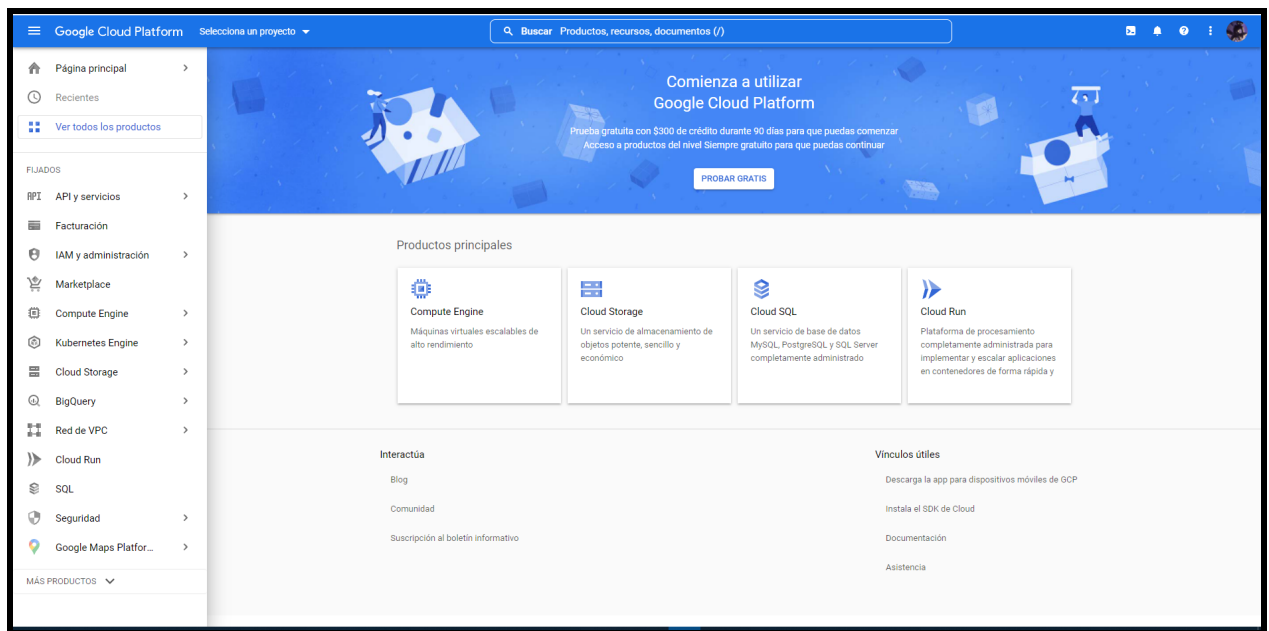


Contornos

Api de Drive

- Primero de todo tendremos que ir a [esta página](#) que es donde tenemos la Consola de Google Cloud para trabajar con nuestras aplicaciones, sacar claves de API, ajustar el OAuth y otras cosas que no cubriremos todo en este pdf. En la próxima imagen se ven los 2 o 3 apartados principales que vamos a trabajar.



- Aquí vemos 3 cosas relativamente importantes para lo que vamos a hacer.
 - La parte de “API y servicios” a la izquierda donde configuraremos casi todo.
 - La parte de Página principal donde podemos ver cosas generales cuando lo creamos.
 - La parte de “Seleccionar el proyecto” donde crearemos el proyecto.

Creando la aplicación

- Una vez le demos a la pestaña de “Seleccionar proyecto” nos dejará crear un proyecto nuevo arriba a la derecha y según le demos solo tendremos que poner nombre al proyecto, elegir organización si queremos (y deberíamos) y modificar la id (**que no podremos más adelante**) si así lo queremos.

The screenshot shows a dialog box titled "Seleccionar un proyecto". In the top right corner, there is a gear icon and the text "PROYECTO NUEVO". Below the title is a search bar with the placeholder text "Buscar en proyectos y carpetas". Under the search bar are three tabs: "RECIENTES", "DESTACADOS", and "TODOS", with "TODOS" being the active tab. Below the tabs is a table with two columns: "Nombre" and "ID". The table contains one row with the text "Sin organización" and the value "0". At the bottom right of the dialog are two buttons: "CANCELAR" and "ABRIR".

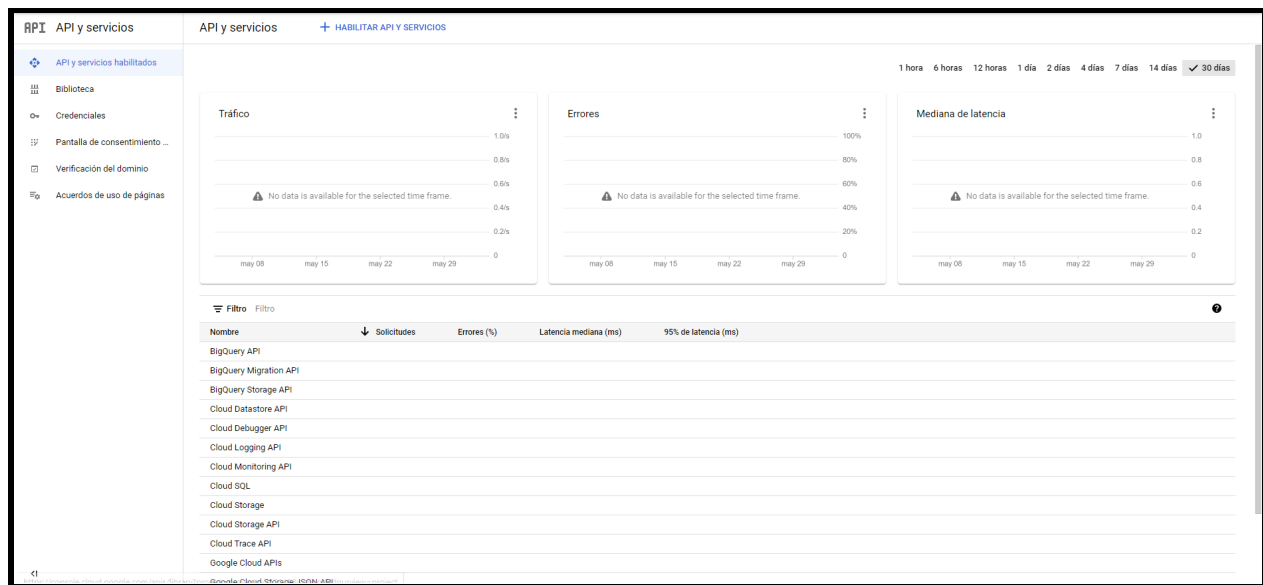
Nombre	ID
Sin organización	0

- Una vez hecho esto iremos a la ventana de API y servicios en la próxima sección, será la parte más importante.

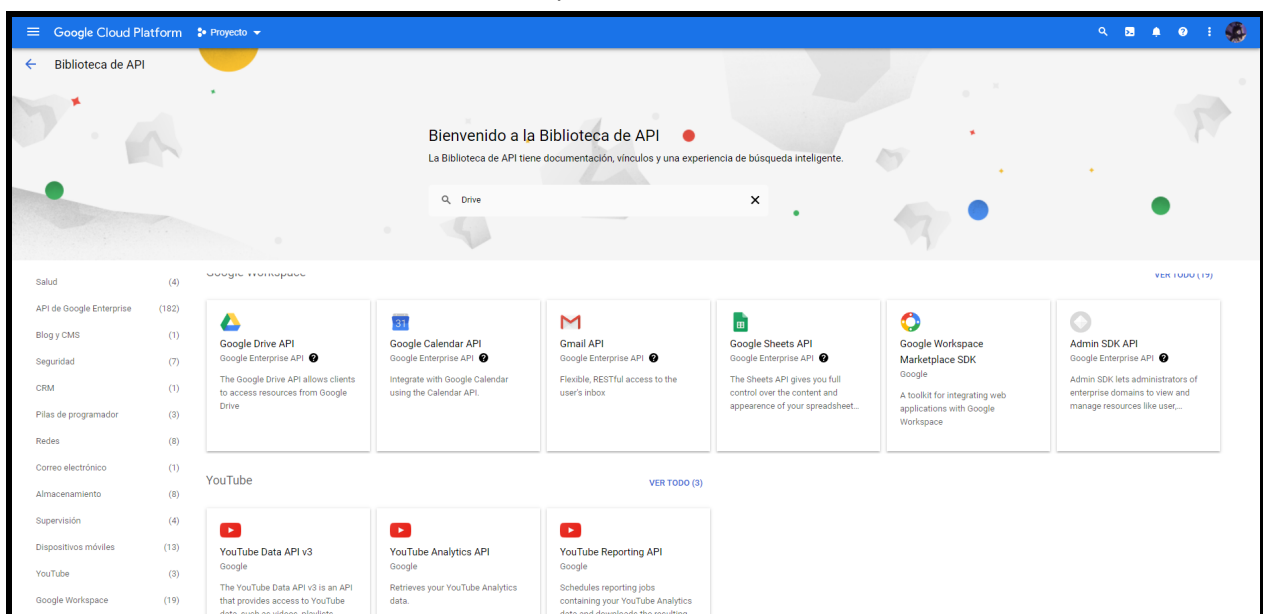
API y servicios

Configurando la biblioteca

- Llegando a la parte de “API y servicios” veremos esta pantalla en la que nos interesará la biblioteca, los credenciales y la pantalla de consentimiento. Esta pantalla además nos aportará información general.

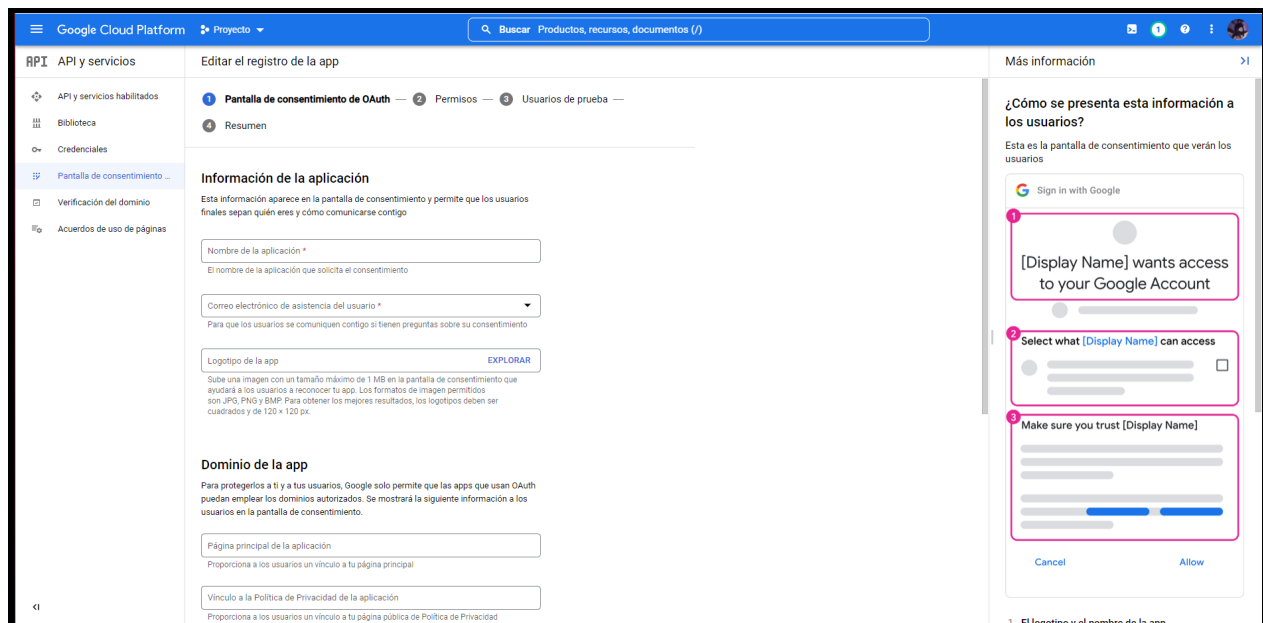


- Llegaremos a la parte de biblioteca donde veremos la siguiente captura, y deberemos buscar la API de Drive, ya sea por las opciones de abajo o con el buscador de arriba. Una vez seleccionado solo tendremos que darle al botón de habilitar.

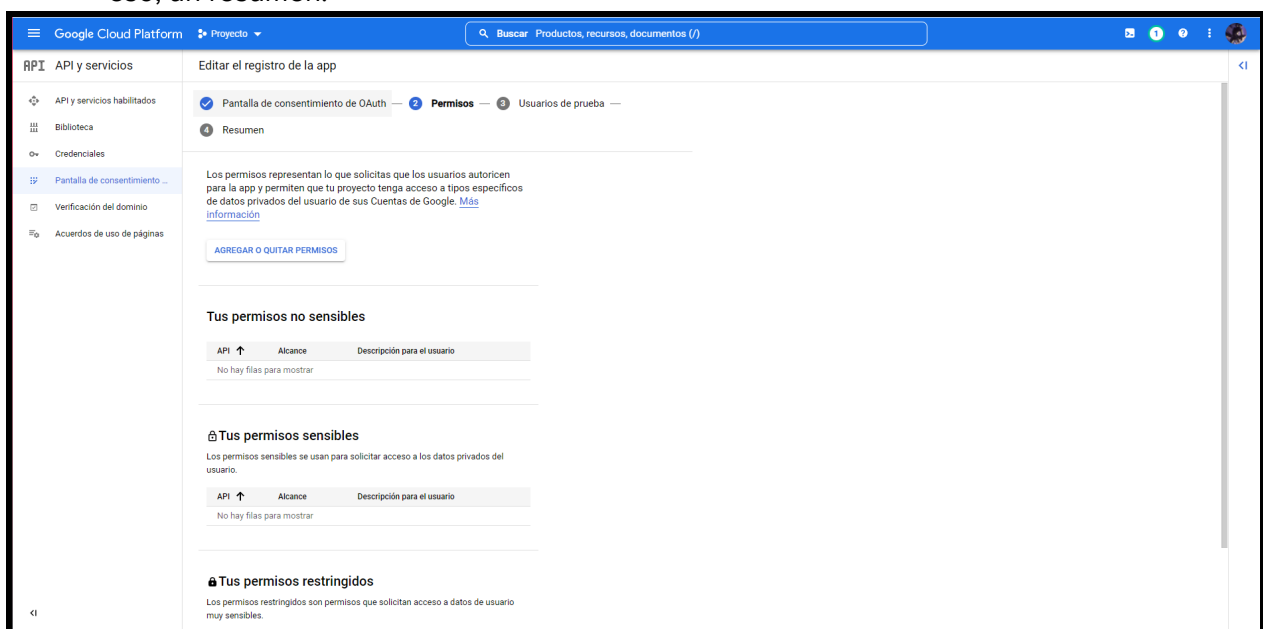


Configurando el OAuth

- Entrando a la parte de OAuth lo primero que veremos es una pantalla que nos preguntará si nuestra aplicación es interna o externa. Si a la hora de crear aplicación le hemos seleccionado una organización podremos poner la aplicación en interna que es lo ideal, si no lo pondremos en externa. Según hagamos esto veremos esta pantalla que en general se explica ella sola. En esta pantalla con poner los campos obligatorios propiamente explicados en la p

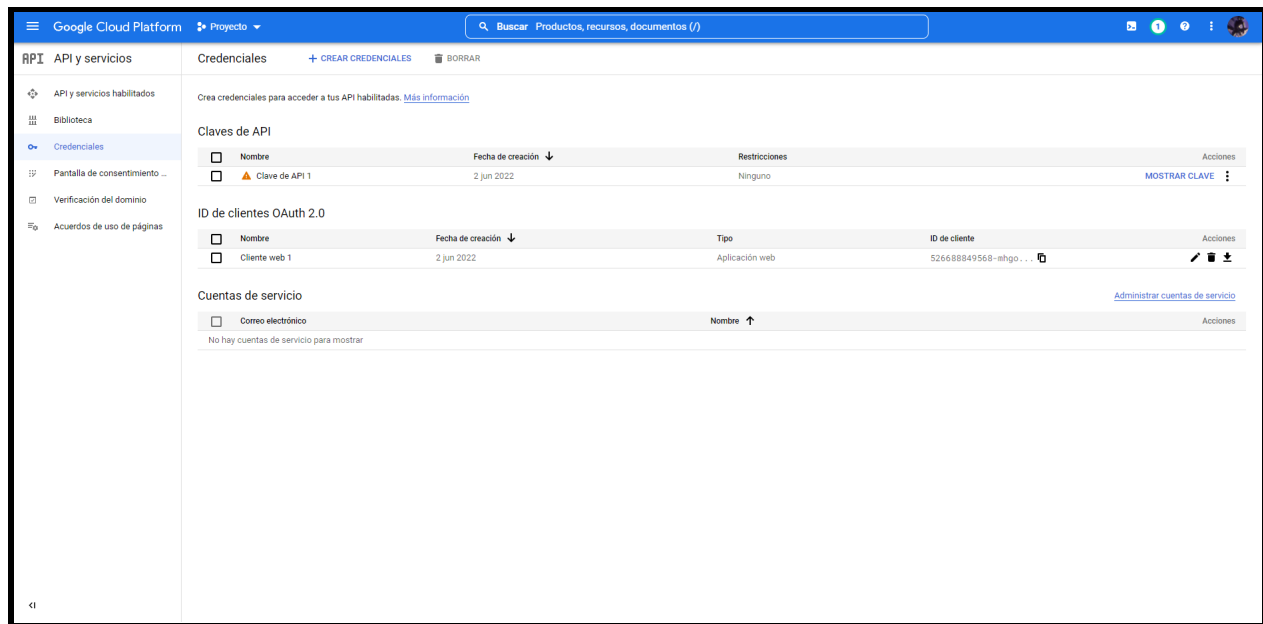


- Lo segundo que veremos son los permisos, que deberemos darles lo que nos interese, en este caso casi todo lo que aplica para Drive. La 3 no os saldrá si es interna y la 4 es eso, un resumen.

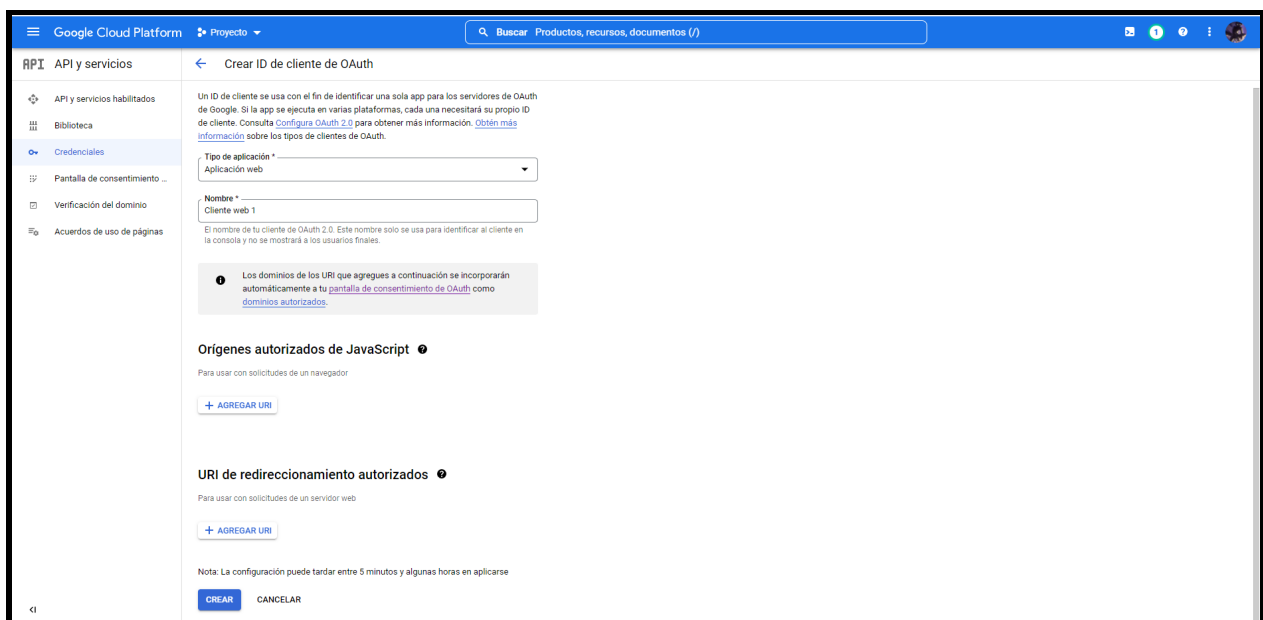


Configurando los credenciales

- Los credenciales lo haremos rápidamente. La siguiente captura es como lo veremos al acabar esos pasos. Primero le daremos a crear credenciales y lo primero será la clave de API. Solo le pondremos un nombre y ya estará.



- Lo segundo será la ID de clientes, que nos pedirá además de un nombre, el tipo de aplicación que va a ser, que lo marcaremos como Aplicación Web. Una vez hecho eso descargaremos el JSON que nos dará opción al acabar de configurar la ID de cliente o que si no podremos hacer desde la captura de arriba en el lápiz. La config se ve abajo.



El código

De donde sacamos la base y cambios base

1. Primero tenemos el código base [aquí](#). Tendremos que hacerle un par de cambios como veremos en la última captura. Importante hacerlo con Gradle como el ejemplo.
 - a. Primero en la línea 34 tendremos que poner solo .DRIVE, no con data como viene.
 - b. Segundo tendremos que añadir las credenciales, poniendo la ruta de las mismas que descargamos anteriormente.
 - c. Tercero, tenemos que poner la userid, que bien podemos sacar del Json o de la página.
- El resto de la aplicación es recomendable ver el código linkeado al principio, esta con comentarios traducidos y se debería entender lo suficiente.

```
30  /**
31   * Global instance of the scopes required by this quickstart.
32   * If modifying these scopes, delete your previously saved tokens/ folder.
33   */
34  private static final List<String> SCOPES = Collections.singletonList(DriveScopes.DRIVE);
35  private static final String CREDENTIALS_FILE_PATH = "/credentials.json";
36
37  /**
38   * Creates an authorized Credential object.
39   * @param HTTP_TRANSPORT The network HTTP Transport.
40   * @return An authorized Credential object.
41   * @throws IOException If the credentials.json file cannot be found.
42   */
43  private static Credential getCredentials(final NetHttpTransport HTTP_TRANSPORT) throws IOException {
44      // Carga la información de credenciales del cliente.
45      InputStream in = DriveQuickstart.class.getResourceAsStream(CREDENTIALS_FILE_PATH);
46      if (in == null) {
47          throw new FileNotFoundException("Resource not found: " + CREDENTIALS_FILE_PATH);
48      }
49      GoogleClientSecrets clientSecrets = GoogleClientSecrets.load(JSON_FACTORY, new InputStreamReader(in));
50
51      // Carga y triggera una credencial
52      GoogleAuthorizationCodeFlow flow = new GoogleAuthorizationCodeFlow.Builder(
53          HTTP_TRANSPORT, JSON_FACTORY, clientSecrets, SCOPES)
54          .setDataStoreFactory(new FileDataStoreFactory(new java.io.File(TOKENS_DIRECTORY_PATH)))
55          .setAccessType("offline")
56          .build();
57      LocalServerReceiver receiver = new LocalServerReceiver.Builder().setPort(8888).build();
58      Credential credential = new AuthorizationCodeInstalledApp(flow, receiver).authorize("847529586564-13a2kfoigs98oa4qcu4ou033tq25t39.apps.googleusercontent.com");
59      //Retorna un objeto autorizado.
60      return credential;
61  }
```