**Descripción conceptual 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 ingresados en el formulario de registro o los 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.

El nombre ***process.env.DB\_NAME*** hace referencia a una variable de entorno que almacena el nombre de la base de datos y se puede encontrar en el archivo ***/.env*** en el código de proyecto en un ambiente de desarrollo y en la configuración de la aplicación en Heroku en un ambiente de pruebas y de producción. Para más información sobre variables de entorno diríjase a las recetas “Receta - Explicación configuración de las variables de entorno para almacenar la configuración de la aplicación en el ambiente de desarrollo” y “Receta - Explicación configuración de las variables de entorno para ambiente de prueba o producción en Heroku”.

La cuenta queda verificada cuando en el campo ***estaVerificado*** de la tabla ***usuarios*** (Imagen 1. Tabla ususarios de la base de datos de la plataforma Dory.) de la base de datos ***process.env.DB\_NAME*** se almacena el entero 1.

En esta receta se presentan los dos métodos de registros implementados en la plataforma Dory:

* En el primer método se recibe la información del usuario a través de un formulario y se envía una solicitud al endpoint /api/usuario/create **API-REST** de la plataforma Dory para crear un nuevo registro.
* En el segundo método el usuario se autentica con su cuenta de Google. Se extrae la información básica de la misma y se envía una solicitud al endpoint /api/usuario/create **API-REST** de la plataforma Dory para crear un nuevo registro.

Los siguientes diagramas (imagen 1, imagen 2) muestran los procesos completos de registro usando formulario y autenticación con cuenta de Google. Los rectángulos azules en estos diagramas representan piezas de software que hacen parte de la plataforma Dory y los cuadros grises representan sistemas externos.

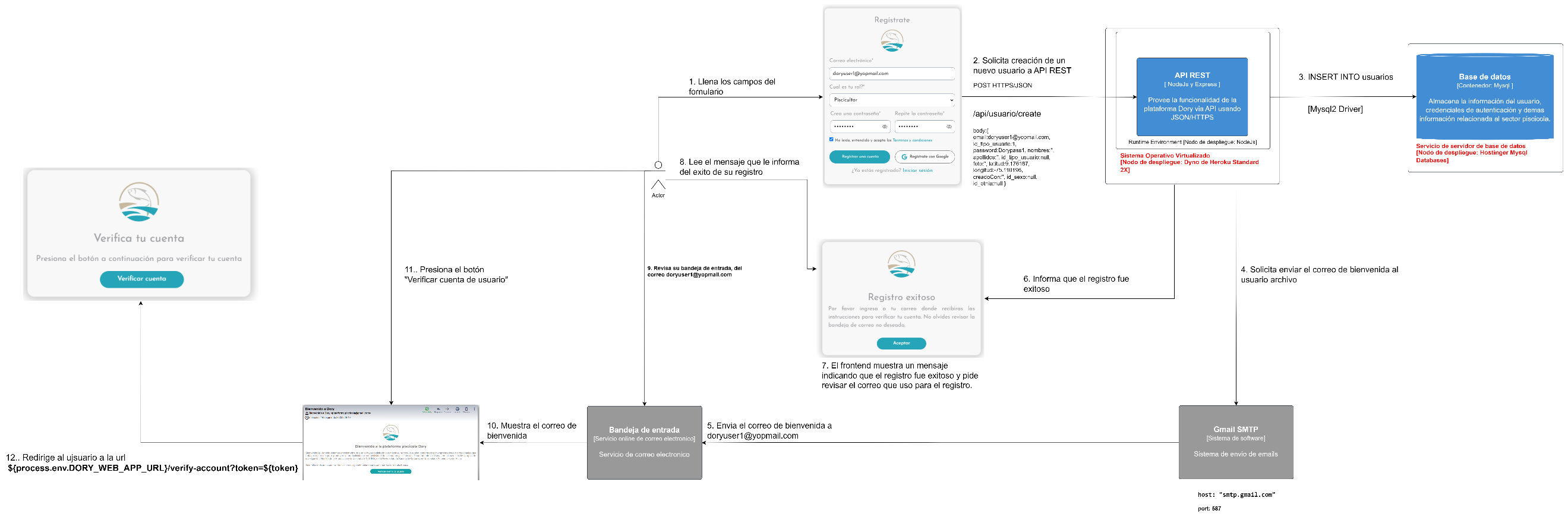


Imagen 1. Proceso completo de registro por formulario en la plataforma Dory

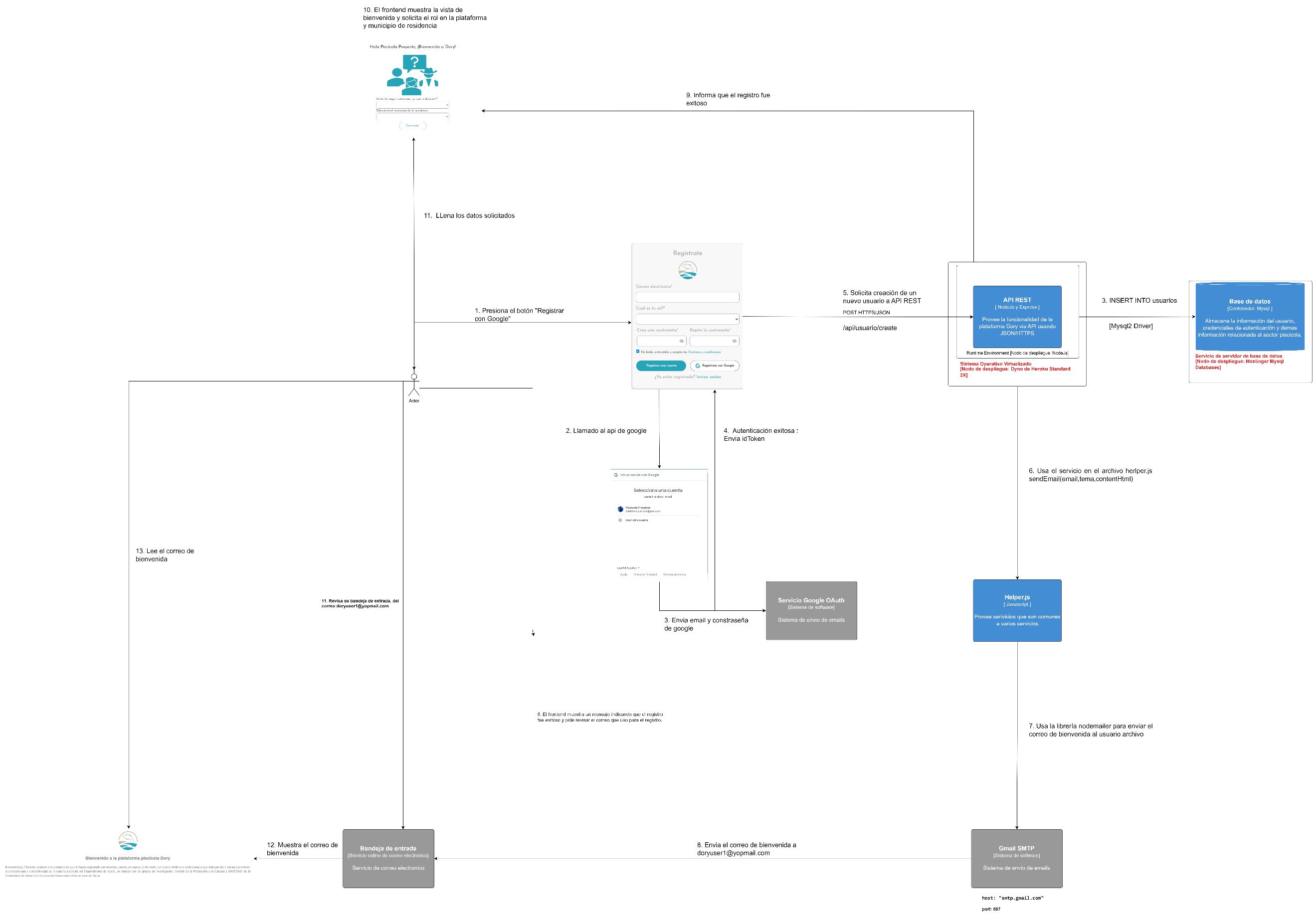


Imagen 2. Proceso de registro completo usando una cuenta de Google.

**Recursos necesarios:**

* Cliente Http
* Base de datos ***process.env.DB\_NAME*** en línea
* API-REST de Dory en línea
* Dory web app en línea
* Cuenta de Google(se usará para obtener el ID de Cliente OAuth 2.0)
* Ver la receta “Token JWT” para conocer sobre tokens JWT.

**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.

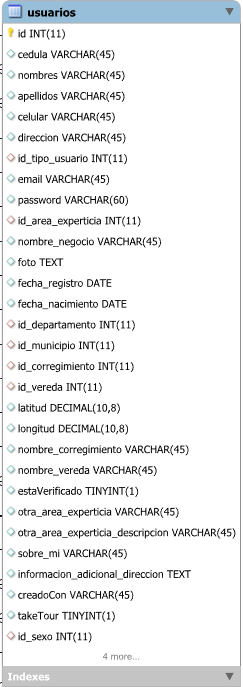


Imagen. Tabla ususarios de la base de datos de la plataforma Dory.

**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 4. 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.

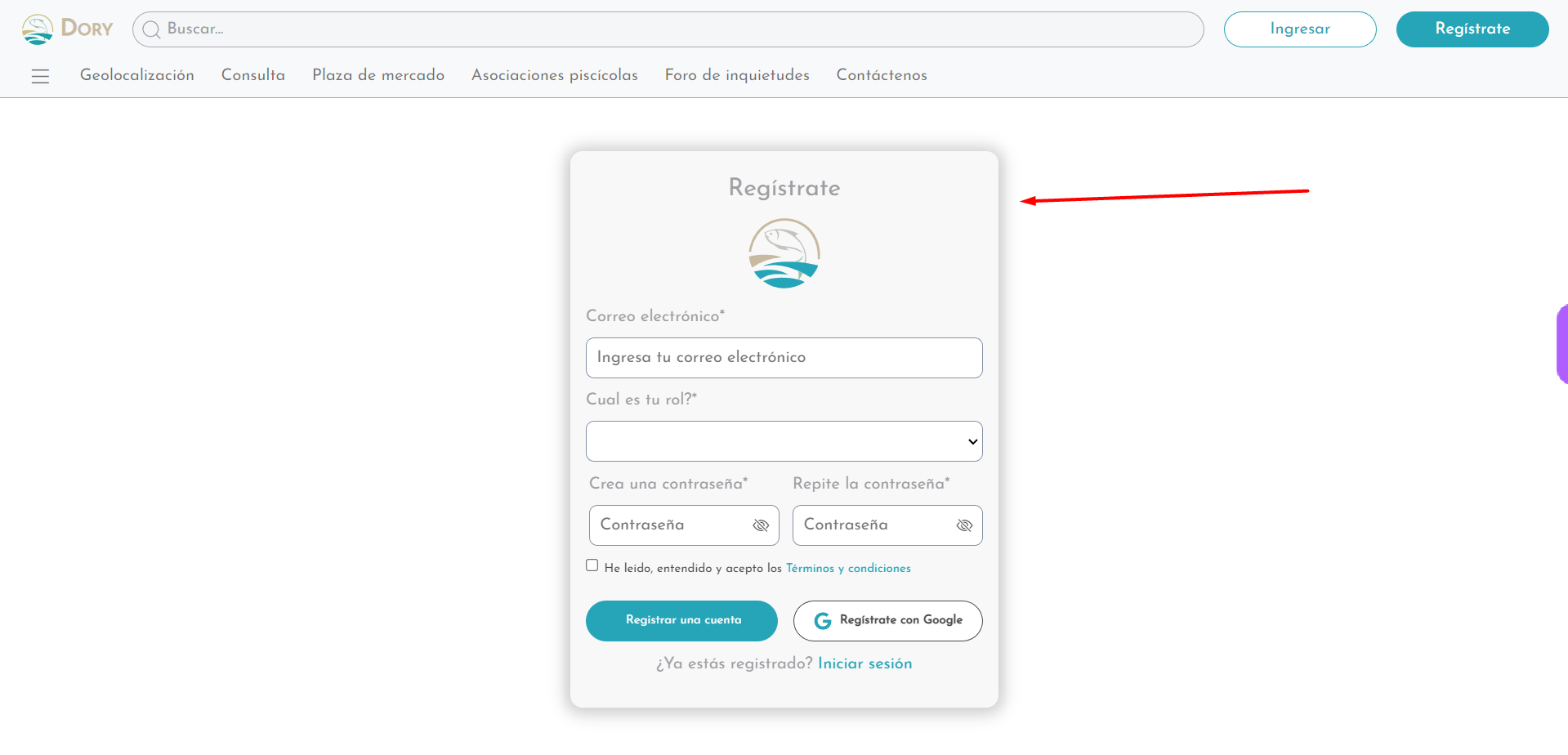


Imagen 5. Formulario de registro de la aplicación web Dory.

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



Imagen 6. 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. A continuación se muestra el proceso que ocurre después de enviar la solicitud de registro.

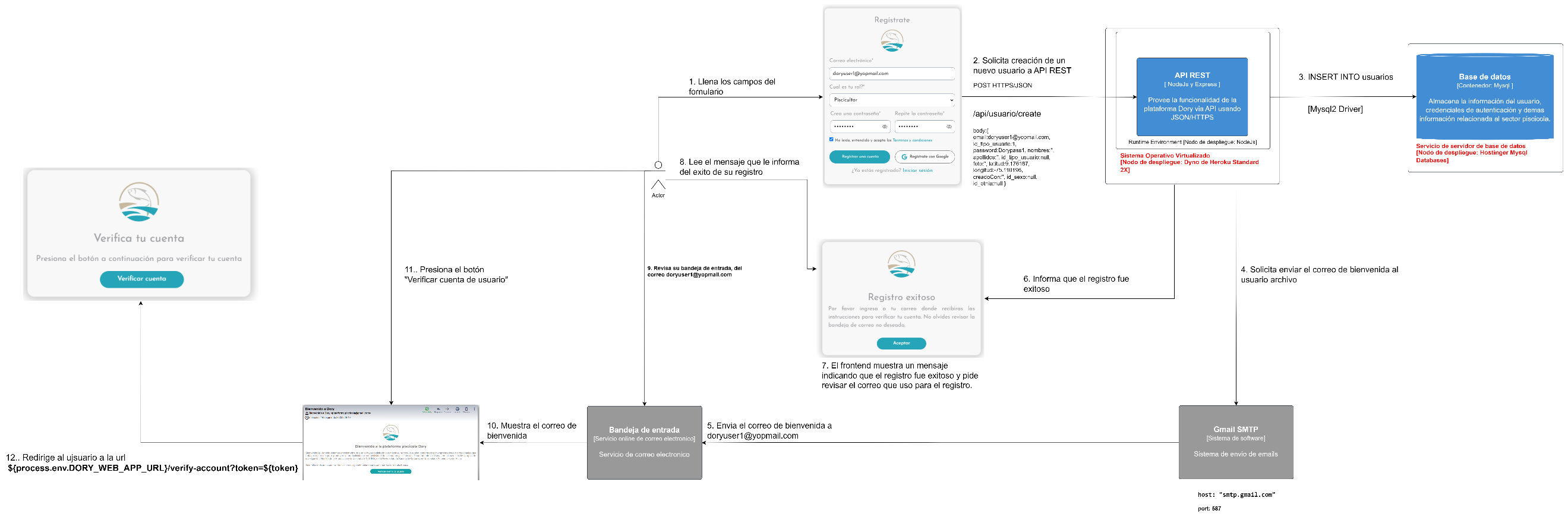


Imagen 6. Diagrama registro por formulario de un usuario en la plataforma Dory

A continuación, se explicará paso a paso de la imagen 6:

Primero veamos el paso 2. Solicita creación de un nuevo usuario al API-REST

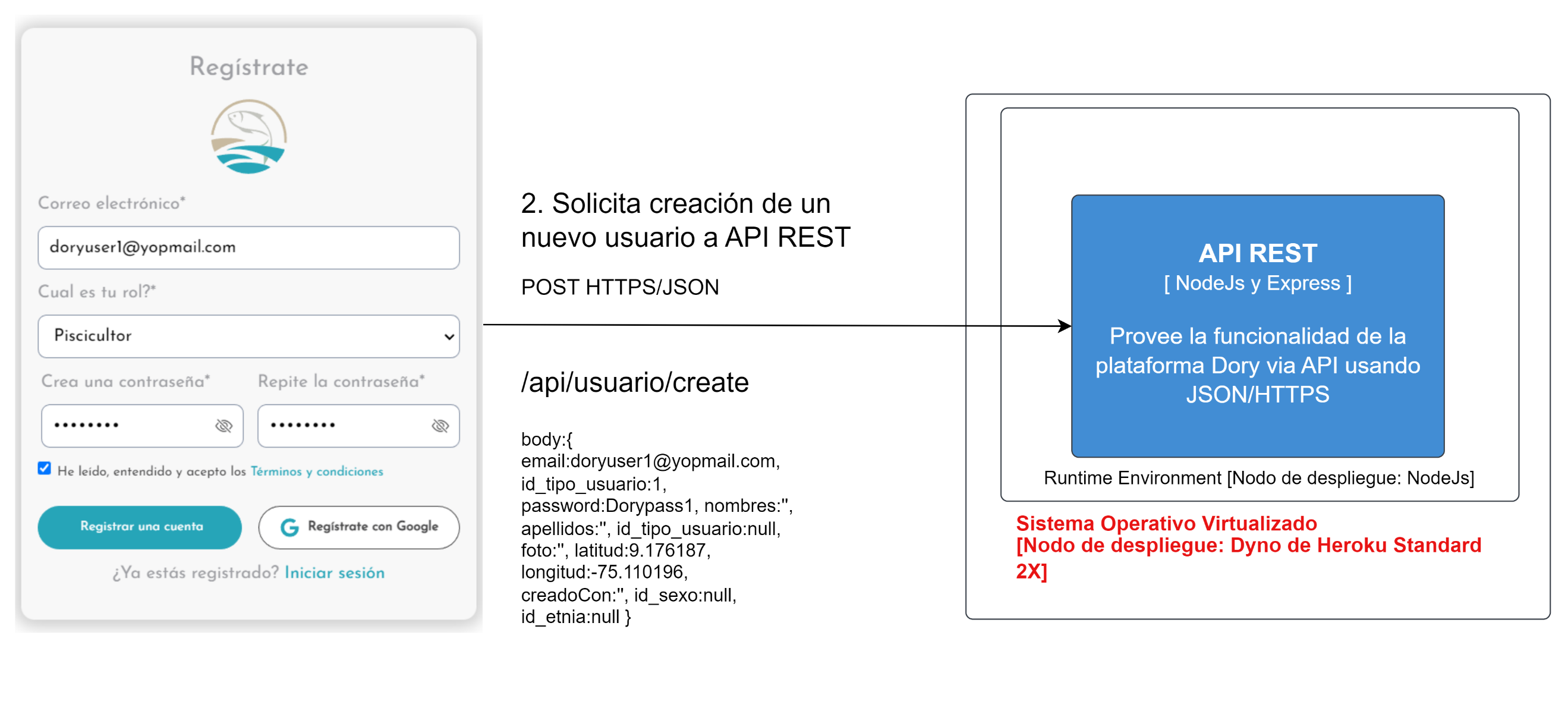


Imagen 8. Formulario solicitándole al API REST la creación de un nuevo usuario

En este paso se envía una solicitud HTTP desde el formulario de registro con datos en formato JSON usando el cliente HttpClient de Angular. El endpoint al que se envía la solicitud es **/api/usuario/create**. Se le envían los datos que el usuario ingreso en el formulario en el body de la solicitud.

body:{

email:doryuser1@yopmail.com,

id\_tipo\_usuario:1,

password:Dorypass1,

nombres:'',

apellidos:'',

id\_tipo\_usuario:null,

foto:'',

latitud:9.176187,

longitud:-75.110196,

creadoCon:'',

id\_sexo:null,

id\_etnia:null

}

Ahora, miremos que pasa cuando llega la solicitud al API REST



Imagen 9. API REST recibiendo la solicitud HTTP que viene del formulario

Ampliemos el cuadro del API REST de la imagen 9 para ver más de cerca lo que pasa dentro.

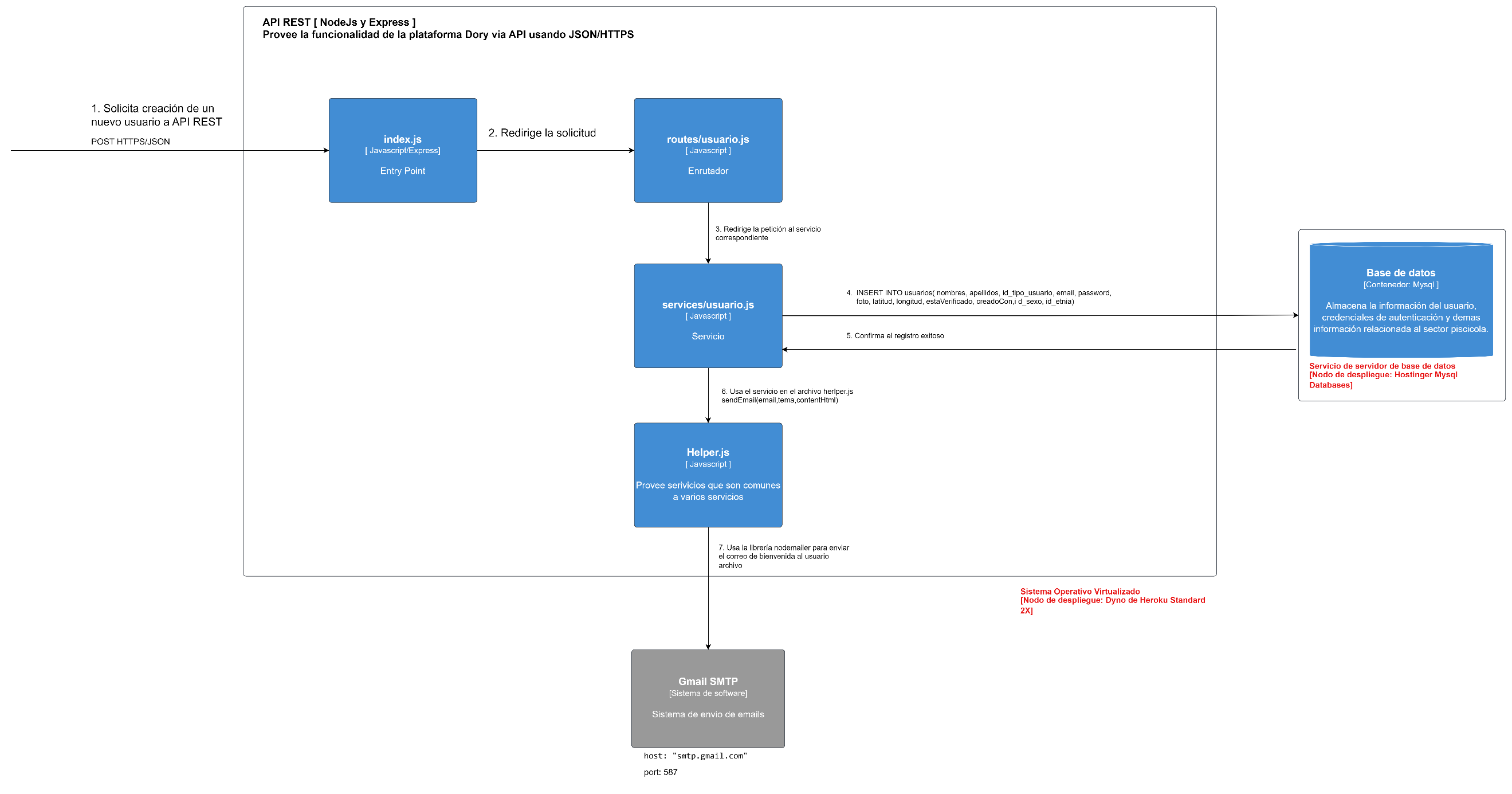


Imagen 10. Zoom del API REST

La solicitud es recibida en punto de entrada(index.js). Aquí a través de express se conoce cuál es el endpoint que el formulario está solicitando y se decide a que enrutador enviar la petición. Como el endpoint solicitado es POST /api/usuario/create y como este contiene el fragmento /api/usuario/ se decide enviar la petición a enrutador routes/usuario.js en el punto 2. Miremos lo que sucede dentro del enrutador:

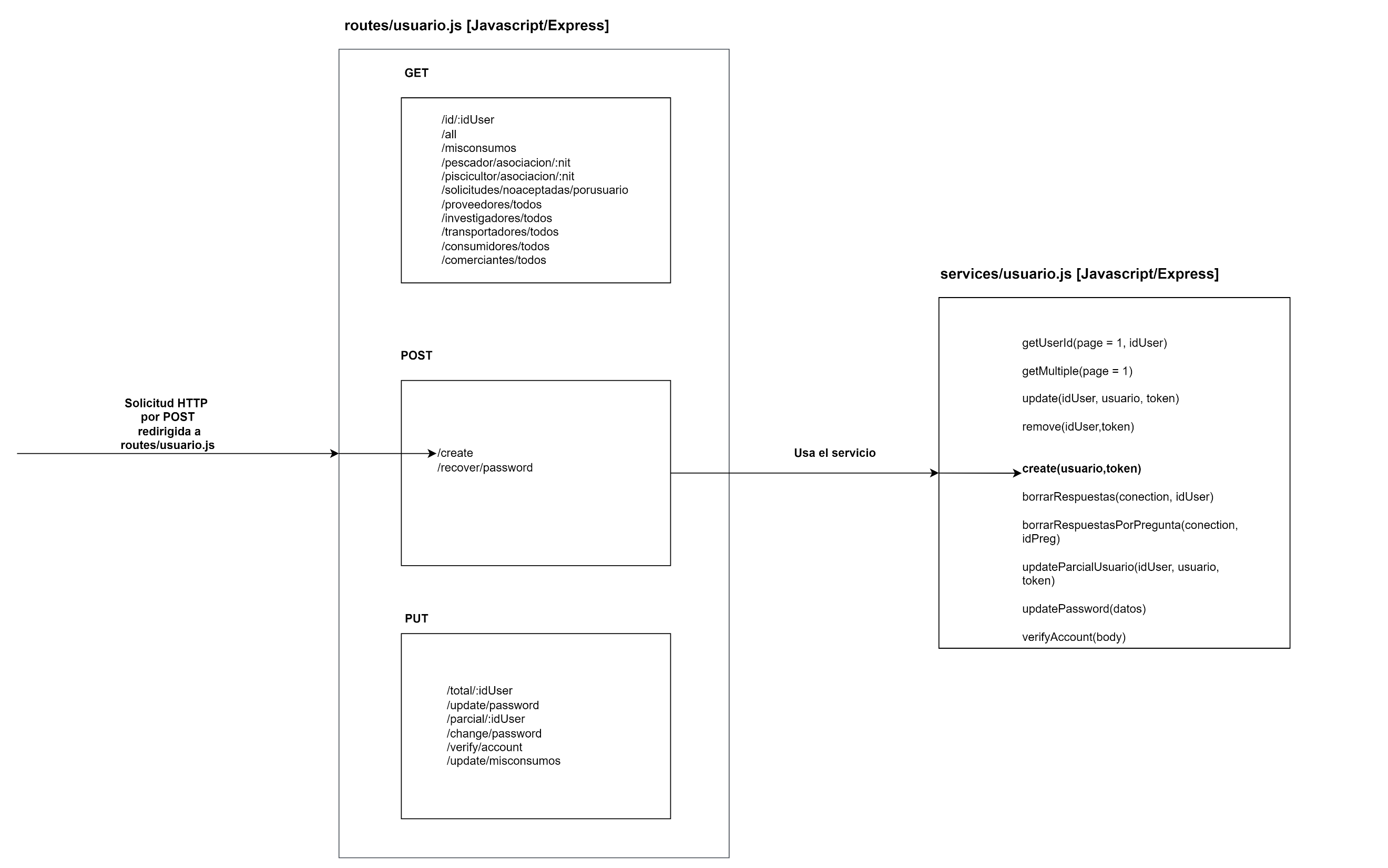


Imagen 11. Funcionamiento interno del enrutador routes/usuario.js

La función del enrutador es seleccionar el servicio que se encargara de resolver la petición. El enrutador contiene múltiples servicios los cuales están agrupados por verbos HTTP(GET, POST, PUT y DELETE). Como la solicitud se hizo usando el verbo POST el enrutador escogerá el servicio solo del grupo POST. Pero como solo debe escoger un solo servicio seleccionará el que este asociado con el endpoint que se está consumiendo. En este caso el escoge el servicio que esta asociado con el fragmento de ruta /create. Por lo tanto el enrutador extrae el body y lo envia al servicio services/usuario.js(punto 3) haciendo el llamado al método create del mismo.

El servicio services/usuario.js en el método create recibe los datos del usuario. Este verifica que todos los datos estén completos. Luego arma la siguiente consulta SQL y la envía usando el Driver Mysql2(https://www.npmjs.com/package/mysql2):

INSERT INTO usuarios(nombres,apellidos,id\_tipo\_usuario,email,password,foto,latitud,longitud,estaVerificado,creadoCon,id\_sexo,id\_etnia) VALUES (?,?,?,?,?,?,?,?,?,?,?,?)

Esta consulta permite insertar los datos ingresados por el usuario en el formulario de registro en la tabla usuarios.

La base de datos responde con un mensaje de éxito cuando realiza la inserción del registro en la tabla usuarios (punto 5).

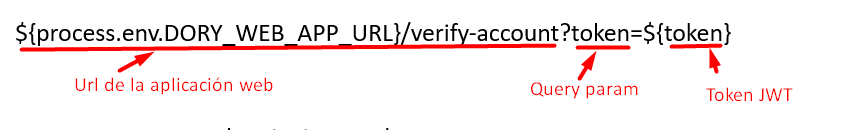
Luego cuando la inserción del registro es exitosa se arma un correo de bienvenida para el usuario y se le adiciona una URL para verificación de la cuenta.



Esta es la URL que se envía en el correo de bienvenida:

${process.env.DORY\_WEB\_APP\_URL}/verify-account?token=${token}

Esta compuesta por los siguientes elementos



Process.env.DORY\_WEB\_APP\_URL: Esta es una variable de entorno donde se encuentra almacenada la url de la aplicación web Dory.

/verify-account: Es el complemento de la url de la aplicación web Dory

token: Es un query param. Es decir, un parámetro que enviaremos por la url a la aplicación y cuyo valor es un token JWT.

Cuando el contenido del correo está preparado es enviado por el servicio services/usuario.js utilizando el servicio sendEmail(email,tema,contentHtml) del modulo herper.js como se muestra en la imagen 10 punto 7. Como se puede observar se le envía por parámetros el email receptor, el tema o asunto y el contenido del mensaje en formato html.

Además de estos datos al servicio sendEmail se le a configurado por defecto los siguientes parámetros:

{

host: "smtp.gmail.com",

port: 587,

secure: false,

auth: {

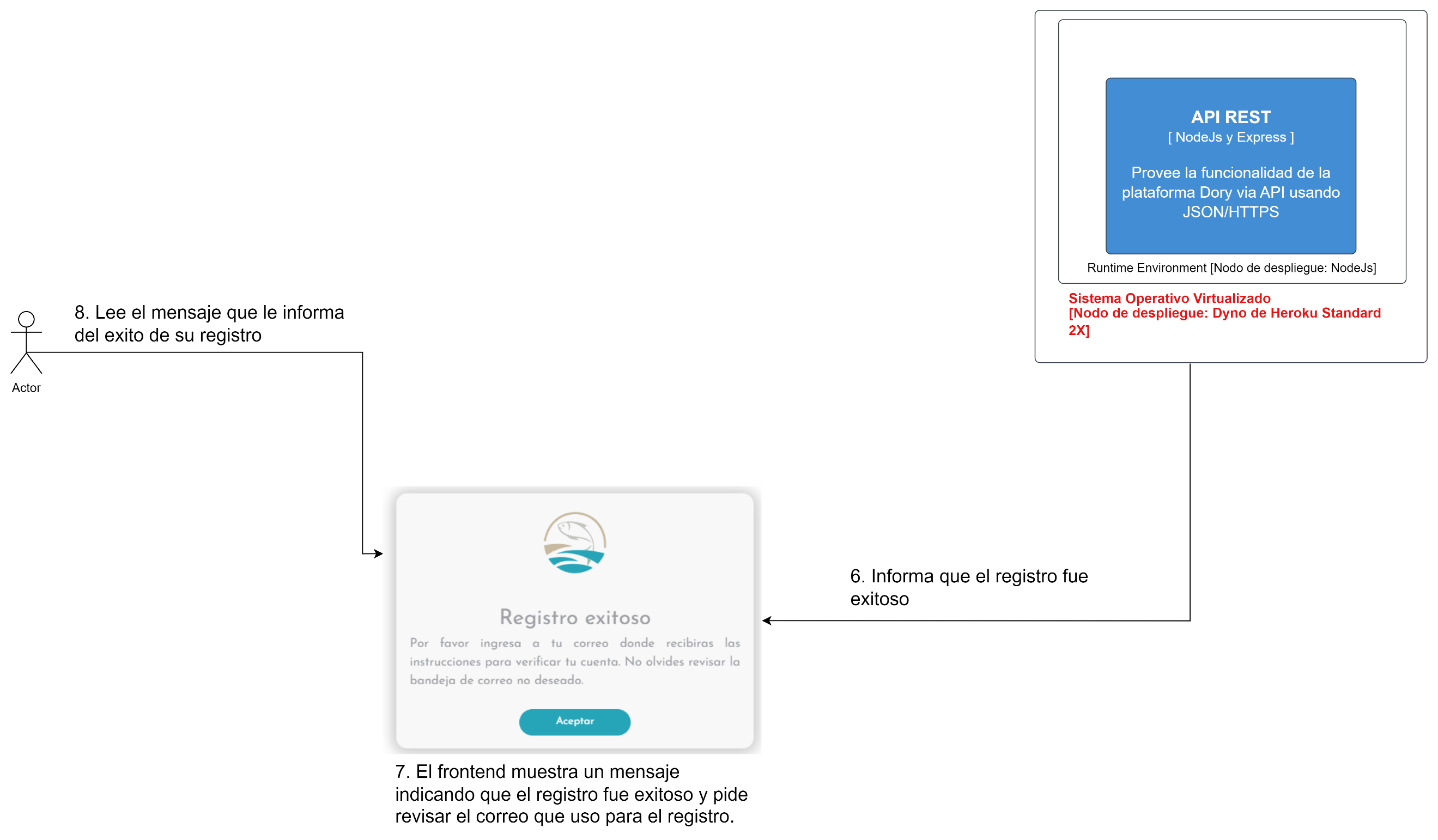
user: "plataforma.piscicola@gmail.com", //dirección desde donde se enviara el correo

pass: process.env.GOOGLE\_PASSWORD\_APP, //contraseña de esta cuenta

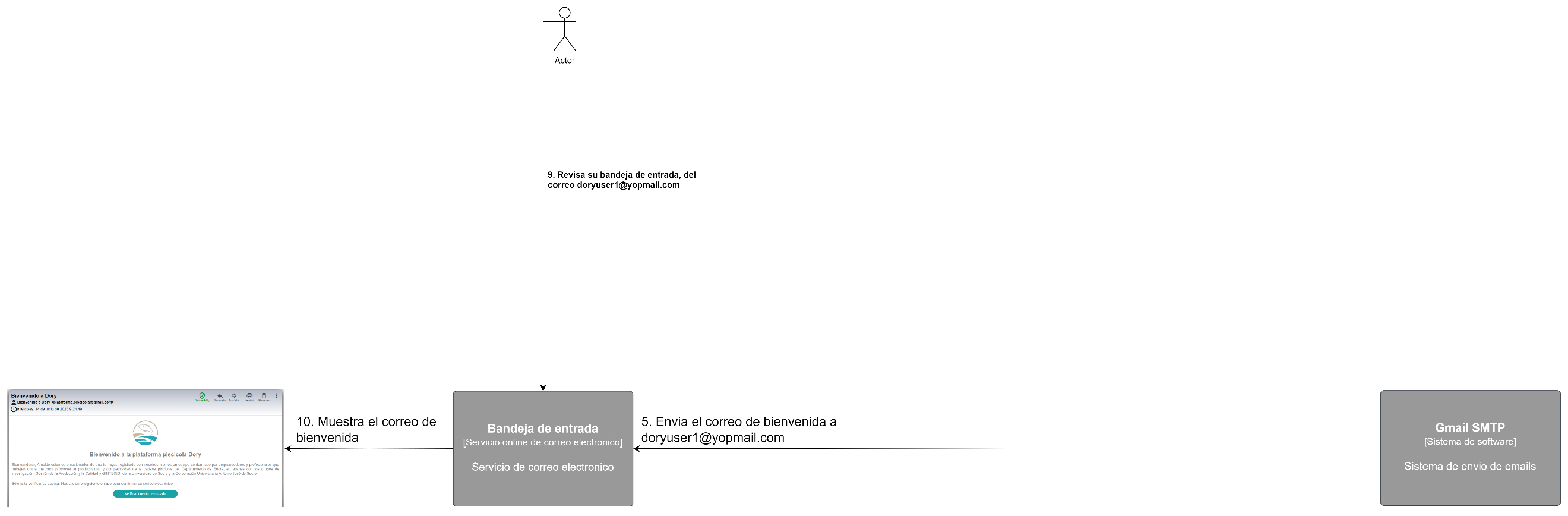
}

El servicio sendEmail utiliza la librería nodemailer y usando los datos anteriores solicita el envío del correo al servidor smtp.gmail.com en el puerto 587 como se indica en los campos host y port respectivamente.

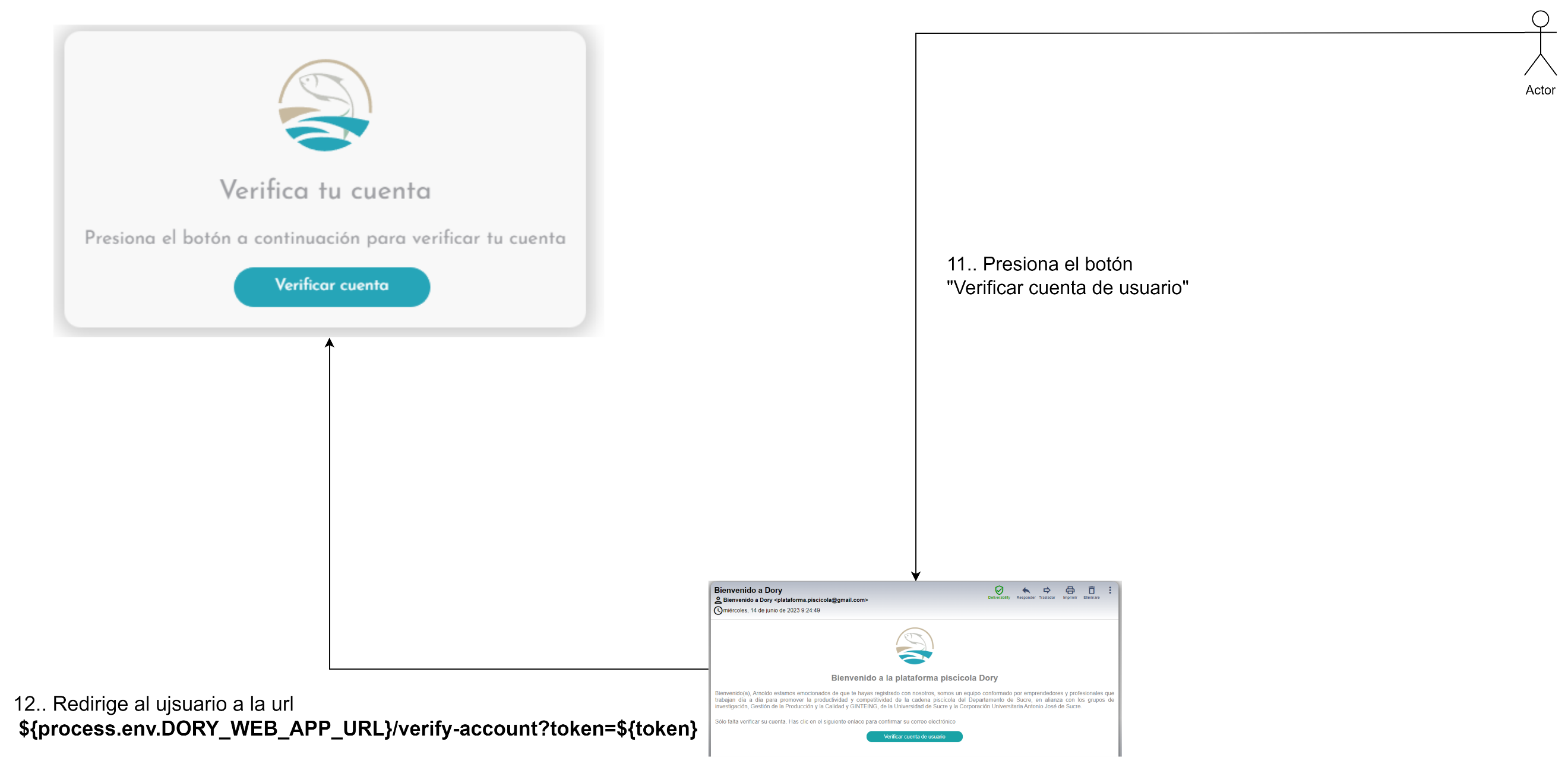
e. El usuario API-REST informa que el registro fue exitoso.



f. El usuario ingresa a su bandeja de entrada y revisa el correo de bienvenida



f. El usuario presiona el botón “Verificar cuenta de usuario”

****

Al presionar el botón “Verificar cuenta de usuario” se le abre la dirección ${process.env.DORY\_WEB\_APP\_URL}/verify-account?token=${token} y esta dirige a la vista del punto 12 que lo invita a verificar la cuenta.

Cuando el usuario presiona el botón “Verificar cuenta” se extrae el token que viene en el query param token en ?token=${token}. Luego se le solicita al endpoint /api/usuario/verify/account y enviándole el token JWT en el body de la petición:

Body: {

token: token

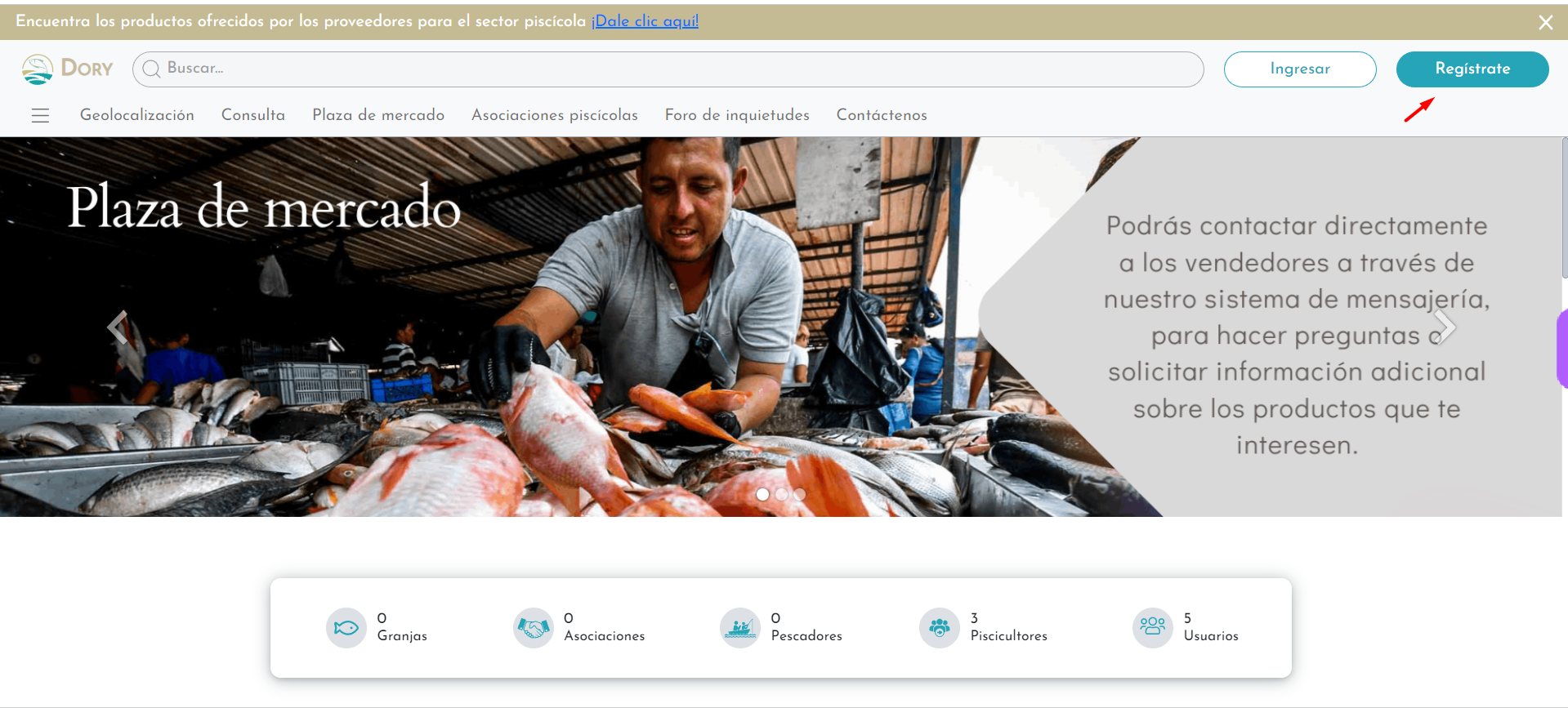
}

Lo que hace el endpoint es validar que el token sea valido y cambiar el estado del campo ***estaVerificado*** de la tabla ***usuarios*** *a 1.* De esta manera queda completado el proceso.

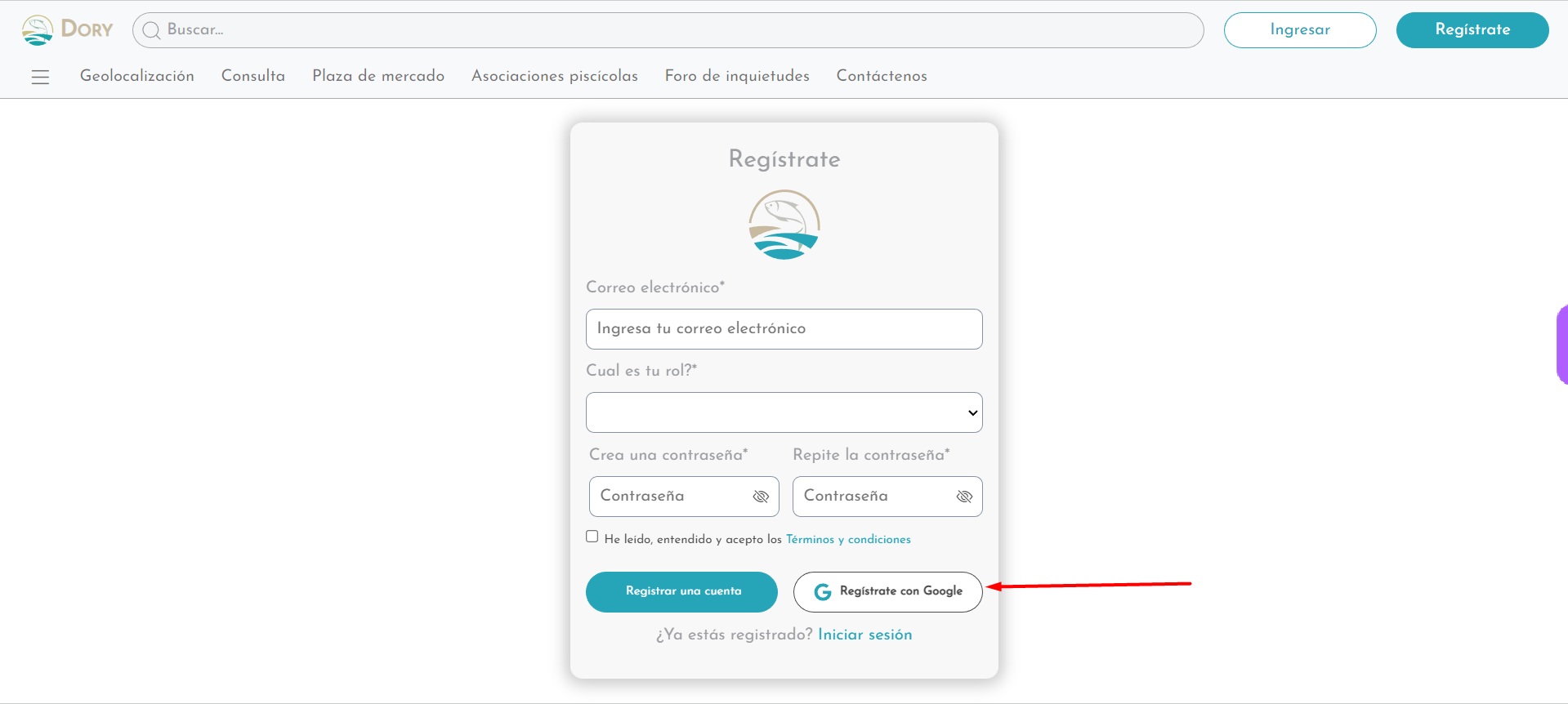
**2. Conocer el método de registro usado por google**

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:

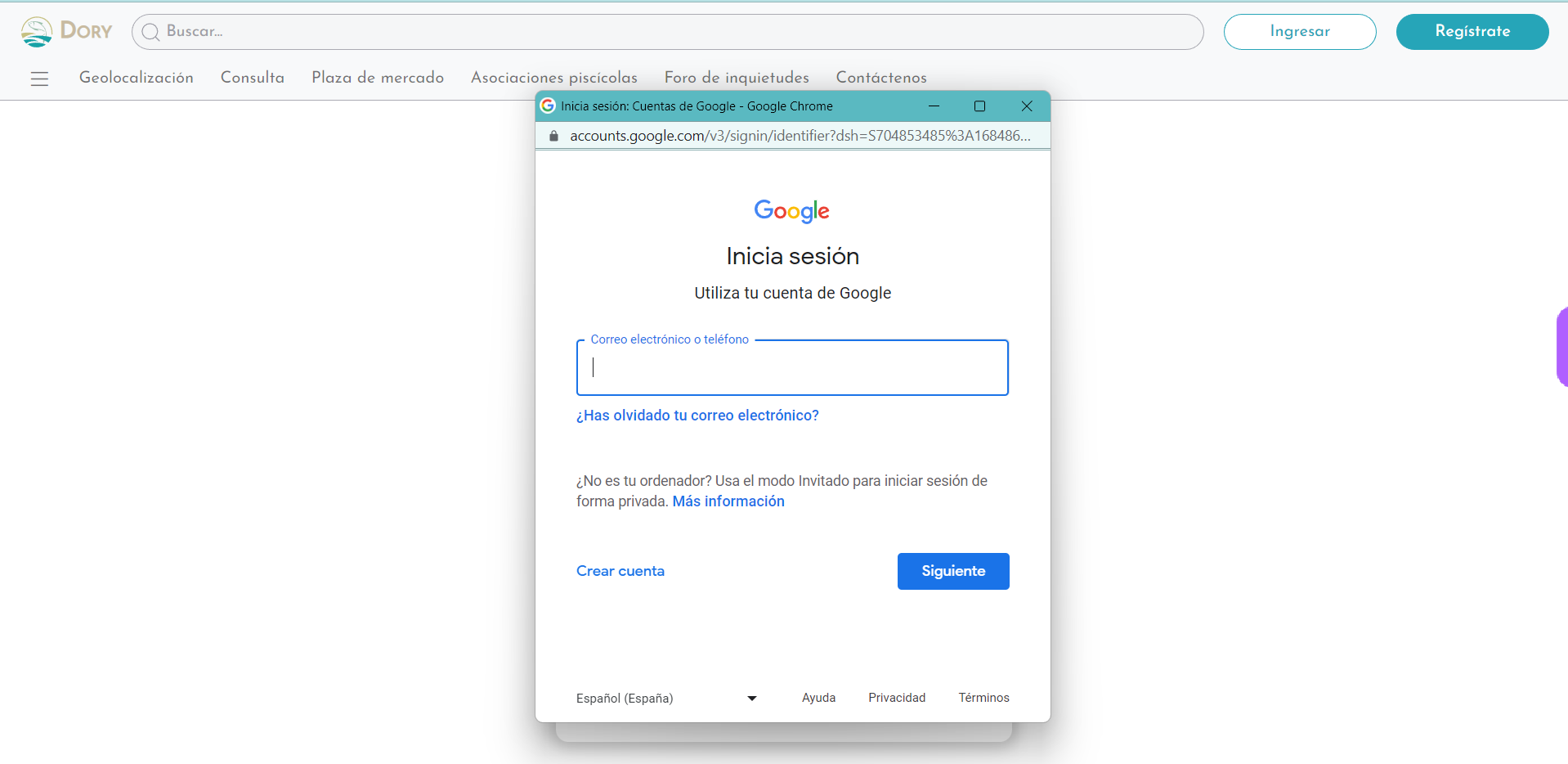
a. El usuario accede a la opción de registro con Google en la aplicación.



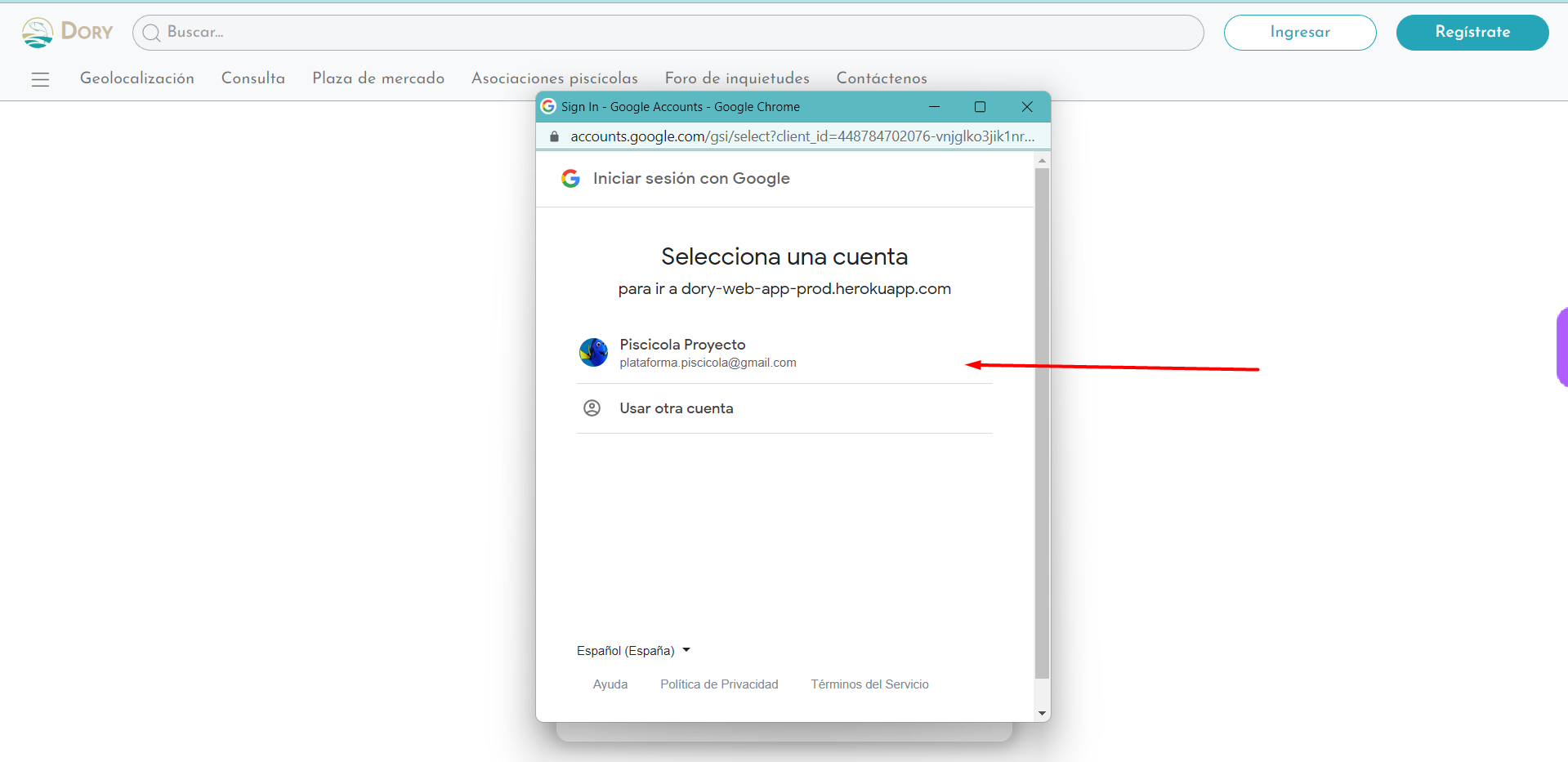
b. Se muestra un botón o enlace que permite al usuario iniciar sesión con su cuenta de Google.



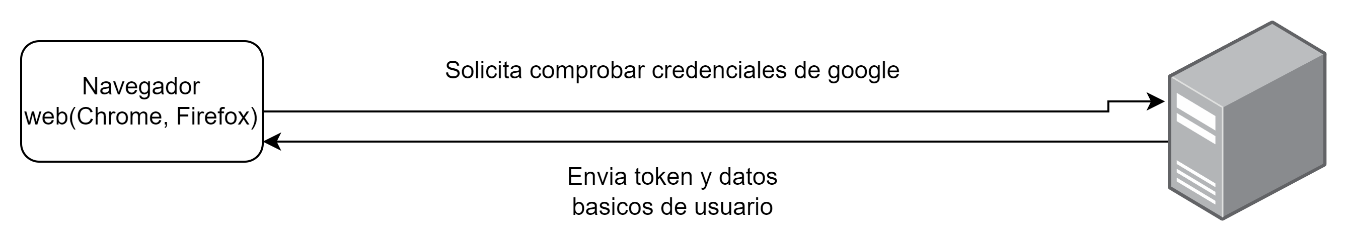
c. Al hacer clic en el botón o enlace, se abre una ventana emergente o se redirige al usuario a la página de inicio de sesión de Google



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 parametro response en el campo credential(response.credential).

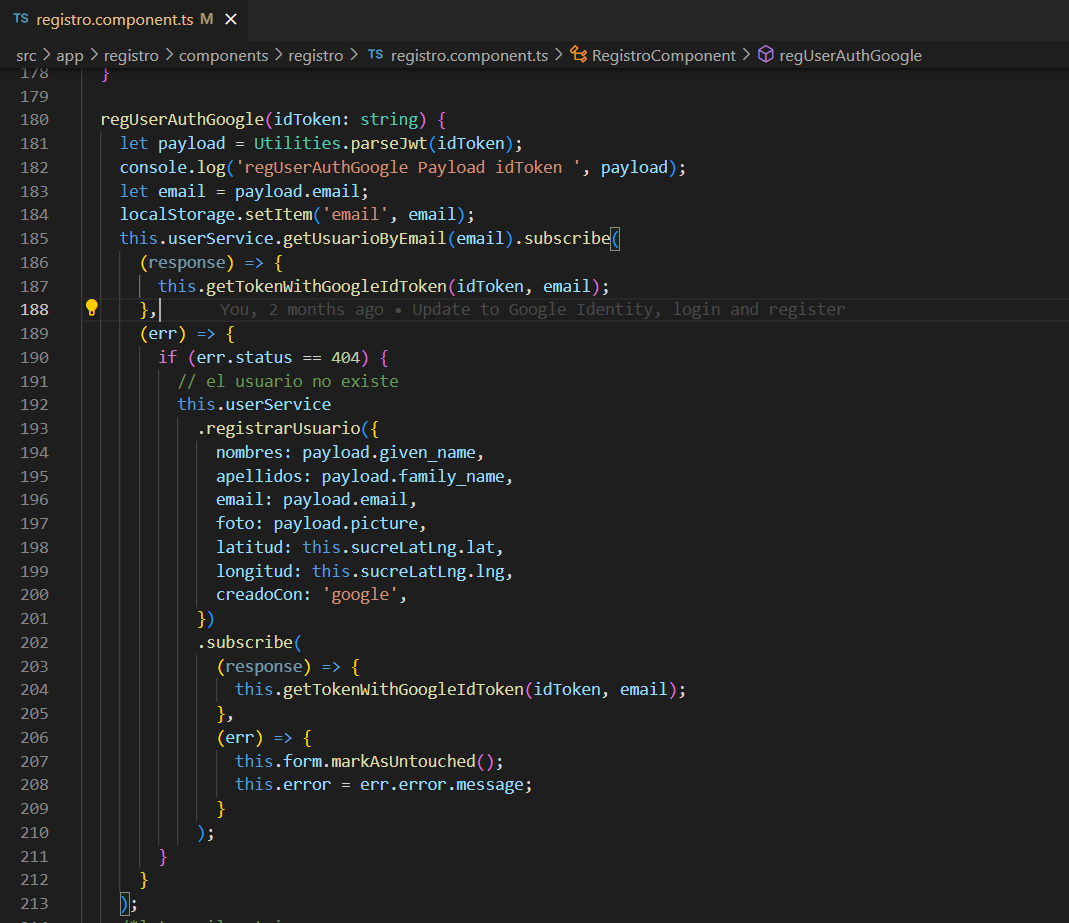
f. Si el usuario es autenticado correctamente, se le solicita al **API-REST** crear una cuenta de usuario en la base de datos ***process.env.DB\_NAME*** con los datos proporcionados por Google.



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(



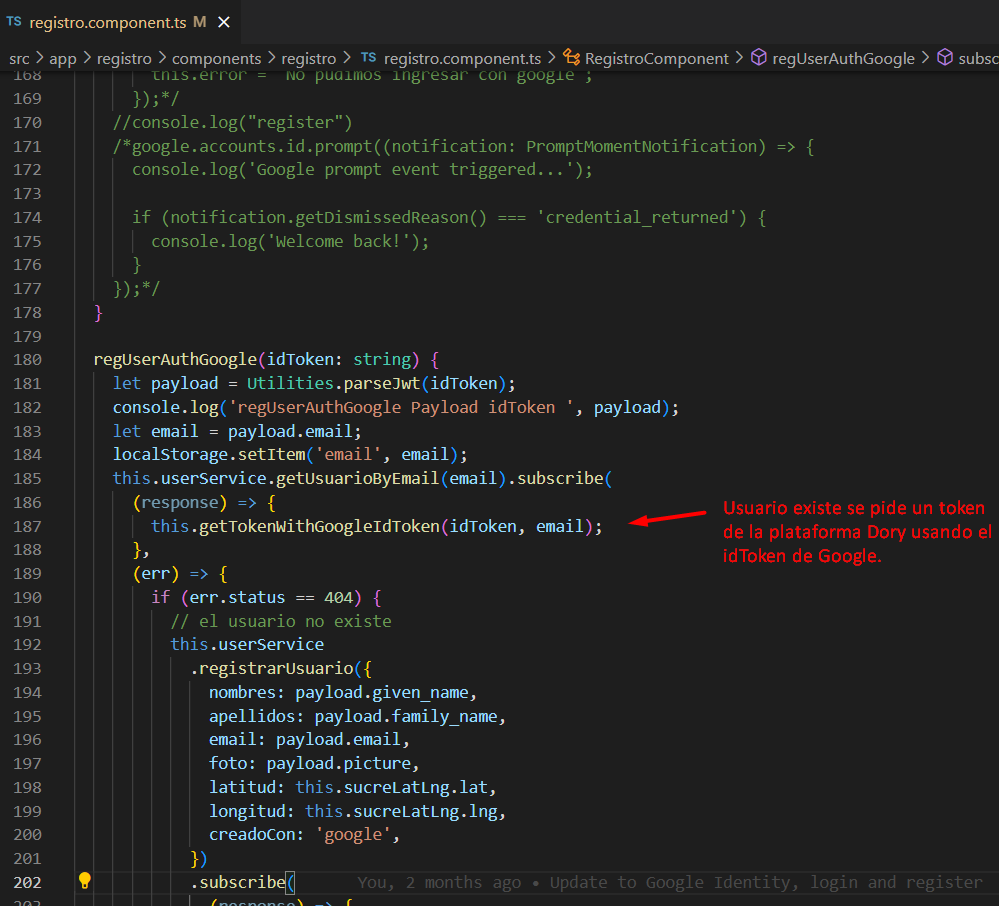
Si no existe se registra en la línea 192 usando el servicio this.userService.registrarUsuario



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.

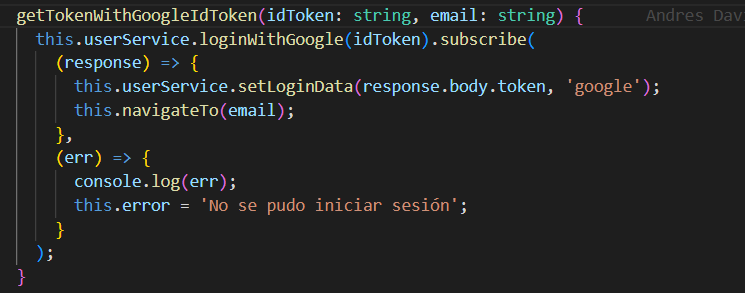
 this.getTokenWithGoogleIdToken(idToken, email);

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

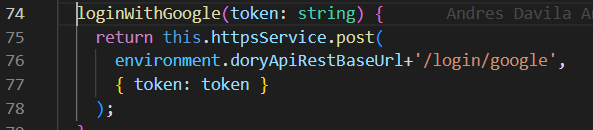


En mas detalle el método getTokenWithGoogleIdToken hace lo siguiente:

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.

