Proyecto "Patitas Felices"

Documentación técnica del proyecto

Integrantes: Diana Sierra

1. Descripción general del proyecto

El proyecto 'Patitas Felices' es una aplicación desarrollada en equipo con el objetivo de gestionar información relacionada con adopciones de mascotas, productos, usuarios y pedidos. El desarrollo se realizó utilizando NestJS, empleando buenas prácticas de modularidad, DTOs y seguridad con JSON Web Tokens (JWT) y bcrypt.

2. Arquitectura del sistema

El backend está estructurado en módulos independientes que permiten mantener la separación de responsabilidades. Cada módulo cuenta con sus controladores, servicios y DTOs, siguiendo el patrón MVC aplicado a NestJS.

- 3. Módulos del proyecto
- 3.1 Usuarios y Autenticación Diana Sierra

Diana Sierra fue responsable del desarrollo del módulo de Usuarios y Autenticación. Implementó las funcionalidades de registro y login, integrando bcrypt para encriptar contraseñas y JWT para la generación de tokens de acceso seguro. También se manejó la validación de datos mediante DTOs.

Endpoints principales:

- POST /auth/register → Registro de nuevos usuarios.
- POST /auth/login → Inicio de sesión con verificación de credenciales.

El proceso fue probado exitosamente en Postman, asegurando respuestas adecuadas para casos de registro, inicio de sesión correcto y manejo de errores.

Integrante: Diana Sofía Sierra Módulo de Usuarios y Autenticación

1. Descripción del módulo

El módulo **Users y Auth** tiene como objetivo gestionar el registro y la autenticación de los usuarios del sistema **Patitas Felices**. Se implementa la seguridad mediante **bcrypt** para el encriptado de contraseñas y **JWT** (**JSON Web Token**) para la generación y validación de tokens, garantizando así la protección de las rutas y datos de los usuarios.

2. Funcionalidades principales

Registro de usuarios con almacenamiento de contraseñas encriptadas.

- Inicio de sesión (login) validando credenciales seguras.
- Generación de token JWT para autenticar solicitudes.
- Validación de contraseñas con bcrypt.
- Estructura modular y reutilizable mediante los servicios AuthService y UsersService.

3. Componentes implementados

auth.service.ts:

Contiene la lógica de registro, validación de contraseñas y generación de tokens JWT.

auth.controller.ts:

Define las rutas /auth/register y /auth/login para registrar e iniciar sesión.

• users.service.ts:

Administra los datos de los usuarios en memoria (lista local) y crea nuevos usuarios.

create-user.dto.ts:

Define la estructura de datos requerida para registrar un nuevo usuario.

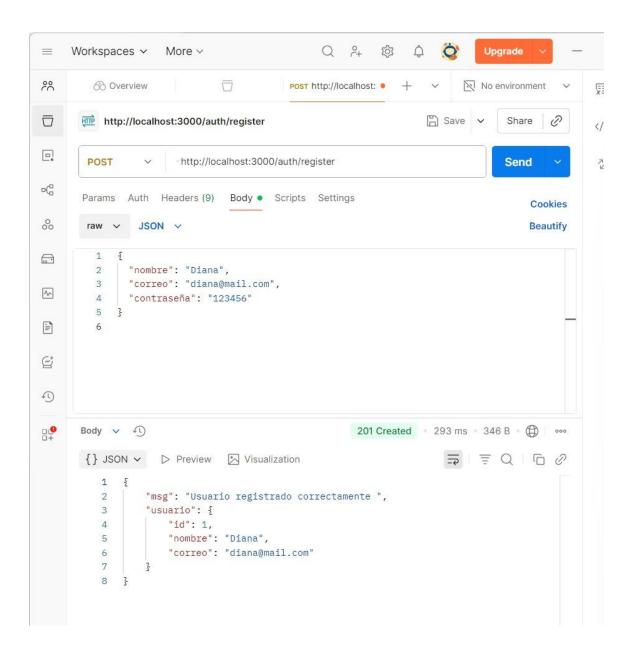
4. Tecnologías utilizadas

- NestJS para la arquitectura del backend.
- bcrypt para el cifrado de contraseñas.
- @nestjs/jwt para la autenticación mediante tokens.
- TypeScript para la tipificación y estructura del código.

5. Pruebas realizadas

Las pruebas se ejecutaron en **Postman**, enviando solicitudes POST a los siguientes endpoints:

Registro:



Login:

```
Q 24 $ $ $ $
                                                                          Upgrade
     Workspaces ∨ More ∨
200
      Overview
                               POST http://loca POST http://loc. • +
                                                                       No environment
                                                                                             F =
                                                                 □ Save ∨
\Box
      http://localhost:3000/auth/login
                                                                              Share
                                                                                             </>
POST
                      http://localhost:3000/auth/login
                                                                              Send
o(0
      Params Auth Headers (9) Body • Scripts Settings
                                                                                  Cookies
000
                JSON V
                                                                                  Beautify
         1
.
         2
              "correo": "diana@mail.com",
         3
               "contraseña": "123456"
1
         4
         5
(m)
0
      Body V
                                                    201 Created = 278 ms = 511 B = 💮 -----
0.0
                                                                       FQ 6
      {} JSON ✓ ▷ Preview ☑ Visualization
          1
                 "msg": "Inicio de sesión exitoso ",
          2
          3
                 "token": "eyJhbGci0iJIUzI1NiIsInR5cCI6IkpXVCJ9.
                     eyJjb3JyZW8i0iJkaWFuYUBtYWlsLmNvbSIsImlhdCI6MTc2MTAyMDA0NiwiZXhwIjoxN
                     zYxMDIzNjQ2fQ.rhzvRqskgZC0LGF1Qgalq0zecRWpRYUiDR6z1D15UK4",
          4
                 "user": {
          5
                     "id": 1,
                     "nombre": "Diana",
         7
                     "correo": "diana@mail.com"
         8
          9 }
```