

# Universidad Don Bosco



## Materia:

Diseño y programación de software multiplataforma

## Alumnos:

Oscar Javier Portillo Barahona PB161930  
Cesar Enrique Regalado Villalta RV210723  
Javier Enrique Hernandez Marquez HM210444  
Juan Jose Fonseca Guandique FG203030

## Profesor:

Alexander Alberto Siguenza

## Tarea:

Primer Proyecto React

Sistema de gestión de proyectos

## **Introduccion:**

Este documento presenta la documentación de un sistema de gestión de proyectos desarrollado con Next.js, que ofrece una experiencia de usuario fluida y segura. El sistema permite a los usuarios registrarse, iniciar sesión y acceder a rutas protegidas, garantizando un entorno controlado y adaptado a sus necesidades.

La autenticación de usuarios se realiza a través de JSON Web Tokens (JWT), proporcionando una forma segura de manejar sesiones y credenciales. Además, se ha implementado el manejo de datos mediante el módulo 'fs' de Node.js, que permite gestionar la información de los usuarios en archivos, asegurando así un almacenamiento eficiente y accesible.

Este sistema no solo facilita la gestión de proyectos, sino que también prioriza la seguridad y la privacidad de los datos, aspectos esenciales en cualquier aplicación moderna. A lo largo de este documento, se detallarán las características del sistema, su arquitectura, y se proporcionarán guías para la instalación y uso, con el objetivo de ofrecer una referencia completa para desarrolladores y usuarios interesados.

## Desarrollo.

1. Registro de usuarios: Permite a los usuarios registrarse con su correo electrónico y contraseña.
2. Inicio de sesión de usuario: Autentica a los usuarios verificando sus credenciales y devuelve un token JWT.
3. Rutas protegidas: Las rutas como el cuadro de mandos están protegidas y sólo son accesibles para usuarios autenticados.
4. Autenticación basada en JWT: El sistema genera un token JWT al iniciar sesión con éxito, que se almacena en el lado del cliente y se utiliza con fines de autenticación.

Interfaz grafica y registro de usuario.



Usuaio registrado.

---

hm210444@alumno.udb.ec	.....	.....	Register
------------------------	-------	-------	----------

Login de usuario.

semita@dps.com	.....	Login
----------------	-------	-------

## Estructura del proyecto.

El proyecto sigue una estructura de carpetas estándar de Next.js, optimizada para facilitar la gestión de la autenticación y las funciones del sistema. A continuación se detalla la organización de los archivos y carpetas más relevantes:

- **pages/api/register.js**: Este archivo gestiona el proceso de registro de usuarios, recibiendo el correo electrónico y la contraseña. Utiliza bcrypt para hacer hash de la contraseña antes de almacenarla en un archivo JSON, garantizando así la seguridad de la información.
- **pages/api/login.js**: Encargado del inicio de sesión, este archivo verifica las credenciales proporcionadas por el usuario. Si las credenciales son válidas, genera y devuelve un token JWT que permite la autenticación en el sistema.
- **pages/index.jsx**: Representa la página de inicio del sistema. Tras un inicio de sesión exitoso, los usuarios son redirigidos a esta página, donde pueden comenzar a interactuar con las funcionalidades del sistema.
- **pages/dashboard.jsx**: Esta es una ruta protegida, accesible únicamente para usuarios que poseen un token JWT válido. Asegura que solo los usuarios autenticados puedan visualizar y gestionar las tareas del proyecto.

Esta estructura permite un desarrollo organizado y escalable, facilitando la implementación y mantenimiento del sistema de gestión de proyectos.

### **Paquetes utilizados.**

- **Next.js:** Framework central utilizado para construir el proyecto, que permite el renderizado del lado del servidor y la creación de rutas API personalizadas, optimizando la experiencia del usuario y la gestión de datos.
- **Axios:** Librería empleada para realizar solicitudes HTTP desde el frontend hacia las rutas API, facilitando la comunicación entre el cliente y el servidor de manera eficiente.
- **Bcryptjs:** Biblioteca que se utiliza para el hash de contraseñas, asegurando que las credenciales de los usuarios se almacenen de manera segura en el sistema.
- **Jsonwebtoken:** Biblioteca diseñada para generar y verificar tokens JWT, garantizando un mecanismo robusto de autenticación y gestión de sesiones de usuario.
- **FS y Path:** Módulos integrados en Node.js que permiten la lectura y escritura en el sistema de archivos, utilizados para almacenar y recuperar datos de usuario desde `users.json`, asegurando una gestión eficiente de la información.

## **Puntos finales de la API**

### **1. POST /api/registro**

Este punto final gestiona el registro de usuarios aceptando un correo electrónico y una contraseña.

- Cuerpo de la solicitud:

```
{  
  "email": "example@example.com",  
  "password": "yourpassword"  
}
```

- Respuesta (en caso de éxito):

```
{  
  "mensaje": "Usuario registrado con éxito"  
}
```

### **2. POST /api/login**

Este punto final gestiona el inicio de sesión del usuario verificando el correo electrónico y la contraseña. Si las credenciales son correctas, se devuelve un token JWT.

- Cuerpo de la solicitud:

```
{  
  "email": "example@example.com",  
  "password": "yourpassword"  
}
```

- Respuesta (en caso de éxito):

```
{  
  "token": "jwt-token-here"  
}
```

## **Mejoras futuras**

- Integración de bases de datos\*\*: En lugar de almacenar los datos del usuario en un archivo JSON, se podría mejorar el sistema integrando una base de datos como MongoDB o PostgreSQL.
- Función de restablecimiento de contraseña\*\*: Añadir una función de restablecimiento de contraseña mejoraría la experiencia de usuario y la seguridad.
- Frontend mejorado\*\*: Mejora de la interfaz de usuario mediante el uso de frameworks CSS como TailwindCSS o Bootstrap para un mejor diseño y capacidad de respuesta.



**Conclusion:**

En resumen, el sistema de gestión de proyectos desarrollado con Next.js ofrece una solución integral para la autenticación y la gestión de usuarios, combinando seguridad y facilidad de uso. A través de la implementación de un registro de usuarios y un mecanismo de inicio de sesión basado en tokens JWT, el sistema garantiza que solo los usuarios autorizados tengan acceso a funciones críticas, como el cuadro de mandos.

La estructura organizada del proyecto, junto con el uso de librerías y módulos eficientes, permite una gestión fluida de las credenciales y los datos de los usuarios. Al utilizar herramientas como bcrypt para el hash de contraseñas y Axios para las solicitudes HTTP, se asegura tanto la integridad de la información como una experiencia de usuario optimizada.

Este sistema no solo cumple con los requisitos fundamentales de autenticación y seguridad, sino que también establece una base sólida para futuras mejoras y expansiones. A medida que las necesidades de gestión de proyectos continúan evolucionando, este enfoque modular y seguro permitirá adaptaciones rápidas y eficientes, contribuyendo al éxito continuo de los usuarios en la gestión de sus proyectos.