

Contenido

1. Descripción del Proyecto.....	2
2. Especificaciones de Requisitos	2
3. Diseño del Sistema	3
4. Plan de Gestión del Proyecto	4
5. Plan de Pruebas	6
6. Manuales de Usuario.....	7
7. Registro de Cambios	8
8. Documentación Técnica	9

Integrantes:

Julio Armando Quintanilla Acosta

Luis David Campos Hernández

Mario Josué Lemus López

Bryan Enrique Padilla Rosa

Dennis Hefziba Martínez

Módulo Web de Recuperación de Contraseñas

1. Descripción del Proyecto

El proyecto consiste en un módulo de inicio de sesión que valida las credenciales de los usuarios utilizando JSON Web Tokens (JWT) para la autorización y acceso a un dashboard el cual es una ruta protegida. Este sistema también incluye la recuperación de contraseñas en caso de olvido.

2. Especificaciones de Requisitos

2.1 Requisitos Funcionales

- **Inicio de Sesión:** El sistema muestra un formulario de inicio de sesión y si las credenciales son correctas, el sistema genera un token JWT para la sesión del usuario.
- **Recuperación de Contraseña:** El sistema envía un correo electrónico al usuario con un enlace de recuperación de contraseña que contiene un token de recuperación. También se actualiza la contraseña del usuario en la base de datos.
- **Seguridad:** El sistema debe asegurar que todas las rutas protegidas solo sean accesibles mediante un token JWT válido y debe asegurar que los tokens tengan una expiración definida.

2.2 Requisitos No Funcionales

- **Rendimiento:** El tiempo de respuesta para la autenticación de usuarios.
- **Mantenibilidad:** El sistema debe estar diseñado de manera modular para facilitar la actualización y el reemplazo de componentes individuales.

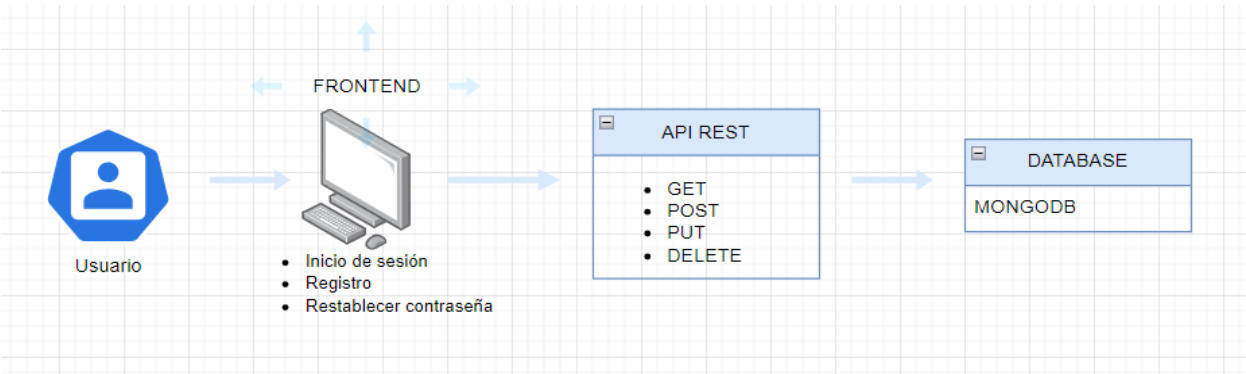
- **Escalabilidad:** La arquitectura del sistema debe permitir una fácil escalabilidad horizontal para manejar aumentos en la carga de trabajo.
- **Disponibilidad:** El sistema debe incluir mecanismos de recuperación ante fallos y backup para asegurar la continuidad del servicio

3. Diseño del Sistema

3.1 Arquitectura del Sistema

- **Frontend:**
 - ✓ **Login Page:** Página donde los usuarios ingresan sus credenciales.
 - ✓ **Forgot Password Page:** Página donde los usuarios ingresan su correo electrónico para solicitar un enlace de recuperación de contraseña.
 - ✓ **Reset Password Page:** Página donde los usuarios ingresan su nueva contraseña utilizando el enlace de recuperación.
- **Backend:**
 - ✓ **Auth Controller:** Maneja la autenticación de usuarios y la generación de tokens JWT
 - ✓ **User Controller:** Maneja la lógica relacionada con los usuarios, incluyendo la actualización de contraseñas.
 - ✓ **Email Service:** Envía correos electrónicos con enlaces de recuperación de contraseña.
- **Base de Datos (MongoDB):**
 - ✓ **Usuarios:** Colección que almacena los datos de los usuarios, incluyendo nombre, apellido, nombre de usuario, contraseña (hashed) y correo electrónico.

3.2 Diagrama de Componentes



4. Plan de Gestión del Proyecto

4.1 Cronograma

Fase	Fecha de inicio	Fecha de fin
Análisis de Requisito	14/07/2024	14/07/2024
Diseño del Sistema	14/07/2024	14/07/2024
Desarrollo del Backend	14/07/2024	17/07/2024
Desarrollo del Frontend	15/07/2024	18/07/2024
Pruebas y Correcciones	15/07/2024	18/07/2024
Implementación y Despliegue	18/07/2024	18/07/2024

4.2 Asignación de Tareas

Tarea	Responsables
-------	--------------

Análisis de Requisitos	<ul style="list-style-type: none"> • Julio Armando Quintanilla Acosta • Mario Lemus López • Brayan Enrique Padilla Rosa • Dennis Erfsiva Martínez • Luis David Campos.
Diseño del Sistema	<ul style="list-style-type: none"> • Julio Armando Quintanilla Acosta • Mario Lemus López • Brayan Enrique Padilla Rosa • Dennis Erfsiva Martínez • Luis David Campos
Desarrollo del Backend	<ul style="list-style-type: none"> • Julio Armando Quintanilla Acosta • Mario Lemus López • Brayan Enrique Padilla Rosa • Dennis Erfsiva Martínez • Luis David Campos
Desarrollo del Frontend	<ul style="list-style-type: none"> • Julio Armando Quintanilla Acosta • Mario Lemus López • Brayan Enrique Padilla Rosa
Pruebas y Correcciones	<ul style="list-style-type: none"> • Julio Armando Quintanilla Acosta • Mario Lemus López • Brayan Enrique Padilla Rosa • Luis David Campos
Implementación y Despliegue	<ul style="list-style-type: none"> • Julio Armando Quintanilla Acosta • Mario Lemus López • Brayan Enrique Padilla Rosa • Dennis Erfsiva Martínez • Luis David Campos

5. Plan de Pruebas

5.1 Estrategia de Pruebas

Pruebas Unitarias.

- Cada función del backend será probada de manera independiente.
- Uso de frameworks: Jest.

Pruebas de Integración:

- Verificación de que los componentes del sistema funcionan correctamente juntos.
- Simulación de flujos completos de usuario.

Pruebas de Usuario:

- Evaluación de la funcionalidad y usabilidad por parte de usuarios reales.
- Recolección de feedback para mejoras.

5.2 Casos de Prueba

1. Inicio de Sesión:

- CT01: Iniciar sesión con credenciales válidas.
- CT02: Iniciar sesión con credenciales inválidas.
- CT03: Intentar iniciar sesión sin ingresar datos.
- CT04: Verificar que el enlace de recuperación contiene un token seguro.

2. Recuperación de Contraseña:

- CT05: Solicitar recuperación con un correo válido.
- CT06: Solicitar recuperación con un correo no registrado.
- CT07: Intentar usar un enlace de recuperación caducado.
- CT08: Intentar usar el mismo enlace de recuperación más de una vez

3. Modificación de Contraseña:

- CT09: Cambiar contraseña con un enlace válido.
- CT10: Intentar cambiar contraseña con un enlace inválido.
- CT11: Verificar que la nueva contraseña funciona en el inicio de sesión.

6. Manuales de Usuario

6.1 Guía de Instalación

1. Tener instalando node y npm ya sea Windows o Linux
2. Tener acceso libre al puerto 3000
3. Instalación de node y npm
4. Descargar el ejecutable de node en <https://nodejs.org/en>
5. Abre el archivo descargado (generalmente, tiene una extensión .msi).
6. Sigue las instrucciones del asistente de instalación.
7. Asegúrate de que la opción para instalar npm esté seleccionada.
8. Acepta las ubicaciones predeterminadas y las configuraciones.
9. Haz clic en "Instalar" y luego en "Finalizar" cuando la instalación se complete.
10. Abre una terminal de comandos (CMD) o PowerShell y utiliza los siguientes comandos para verificar la instalación
11. node -v
12. npm -v.
13. Instalación en Linux
14. Actualizar con el comando `sudo apt update`
15. `curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -`
16. `sudo apt install -y nodejs`
17. Verificar con los siguientes comandos
18. node -v
19. npm -v
20. Utilizar el siguiente comando para instalar todas las dependencias `npm install`
21. En la carpeta root del proyecto buscar archivo .env y agregar la siguiente información

- 22. PORT=3000
- 23. SECRET_KEY=clave_secreata
- 24. MONGO_URI=mongodb://localhost:27017/Usuarios
- 25. EMAIL_USER=usuario@mail.com
- 26. EMAIL_PASS=contraseña personal
- 27. EMAIL_SERVICE=hotmail

(**Nota:** el apartado EMAIL_SERVICE varía dependiendo del tipo de correo que se utiliza en caso de Outlook es Hotmail, caso de Gmail es Gmail y Yahoo! es Yahoo!)

- 28. En la terminal ya sea cmd o la terminal de preferencia de Linux correr el siguiente comando
- 29. node ./app.js
- 30. La página estará en <http://localhost/3000>

6.2 Guía del Usuario Final

1. Al ingresar a la página de inicio el usuario tendrá un formulario donde podrá ingresar o crear una cuenta para ingresar el dashboard esta página solo será accesible si y solo si el usuario se ha loggeado previamente en un periodo no mayor a 15s.
2. La creación de usuario pedirá los siguientes campos, nombre, apellido, correo usuario y contraseña donde usuario y correo deben ser únicos para cada nuevo usuario.
3. La recuperación de contraseñas pedirá el correo asociado con la cuenta una vez llenado ese campo y enviada la solicitud, el usuario obtendrá un link a su correo y tendrá de 15 min para poder cambiar su contraseña.

7. Registro de Cambios

v1.0.0 (2024-07-10)

- Implementación inicial de la API REST de registro de usuarios.

- Funcionalidades: log de usuarios, creación de usuarios, cambio de contraseñas,
- restricción por JWT.
- Envío de correo

8. Documentación Técnica

8.1 Estructura del Código

- Utiliza camelCase para nombrar variables y funciones. Ejemplo: num1, calculateResult.
- PascalCase para Nombres de Clases
- Constantes en MAYÚSCULAS
- Usar tabuladores para la indentación.
- Uso de Comillas Simples para llamadas a librerías y dobles para mensajes

8.2 Comentarios en el Código

Comentarios en el Código Funciones, objetos y variables deben ser comentadas antes de hacer la declaración para mejor comprensión y mantenimiento.