**TP Nro. 10 Autenticación OAuth2 mediante Google**

**Paso 1: Configurando OAuth2 con Google como Autenticador**

Configuramos application.properties para acceder mediante Google con OAuth2 (ya lo vamos a dejar pre-seteado en formato variables de entorno y luego reemplazamos sus valores).

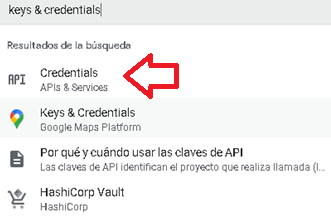
#Configuraciones de OAuth con Google

spring.security.oauth2.client.registration.google.client-id=${GOOGLE\_CLIENT\_ID}

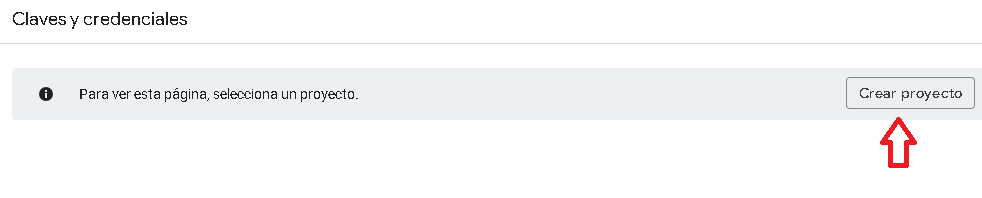
spring.security.oauth2.client.registration.google.client-secret=${GOOGLE\_CLIENT\_SECRET}

Para obtener el client\_id y el client\_secret debemos acceder a nuestra cuenta de Google mediante [este link](https://console.cloud.google.com/) (en caso de que tengas problemas con el link, podés googlear “**Google Console**” y la primera opción de respuesta te redirigirá a la web de Google para generar las keys).

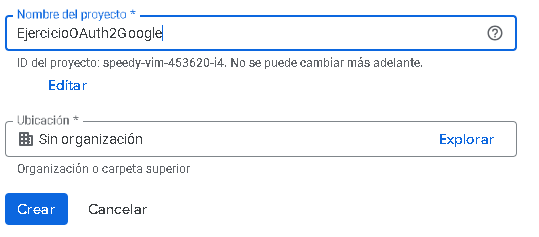
Una vez ingresado, buscamos en el buscador “keys & credentials” y elegimos la opción API Credencials (Api & Services)



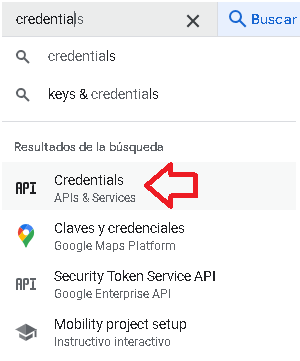
Una vez que nos redirija, vamos a crear un nuevo proyecto:



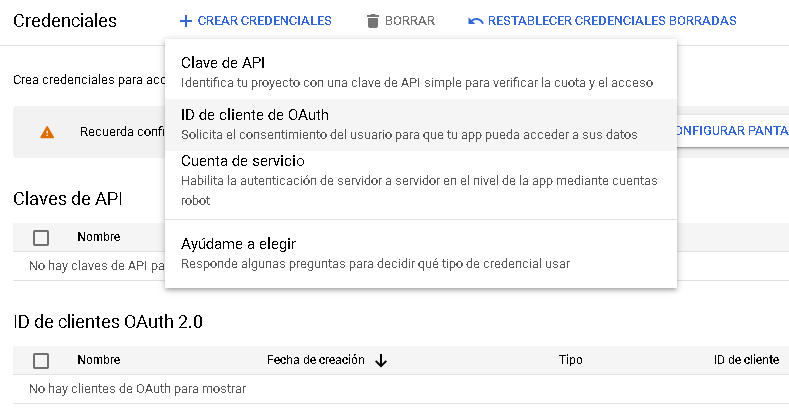
Cargamos los datos correspondientes y luego hacemos click en “Crear”



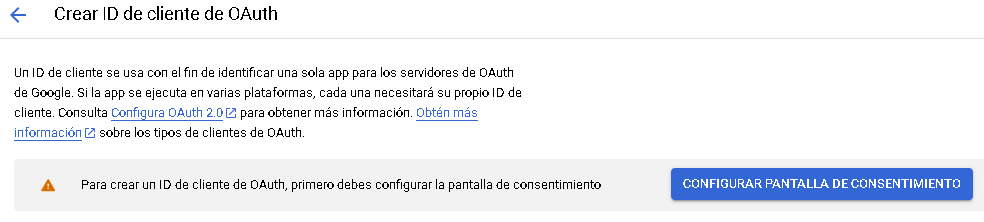
Una vez todo creado, vamos a la configuración del proyecto y buscamos por “Credentials”:



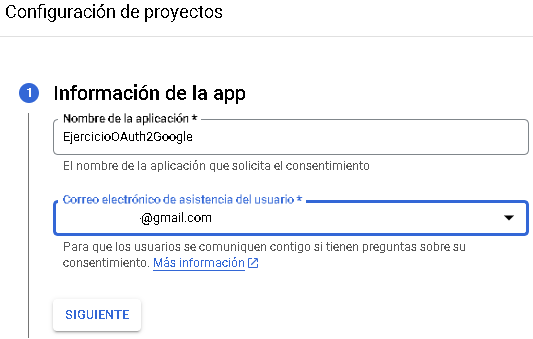
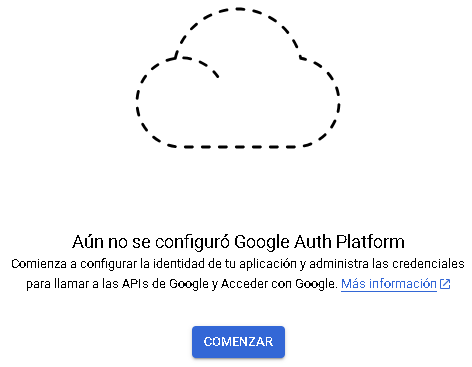
Despues, vamos al botón “Crear Credenciales”. Luego vamos a ir a Id de Cliente OAuth.

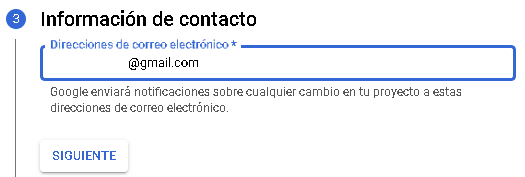


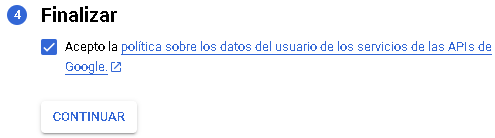
Una vez allí nos va a pedir configurar la pantalla de consentimiento, lo cual vamos a hacer haciendo click en el botón correspondiente:



Luego nos pide que configuremos Google Auth Platform



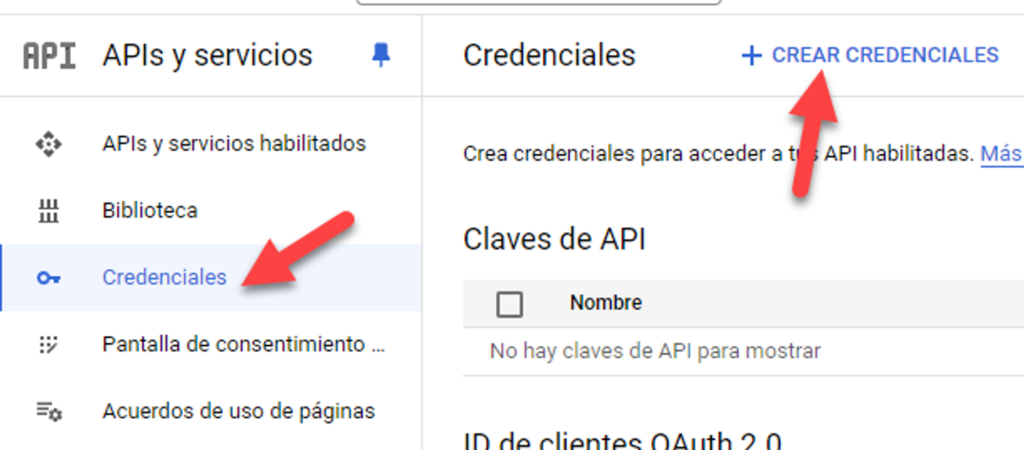




Siempre podemos volver a esta pantalla de consentimiento para editarla:

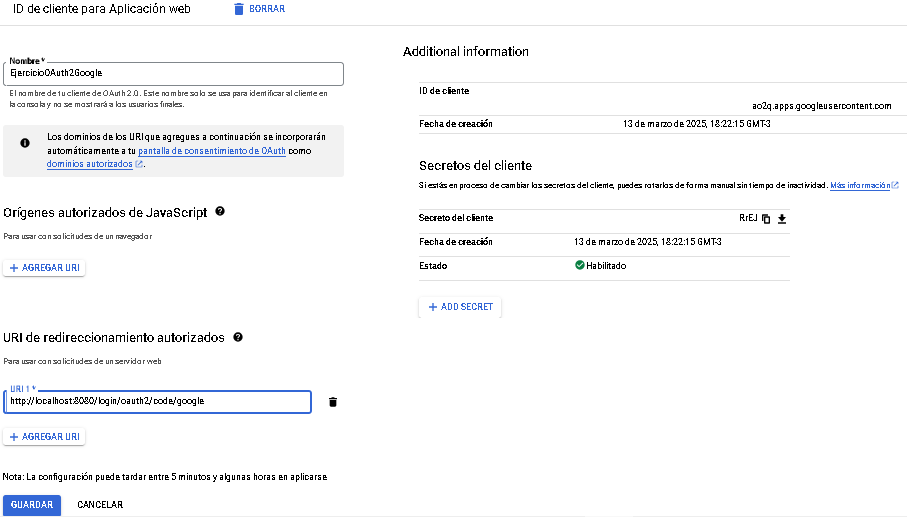


Ahora que ya tenemos creado nuestra pantalla de consentimiento, volveremos nuevamente al panel principal y allí haremos click en Credenciales >> Crear Credenciales y luego Id de Cliente de OAuth



A partir de esto, seleccionaremos que la aplicación es de tipo web, luego elegiremos un nombre y por último indicaremos la url correspondiente de OAuth2 para comunicar con Google (de igual manera que lo hicimos con GitHub donde tomamos como referencia la url ofrecida por la documentación).

Una vez configurado todo esto, Google nos dará la id de cliente y su secreto:



Tene en cuenta que la URI de redireccionamiento es:

<http://localhost:8080/login/oauth2/code/google>

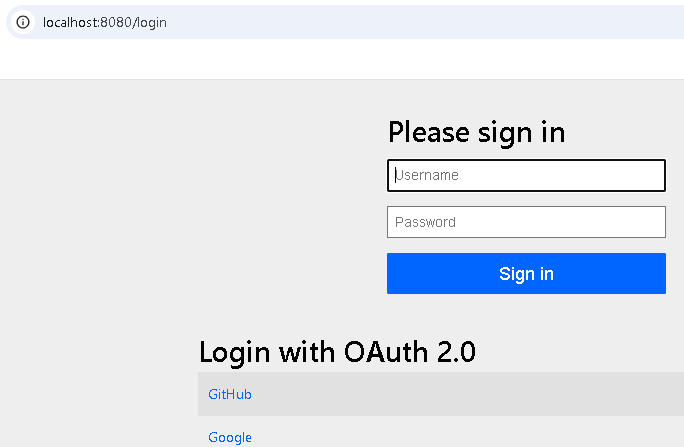
Lo último que nos queda es ir a nuestras variables de entorno y pegar los valores copiados de ambos campos:

spring.security.oauth2.client.registration.google.client-id=${GOOGLE\_CLIENT\_ID}

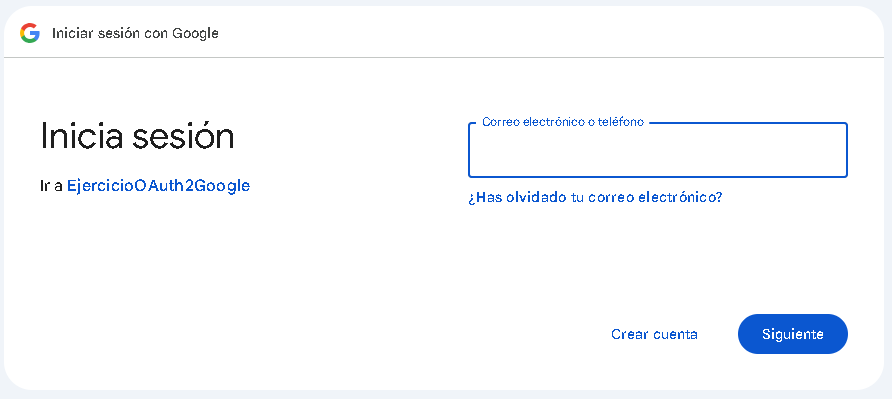
spring.security.oauth2.client.registration.google.client-secret=${GOOGLE\_CLIENT\_SECRET}

**Paso 2: Probar Login con Google**

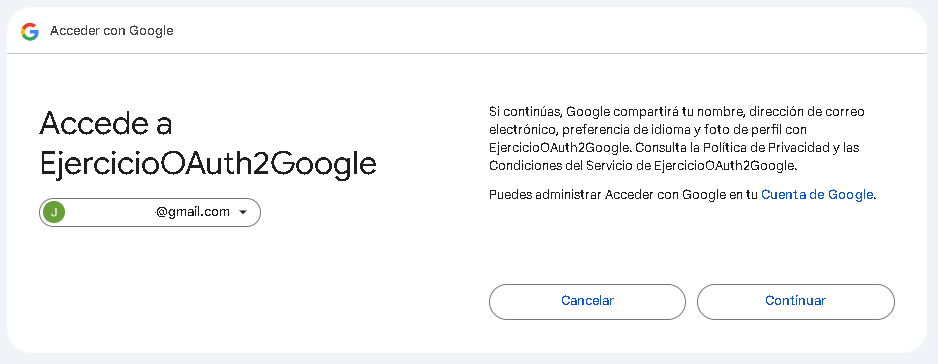
Si ejecutamos nuestro proyecto y levantamos en el navegador (http://localhost:8080), ahora en el login tendremos una nueva forma de autenticación, mediante Google:



Nos va a pedir que seleccionemos una cuenta para autenticarnos, elegiremos una cuenta de Google que tengamos que sea válida:



Luego nos pedirá que le demos permisos de acceso mediante esa cuenta (lo hacemos):



Y si todo sale bien nos redirigirá al directorio raíz de nuestra aplicación:



Si probamos entrar el endpoint securizado, podemos hacerlo sin problema alguno:

