

Proyecto Backend - Huellitas Felices

Este proyecto es una aplicación backend desarrollada con Node.js y Express, orientada a la gestión integral de una petshop.

Permite administrar productos, mascotas y turnos de atención para servicios veterinarios y peluquería, utilizando almacenamiento en archivos JSON y vistas dinámicas generadas con Pug.

Funcionalidades principales

Login con control de sesión

- Validación de usuarios desde archivo `usuarios.json`
- Sesiones con `express-session`
- Acceso a funcionalidades según el rol (administrador o empleado)

Módulo de Productos

- Alta, modificación, visualización y baja de productos
- Campos: ID, precio, stock, categoría
- Vista detallada por producto
- Protección con `autorizar(['administrador'])` para modificar/eliminar

Módulo de Mascotas

- Registro, edición, listado y eliminación de mascotas
- Búsqueda por nombre o especie con vista filtrada
- Campos: ID, nombre, especie, raza, nacimiento, responsable

Módulo de Turnos

- Registro y gestión de turnos para servicios veterinarios o peluquería
 - Campos: ID, tipo, fecha, hora, mascota, dueño
-

Buscador de Mascotas

- Ruta: `GET /mascotas/buscar`
- Parámetros: `?nombre=...` y/o `?especie=...`
- Vista de resultados con tabla
- Formulario visual disponible desde `/formulario-busqueda`

⚙️ □ Funcionamiento técnico

La lógica del backend está implementada en `index.js`, que actúa como punto central del servidor Express.

A continuación, se detallan los componentes clave:

Servidor Express

- Maneja todas las rutas (GET y POST) para productos, mascotas y turnos.
- Organiza la navegación mediante rutas dedicadas para vistas y formularios.

Middleware

- `express.urlencoded()` y `express.json()` para interpretar datos entrantes.
- `express-session` para manejar sesiones de usuarios.

Sistema de Login

- Validación de credenciales desde archivo `usuarios.json`
- Almacenamiento del nombre de usuario y rol en sesión

Autorización por rol

- Middleware `autorizar([roles])` protege rutas según permisos
- Solo administradores pueden eliminar/modificar datos

Archivos JSON como base de datos

- Lectura y escritura con `fs/promises`
- Funciones como `leerProductos`, `leerMascotas`, `leerTurnos` encapsulan esta lógica

Motor de vistas Pug

- Separación de vistas por entidad (producto, mascota, turno)
- Vistas de edición, errores, formularios y listado

Buscador de mascotas

- Ruta: `/mascotas/buscar`
- Filtra resultados por nombre y/o especie
- Resultados renderizados en `mascotas_busqueda.pug`

📁 Estructura del proyecto

```
huellitas_felices
├─ index.js
```

```
├── Producto.js / Mascota.js / Turno.js
├── data/
│   ├── productos.json
│   ├── mascotas.json
│   ├── turnos.json
│   └── usuarios.json
├── views/
│   ├── productos.pug / mascotas.pug / turnos.pug
│   ├── editar-*.pug / error-*.pug
│   ├── login.pug / menu.pug / bienvenida.pug
│   ├── mascotas_busqueda.pug
│   └── formulario_busqueda_mascotas.pug
└── package.json
```

► ☐ Cómo ejecutar el proyecto

1. Instalar dependencias:

```
npm install
npm install express-session
```

2. Ejecutar el servidor:

```
npm start
```

3. Ingresar desde el navegador:

```
http://localhost:3000/
```



Usuarios habilitados

– Administrador

- Usuario: **Claudia**
- Contraseña: **12345**

– Empleado

- Usuario: **Mariana**
 - Contraseña: **12345**
-



Pruebas con Thunder Client

El archivo **Uso_Thunder_Client.pdf** adjunto incluye pruebas prácticas con ejemplos de:

- Alta de producto
- Eliminación protegida según rol
- Filtros de búsqueda de mascotas

Tecnologías utilizadas

- Node.js
- Express
- Pug
- fs/promises
- express-session
- Thunder Client (archivo pdf)

Autoría

Trabajo realizado por el Grupo 4 - Comisión D en el marco del caso práctico “Petshop Huellitas Felices”.