Manual de Funcionalidad del Sistema de Gestión Veterinaria

Este documento describe las funcionalidades básicas del prototipo inicial del sistema de gestión para una clínica veterinaria, basándose en las pantallas de la interfaz de usuario, la documentación de la API en Swagger y el esquema de la base de datos SQL.

1. Registro de Usuarios

Esta pantalla permite la creación de nuevos usuarios en el sistema, que pueden ser propietarios de mascotas o personal de la clínica. El formulario solicita la información esencial para el perfil de un usuario:

- **Nombre**
- **Apellido**
- **Correo**
- **Teléfono**
- **Dirección**

Una vez completados los datos, el botón "Crear" enviará la información para su registro en el sistema. En el backend, esta acción se maneja a través del **endpoint POST** api/veterinaria/usuarios.

2. Registro y Consulta de Mascotas

Esta sección del sistema está diseñada para gestionar la información de las mascotas de los clientes. El formulario de "Registro y Consulta de Mascotas" permite

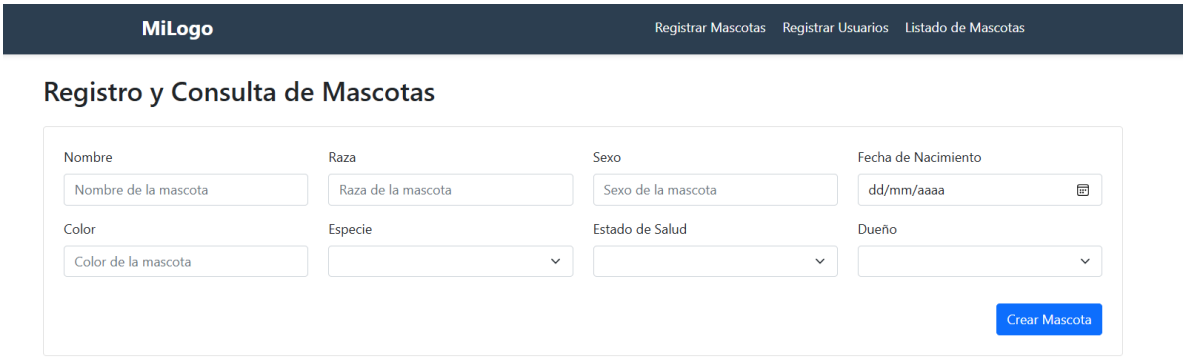
ingresar los datos de una nueva mascota, los cuales se almacenan en la tabla Mascotas del esquema de la base de datos.

2.1. Funcionalidad de Registro

El formulario de registro de mascotas incluye los siguientes campos:

- **Nombre**
- **Raza**
- **Sexo**
- **Fecha de Nacimiento**
- **Color**
- **Especie**: Este campo es un desplegable que se alimenta de la tabla de catálogo Especie. La API tiene un **endpoint GET** específico (api/veterinaria/especies) para obtener esta información.
- **Estado de Salud**: Un desplegable que utiliza los valores de la tabla de catálogo EstadoSalud. Para obtener estos datos, la API utiliza el **endpoint GET** api/veterinaria/estadoSalud.
- **Dueño**: Un desplegable que muestra los usuarios existentes en el sistema para asociar la mascota con su propietario. Los datos se obtienen del **endpoint GET** api/veterinaria/usuarios.

La información se guarda en la base de datos mediante el **endpoint POST** api/veterinaria/mascotas.

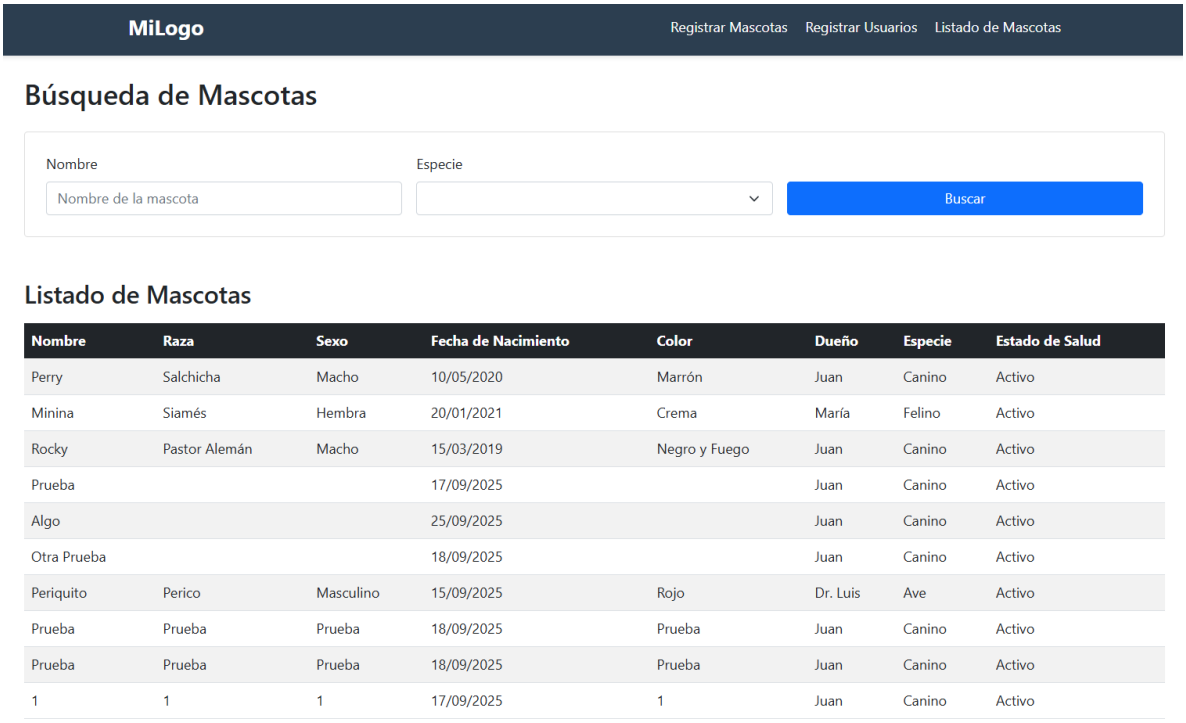


2.2. Funcionalidad de Búsqueda y Listado

La pantalla de "Búsqueda de Mascotas" ofrece la posibilidad de buscar mascotas por nombre o especie. Los resultados se muestran en un listado detallado que incluye la siguiente información:

- **Nombre**
- **Raza**
- **Sexo**
- **Fecha de Nacimiento**
- **Color**
- **Dueño**
- **Especie**
- **Estado de Salud**

Para obtener este listado de mascotas, la interfaz de usuario se comunica con el **endpoint GET** api/veterinaria/mascotas.



---

3. Estructura de la Base de Datos

El sistema utiliza una base de datos relacional para organizar la información. El esquema proporcionado incluye tablas clave para la gestión de la clínica:

Tablas de Catálogo

Estas tablas son esenciales para mantener la **consistencia de los datos** y evitar el texto libre. Funcionan como listas de opciones predefinidas para otros campos en el sistema:

- Estado: Define si un registro está activo, inactivo, pendiente, etc.
- Especialidad: Almacena las especialidades de los veterinarios.
- EstadoSalud: Contiene los posibles estados de salud de una mascota.
- Especie: Enumera las especies de animales que atiende la clínica.
- MotivosVisita: Describe las razones comunes de una visita a la clínica.
- Roles: Define los roles de los usuarios para la gestión de permisos.

Tablas Principales

Estas tablas almacenan la información central del negocio:

- Usuarios: Contiene los datos de todos los usuarios del sistema.
- Veterinarios: Extiende la tabla de Usuarios con información específica del personal veterinario, como su número de licencia y especialidad.
- Mascotas: Almacena la información de las mascotas, vinculada a un dueño de la tabla Usuarios.
- Visitas: Registra cada visita de una mascota, incluyendo el veterinario que la atendió, el motivo y los costos.
- Citas: Gestiona las citas programadas para las mascotas.

Tablas de Procedimientos y Facturación

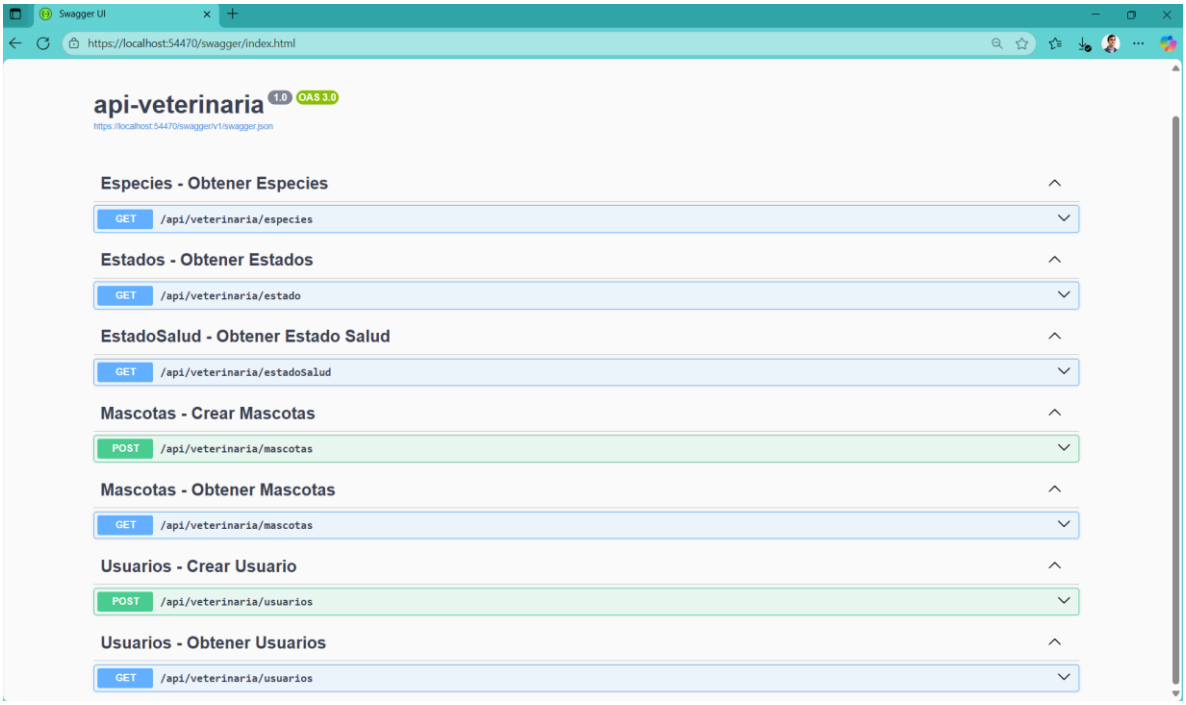
Estas tablas se encargan de registrar los servicios y tratamientos proporcionados y generar facturas:

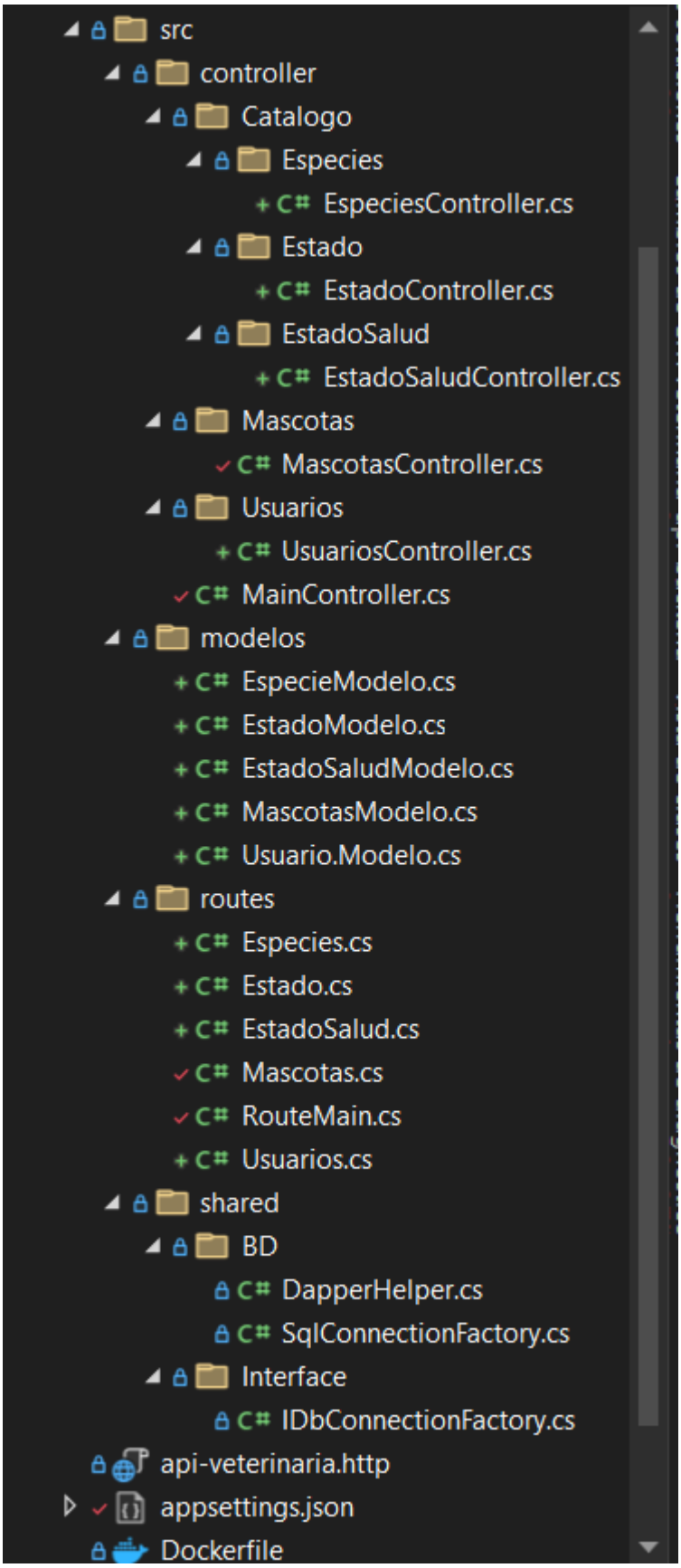
- ProcedimientosMedicos: Almacena tratamientos y vacunas, diferenciados por un campo EsVacuna.
- VisitaProcedimientos: Detalla los procedimientos realizados en cada visita.
- Servicios: Define servicios adicionales como peluquería o venta de productos.
- Facturas y DetalleFactura: Permiten la creación y el desglose de facturas para los clientes.

4. Arquitectura del Sistema

El prototipo sigue una arquitectura de **microservicios**, con una clara separación entre el **Front End** (desarrollado con Angular) y el **Back End** (implementado en C#). Los **endpoints de la API** están documentados con **Swagger**, lo que facilita la comunicación entre los componentes del sistema y el desarrollo futuro.

El diseño modular del backend, con carpetas para controller, modelos, y routes, muestra una organización limpia que promueve la escalabilidad y el mantenimiento del código. La capa de persistencia de datos está separada y utiliza DapperHelper para interactuar con la base de datos.




src
controller
Catalogo
Especies
EspeciesController.cs
Estado
EstadoController.cs
EstadoSalud
EstadoSaludController.cs
Mascotas
MascotasController.cs
Usuarios
UsuariosController.cs
MainController.cs
modelos
EspecieModelo.cs
EstadoModelo.cs
EstadoSaludModelo.cs
MascotasModelo.cs
Usuario.Modelo.cs
routes
Especies.cs
Estado.cs
EstadoSalud.cs
Mascotas.cs
RouteMain.cs
Usuarios.cs
shared
BD
DapperHelper.cs
SqlConnectionFactory.cs
Interface
IDbConnectionFactory.cs
api-veterinaria.http
appsettings.json
Dockerfile