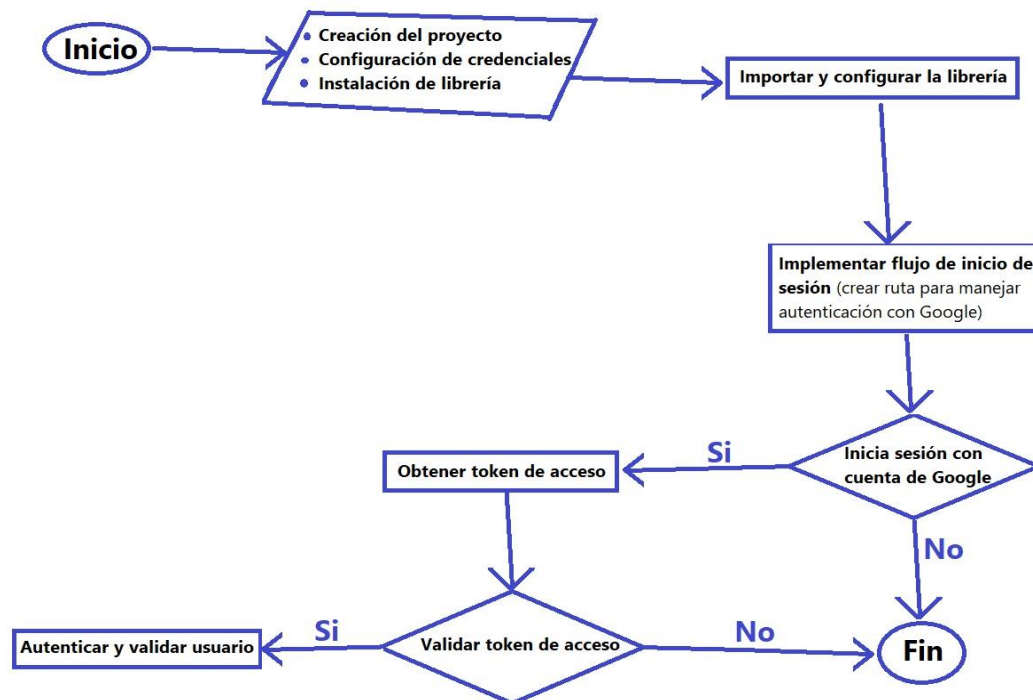


# Como funciona el servicio de inicio de sesión con Google en el api rest de la plataforma web Dory

## Objetivo:

Proporcionar una explicación de cómo funciona el servicio de inicio de sesión con Google en el api rest de la plataforma web Dory.



## Recursos necesarios:

- Api rest de la plataforma web Dory.
- Cuenta de desarrollador Google con acceso al servicio de autenticación de Google.
- Acceso a internet.
- Un entorno de desarrollo con Node.js versión 16.14.1 y npm versión 8.5.0 instaladas en el sistema.
- Express versión 4.17.1

## Piezas de software:

- Librería de autenticación de Google (Google-auth-library) versión 7.14.1 instalada en el proyecto.

## Pasos:

1. Abrir el api rest de la plataforma web Dory:
  - Abrir el entorno de desarrollo del api rest de la plataforma web Dory.
2. Configurar el servicio de autenticación en Google:
  - El cliente obtiene un ID DE CLIENTE OAUTH2.0
  - El cliente obtiene un idToken de Google.
3. Importar y configurar la librería:
  - En el archivo del servicio de inicio de sesión con google de la aplicación web Dory, importar la librería Google Auth Library.
  - Configura la librería utilizando las credenciales generadas (ver imagen 2).

```
const { OAuth2Client } = require('google-auth-library');  
  
const CLIENT_ID = 'TU_CLIENT_ID';  
  
const client = new OAuth2Client(CLIENT_ID);
```

Imagen 2. Configuración de Librería 'google-auth-library'

Asegúrate de reemplazar 'TU\_CLIENT\_ID' con el ID de cliente proporcionado por Google.

4. Implementar el flujo de inicio de sesión:
  - Crea una ruta o controlador en tu aplicación web para manejar la autenticación con Google.

```
router.post('/google', async function(req, res, next) {  
  try {  
    res.json(await login.loginWithGoogle(req));  
  } catch (err) {  
    console.error(`Error al loguear el usuario con google`, err.message);  
    next(err);  
  }  
});
```

Imagen 3. Ejemplo de una ruta '/google'

5. Obtener el token de acceso:

- En el servicio de inicio de sesión de Google se debe recibir el idToken de Google.

6. Validar el token de acceso:

- Para asegurarte de que el token de acceso es válido y proviene de Google, puedes utilizar la librería Google Auth Library para validar el token del usuario (ver imagen 5); después asignar un token propio de la aplicación web Dory a ese usuario.

```
const token = req.body.token;
const ticket = await client.verifyIdToken({
  idToken: token,
  audience: [CLIENT_ID_1, CLIENT_ID_2]
});
const payload = ticket.getPayload();
let email = payload.email;
let creadoCon = 'Google';
const newToken = await helper.createToken(rows[0], ONE_YEAR_MILLISECONDS);
return {
  token: newToken,
  authenticated: true,
  creadoCon
};
```

Validación de token

Creación de nuevo token de usuario

Imagen 5. Validación de token de usuario

7. Utilizar el inicio de sesión con Google en tu aplicación:

- A partir de este punto, puedes utilizar el inicio de sesión con Google en tu aplicación web para autenticar y autorizar a los usuarios.
- Por ejemplo, puedes guardar el ID de usuario de Google en tu base de datos y utilizarlo para identificar a los usuarios en tu aplicación.