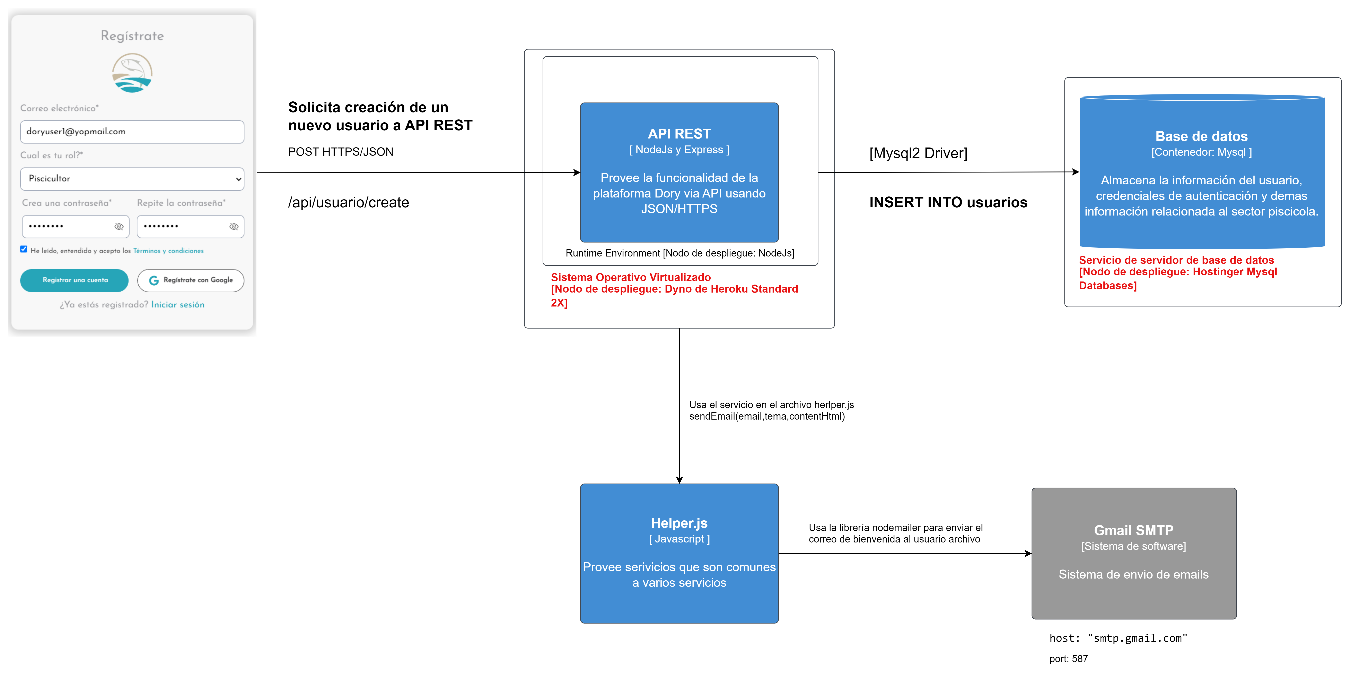
**Codificación del proceso de registro en Dory**

**Objetivo:** Conocer cuáles fueron los métodos implementados para registrar usuarios en la aplicación web Dory

El proceso de registro de la plataforma Dory se lleva a cabo cuando los datos que el usuario llenó en el formulario de registro o que están en su cuenta de Google quedan almacenados en la tabla *usuarios* de la base de datos ***process.env.DB\_NAME*** y su cuenta queda verificada.



Cuando el registro es exitoso el API REST envía un correo a través del servicio Gmail SMTP de Google utilizando la librería nodemailer.

A continuación, se presentan los dos métodos de registros implementados en la plataforma Dory:

* El primer método implementa un registro realizando una solicitud al endpoint /api/usuario/create **API-REST** de la plataforma Dory.
* El segundo método solicita información del usuario a Google y la envía al **API-REST**.

Para conocer detalles sobre el consumo del **API-REST** puedes revisar la receta – “Explicación como se conecta la aplicación Dory web con el API-REST de la plataforma”

**Recursos necesarios:**

* HttpClient de Angular
* Servicio del servicio usuario.service.ts
* Cuenta de Google
* Base de datos ***process.env.DB\_NAME*** en línea
* API-REST de Dory

**Pasos:**

**0. Conocer la base de datos tabla usuarios**

La siguiente imagen muestra la tabla (tabla *usuarios*) donde se almacenarán los datos del registro. Para este proceso solo se necesitaran los campos *email, id\_tipo\_usuario, password*. Los demás datos se pueden llenar cuando el usuario se encuentre dentro de la plataforma.

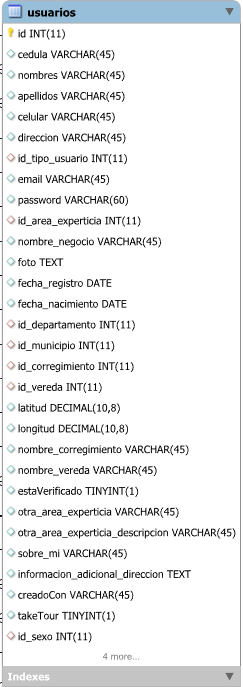


Imagen 1. Tabla usuarios de la base de datos de la plataforma Dory.

**1. Conocer el método de registro usando formularios**

En la aplicación web Dory, se implementó un método de registro utilizando formularios estándar. Los pasos para el registro mediante formularios son los siguientes:

a. El usuario accede a la página de registro en la aplicación.



Imagen 2. Pagina home de la aplicación web Dory.

b. Se muestra un formulario que solicita al usuario ingresar su correo electrónico, rol en la plataforma y contraseña.

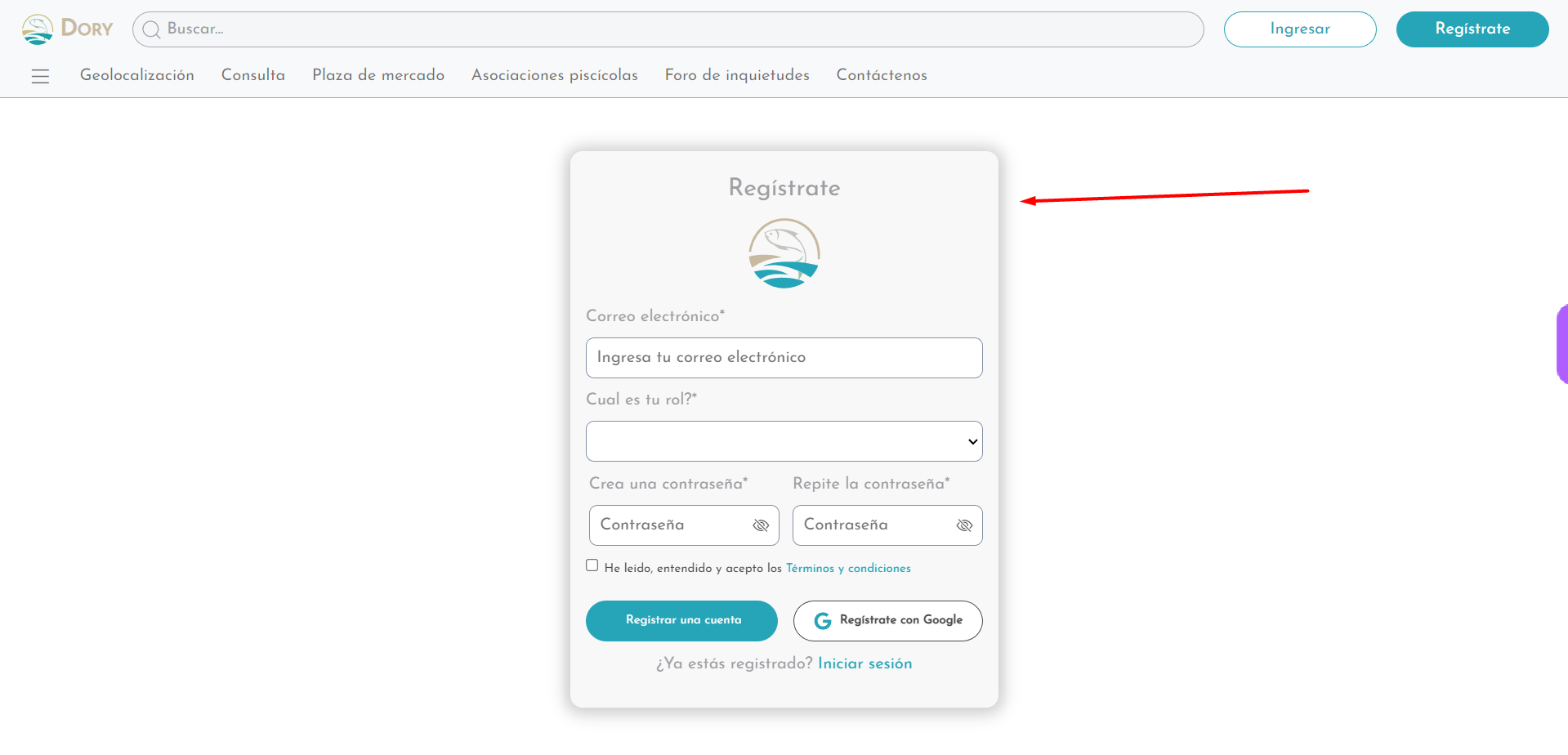


Imagen 3. Formulario de registro de la aplicación web Dory.

c. El usuario completa los campos del formulario con la información solicitada

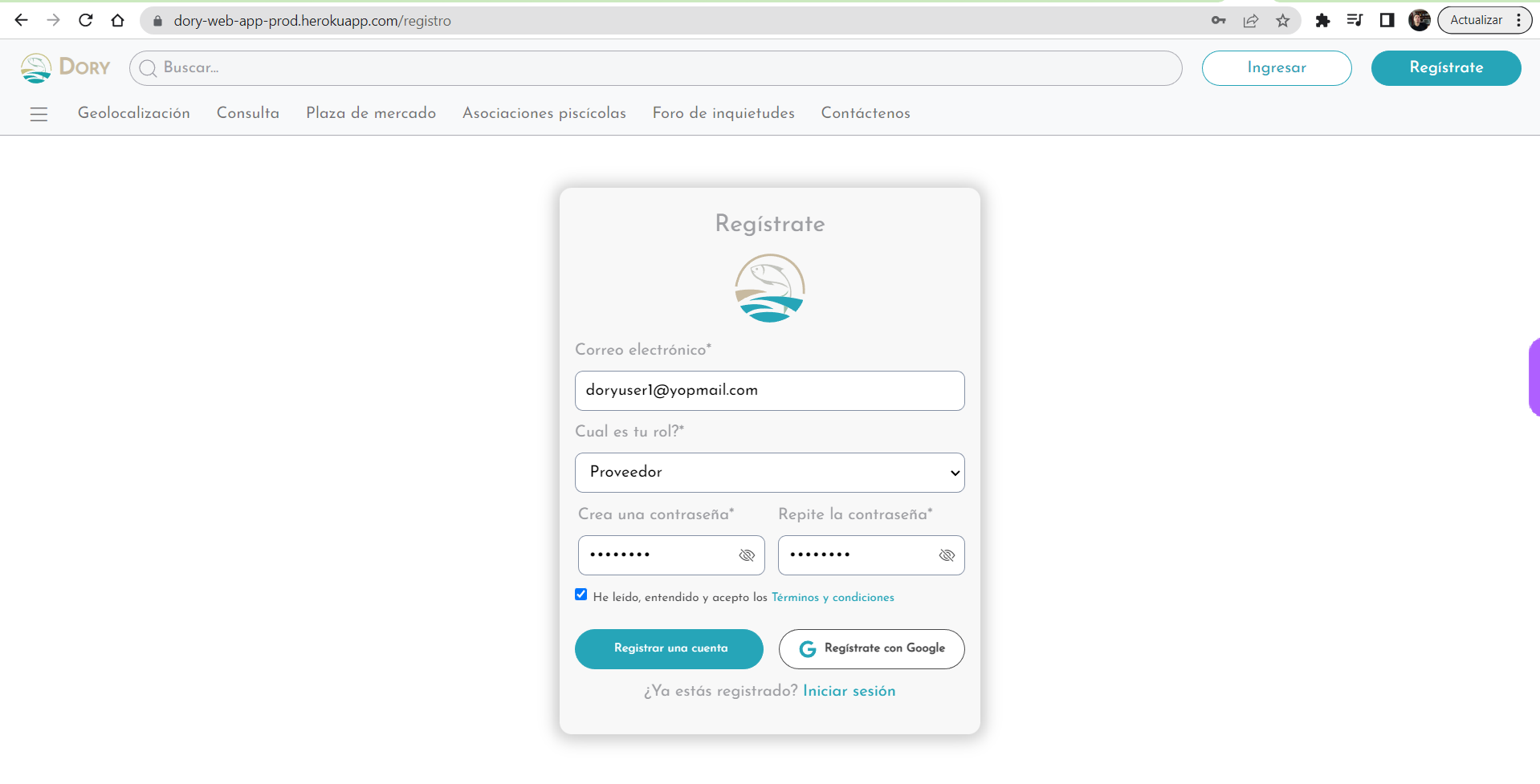


Imagen 4. Datos del formulario ingresados por el usuario.

d. El componente RegistroComponent verifica los datos ingresados en el formulario y envía una solicitud al API REST para crear un nuevo usuario y este a su vez envía los datos a la base de datos.

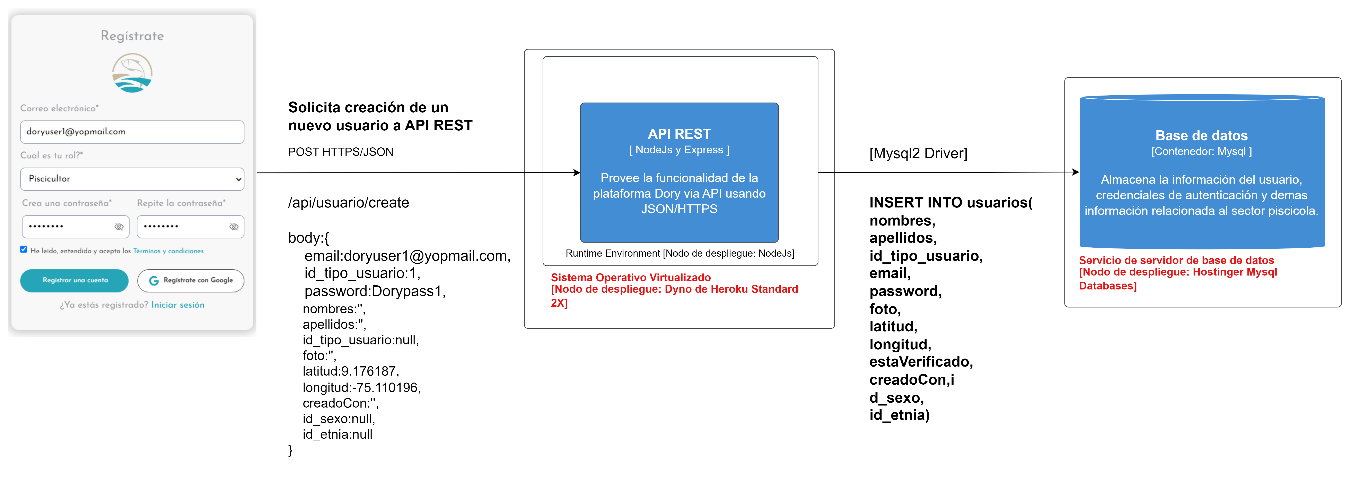


Imagen 5. Diagrama registro de un usuario en la plataforma Dory

A continuación, se explicará paso a paso la Imagen 5:

Primero veamos el paso **Solicita creación de un nuevo usuario al API-REST**

El proceso de solicitar un nuevo usuario al API-REST comienza en el componente RegistroComponent(/registro/components/registro.component.ts) en el método ***onSubmit().*** Este captura y verifica que los datos ingresados en el formulario sean correctos. Luego arma el objeto newUser con la siguiente estructura:

  newUser:{

    cedula: '',

        nombres: '',

        apellidos: '',

        email: 'doryuser1@yopmail.com',

        tipoUsuario:1,

        fechaNac: '',

        celular: '',

        password: 'Dorypass1',

        matchPassword: 'Dorypass1',

        departamento: '',

        municipio: '',

        corregimiento:'',

        vereda:'',

        terms: true

  }

Y lo envía a /services/usuario.service.ts a través del método registrarUsuario(usuario).

this.usuarioService.registrarUsuario(newUser)

El servicio usuario.service.ts arma un objeto body en el método registrarUsuario(usuario) con la siguiente estructura a partir del objeto newUser, extrayendo solo los datos necesarios.

body = {

      nombres: usuario.nombres || '',

      apellidos: usuario.apellidos || '',

      id\_tipo\_usuario: usuario.tipoUsuario || null,

      email: usuario.email || '',

      password: usuario.password || '',

      foto: usuario.foto || '',

      latitud: usuario.latitud || 0,

      longitud: usuario.longitud || 0,

      creadoCon: usuario.creadoCon || '',

      id\_sexo:null,

      id\_etnia:null

    };

Luego envía a través del servicio HttpsServices (/services/https.service.ts) la solicitud de creación de un nuevo usuario enviándole el objeto body que es donde se encuentra los datos a registrar.

this.httpsService.post(

      environment.doryApiRestBaseUrl+'/usuario/create',

body

);

Para conocer mas detalles de como funciona el servicio HttpsService consulte la receta “Explicación del funcionamiento del servicio HttpsService”

**El API-REST recibe los datos**

El API-REST recibe la solicitud haciendo uso del framework express.js en la ruta **/api/usuario** y reenvía la petición al enrutador /routes/usuario.js la cual internamente elige entre todas sus rutas enviarla por la siguiente ya que coincide con la ruta y verbo POST solicitados por la aplicación web.

router.post('/create', async function(req, res, next) {

  try {

    var token=req.headers.authorization;

    res.json(await usuario.create(req.body,token));

  } catch (err) {

    console.error(`Error al registrar el usuario`, err.message);

    next(err);

  }

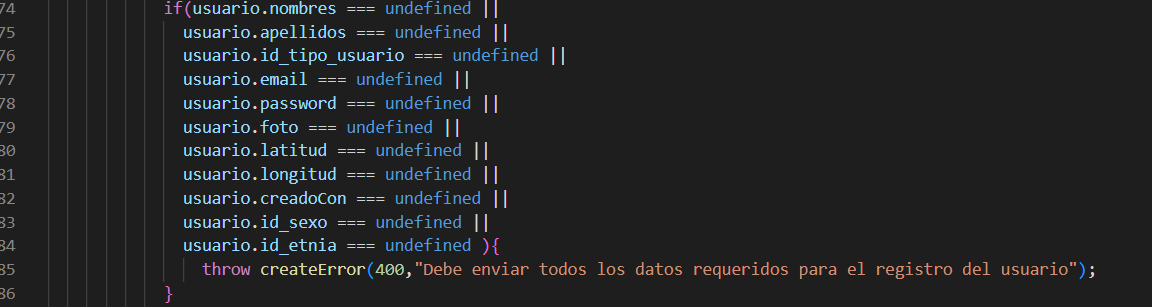
});/\* Create del usuario\*/

Dentro de esta ruta elegida por el enrutador se llama al método ***create*** del servicio /services/usuario.js y se le envia el body enviado desde el formulario de registro:

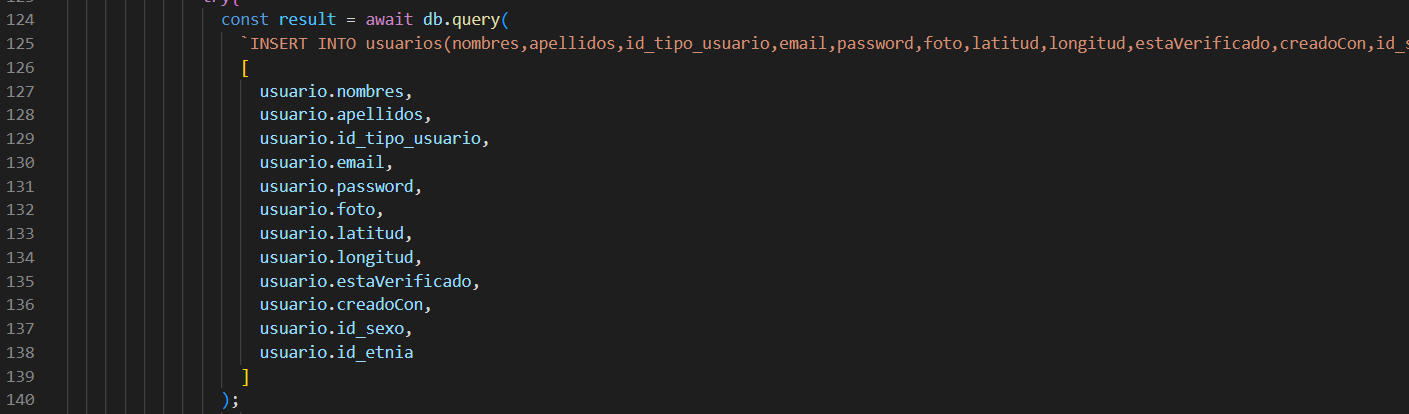
 res.json(await usuario.create(req.body,token));

En el método create del servicio usuario.js sucede lo siguiente:

* Se verifica que en los datos enviados en el body estén los necesarios:

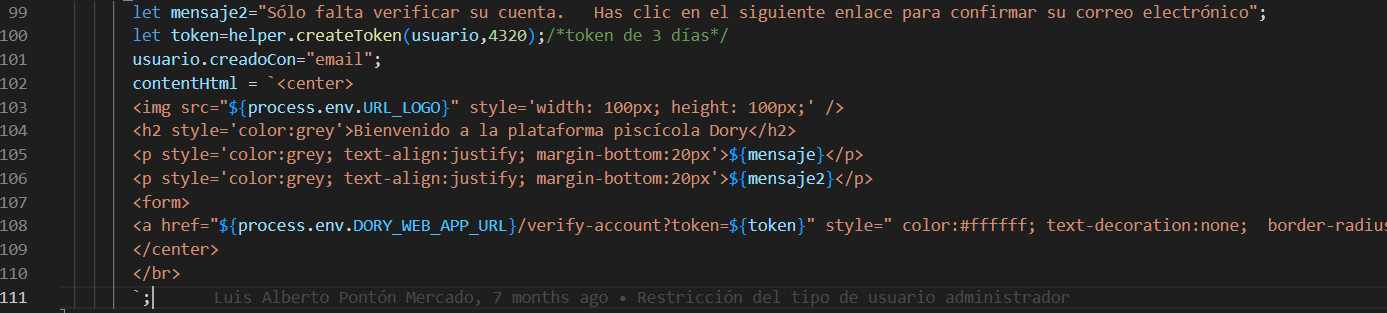


Se insertan los datos del body en la base de datos en la tabla usuarios:

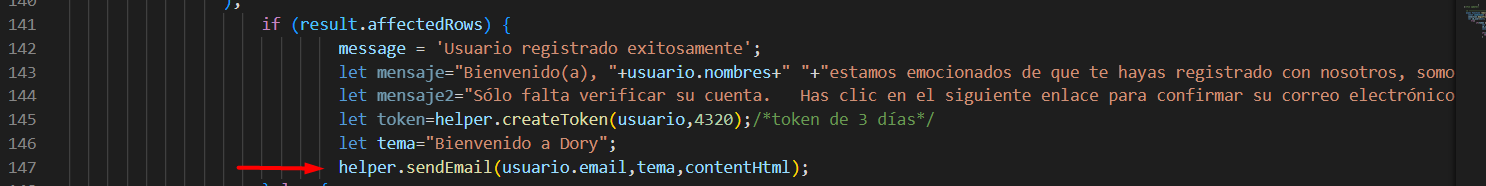


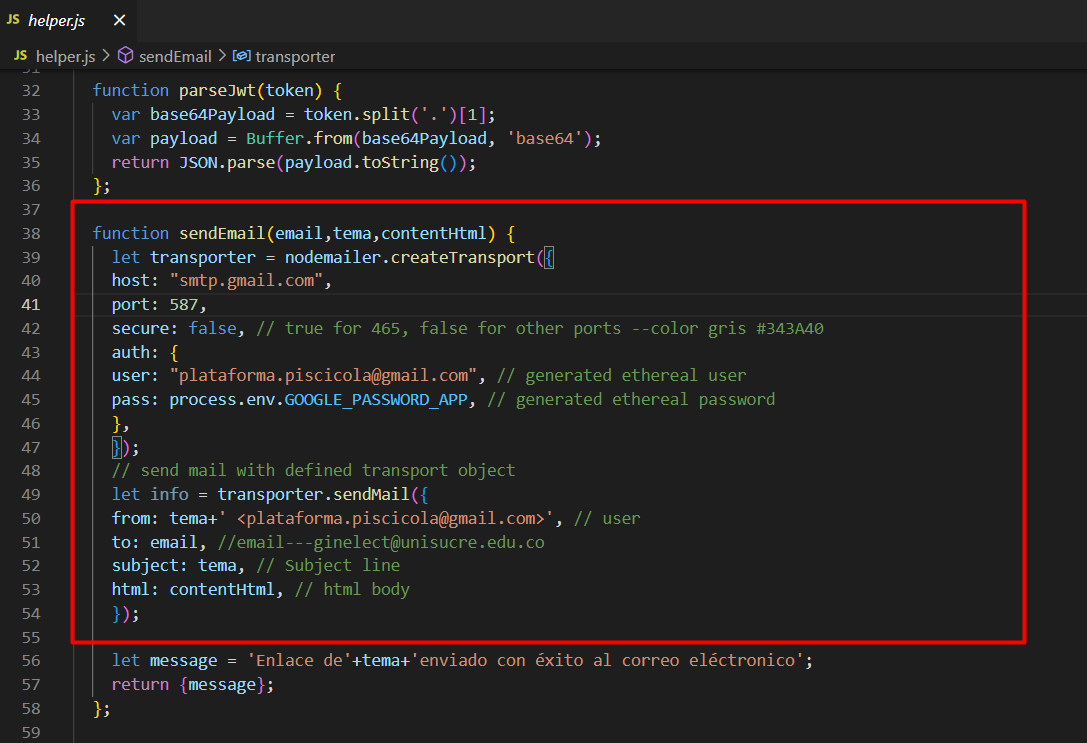
* Se prepara un mensaje de bienvenida que será enviado por correo y se le concatena con un enlace con un token de 3 días que permitirá al usuario verificar su cuenta.

let mensaje="Bienvenido(a), "+usuario.nombres+" "+"estamos emocionados de que te hayas registrado con nosotros, somos un equipo conformado por emprendedores y profesionales que trabajan día a día para promover la productividad y competitividad de la cadena piscícola del Departamento de Sucre, en alianza con los grupos de investigación, Gestión de la Producción y la Calidad y GINTEING, de la Universidad de Sucre y la Corporación Universitaria Antonio José de Sucre.";



* Por último se envía por SMTP el correo de bienvenida.

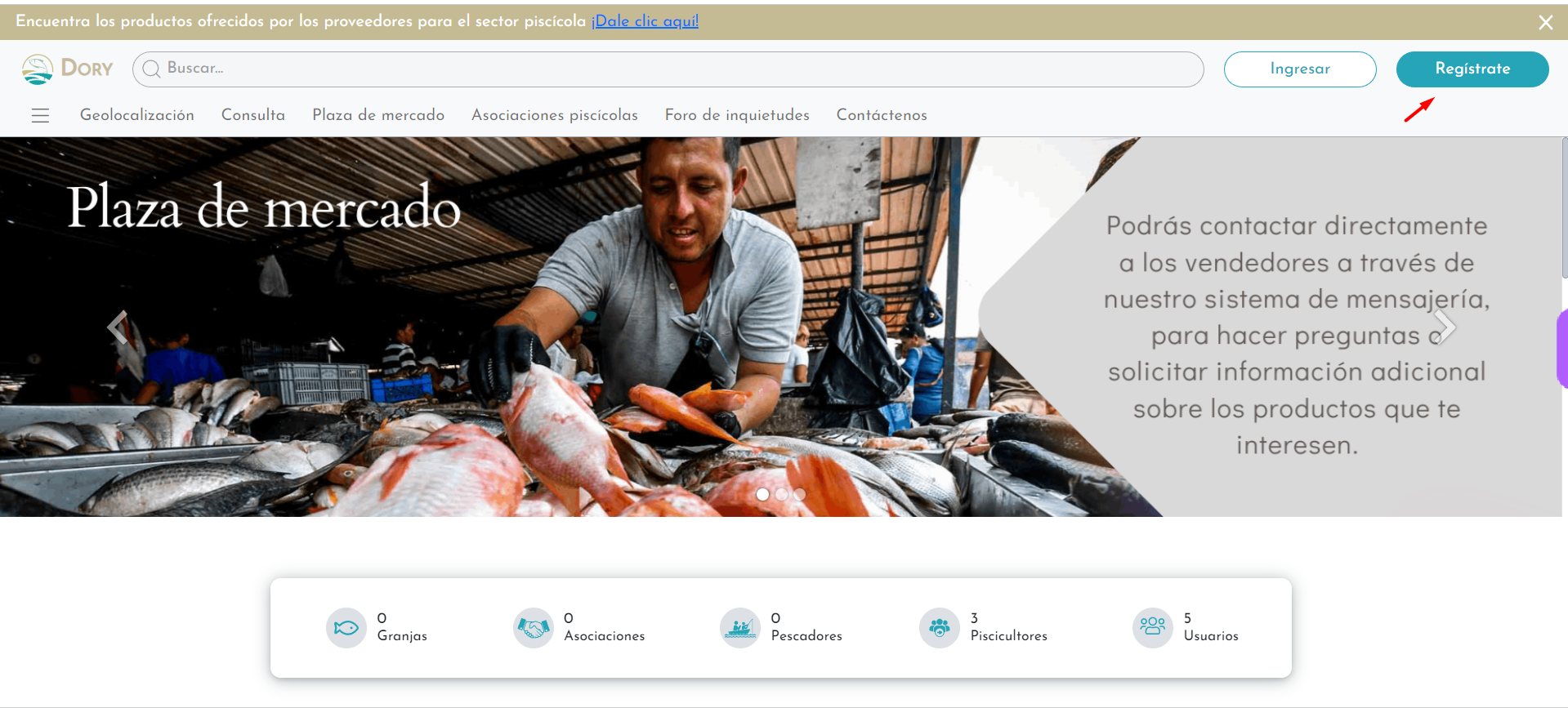




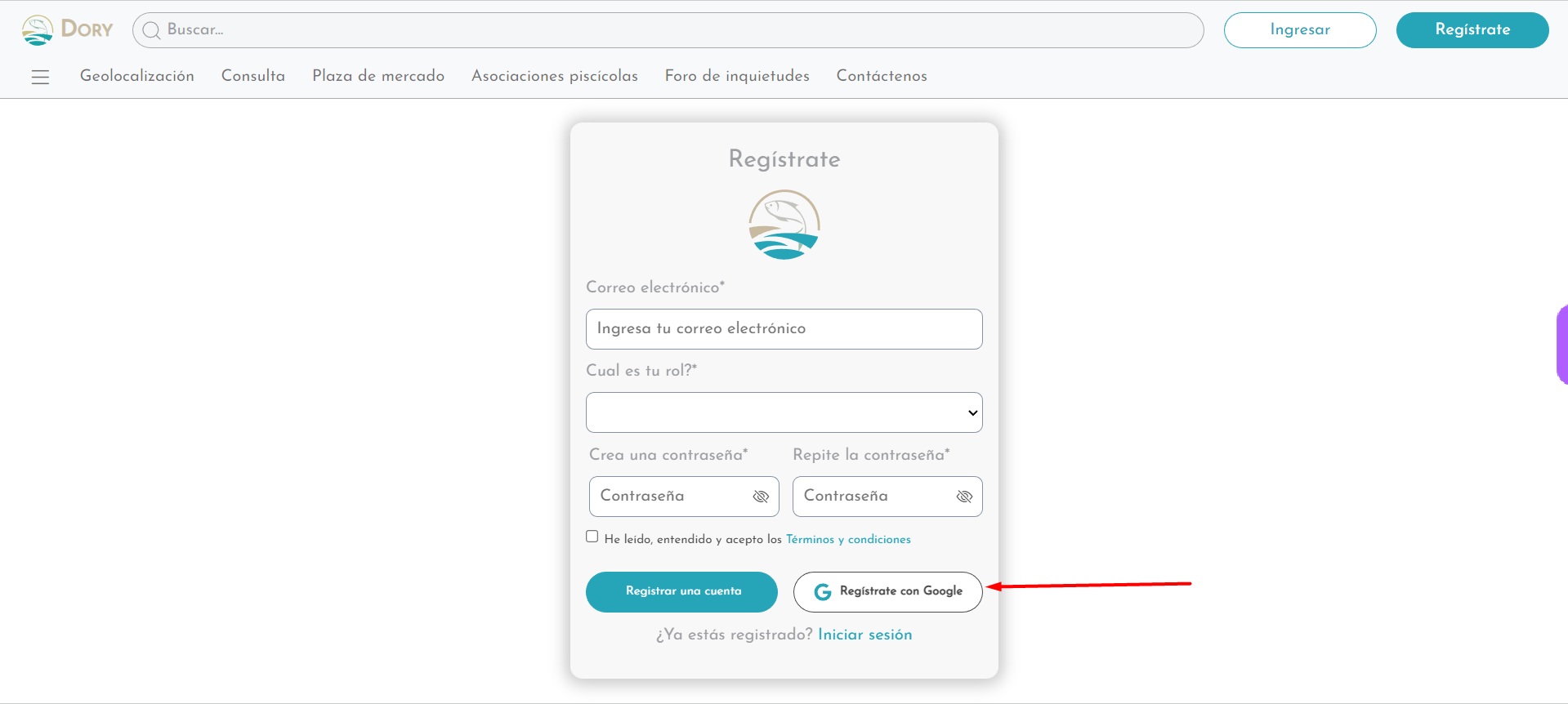
**2. Conocer el método de registro usado por google**

En la aplicación web Dory, también se implementó un método de registro utilizando los servicios de autenticación de Google. Los pasos para el registro con Google son los siguientes:

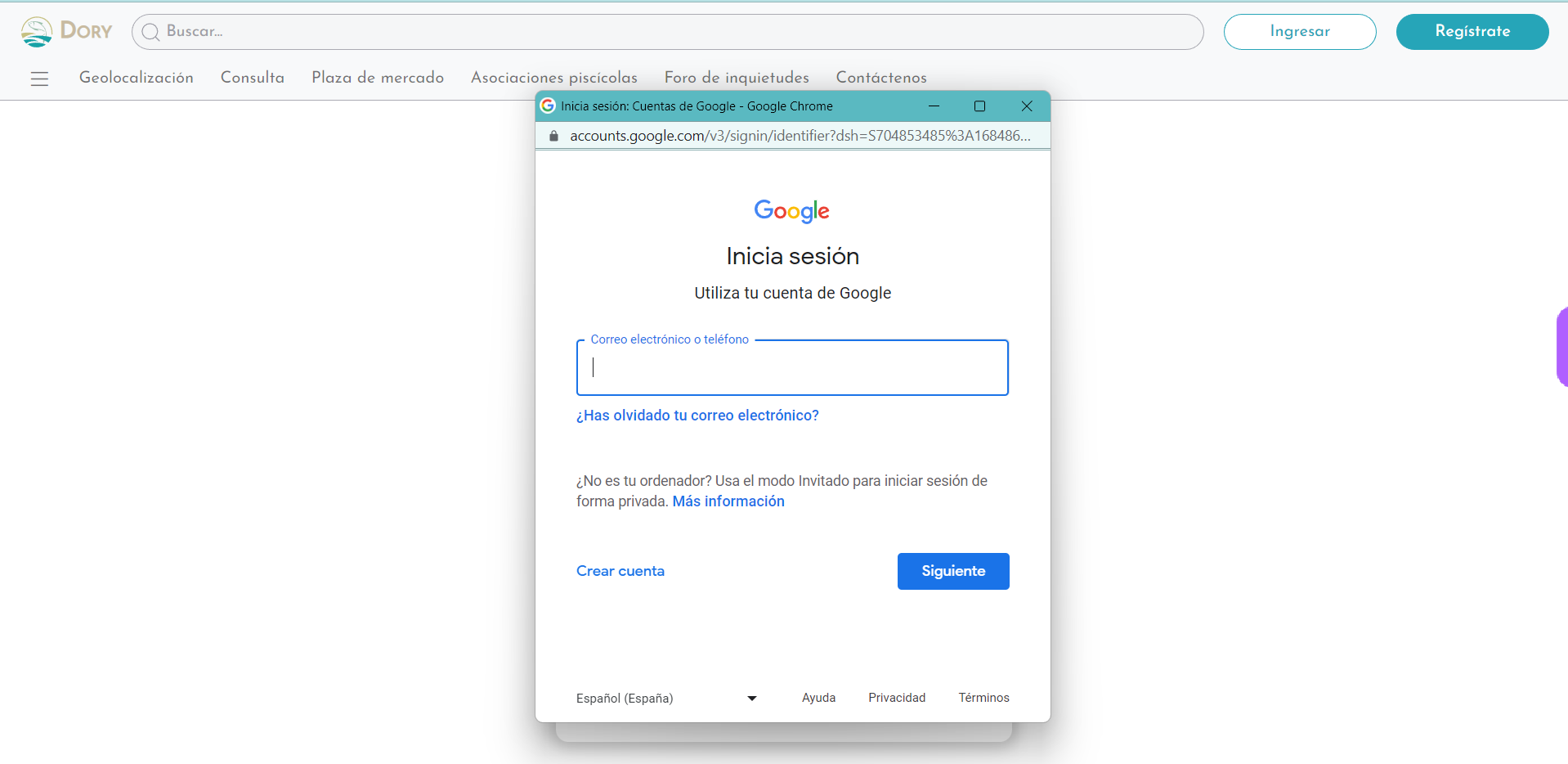
a. El usuario accede a la opción de registro con Google en la aplicación.



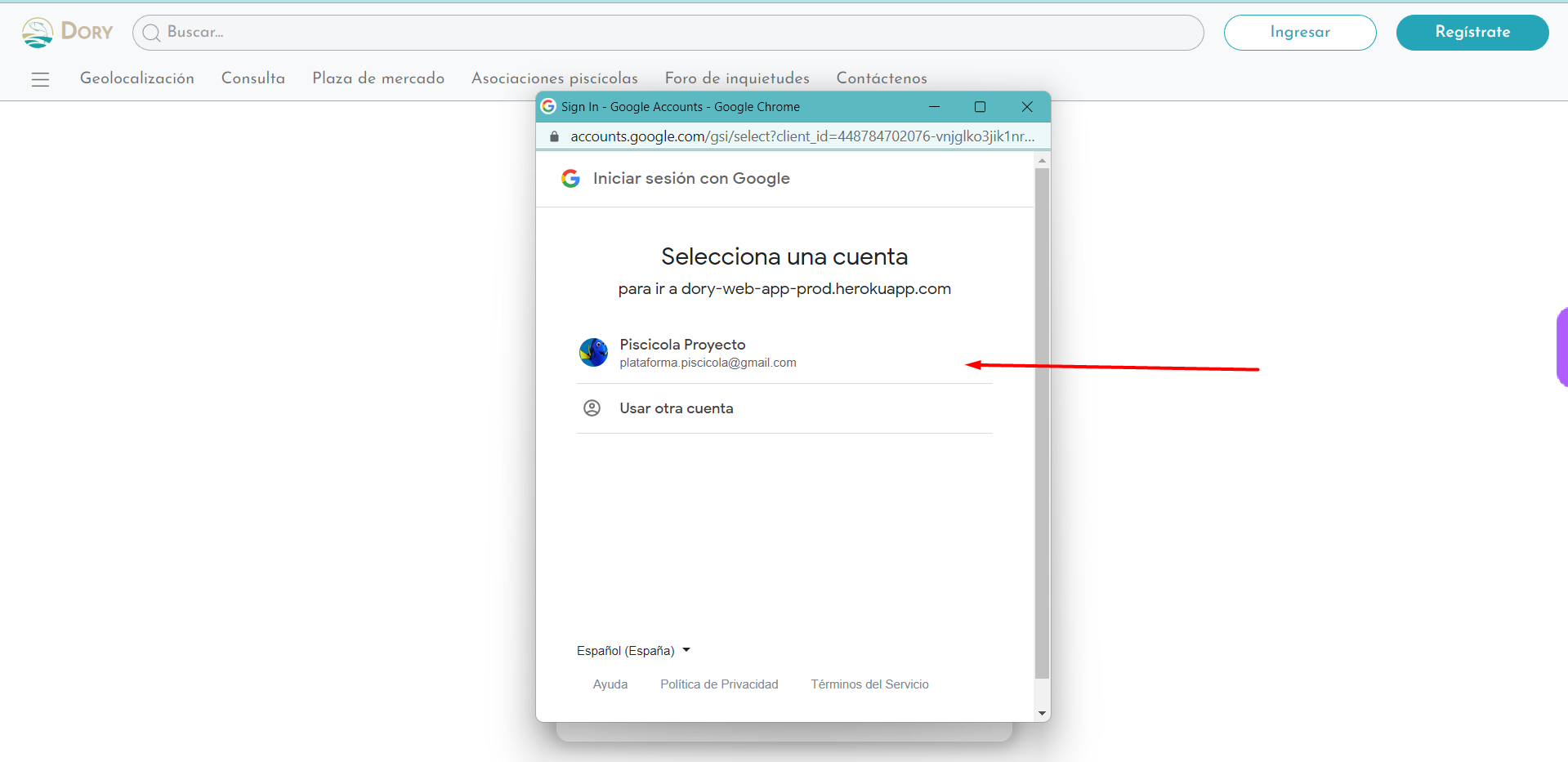
b. Se muestra un botón o enlace que permite al usuario iniciar sesión con su cuenta de Google.



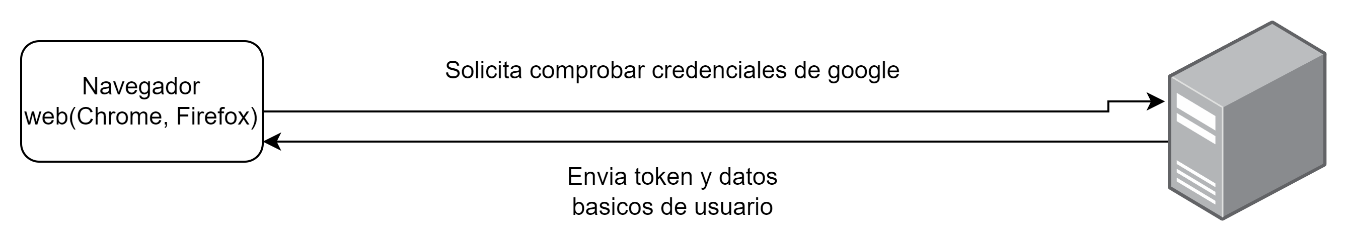
c. Al hacer clic en el botón o enlace, se abre una ventana emergente o se redirige al usuario a la página de inicio de sesión de Google

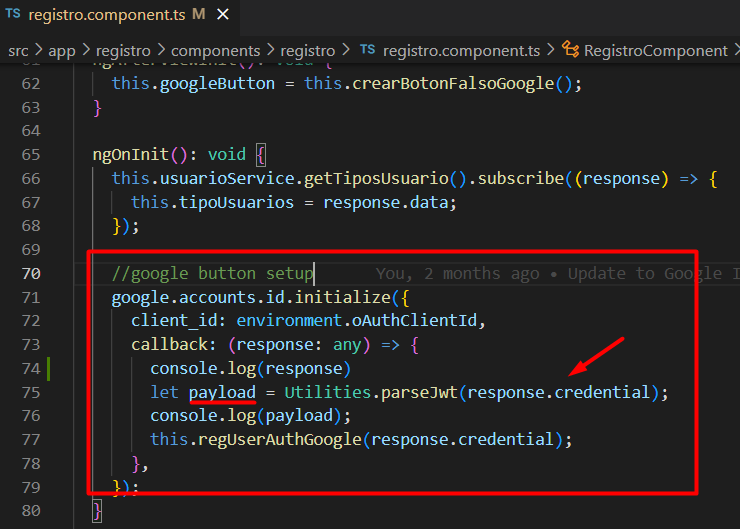


d. El usuario ingresa sus credenciales de inicio de sesión de Google.



e. Si el inicio de sesión es exitoso, Google devuelve un token de acceso a la aplicación web Dory.





En el callback que se ejecuta luego de que la autenticación fue exitosa recibimos en idToken en el parámetro response en el campo credential(response.credential).

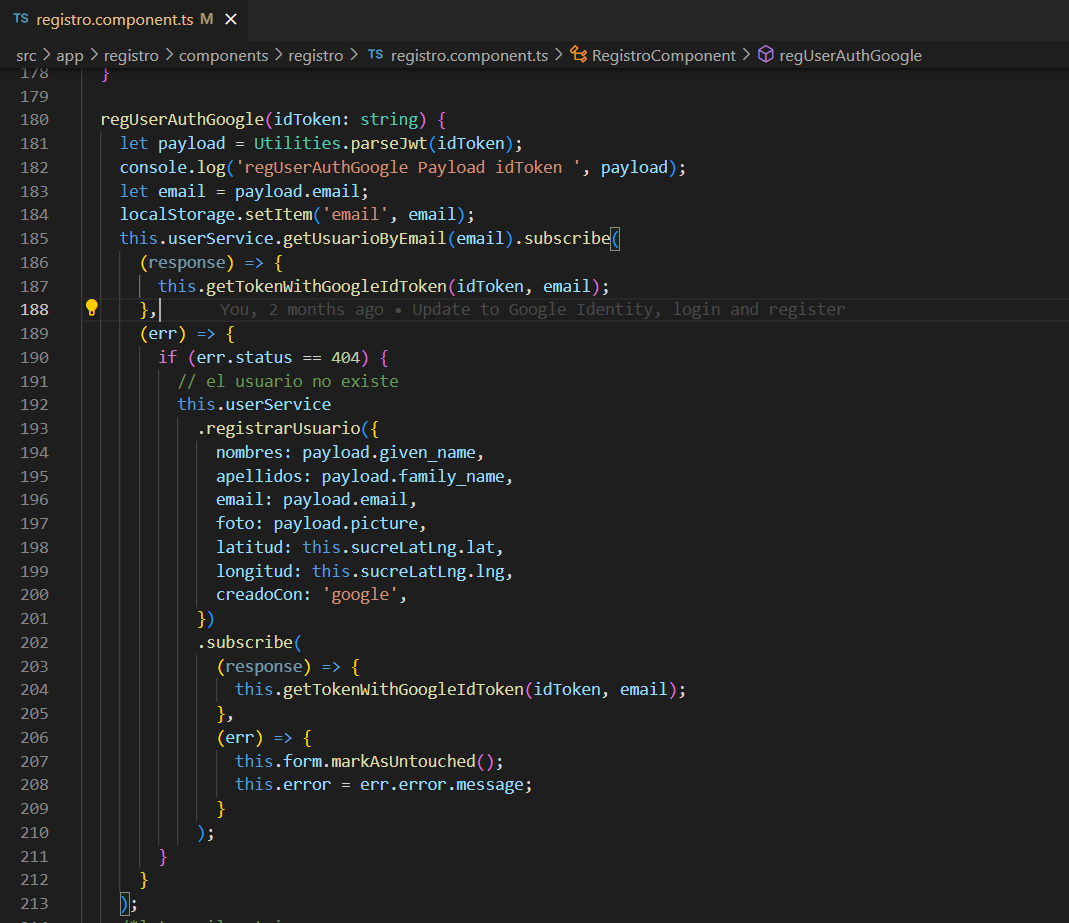
f. Si el usuario es autenticado correctamente, se le solicita al **API-REST** crear una cuenta de usuario en la base de datos ***process.env.DB\_NAME*** con los datos proporcionados por Google.



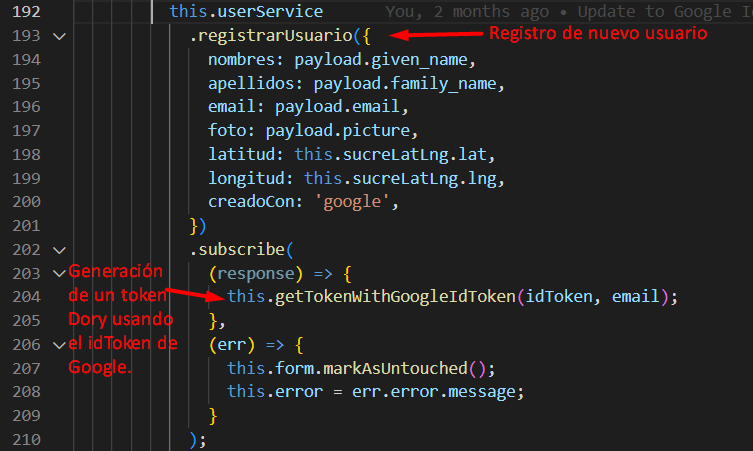
La solicitud de creación de un usuario al **API-REST** se realiza en el método regUserAuthGoogle(idToken) del componente registro.component.ts

Lo primero que hace el método es comprobar en la línea 185 si el usuario existe en la base de datos ***process.env.DB\_NAME***

this.userService.getUsuarioByEmail(email).subscribe(



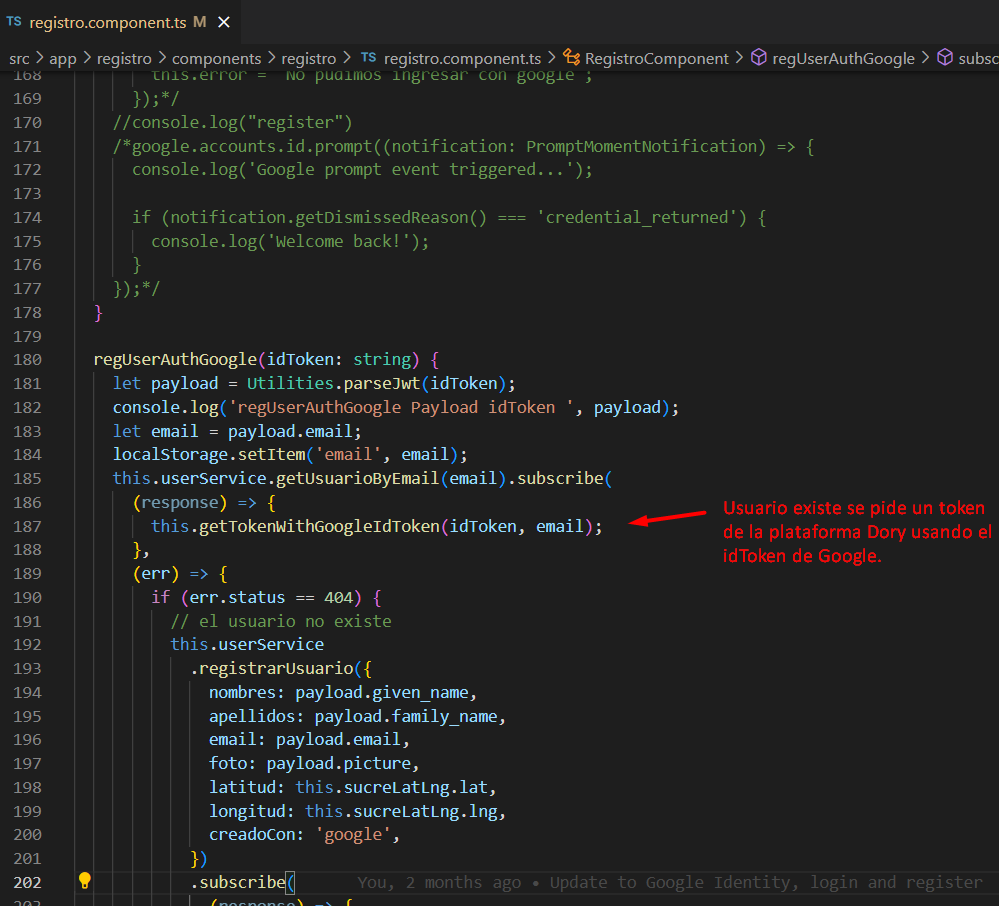
Si no existe se registra en la línea 192 usando el servicio this.userService.registrarUsuario



Luego de que el registro se lleve con éxito se le solicita al **API-REST** en la línea 204 obtener un token de la plataforma Dory. El cual servirá para consumir los demás servicios.

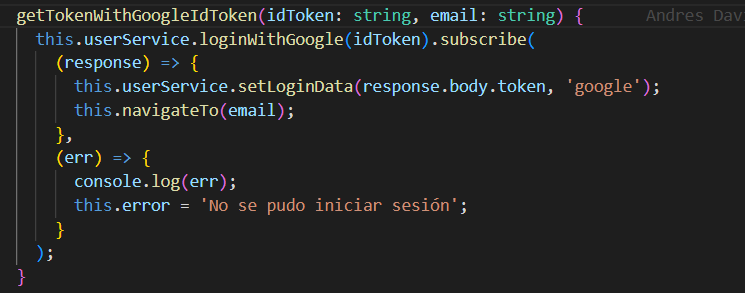
 this.getTokenWithGoogleIdToken(idToken, email);

En caso de que el usuario exista se pide un token de la plataforma directamente en la línea de código 187.

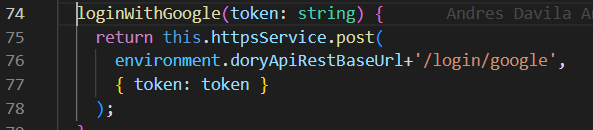


En más detalle el método getTokenWithGoogleIdToken hace lo siguiente:

Consume el servicio this.userService.loginWithGoogle(idToken) y le envía por parámetro el idToken



El servicio this.userService.loginWithGoogle solicita a su vez al endpoint /api/login/google usando el verbo POST y enviándole en el body el token.



g. El usuario es redirigido a su panel de administración en la aplicación web Dory.

Cuando estos procesos terminan exitosamente se realiza la siguiente redirección, enviado al welcome o al dashboard dependiendo de si los datos del usuario están completos o no.

