1. **Historial de cambios**

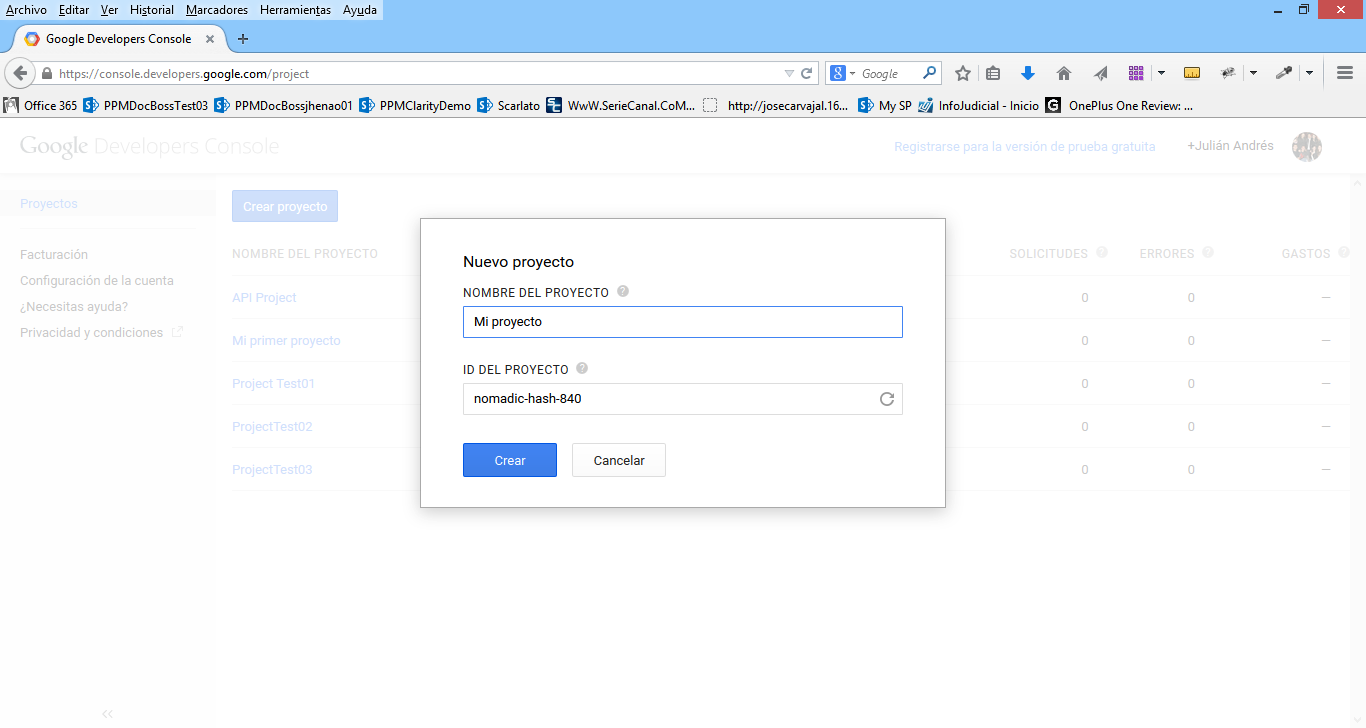
| **Fecha** | **Versión** | **Autor** | **Descripción del Cambio** |
| --- | --- | --- | --- |
| 13/05/2015 | 1.0 | Julián Andrés Henao Taborda | Creación del documento |

1. **Introducción**

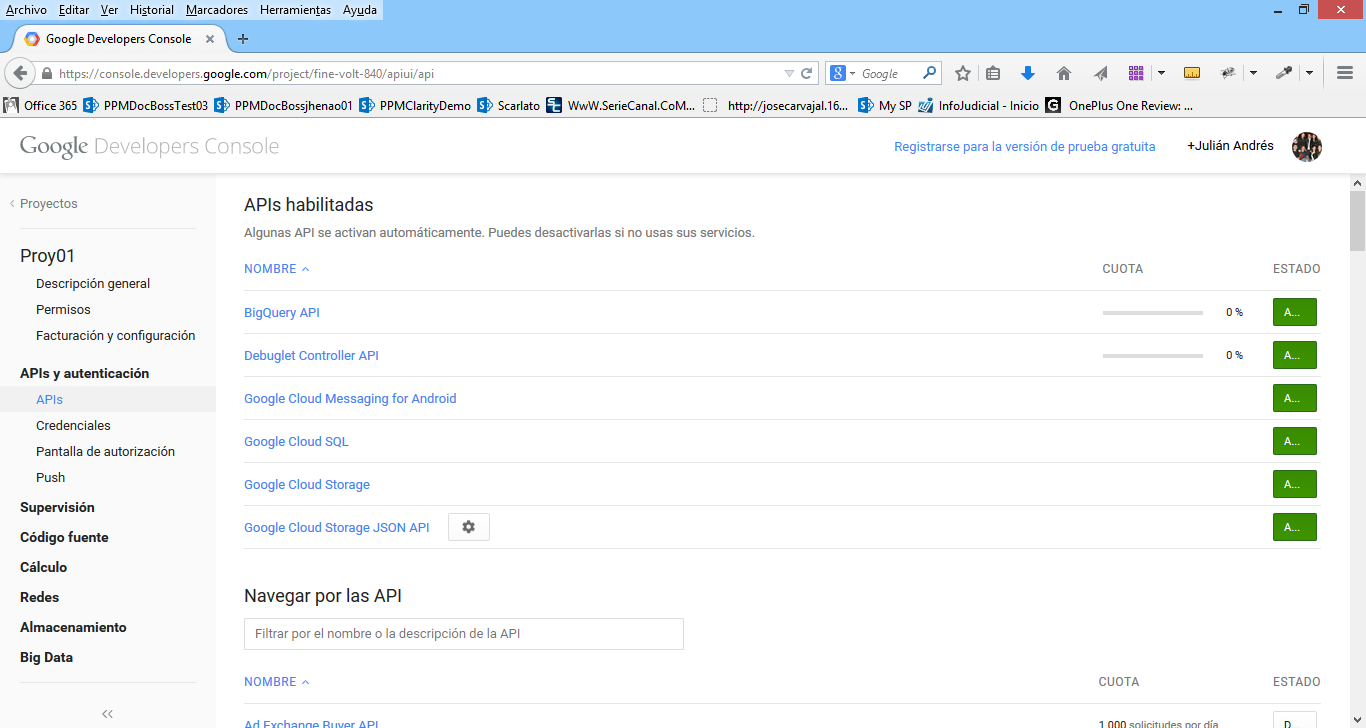
El objetivo del tutorial es el desarrollo de una aplicación cliente para sistema operativo Android para hacer uso de Extensible Messaging and Presence Protocol*.*

Después de seguir este tutorial se obtendrá una aplicación que una vez instalada en un dispositivo podrá enviar mensajes que a su vez serán devueltos al dispositivo mediante una aplicación servidor.

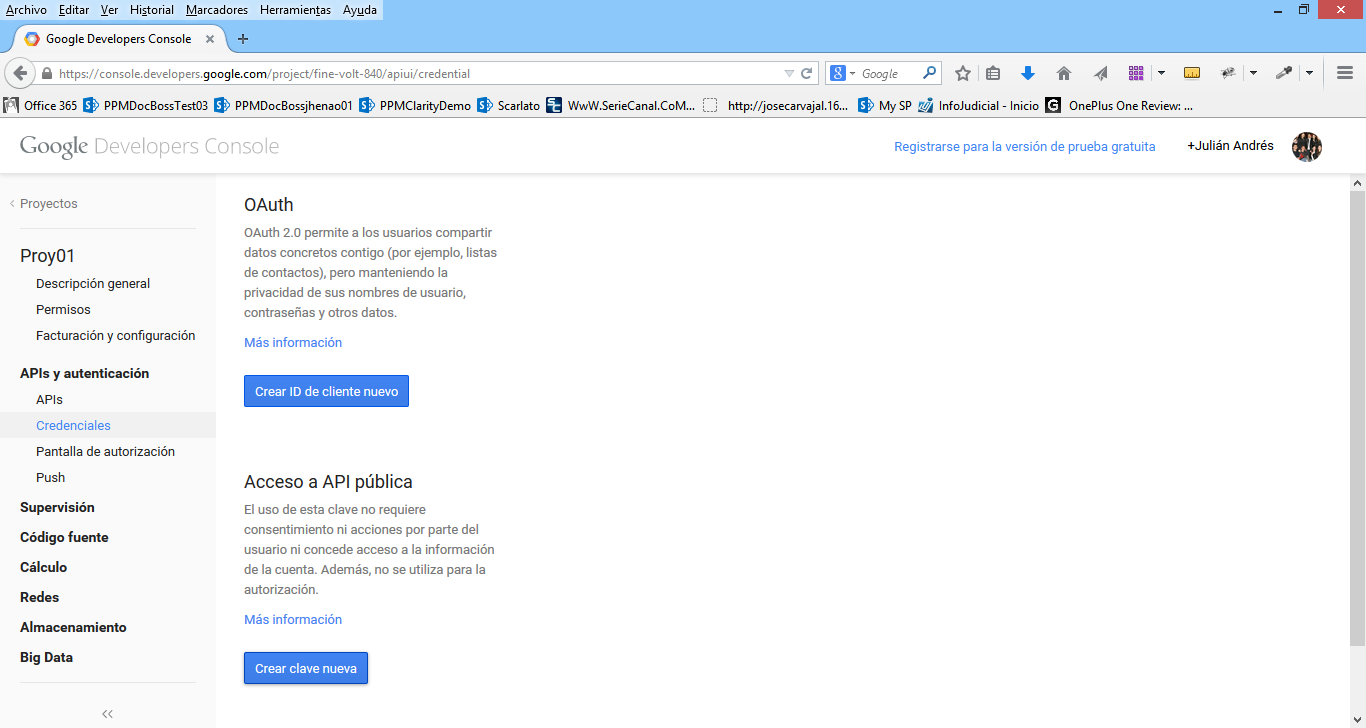
1. Antes de proceder con la instalación es necesario registrar una cuenta de Google en la Consola de Google (<https://console.developers.google.com/project/>).
   1. Crear un proyecto



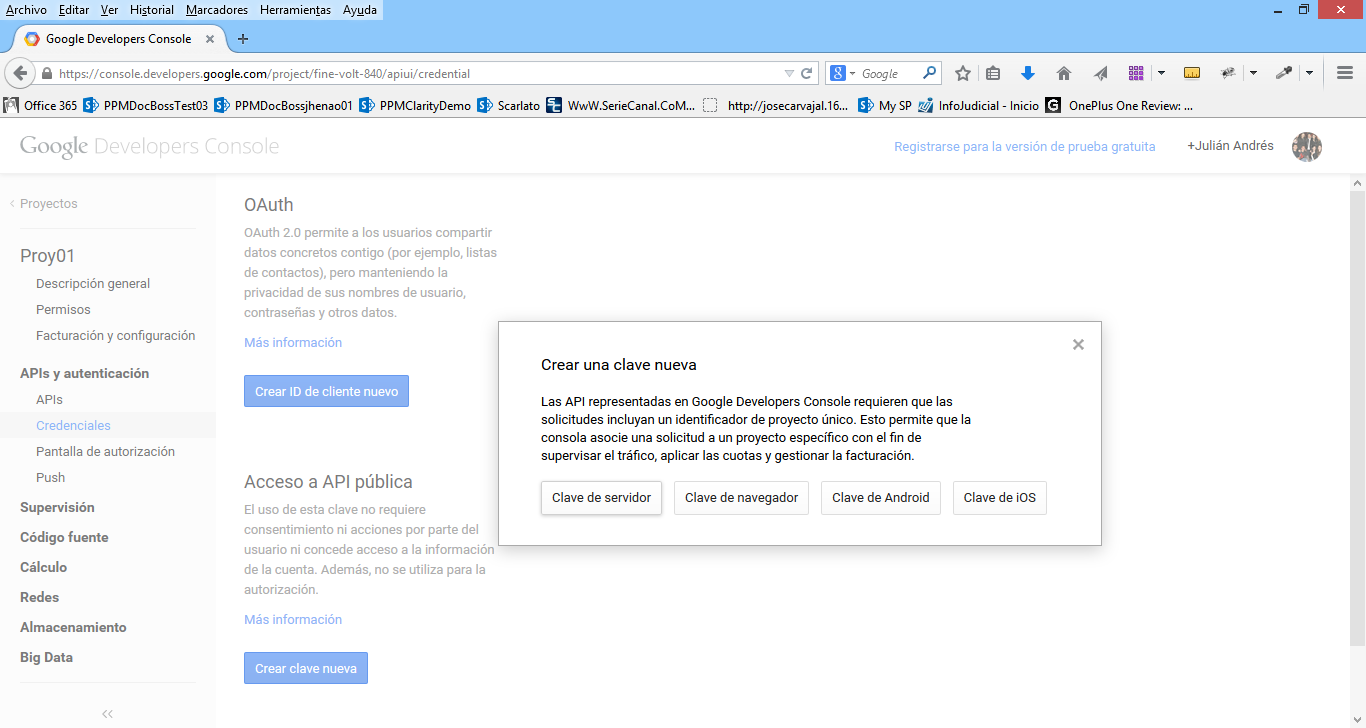
* 1. Acceder al proyecto y en “APISs y autenticación” seleccionar “APIs” y habilitar “Google Cloud Messaging for Android”.

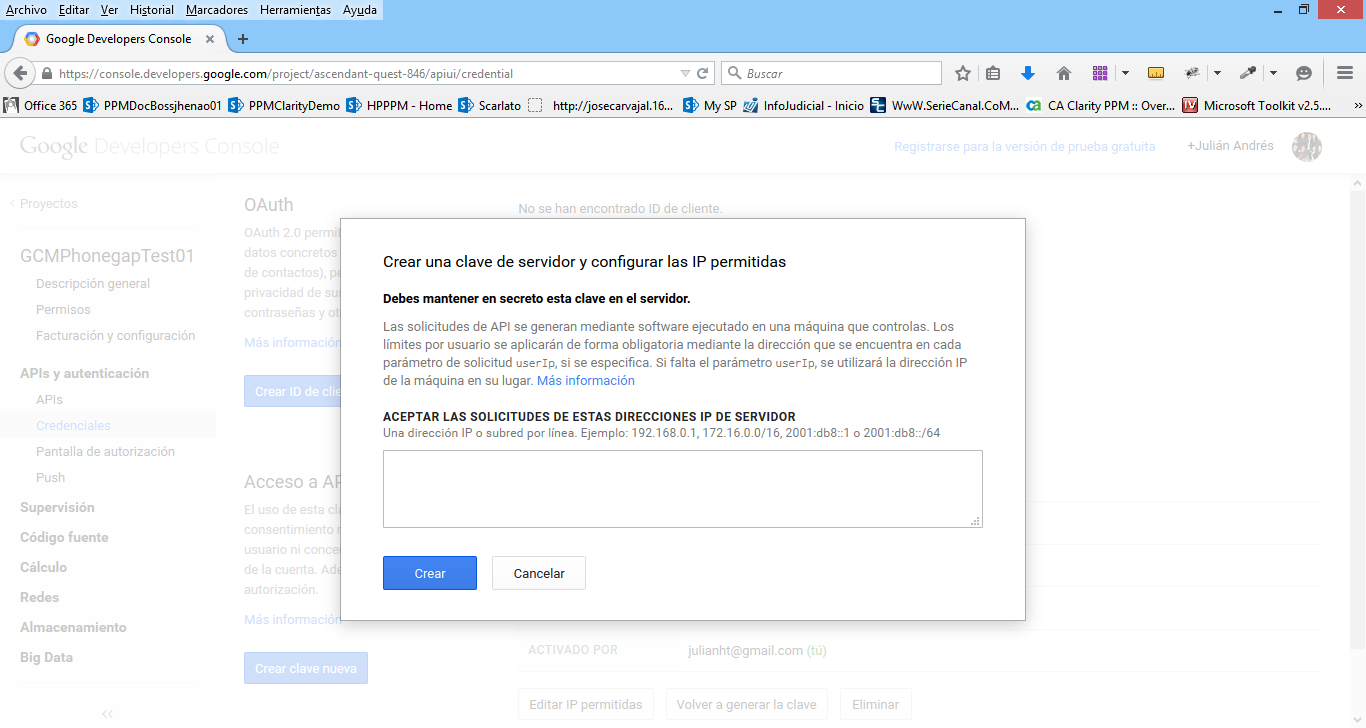


* 1. En “APIs y autenticación” seleccionar “Credenciales” y “Crear clave nueva”.

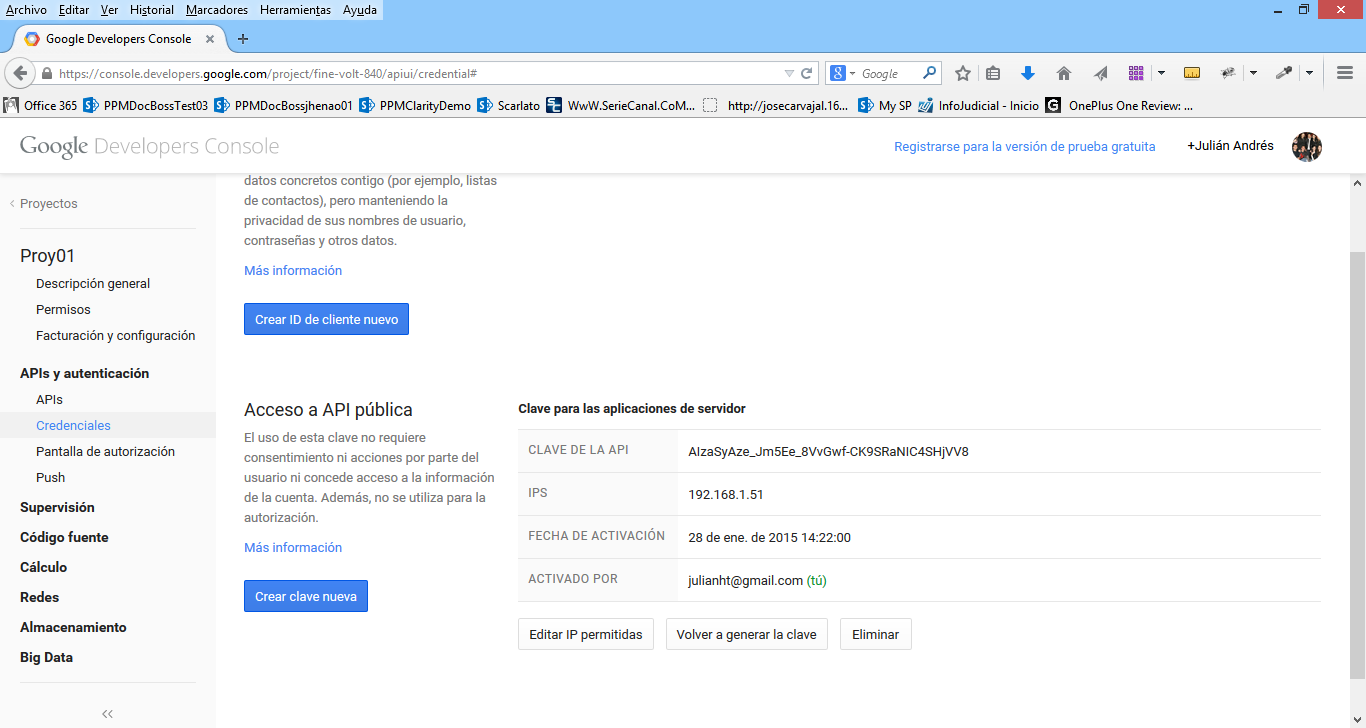


* 1. Seleccionar “Clave servidor” y “Crear”.

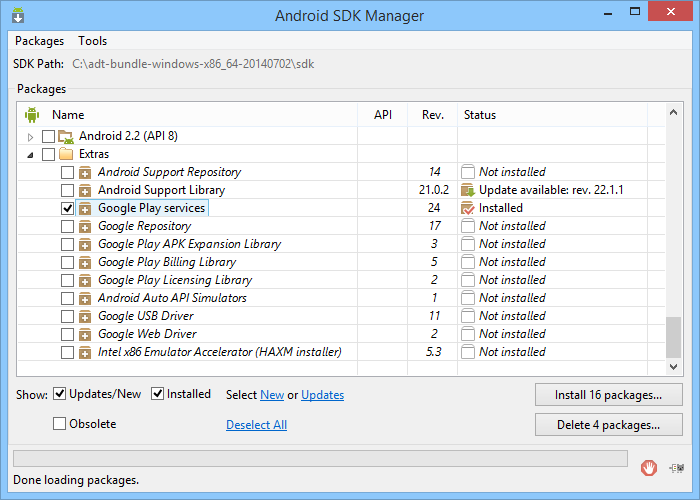




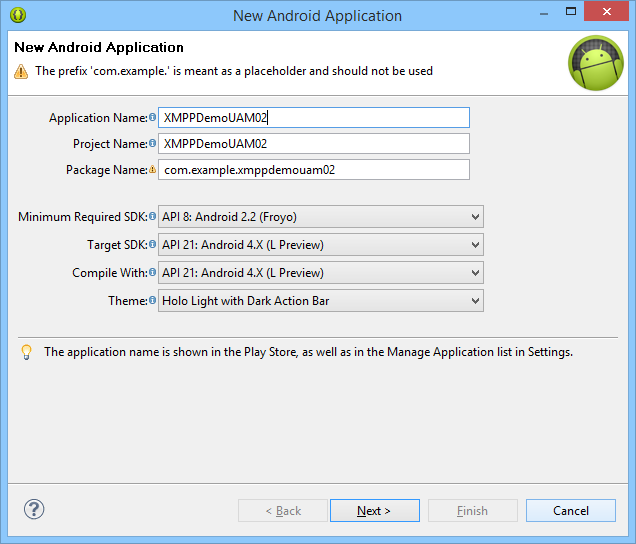
* 1. Una vez creada copiar el código definido en Clave de la API (Será usado más adelante).

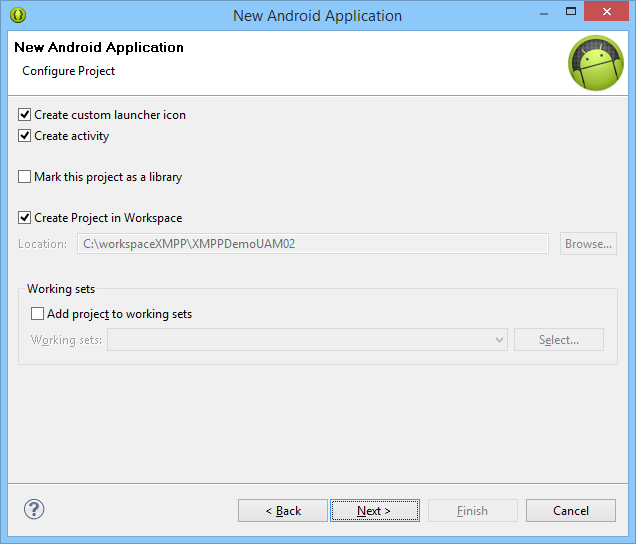


1. Descargar Eclipse SDK.
2. Ir al Android SDK Manager y descargar “Google Play Services”.

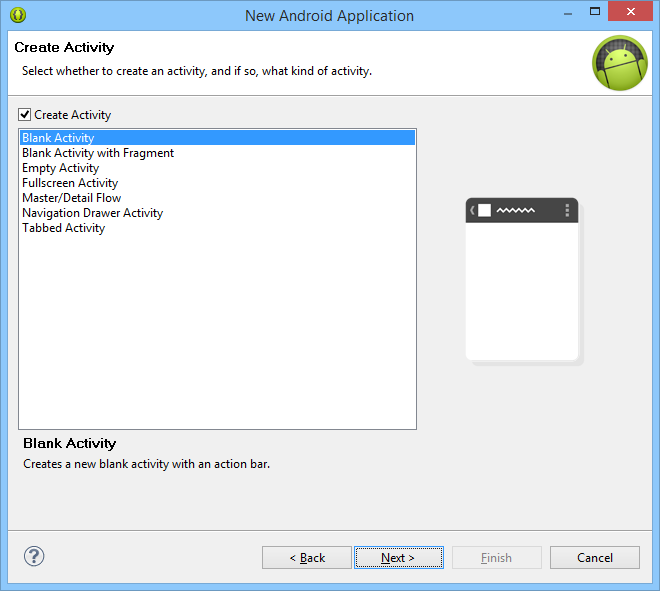


1. En Eclipse crear un nuevo proyecto tipo -> Android Application Project.



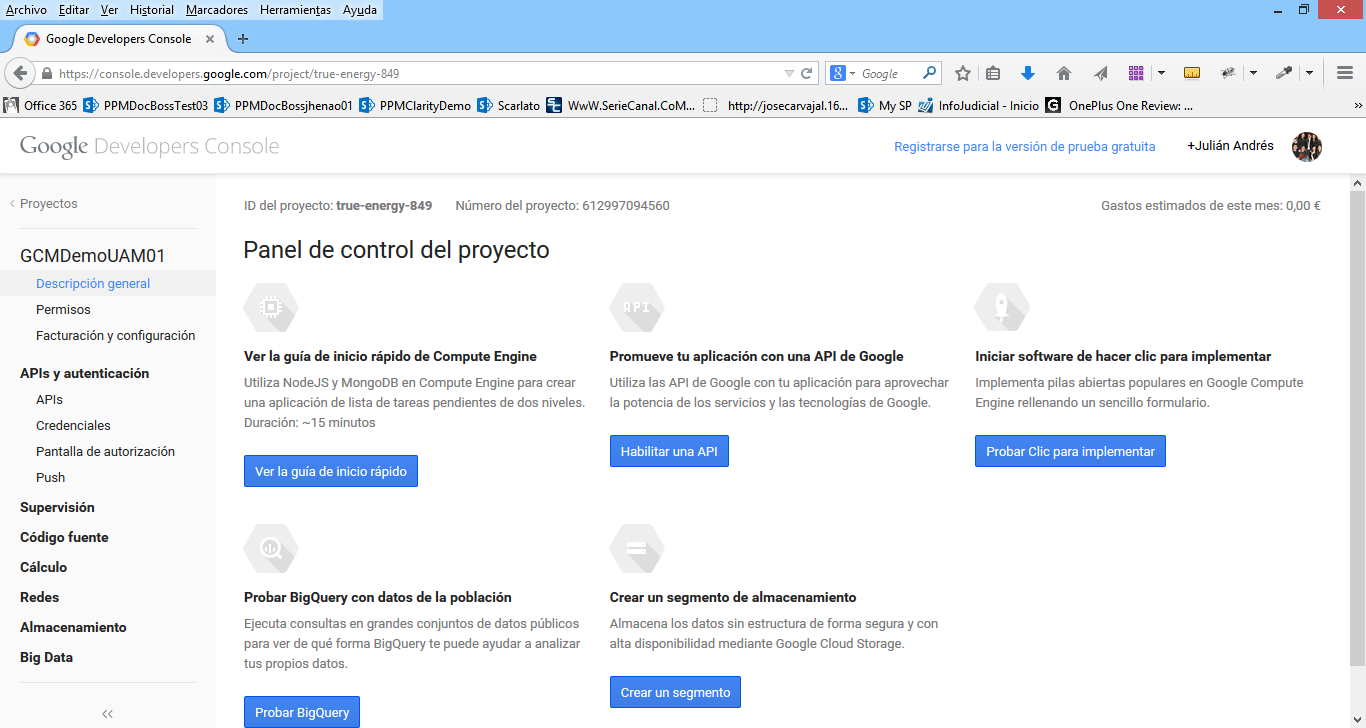


1. Crear “Blank Activity”



1. Copiar el archivo “google-play-services.jar” ubicado en el directorio SKD donde se instaló “Google Play Services” (sdk\extras\google\google\_play\_services\ libproject\google-play-services\_lib\libs\ google-play-services.jar) y pegarlo en la carpeta “libs” del proyecto recientemente creado.
2. Reemplazar el contenido de la clase creada por el siguiente código

Reemplazar el valor de la variable *GOOGLE\_USERNAME* por el ID del proyecto creado en la consola de Google:



**En esta clase se programan los eventos de los botones de registro del dispositivo para el envío de los mensajes desde la aplicación servidor**

**package** com.example.xmppdemouam;

**import** java.io.IOException;

**import** java.util.concurrent.atomic.AtomicInteger;

**import** android.app.Activity;

**import** android.content.Context;

**import** android.content.SharedPreferences;

**import** android.content.pm.PackageInfo;

**import** android.content.pm.PackageManager.NameNotFoundException;

**import** android.os.AsyncTask;

**import** android.os.Bundle;

**import** android.text.TextUtils;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.TextView;

**import** android.widget.Toast;

**import** com.google.android.gms.gcm.GoogleCloudMessaging;

**import** android.view.Menu;

**import** android.view.MenuItem;

**public** **class** RegisterActivity **extends** Activity {

Button btnGCMRegister;

Button btnXmppRegiser;

Button btnSendMessage;

GoogleCloudMessaging gcm;

Context context;

String regId;

AsyncTask<Void, Void, String> sendTask;

AtomicInteger ccsMsgId = **new** AtomicInteger();

**static** **final** String *GOOGLE\_PROJECT\_ID* = "156429511339";

**public** **static** **final** String *REG\_ID* = "";

**private** **static** **final** String *APP\_VERSION* = "1.0";

**static** **final** String *TAG* = "RegisterActivity";

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_register*);

context = getApplicationContext();

btnGCMRegister = (Button) findViewById(R.id.*btnGCMRegister*);

btnGCMRegister.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View arg0) {

**if** (TextUtils.*isEmpty*(regId)) {

regId = registerGCM();

TextView txtIdRegistro = (TextView)findViewById(R.id.*registerid*);

txtIdRegistro.setText(regId);

} **else** {

Toast.*makeText*(getApplicationContext(),

"Ya se encuentra registrado en el servidor GCM!",

Toast.*LENGTH\_LONG*).show();

}

}

});

btnXmppRegiser = (Button) findViewById(R.id.*btnXmppRegiser*);

btnXmppRegiser.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View arg0) {

**if** (TextUtils.*isEmpty*(regId)) {

Toast.*makeText*(getApplicationContext(), "RegId vacío!",

Toast.*LENGTH\_LONG*).show();

} **else** {

sendMessage("REGISTER");

TextView txtIdRegistro = (TextView)findViewById(R.id.*registerid*);

txtIdRegistro.setText(regId);

}

}

});

btnSendMessage = (Button) findViewById(R.id.*btnSendMessage*);

btnSendMessage.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** **void** onClick(View arg0) {

**if** (TextUtils.*isEmpty*(regId)) {

Toast.*makeText*(getApplicationContext(), "RegId vacío!",

Toast.*LENGTH\_LONG*).show();

} **else** {

sendMessage("ECHO");

TextView txtIdRegistro = (TextView)findViewById(R.id.*registerid*);

txtIdRegistro.setText(regId);

}

}

});

}

@Override

**public** **boolean** onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.*register*, menu);

**return** **true**;

}

@Override

**public** **boolean** onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

**int** id = item.getItemId();

**if** (id == R.id.*action\_settings*) {

**return** **true**;

}

**return** **super**.onOptionsItemSelected(item);

}

**private** **void** sendMessage(**final** String action) {

sendTask = **new** AsyncTask<Void, Void, String>() {

@Override

**protected** String doInBackground(Void... params) {

Bundle data = **new** Bundle();

data.putString("ACTION", action);

data.putString("CLIENT\_MESSAGE", "Hola servidor XMPP!");

String id = Integer.*toString*(ccsMsgId.incrementAndGet());

**try** {

gcm.send(*GOOGLE\_PROJECT\_ID* + "@gcm.googleapis.com", id, data);

} **catch** (IOException e) {

e.printStackTrace();

}

**return** "Mensaje envíado";

}

@Override

**protected** **void** onPostExecute(String result) {

sendTask = **null**;

Toast.*makeText*(getApplicationContext(), result,

Toast.*LENGTH\_LONG*).show();

}

};

sendTask.execute(**null**, **null**, **null**);

}

**public** String registerGCM() {

gcm = GoogleCloudMessaging.*getInstance*(**this**);

regId = getRegistrationId(context);

**if** (TextUtils.*isEmpty*(regId)) {

registerInBackground();

}

**else** {

Toast.*makeText*(getApplicationContext(), "Dispositivo ya registrado",

Toast.*LENGTH\_LONG*).show();

TextView txtIdRegistro = (TextView)findViewById(R.id.*registerid*);

txtIdRegistro.setText(regId);

}

**return** regId;

}

**private** String getRegistrationId(Context context) {

**final** SharedPreferences prefs = getSharedPreferences(RegisterActivity.**class**.getSimpleName(), Context.*MODE\_PRIVATE*);

String registrationId = prefs.getString(*REG\_ID*, "");

**if** (registrationId.isEmpty()) {

**return** "";

}

**int** registeredVersion = prefs.getInt(*APP\_VERSION*, Integer.*MIN\_VALUE*);

**int** currentVersion = *getAppVersion*(context);

**if** (registeredVersion != currentVersion) {

**return** "";

}

**return** registrationId;

}

**private** **static** **int** getAppVersion(Context context) {

**try** {

PackageInfo packageInfo = context.getPackageManager()

.getPackageInfo(context.getPackageName(), 0);

**return** packageInfo.versionCode;

} **catch** (NameNotFoundException e) {

**throw** **new** RuntimeException(e);

}

}

**private** **void** registerInBackground() {

**new** AsyncTask<Void, Void, String>() {

@Override

**protected** String doInBackground(Void... params) {

String msg = "";

**try** {

**if** (gcm == **null**) {

gcm = GoogleCloudMessaging.*getInstance*(context);

}

regId = gcm.register(*GOOGLE\_PROJECT\_ID*);

msg = "Dispositivo registrado. Registration ID: " + regId;

storeRegistrationId(context, regId);

} **catch** (IOException ex) {

msg = "Error :" + ex.getMessage();

}

**return** msg;

}

@Override

**protected** **void** onPostExecute(String msg) {

Toast.*makeText*(getApplicationContext(),

"Registrado con el servidor GCM" + msg, Toast.*LENGTH\_LONG*)

.show();

}

}.execute(**null**, **null**, **null**);

}

**private** **void** storeRegistrationId(Context context, String regId) {

**final** SharedPreferences prefs = getSharedPreferences(

RegisterActivity.**class**.getSimpleName(), Context.*MODE\_PRIVATE*);

**int** appVersion = *getAppVersion*(context);

SharedPreferences.Editor editor = prefs.edit();

editor.putString(*REG\_ID*, regId);

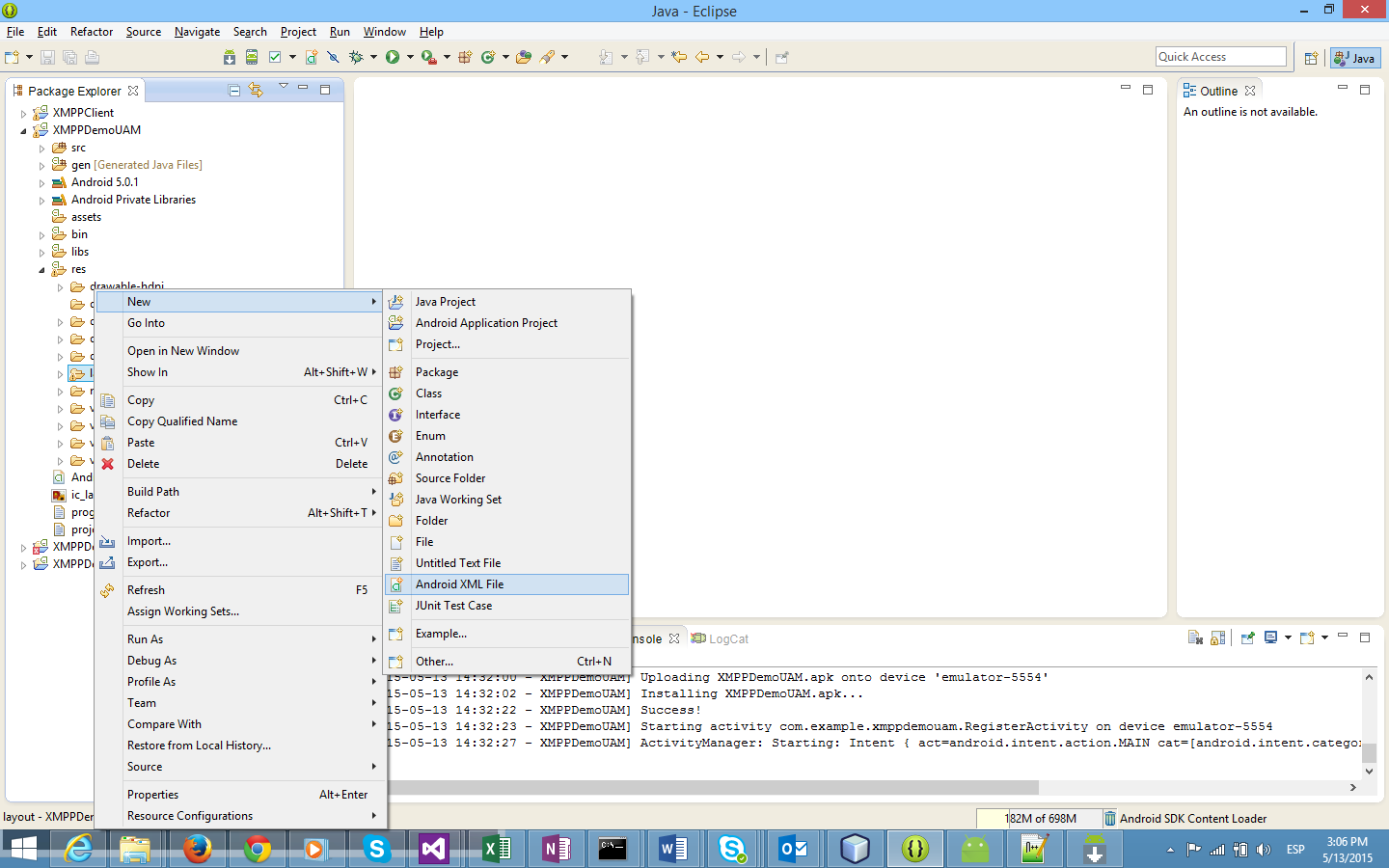
editor.putInt(*APP\_VERSION*, appVersion);

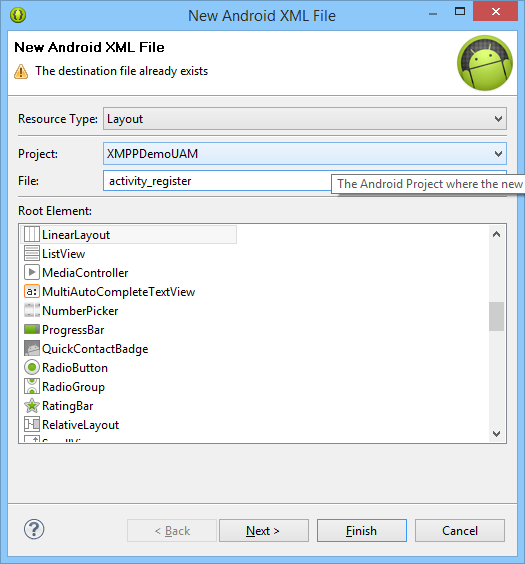
editor.commit();

}

}

1. Proceder a crear la plantilla gráfica para la clase. Creando un “Android XML File” en el directorio de layouts:





1. Reemplazar el contenido del XML creado por el código:

**En el xml se definen los controles android de la interfaz gráfica.**

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:orientation=*"vertical"* android:background=*"#ffffff"*>

<Button android:id=*"@+id/btnGCMRegister"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Registrarse en Google GCM CCS Server"*

android:layout\_margin=*"10dip"*/>

<Button android:id=*"@+id/btnXmppRegiser"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"Registrarse en XMPP Server"*

android:layout\_margin=*"10dip"*/>

<Button

android:id=*"@+id/btnSendMessage"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_margin=*"10dip"*

android:text=*"Enviar mensaje a XMPP Server"* />

<TextView

android:id=*"@+id/registerid"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/btnSendMessage"*

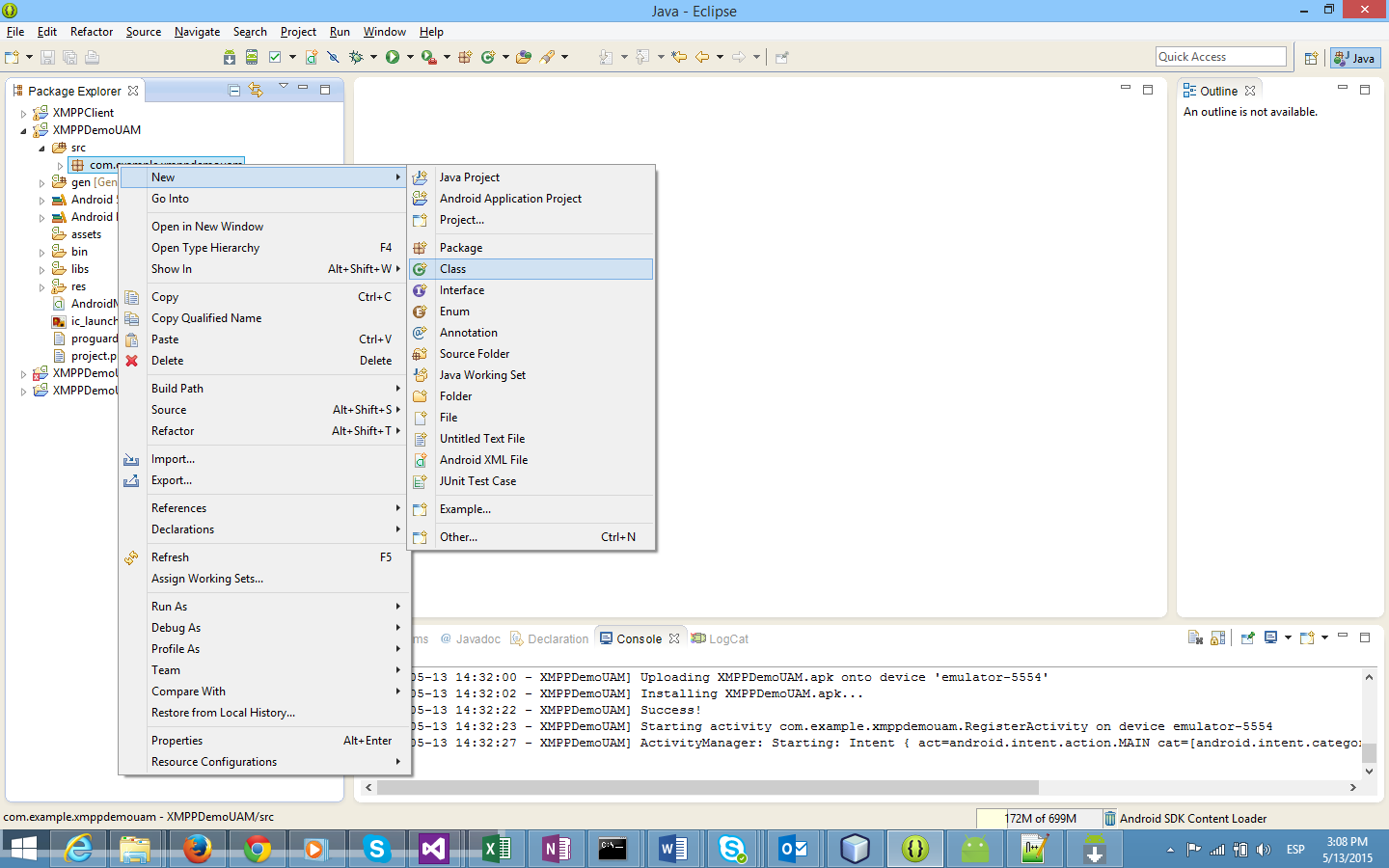
android:layout\_centerHorizontal=*"true"*

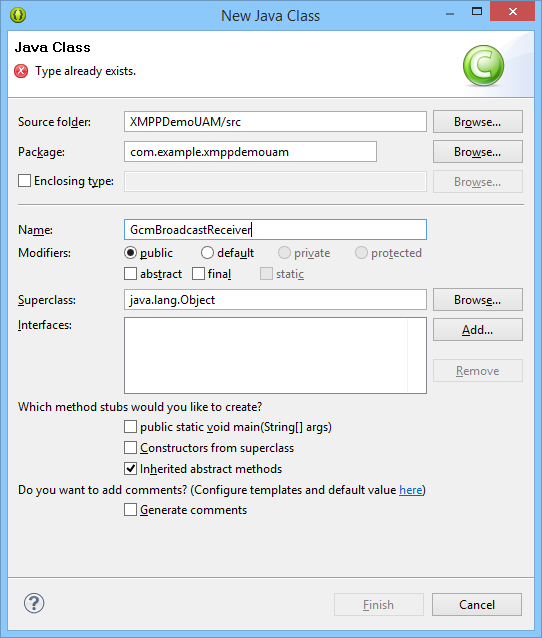
android:textIsSelectable=*"true"*

android:layout\_marginTop=*"36dp"* />

</LinearLayout>

1. Crear una nueva clase en el proyecto:





1. Reemplazar el contenido de la clase creada por el código:

**package** com.example.xmppdemouam;

**import** android.app.Activity;

**import** android.content.ComponentName;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.support.v4.content.WakefulBroadcastReceiver;

**public** **class** GcmBroadcastReceiver **extends** WakefulBroadcastReceiver {

@Override

**public** **void** onReceive(Context context, Intent intent) {

ComponentName comp = **new** ComponentName(context.getPackageName(),

GCMNotificationIntentService.**class**.getName());

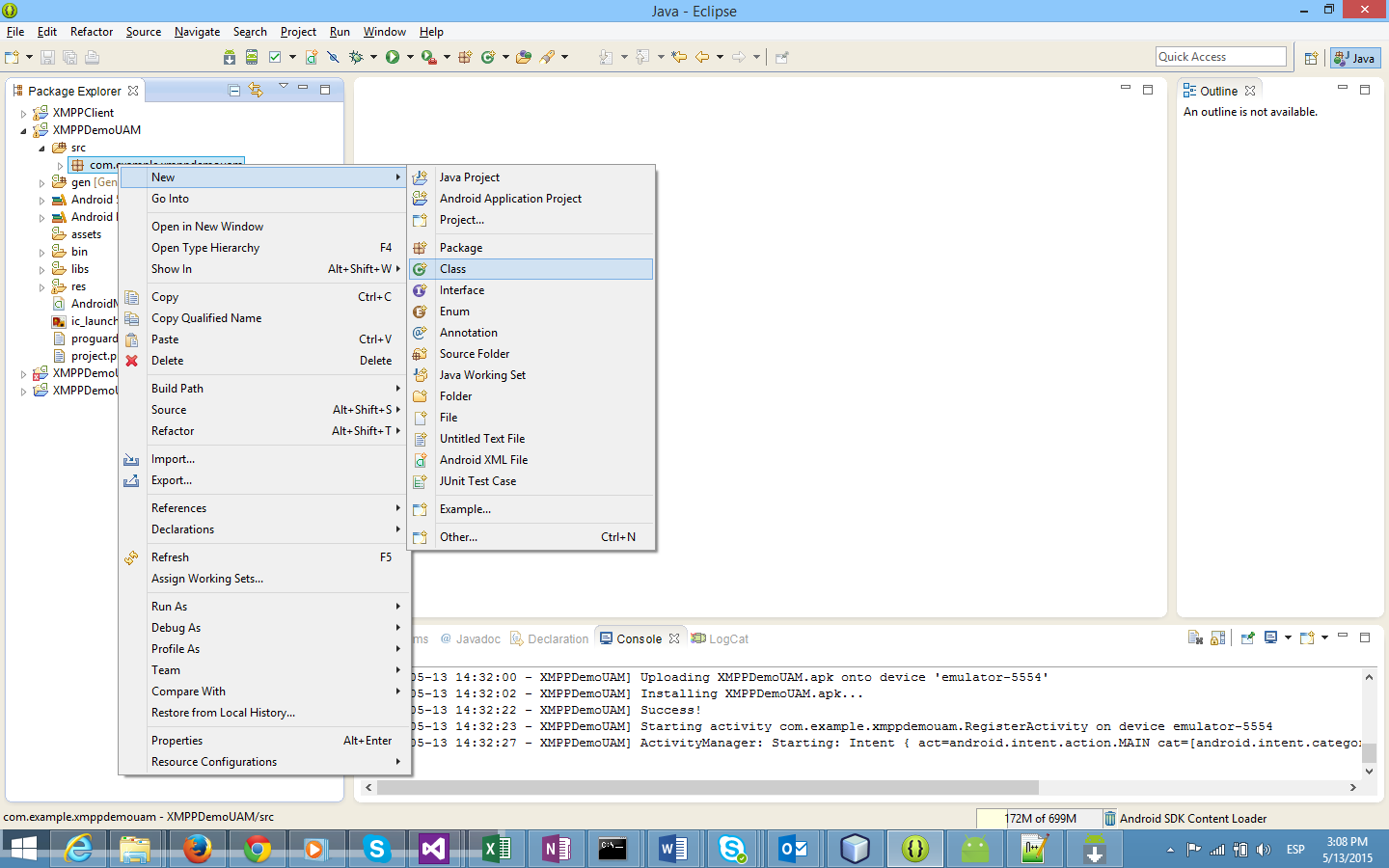
*startWakefulService*(context, (intent.setComponent(comp)));

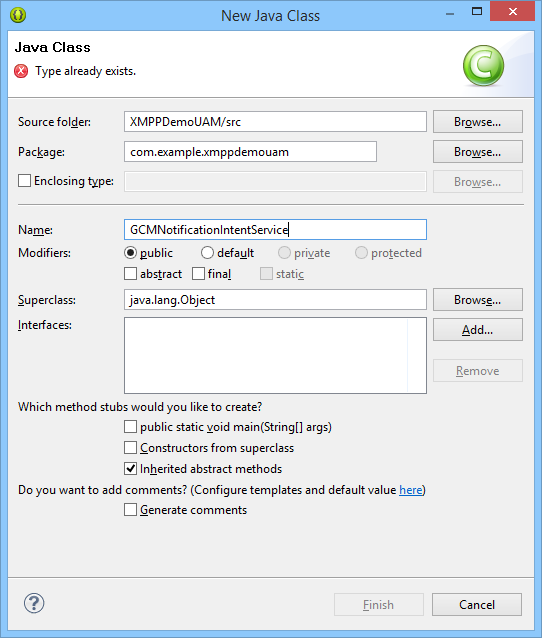
setResultCode(Activity.*RESULT\_OK*);

}

}

1. Crear una nueva clase en el proyecto:





1. Reemplazar el contenido de la clase creada por el código:

**En esta clase se definen la notificación de mensajes.**

package com.example.xmppdemouam;

import android.app.IntentService;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Context;

import android.content.Intent;

import android.os.Bundle;

import android.support.v4.app.NotificationCompat;

import com.google.android.gms.gcm.GoogleCloudMessaging;

public class GCMNotificationIntentService extends IntentService {

public static final int NOTIFICATION\_ID = 1;

private NotificationManager mNotificationManager;

NotificationCompat.Builder builder;

public GCMNotificationIntentService() {

super("GcmIntentService");

}

public static final String TAG = "GCMNotificationIntentService";

@Override

protected void onHandleIntent(Intent intent) {

Bundle extras = intent.getExtras();

GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(this);

String messageType = gcm.getMessageType(intent);

if (!extras.isEmpty()) {

if (GoogleCloudMessaging.MESSAGE\_TYPE\_SEND\_ERROR

.equals(messageType)) {

sendNotification("Send error: " + extras.toString());

} else if (GoogleCloudMessaging.MESSAGE\_TYPE\_DELETED

.equals(messageType)) {

sendNotification("Deleted messages on server: "

+ extras.toString());

} else if (GoogleCloudMessaging.MESSAGE\_TYPE\_MESSAGE

.equals(messageType)) {

sendNotification("SERVER\_MESSAGE: " + extras.get("SERVER\_MESSAGE"));

}

}

GcmBroadcastReceiver.completeWakefulIntent(intent);

}

private void sendNotification(String msg) {

mNotificationManager = (NotificationManager) this

.getSystemService(Context.NOTIFICATION\_SERVICE);

PendingIntent contentIntent = PendingIntent.getActivity(this, 0,

new Intent(this, RegisterActivity.class), 0);

NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(

this)//.setSmallIcon(R.drawable.gcm\_cloud)

.setContentTitle("GCM XMPP Message")

.setStyle(new NotificationCompat.BigTextStyle().bigText(msg))

.setContentText(msg);

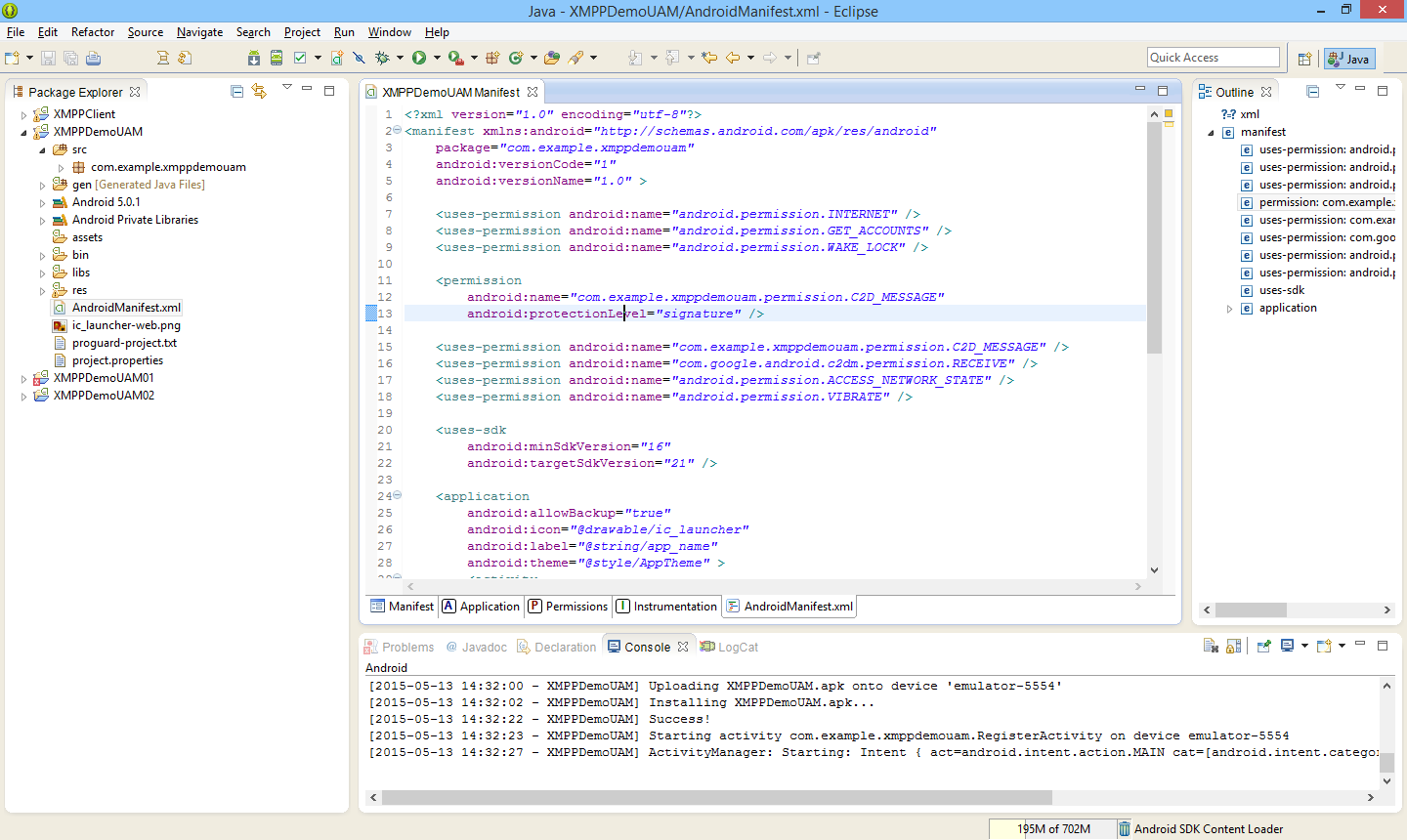
mBuilder.setContentIntent(contentIntent);

mNotificationManager.notify(NOTIFICATION\_ID, mBuilder.build());

}

}

1. Procedemos a modificar el archivo “**AndroidManifest.xml**”.



1. Reemplazamos el contenido del archivo XML por el código:

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*

package=*"com.example.xmppdemouam"*

android:versionCode=*"1"*

android:versionName=*"1.0"* >

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

<uses-permission android:name=*"android.permission.GET\_ACCOUNTS"* />

<uses-permission android:name=*"android.permission.WAKE\_LOCK"* />

<permission

android:name=*"com.example.xmppdemouam.permission.C2D\_MESSAGE"*

android:protectionLevel=*"signature"* />

<uses-permission android:name=*"com.example.xmppdemouam.permission.C2D\_MESSAGE"* />

<uses-permission android:name=*"com.google.android.c2dm.permission.RECEIVE"* />

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

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

<uses-sdk

android:minSdkVersion=*"16"*

android:targetSdkVersion=*"21"* />

<application

android:allowBackup=*"true"*

android:icon=*"@drawable/ic\_launcher"*

android:label=*"@string/app\_name"*

android:theme=*"@style/AppTheme"* >

<activity

android:name=*".RegisterActivity"*

android:label=*"@string/app\_name"* >

<intent-filter>

<action android:name=*"android.intent.action.MAIN"* />

<category android:name=*"android.intent.category.LAUNCHER"* />

</intent-filter>

</activity>

<receiver

android:name=*".GcmBroadcastReceiver"*

android:permission=*"com.google.android.c2dm.permission.SEND"* >

<intent-filter>

<action android:name=*"com.google.android.c2dm.intent.RECEIVE"* />

<action android:name=*"com.google.android.c2dm.intent.REGISTRATION"* />

<category android:name=*"com.example.xmppdemouam"* />

</intent-filter>

</receiver>

<service android:name=*".GCMNotificationIntentService"* />

</application>

</manifest>

1. Instalar la aplicación en el dispositivo para obtener el “**Id del dispositivo**”.



1. Copiar el id del dispositivo para pegarlo en la aplicación servidor.
2. Dar click en los botones “Registrarse en XMPP Server/Enviar mensaje a XMPP server” para comunicarse con la aplicación servidor.