Ejercicio BD NOSQL 🡪 Proyecto CINE

1. Inicialización Entorno
   1. Npm install -y
   2. npm install express mongoose body-parser bcrypt jsonwebtoken
2. Creamos app.js
3. Creamos routes
   1. // routes/auth.js
   2. const express = require('express');
   3. const router = express.Router();
   4. const bcrypt = require('bcrypt');
   5. const jwt = require('jsonwebtoken');
   6. const User = require('../models/User');
   7. // Ruta para registrar un usuario
   8. router.post('/register', async (req, res) => {
   9. try {
   10. const { dni, nombres, apellidos, fechaNacimiento, rol, telefono, correo, contraseña, nombreUsuario } = req.body;
   11. // Verificar si el usuario ya existe
   12. const existingUser = await User.findOne({ dni });
   13. if (existingUser) {
   14. return res.status(400).json({ message: 'El usuario ya existe' });
   15. }
   16. // Hash de la contraseña
   17. const hashedPassword = await bcrypt.hash(contraseña, 10);
   18. // Crear un nuevo usuario
   19. const user = new User({
   20. dni,
   21. nombres,
   22. apellidos,
   23. fechaNacimiento,
   24. rol,
   25. telefono,
   26. correo,
   27. contraseña: hashedPassword,
   28. nombreUsuario,
   29. });
   30. await user.save();
   31. res.status(201).json({ message: 'Usuario registrado exitosamente' });
   32. } catch (error) {
   33. res.status(500).json({ error: 'Error interno del servidor' });
   34. }
   35. });
   36. module.exports = router;
   37. //COnfiguración de rutas
   38. app.use('/api/auth', require('./routes/auth'));
4. Creamos models
   1. // models/User.js
   2. const mongoose = require('mongoose');
   3. const userSchema = new mongoose.Schema({
   4. dni: String,
   5. nombres: String,
   6. apellidos: String,
   7. fechaNacimiento: Date,
   8. rol: String,
   9. telefono: String,
   10. correo: String,
   11. contraseña: String,
   12. nombreUsuario: String,
   13. });
   14. module.exports = mongoose.model('User', userSchema);
5. Configuración de rutas
   1. app.use('/api/auth', require('./routes/auth'));
6. Ejecución
   1. node app.js