

Prueba Técnica

Aprendiz:

Santiago Castañeda Castaño

Instructor:

Richard Alexander Betancur

Ficha:

2694526

Web MERN/MEAN STACK

Vacante: Desarrollador

SENA

27/11/2023

Documentación del Proyecto "NotesApp"

Objetivo del Proyecto

Desarrollar una aplicación de notas que consta de un frontend y un backend. El frontend permitirá a los usuarios crear notas, visualizarlas y eliminarlas. El backend gestionará la lógica del negocio, almacenando las notas y usuarios en una base de datos MongoDB.

Estructura del Proyecto

```
stack_react/
|
|└─ notes_app_sps/
|  |└─ backend/
|  |  |└─ models/
|  |  |  |└─ Note.js
|  |  |  |└─ User.js
|  |  |└─ routes/
|  |  |  |└─ noteRoutes.js
|  |  |  |└─ userRoutes.js
|  |  └─ server.js
|  |
|  |└─ frontend/
|  |  |└─ public/
|  |  |  └─ index.html
|  |  |└─ src/
|  |  |  |└─ components/
|  |  |  |  |└─ CreateUser.js
|  |  |  |  |└─ CreateNote.js
|  |  |  |  └─ Notes.js
|  |  |  └─ App.js
|  |  └─ index.js
```

```

| | └─ package.json
| | └─ ...
| |
| └─ stack_react_notes_app_sps.postman_collection.json
|
└─ ...

```

- **backend/**: Contiene el código del servidor Node.js y la configuración de la base de datos MongoDB.
- **frontend/**: Contiene el código del cliente React.
- **stack_react_notes_app_sps.postman_collection.json**: Una colección de Postman que contiene ejemplos de solicitudes para probar las API del backend.

Backend

Configuración del Servidor

En el archivo **backend/server.js**, se establece la conexión al servidor y se configuran las rutas.

```

const express = require('express');
const cors = require('cors');
const mongoose = require('mongoose');
const noteRouter = require('./routes/noteRoutes');
const userRouter = require('./routes/userRoutes');

```

```

const app = express();
const PORT = process.env.PORT || 5000;

```

```

app.use(cors());

```

```

app.use(express.json());

```

```

mongoose.connect('mongodb://localhost:27017', {

```

```
useNewUrlParser: true,  
useUnifiedTopology: true,  
});  
  
app.use('/servicio/api_notes_app/notes', noteRouter);  
app.use('/servicio/api_notes_app/users', userRouter);  
  
app.listen(PORT, () => {  
  console.log(`Servidor en ejecución en el puerto ${PORT}`);  
});
```

Definición de Modelos y Rutas

- **backend/models/Note.js**: Define el modelo de datos para las notas.
- **backend/models/User.js**: Define el modelo de datos para los usuarios.
- **backend/routes/noteRoutes.js**: Define las rutas y controladores para las notas.
- **backend/routes/userRoutes.js**: Define las rutas y controladores para los usuarios.

Frontend

Componentes React

CreateUser.js

Permite a los usuarios crear nuevos usuarios y muestra la lista de usuarios existentes.

CreateNote.js

Permite a los usuarios crear nuevas notas. Incluye un formulario con un selector de usuarios, título, contenido, fecha y un botón de guardar.

Notes.js

Muestra la lista de notas existentes con la opción de eliminar cada nota.

Configuración de las Solicitudes a la API

En los componentes **CreateUser.js**, **CreateNote.js** y **Notes.js**, se utilizan solicitudes Axios para interactuar con el backend. Asegúrate de ajustar las URL según la configuración de tu servidor.

```
// Ejemplo de URL en CreateUser.js
axios.get('http://localhost:5000/servicio/api_notes_app/users')
  .then(response => setUsers(response.data))
  .catch(error => console.error('Error al obtener la lista de usuarios:',
error));
```

```
// Ejemplo de URL en CreateNote.js
axios.post('http://localhost:5000/servicio/api_notes_app/notes', { /*
datos de la nota */ })
  .then(response => {
    console.log('Nota creada exitosamente:', response.data);
    // Lógica adicional si es necesario
  })
  .catch(error => console.error('Error al crear la nota:', error));
```

```
// Ejemplo de URL en Notes.js
axios.get('http://localhost:5000/servicio/api_notes_app/notes')
  .then(response => setNotes(response.data))
  .catch(error => console.error('Error al obtener la lista de notas:', error));
```

Ejecución del Proyecto

1. Backend:

- Navega a la carpeta **backend** en tu terminal.
- Ejecuta **npm install** para instalar las dependencias.
- Ejecuta **npm start** para iniciar el servidor backend.

2. Frontend:

- Navega a la carpeta **frontend** en otra terminal.
- Ejecuta **npm install** para instalar las dependencias.
- Ejecuta **npm start** para iniciar la aplicación frontend.

Postman Collection

Se proporciona una colección de Postman (**stack_react_notes_app_sps.postman_collection.json**) con ejemplos de solicitudes para probar las API del backend. Importa esta colección en tu cliente de Postman para realizar pruebas y verificar el funcionamiento del backend.