**Como implementar el servicio de registro en el api rest de la aplicación web Dory**

**Objetivos:**

Implementar el servicio de registro de la plataforma web Dory.

Diagrama

Descripción generada automáticamente**Recursos necesarios:**

* Node.js, Visual Studio Code, Express.
* Cuenta de servicio SMTP de Google.
* API REST para la comunicación con la plataforma web y la base de datos.
* Base de datos para almacenar los datos de registro de los usuarios.
* Cliente de correo electrónico para recibir el mensaje de bienvenida.

Piezas de software necesarias:

* Librería de encriptación de contraseñas (bcrypt).
* Librería para utilizar token (JWT- simple)
* Librería para manejo de errores (http-errors)
* Librería para envió de mensajes a correo electrónicos (nodemailer)

**Pasos:**

1. Configurar el entorno de desarrollo. Ver (Como preparar el entorno de desarrollo).
2. En el archivo principal “index.js” se debe tener la configuración del servidor de Express (ver imagen 1).

Texto

Descripción generada automáticamente

Imagen 1. *Configuración del servidor Express*

1. *Se debe tener implementado la ruta para registrar un usuario en el archivo principal “index.js” (ver imagen 2).*

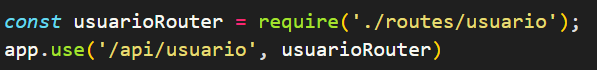


Imagen 2. *Implementación de la ruta para el registro de usuario en el “index.js”*

1. Se debe tener definido la función “createToken” para generar un nuevo token de aplicación en el archivo “helper.js” (ver imagen 3).

Texto

Descripción generada automáticamente

*Imagen 3. Función “createToken” en archivo “helper.js”*

1. Se debe tener definido la función “sendEmail” para el envió de mensajes al correo electrónico del usuario en el archivo “helper.js” (ver imagen 3).

Texto

Descripción generada automáticamente

*Imagen 4. Función “sendEmail” en archivo “helper.js”*

1. Se debe tener el archivo “auth.js” creado en el directorio “middelware” ubicado en la raíz principal de proyecto.
2. En el archivo “auth.js” debe estar implementado la función para validar el token de usuario y ser exportada para ser usado en otros archivos (ver imagen 5).

Texto

Descripción generada automáticamente

Imagen 5. *Función validarToken en el archivo “auth.js”*

1. En el directorio “routes” ubicado en la raíz principal del proyecto debe estar creado el archivo de rutas “usuario.js”.
2. En el archivo “usuario.js” debe estar creado el endpoint para registrar un usuario y debe tener el llamado al servicio para realizar dicho proceso (ver imagen 6).

Texto

Descripción generada automáticamente

Imagen 6. *Endpoint para registrar un usuario en el archivo de rutas “usuario.js”*

1. En el directorio “Services” ubicado en la raíz principal del proyecto, debe estar el archivo de servicios de usuarios llamado “usuario.js”.
2. En el archivo de servicios “usuario.js” se deben importar las librerías y archivos necesarios para la implementación del servicio de registro de usuario (ver imagen 7).

Texto

Descripción generada automáticamente

Imagen 7. *Importación de librerías y archivos en el servicio de usuarios “usuario.js”*

1. El servicio para registrar al usuario recibe el token y los datos del usuario (ver imagen 8).

* Se verifica la autenticidad y validez de los datos enviados por el usuario.
* Se procesa la solicitud y se extrae los datos del registro del usuario.
* Se comprueba si el token pertenece a un administrador.

Texto

Descripción generada automáticamenteImagen 8. *Verificación de token y datos del usuario*

1. Se realiza el contenido del mensaje que se enviará al correo electrónico del usuario para terminar su registro (ver imagen 9).

Texto

Descripción generada automáticamente

Imagen 9. *Contenido html del mensaje enviado al correo electrónico del usuario.*

1. Se encripta la contraseña del usuario (ver imagen 10).

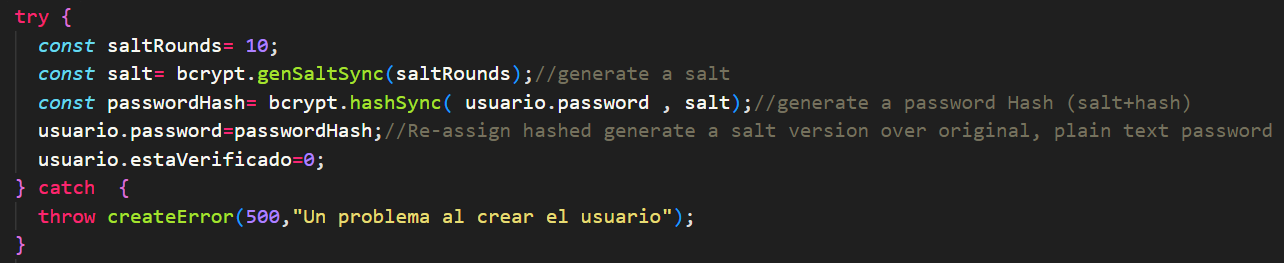
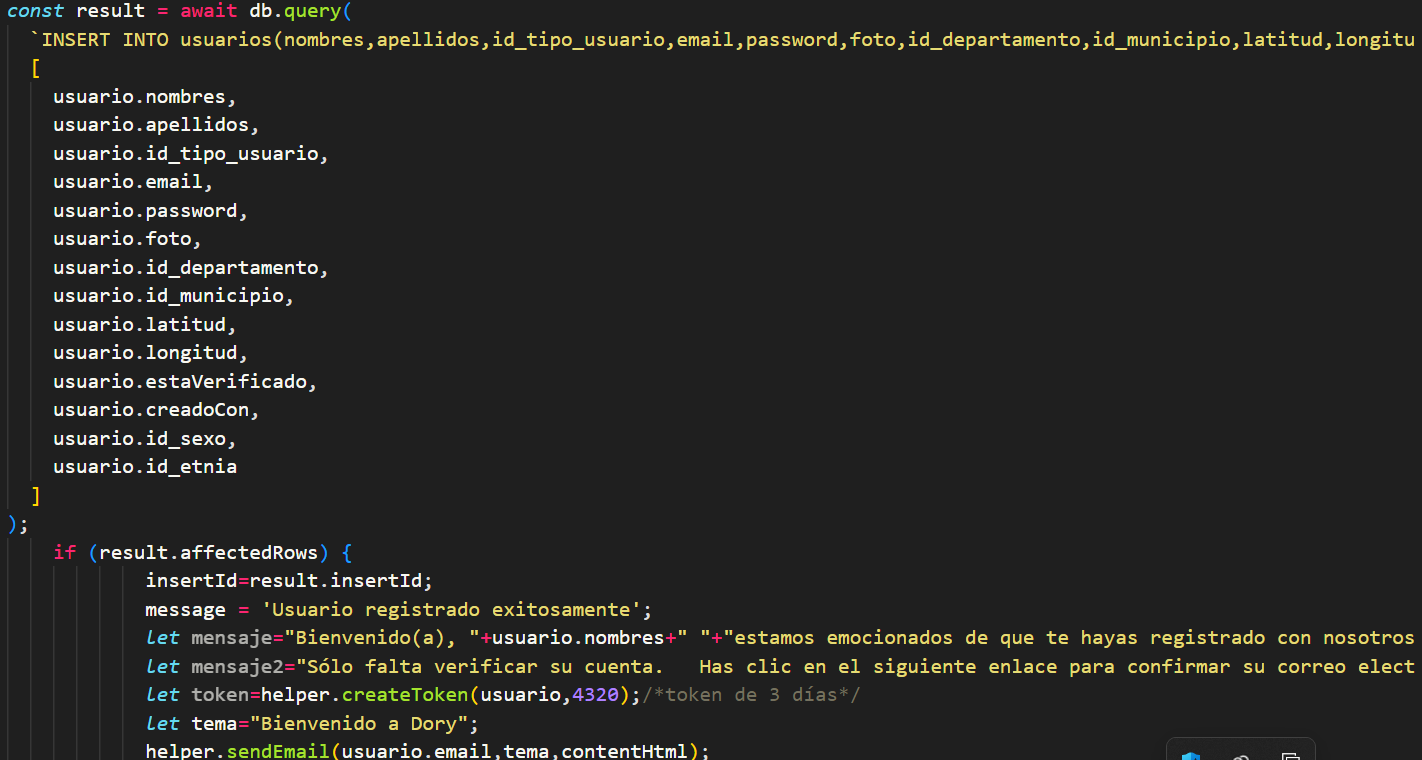


Imagen 10. *Encriptación de la contraseña del usuario*

1. Se realiza proceso del registro de los datos del usuario en la base de datos y se envía el mensaje de registro exitoso al correo electrónico (ver imagen 11); en caso contrario se envía un mensaje de error (ver imagen 12).

**

*Imagen 11. Registro de usuario y envió de mensaje al correo electrónico*



Imagen 12. *Mensaje de error del registro*

1. El usuario recibe el mensaje de bienvenida en su correo electrónico (ver imagen 13).Interfaz de usuario gráfica, Texto

   Descripción generada automáticamente

Imagen 13. *Mensaje de bienvenida recibido en el email de usuario*