

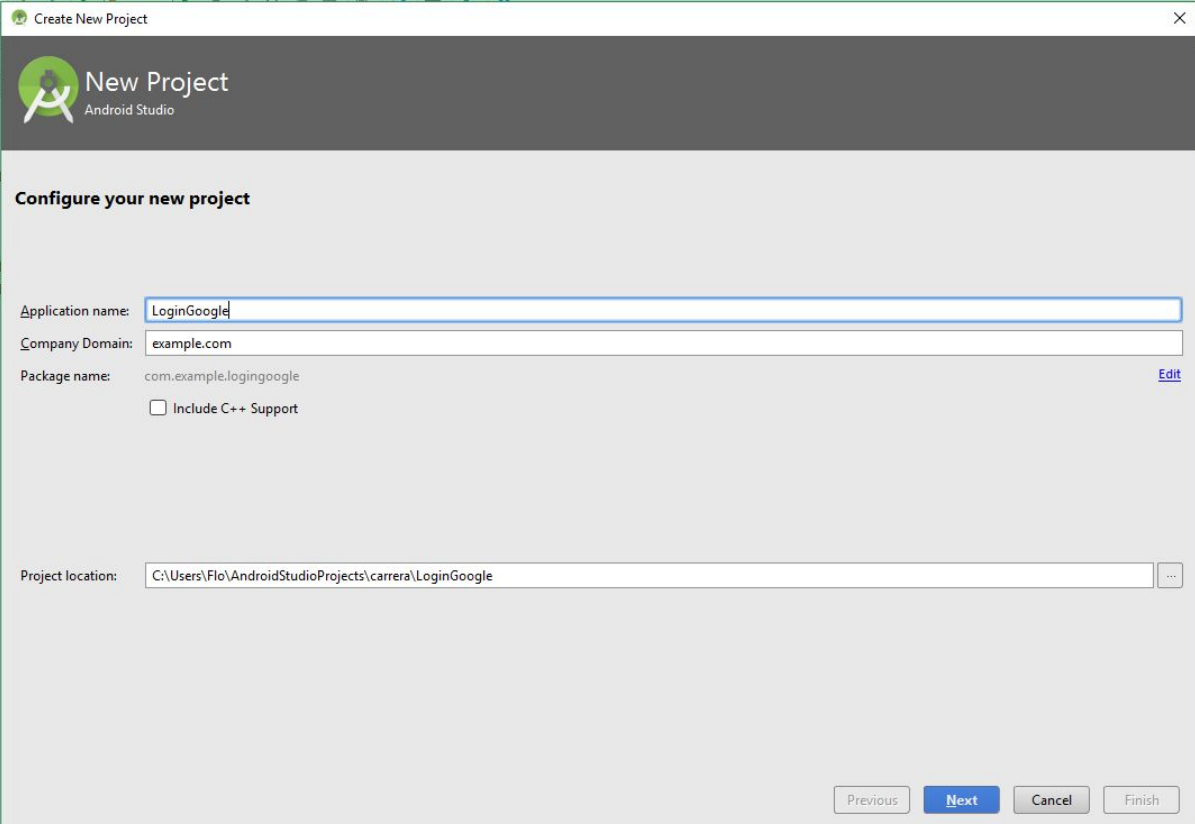
Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería en Computación  
Electiva Dispositivos Móviles  
Profesora Adriana Álvarez  
Laboratorio: Integración con Facebook

### Requisitos:

- 1) Dispositivo con Android 2.3 o más nuevo
- 2) Tener foto de perfil en Google+

### Laboratorio:

- Crear un nuevo proyecto en Android Studio.



Create New Project

New Project  
Android Studio

Configure your new project

Application name: LoginGoogle

Company Domain: example.com

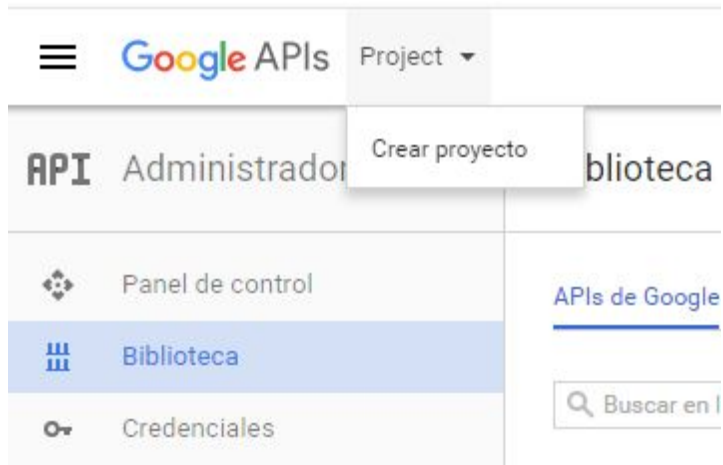
Package name: com.example.logingoogle [Edit](#)

☐ Include C++ Support

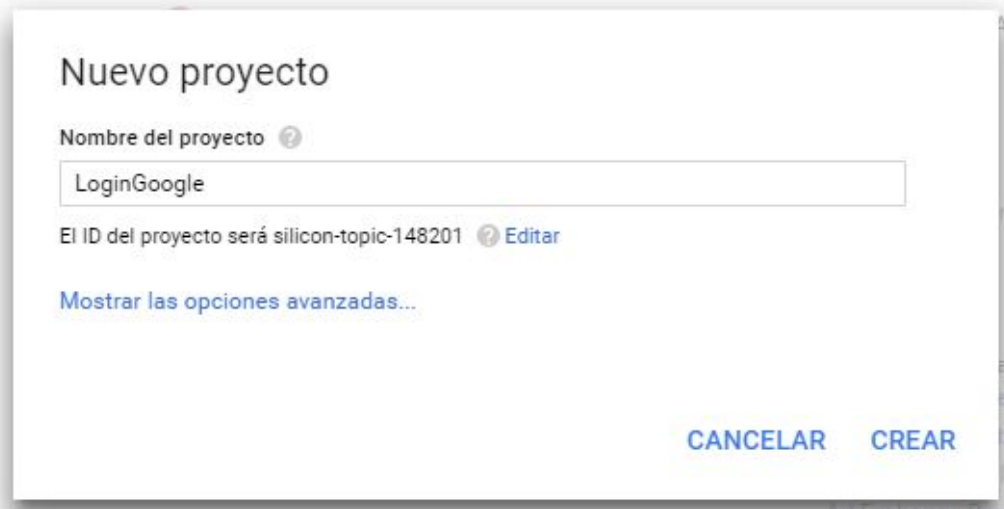
Project location: C:\Users\Flo\AndroidStudioProjects\carrera>LoginGoogle

Previous Next Cancel Finish

- Ir a este [link](#) y crear un nuevo proyecto



- Nombrar el proyecto y aceptar los términos.



- Ahora se necesita el archivo de configuración. En el siguiente [link](#) seleccionar el app que se creó, además hay que escribir el nombre del paquete, y seleccionar el país.

## Create or choose an app

App name

LoginGoogle

Services will be added to your existing project in the [Google Developers Console](#).

Android package name

com.example.logingoogle

☒ Share your [Google Mobile Developer Services](#) data with Google to help improve Google's products and services. This includes sharing with Google technical support, account specialists, and anonymous data for benchmarking. If you disable this option, data can still flow to other Google products that are explicitly added.

Your country/region: Costa Rica

By proceeding and clicking the button below, you agree and understand that you are using the [Google Mobile Developer Services](#) in your app, and agree to its [terms of service](#). These terms govern only your use of the Google Mobile Developer Services. If you use any additional Google API or service, you agree that the applicable [terms of service](#) will govern with respect to such [APIs and services](#).

CONTINUE TO  
Choose and configure services →

- Después se necesita el certificado SHA-1, por lo que hay que utilizar la línea de comandos, en esta hay que ir al fichero bin del jdk y luego ejecutar la siguiente instrucción. (Cambiar USERNAME por su nombre de usuario)

```
1 keytool -list -v -keystore c:\users\USERNAME\.android\debug.keystore -alias  
androiddebugkey -storepass android -keypass android
```

```
Select Command Prompt

C:\Users\Flo>cd C:\Program Files\Java\jdk1.8.0_101\bin




C:\Program Files\Java\jdk1.8.0_101\bin>keytool -list -v -keystore c:\users\flo\android\debug.keystore -alias androiddebugkey -storepass android -keypass android

Alias name: androiddebugkey
Creation date: Jul 28, 2016
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=US, O=Android, CN=Android Debug
Issuer: C=US, O=Android, CN=Android Debug
Serial number: 1
Valid from: Thu Jul 28 13:49:02 CST 2016 until: Sat Jul 21 13:49:02 CST 2046
Certificate fingerprints:
  MD5: D3:68:DE:CC:01:21:4E:85:DC:47:D8:BB:87:49:76:2A
  SHA1: FA:41:C1:BB:D7:7E:9D:B9:A4:D6:DB:B8:90:D2:01:CE:11:3B:F2:7C
  SHA256: 09:FC:A0:E7:FD:51:61:1F:96:75:24:DA:64:98:34:F9:B6:9A:72:9F:76:FF:F9:25:D0:D6:26:B4:49:AF:6F:AC
Signature algorithm name: SHA1withRSA
Version: 1

C:\Program Files\Java\jdk1.8.0_101\bin>
```

- Hay que utilizar el certificado para agregar el servicio de Sign-In. Vamos a la siguiente pantalla al habilitar Google Sign-In.

Select which Google services you'd like to add to your app below.



Google Sign-In

Analytics

Cloud Messaging

### Google Sign-In

Get users into your app quickly and securely

[LEARN MORE](#)

To use Google Sign-In, you'll need to provide the SHA-1 of your signing certificate so we can create an OAuth2 client and API key for your app.

Android Signing Certificate SHA-1

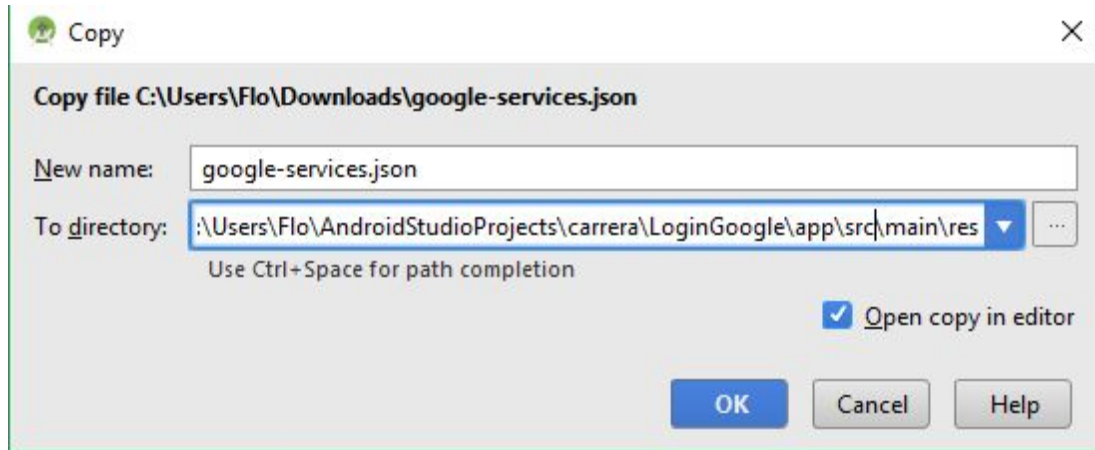
**FA:41:C1:BB:D7:7E:9D:B9:A4:D6:DB:B8:90:D2:01:CE:11:3B:F2:7C**

[How do I find my SHA-1?](#)

**ENABLE GOOGLE SIGN-IN**

- Luego generar el archivo de configuración y descargar google-services.json

- En Android Studio, hay que copiar el .json y pegarlo en el proyecto (dentro del directorio app/).



- Ahora en el build.gradle (el de Project:LoginGoogle) agregar las siguientes dependencias.

```
1 classpath 'com.google.gms:google-services:1.5.0-beta2'
```

- En el build.gradle (Module:app) agregar las siguientes líneas y sincronizar el proyecto.

```
1 compile 'com.google.android.gms:play-services-auth:8.3.0'
```

```
2 compile 'com.mcxiaoke.volley:library-aar:1.0.0'
```

- Ahora en activity\_main.xml escribir el siguiente código.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.logingoogle.MainActivity">
```

```
<com.google.android.gms.common.SignInButton
    android:id="@+id/sign_in_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" />
```

```
<com.android.volley.toolbox.NetworkImageView
    android:id="@+id/profileImage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<TextView
    android:text="Name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<TextView
    android:id="@+id/textViewName"
    android:textStyle="bold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<TextView
    android:text="email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<TextView
    android:id="@+id/textViewEmail"
    android:textStyle="bold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
</LinearLayout>
```



- En el MainActivity escribimos el siguiente código.

```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.NetworkImageView;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;

public class MainActivity extends AppCompatActivity implements View.OnClickListener,
    GoogleApiClient.OnConnectionFailedListener {

    private SignInButton signInButton;
    private GoogleSignInOptions gso;
    private GoogleApiClient mGoogleApiClient;
    private int RC_SIGN_IN = 100;
    private TextView textViewName;
    private TextView textViewEmail;
    private NetworkImageView profilePhoto;
    private ImageLoader imageLoader;

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

        textViewName = (TextView) findViewById(R.id.textViewName);
        textViewEmail = (TextView) findViewById(R.id.textViewEmail);
        profilePhoto = (NetworkImageView) findViewById(R.id.profileImage);

        gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
            .requestEmail()
            .build();

        signInButton = (SignInButton) findViewById(R.id.sign_in_button);
        signInButton.setSize(SignInButton.SIZE_WIDE);
        signInButton.setScopes(gso.getScopeArray());

        mGoogleApiClient = new GoogleApiClient.Builder(this)
            .enableAutoManage(this /* FragmentActivity */, this /* OnConnectionFailedListener */)
            .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
            .build();

        signInButton.setOnClickListener(this);
    }
}
```

```

private void signIn() {
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
    startActivityForResult(signInIntent, RC_SIGN_IN);
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RC_SIGN_IN) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        handleSignInResult(result);
    }
}

```

```

private void handleSignInResult(GoogleSignInResult result) {
    if (result.isSuccess()) {
        GoogleSignInAccount acct = result.getSignInAccount();

        textViewName.setText(acct.getDisplayName());
        textViewEmail.setText(acct.getEmail());

        imageLoader = CustomVolleyRequest.getInstance(this.getApplicationContext())
            .getImageLoader();

        imageLoader.get(acct.getPhotoUrl().toString(),
            ImageLoader.getImageListener(profilePhoto,
                R.mipmap.ic_launcher,
                R.mipmap.ic_launcher));

        profilePhoto.setImageUrl(acct.getPhotoUrl().toString(), imageLoader);
    } else {
        Toast.makeText(this, "Login Failed", Toast.LENGTH_LONG).show();
    }
}

```

```

@Override
public void onClick(View v) {
    if (v == signInButton) {
        signIn();
    }
}

```

```

@Override
public void onConnectionFailed(ConnectionResult connectionResult) {
}

```



- Crear la clase CustomVolleyRequest.java, que se encargará de cargar una imagen de un URL.

```
import android.content.Context;
import android.graphics.Bitmap;
import android.support.v4.util.LruCache;
import com.android.volley.Cache;
import com.android.volley.Network;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.BasicNetwork;
import com.android.volley.toolbox.DiskBasedCache;
import com.android.volley.toolbox.HurlStack;
import com.android.volley.toolbox.ImageLoader;

public class CustomVolleyRequest {

    private static CustomVolleyRequest customVolleyRequest;
    private static Context context;
    private RequestQueue requestQueue;
    private ImageLoader imageLoader;

    private CustomVolleyRequest(Context context) {
        this.context = context;
        this.requestQueue = getRequestQueue();

        imageLoader = new ImageLoader(requestQueue,
            new ImageLoader.ImageCache() {
                private final LruCache<String, Bitmap>
                    cache = new LruCache<String, Bitmap>(20);

                @Override
                public Bitmap getBitmap(String url) {
                    return cache.get(url);
                }

                @Override
                public void putBitmap(String url, Bitmap bitmap) {
                    cache.put(url, bitmap);
                }
            });
    }
}
```

```

public static synchronized CustomVolleyRequest getInstance(Context context) {
    if (customVolleyRequest == null) {
        customVolleyRequest = new CustomVolleyRequest(context);
    }
    return customVolleyRequest;
}

public RequestQueue getRequestQueue() {
    if (requestQueue == null) {
        Cache cache = new DiskBasedCache(context.getCacheDir(), 10 * 1024 * 1024);
        Network network = new BasicNetwork(new HurlStack());
        requestQueue = new RequestQueue(cache, network);
        requestQueue.start();
    }
    return requestQueue;
}

public ImageLoader getImageLoader() {
    return imageLoader;
}
}

```

- Para finalizar, se le otorgan permisos de internet en el archivo manifest.

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

NOTA: Es importante tener foto de perfil en Google+ para el correcto funcionamiento del laboratorio.