Contenido

[1. Configurar el backend con Node.js y Express.js 1](#_Toc140669302)

[a. Crea una nueva carpeta para tu proyecto y ábrela en tu editor de código. 1](#_Toc140669303)

[b. Instala los paquetes necesarios 1](#_Toc140669304)

[c. Configura Express.js en server.js e importa las dependencias necesarias: 1](#_Toc140669305)

[2. Configurar la base de datos MongoDB 2](#_Toc140669306)

[a. configura la conexión con MongoDB 2](#_Toc140669307)

[3. Crear los modelos de datos 3](#_Toc140669308)

Crear un modelo de base de datos, relacional en mi caso

# Configurar el backend con Node.js y Express.js

## Crea una nueva carpeta para tu proyecto y ábrela en tu editor de código.

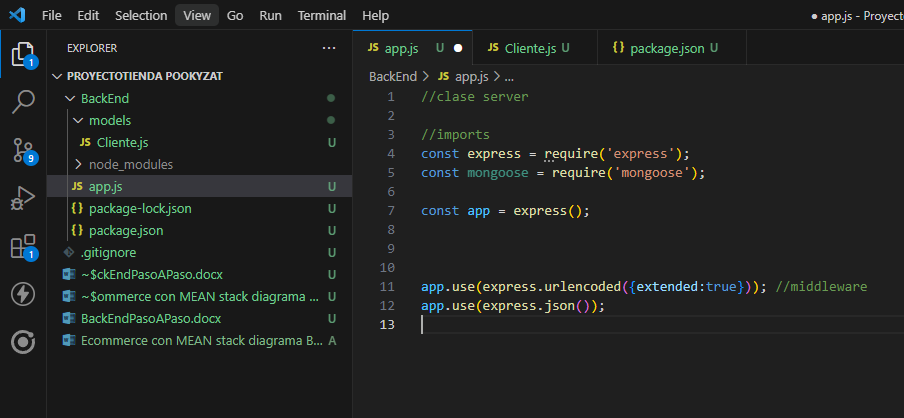
Inicializa un nuevo proyecto de Node.js ejecutando npm init en la terminal y siguiendo las instrucciones.

## **Instala los paquetes necesarios**

ejecutando npm install express mongoose en la terminal.

## Configura Express.js en server.js e importa las dependencias necesarias:

1. //clase server
2. //imports
3. const express = require('express');
4. const mongoose = require('mongoose');
5. const app = express();
7. app.use(express.urlencoded({extended:true})); //middleware
8. app.use(express.json());



# Configurar la base de datos MongoDB

Asegúrate de tener MongoDB instalado y ejecutándose.

## configura la conexión con MongoDB

En el archivo app.js del backend, configura la conexión con MongoDB utilizando Mongoose.

//coneccion a la base de datos

const uri='mongodb://127.0.0.1:27017/Pookyzat'

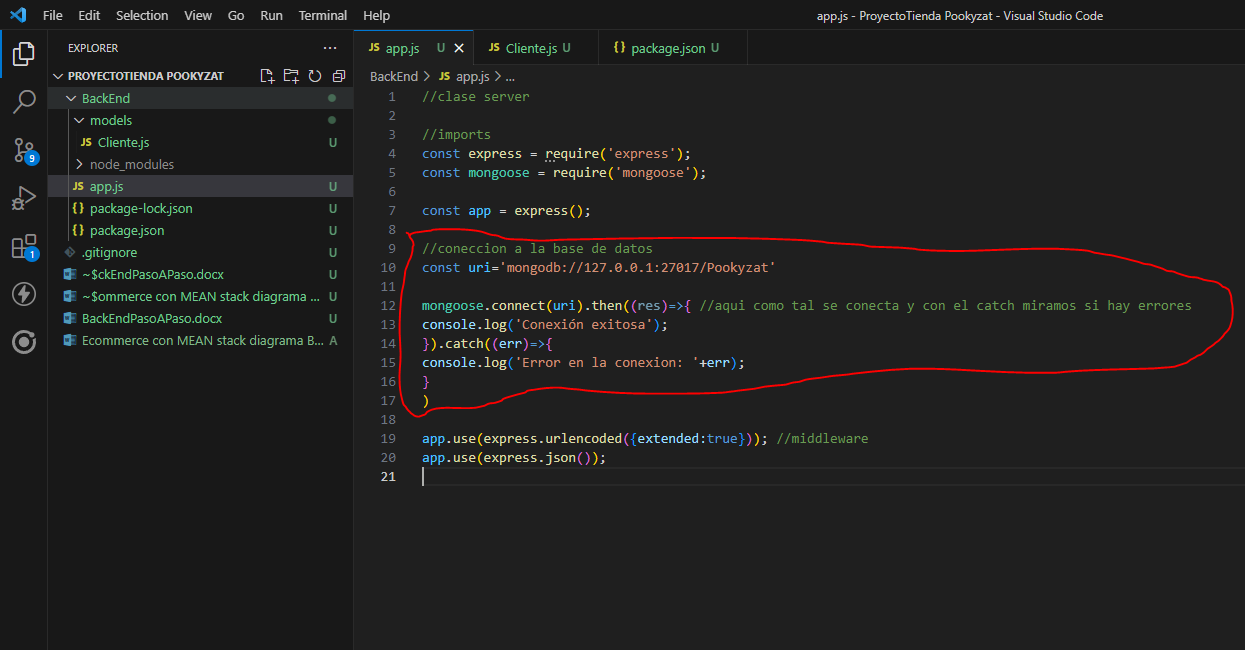
mongoose.connect(uri).then((res)=>{ //aqui como tal se conecta y con el catch miramos si hay errores

console.log('Conexión exitosa');

}).catch((err)=>{

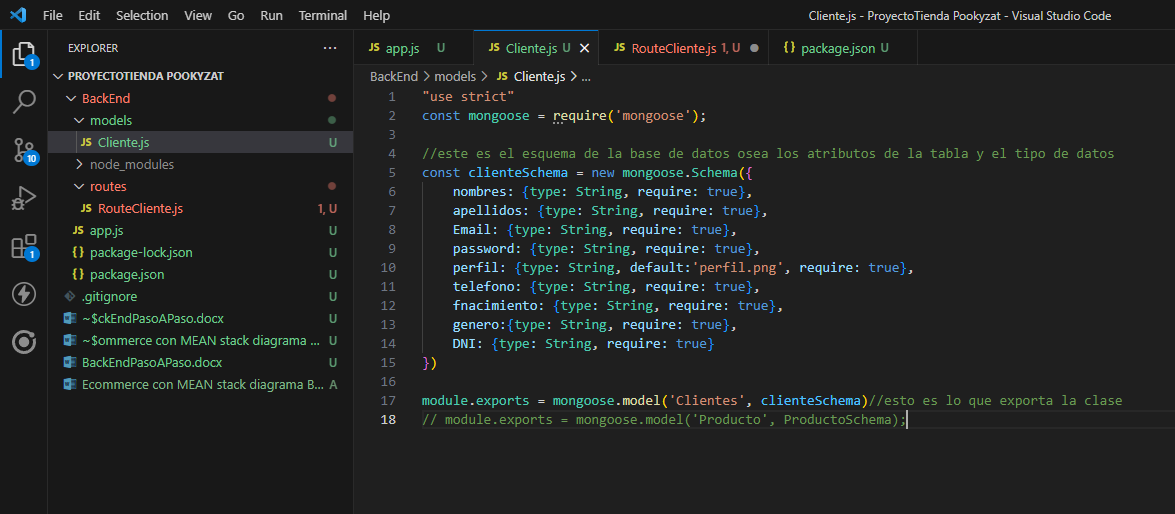
console.log('Error en la conexion: '+err);

}



# Crear los modelos de datos

Crea los schemas con mongoose en una carpeta llamada models



"use strict"

const mongoose = require('mongoose');

//este es el esquema de la base de datos osea los atributos de la tabla y el tipo de datos

const clienteSchema = new mongoose.Schema({

    nombres: {type: String, require: true},

    apellidos: {type: String, require: true},

    Email: {type: String, require: true},

    password: {type: String, require: true},

    perfil: {type: String, default:'perfil.png', require: true},

    telefono: {type: String, require: true},

    fnacimiento: {type: String, require: true},

    genero:{type: String, require: true},

    DNI: {type: String, require: true}

})

module.exports = mongoose.model('Clientes', clienteSchema)//esto es lo que exporta la clase

# Implementar las rutas y controladores

## Routing

Crea las rutas para las operaciones CRUD (crear, leer, actualizar, eliminar) en el archivo index.js.

Define los controladores correspondientes para cada ruta, donde se ejecutarán las acciones en la base de datos.

//Primer paso a paso

Aquí tienes un paso a paso para crear un CRUD utilizando el stack MEAN (MongoDB, Express.js, Angular, Node.js):

Paso 1: Configurar el backend con Node.js y Express.js

* Crea una nueva carpeta para tu proyecto y ábrela en tu editor de código.
* Inicializa un nuevo proyecto de Node.js ejecutando **npm init** en la terminal y siguiendo las instrucciones.
* Instala los paquetes necesarios ejecutando **npm install express mongoose** en la terminal.
* Crea un archivo **index.js** (o cualquier otro nombre) para tu backend y configura un servidor Express.js básico con las rutas y controladores para el CRUD.

Paso 2: Configurar la base de datos MongoDB

* Asegúrate de tener MongoDB instalado y ejecutándose.
* En el archivo **index.js** del backend, configura la conexión con MongoDB utilizando Mongoose.

Paso 3: Crear los modelos de datos

* Define los esquemas de tus datos en Mongoose, que representarán los documentos de MongoDB.
* Crea los modelos utilizando los esquemas definidos, los cuales te permitirán interactuar con la base de datos.

Paso 4: Implementar las rutas y controladores

* Crea las rutas para las operaciones CRUD (crear, leer, actualizar, eliminar) en el archivo **index.js**.
* Define los controladores correspondientes para cada ruta, donde se ejecutarán las acciones en la base de datos.

Paso 5: Configurar el frontend con Angular

* Crea una nueva carpeta para el frontend en tu proyecto y ábrela en tu editor de código.
* Instala Angular CLI globalmente ejecutando **npm install -g @angular/cli** en la terminal.
* Crea un nuevo proyecto de Angular ejecutando **ng new nombre-proyecto** en la terminal y sigue las instrucciones.
* Navega al directorio del proyecto Angular con **cd nombre-proyecto**.

Paso 6: Crear componentes y servicios

* Crea los componentes necesarios para las vistas de tu CRUD, como lista de elementos, formulario de creación/edición, etc.
* Crea servicios Angular para interactuar con el backend, utilizando HTTP para realizar las solicitudes al servidor.

Paso 7: Configurar las rutas y enlaces

* Configura las rutas en Angular para mapear las URL a los componentes correspondientes.
* Crea enlaces en tus vistas para navegar entre las diferentes páginas y acciones del CRUD.

Paso 8: Implementar las funcionalidades del CRUD en el frontend

* Utiliza los servicios Angular para realizar las solicitudes al backend y realizar las operaciones CRUD (crear, leer, actualizar, eliminar) en los componentes correspondientes.

Paso 9: Ejecutar la aplicación

* Inicia el servidor backend ejecutando **node index.js** en la terminal en la carpeta del backend.
* Inicia la aplicación Angular ejecutando **ng serve** en la terminal en la carpeta del frontend.
* Abre tu navegador y accede a **http://localhost:4200** para ver y probar tu aplicación CRUD.

¡Felicidades! Ahora tienes un CRUD básico utilizando el stack MEAN. Puedes continuar añadiendo más funcionalidades, validaciones y mejoras según tus necesidades.