RegistrationIntentService.class));
        }
    }
}

```




5. Agregamos el Permiso de Internet

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.powerwtechnology.firebasecloudmessaging">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
```



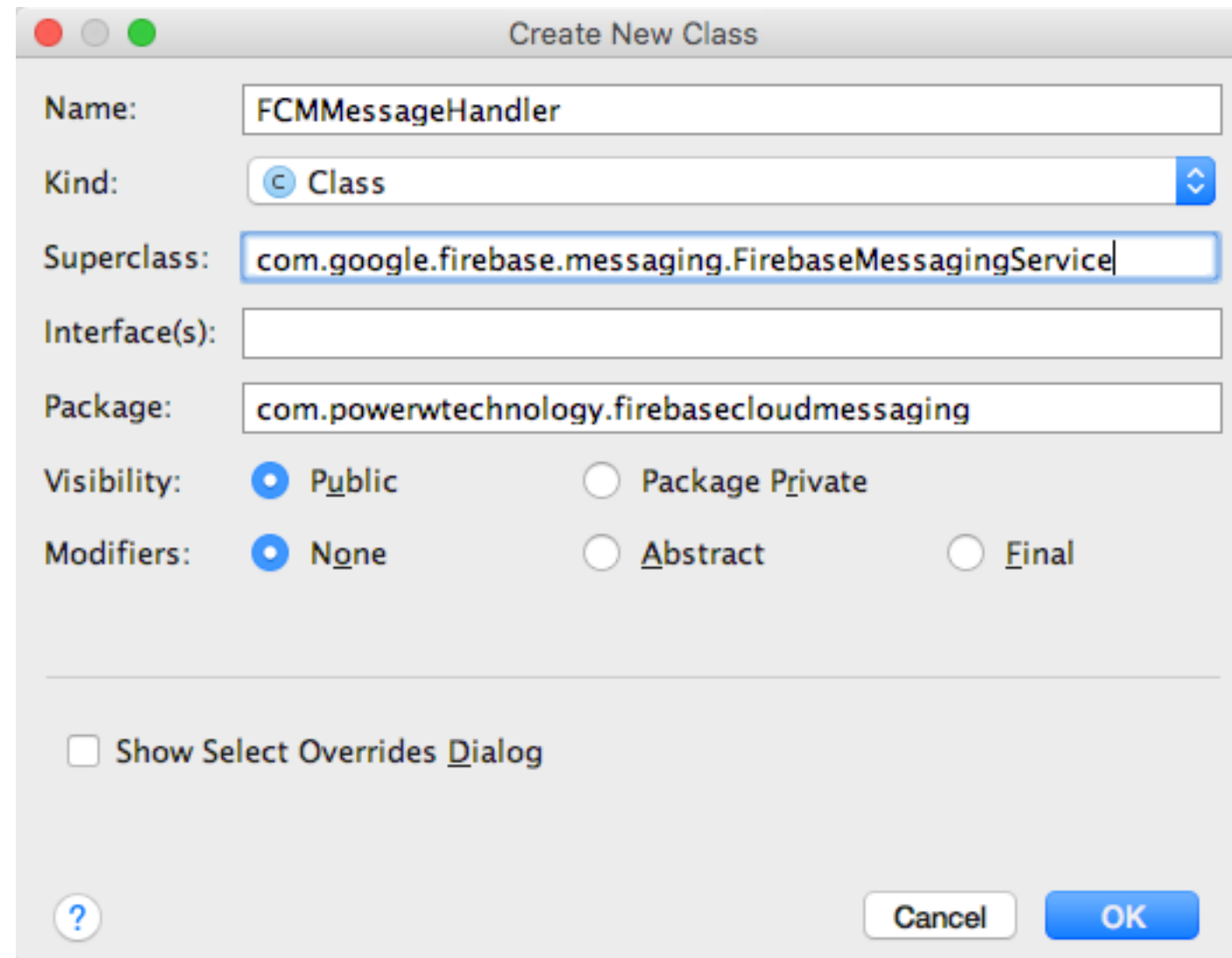
Aquí ya podemos correr la
aplicación y obtener
nuestro token



Pero ahora necesitamos recibir
y mostrar la notificación

Si este fuera el caso.

1. Creamos un Firebase Messaging Service



Create New Class

Name:

Kind: Class

Superclass:

Interface(s):

Package:

Visibility: ☒ Public ☐ Package Private

Modifiers: ☒ None ☐ Abstract ☐ Final

☐ Show Select Overrides Dialog

? Cancel OK


```

public class FCMMessageHandler extends FirebaseMessagingService {

    public static final int MESSAGE_NOTIFICATION_ID = 435345;

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        Map<String, String> data = remoteMessage.getData();
        String from = remoteMessage.getFrom();


        RemoteMessage.Notification notification = remoteMessage.getNotification();
        createNotification(notification);
    }



    // Creates notification based on title and body received
    private void createNotification(RemoteMessage.Notification notification) {
        Context context = getBaseContext();
        NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(context)
            .setSmallIcon(R.mipmap.ic_launcher).setContentTitle(notification.getTitle())
            .setContentText(notification.getBody());
        NotificationManager mNotificationManager = (NotificationManager) context
            .getSystemService(Context.NOTIFICATION_SERVICE);
        mNotificationManager.notify(MESSAGE_NOTIFICATION_ID, mBuilder.build());
    }
}

```




Probamos el envío de una notificación desde Firebase Console


 **Firebase**


 **FCMExample** 


Overview


 **Analytics**


DEVELOP


 **Auth**


 **Database**

 **Storage**


 **Hosting**

 **Remote Config**




 **Test Lab**

 **Crash**

GROW

 **Notifications**

Aplicaciones móviles FCMExample

 **com.powerwtechnology.fi...**  

Analytics (últimos 30 días)

0
Usuarios activos/mes

0 \$
Compras en la aplicación

Bloqueos (30 días)

0
Usuarios afectados

0
Errores



Administra las campañas de notificaciones
y envía mensajes para implicar a los
usuarios adecuados en el momento más
oportuno

 [Más información](#)

ENVÍA TU PRIMER MENSAJE

MENSAJE NUEVO

Mensaje

Estado ?

Fecha
de entrega
?

Plataforma

Estimación
de destinos
?

% de
mensajes
abiertos
?

Este es nuestro mensaje.

✓ Completado

27 ago. 2016
16:02



—

—

Filas por página: 25 ▼

1 - 1 de 1



Comunidad Desarrolladores en Tecnologías
< Google > en Guatemala



Implementando el Cliente Servidor

```
{ "notification": {  
  "title": "Portugal vs. Denmark",  
  "text": "5 to 1"  
},  
  "to" : "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1..."  
}
```

```
public static void main(String[] args) {
    String response = "";
    String uri = "https://fcm.googleapis.com/fcm/send";
    try {
        URL url = new URL(uri);
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();

        connection.setDoInput(true);
        connection.setDoOutput(true);
        connection.setRequestMethod("POST");
        connection.setRequestProperty("Content-Type", "application/json");
        connection.setRequestProperty("Authorization", "key=AIZA5yD1IfcUr9EP8SWj33KL21m_EinLrBntxLs");

        OutputStream outputStream = new BufferedOutputStream(connection.getOutputStream());
        BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(outputStream, "utf-8"));
        writer.write("{\"to\" : \"f8uLGpKcUWo:APA91bGA0yIjxasih0lp78kcUrMuJ5TeW6jCKN6lAk9jOUu30ITfLYj9fgwC4fq7x_hFtTiW0Ikrxs8m4z-SYw\" : \"notification\": {\"title\": \"Este es el titulo\", \"text\": \"Este es el texto\"}}");
        writer.flush();
        writer.close();
        outputStream.close();
    }
}
```



```

    int responseCode = connection.getResponseCode();
    InputStream inputStream;
    if(responseCode < HttpURLConnection.HTTP_BAD_REQUEST){
        inputStream = connection.getInputStream();
    }else{
        inputStream = connection.getErrorStream();
    }

    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));
    String temp = "";
    while ((temp = bufferedReader.readLine()) != null) {
        response += temp;
    }

} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e){
    e.printStackTrace();
}

System.out.println(String.format("Response: %s", response));
}

```