



Desarrollo Backend NetCore Filtro Final

Introducción al Proyecto de Administración de Veterinaria

El proyecto de desarrollo de software tiene como objetivo principal la creación de un sistema de administración para una veterinaria. Este sistema permitirá a los administradores y al personal de la veterinaria gestionar de manera eficiente y efectiva todas las actividades relacionadas con la atención de mascotas y la gestión de clientes.

El sistema contará con diferentes módulos que abarcarán áreas clave como el registro de pacientes, la programación de citas, el seguimiento de tratamientos médicos, la gestión de inventario de medicamentos y productos, así como la generación de reportes y estadísticas relevantes para la toma de decisiones.

El desarrollo se realizará utilizando la tecnología NetCore Version 7.0, que proporciona un entorno robusto y escalable para la creación de aplicaciones web. Se implementarán las mejores prácticas de desarrollo de software para garantizar la calidad y fiabilidad del sistema.

El proyecto de administración de veterinaria tiene como objetivo mejorar la eficiencia y la experiencia del cliente, al tiempo que facilita la gestión interna de la veterinaria. Se espera que este sistema contribuya positivamente al crecimiento y éxito del negocio.

Requerimientos Funcionales

El proyecto de administración de veterinaria debe cumplir con los siguientes requerimientos funcionales:

1. Autenticación y autorización:
 - El sistema debe implementar protección en los endpoints utilizando JWT (JSON Web Tokens). El token tiene una duracion de 1 minuto.
 - Se debe implementar refresh token.
 - Debe restringir las peticiones a los endpoints según los roles de los usuarios.
2. Se debe permitir realizar procesos de creacion, edicion, eliminacion y listado de informacion de cada una de las tablas
3. El backend debe permitir restringir peticiones consecutivos usando tecnicas de limitacion por IP.
4. El backend debe permitir realizar la paginacion en las peticiones get de todos los controladores.
5. Los controladores deben implementar 2 versiones diferentes (Query y Header)

6. Lista

Estructura de base de datos

A continuación se muestra la estructura de la base de datos normalizada para el proyecto de administración de veterinaria:

Tabla: Usuarios

- ID_Usuario (PK)
- Nombre
- CorreoElectronico
- Contraseña

Tabla: Roles

- ID_Rol (PK)
- Nombre

Tabla: RolesUsuarios

- ID_RolUsuario (PK)
- ID_Usuario (FK)
- ID_Rol (FK)

Tabla: Citas

- ID_Cita (PK)
- ID_Mascota (FK)
- Fecha
- Hora
- Motivo
- ID_Veterinario (PK)

Tabla: TratamientosMedicos

- ID_Tratamiento (PK)
- ID_cita(FK)
- ID_medicamento
- Dosis
- FechaAdministracion
- Observacion

Tabla: Medicamento

- ID_medicamento (PK)
- Nombre
- CantidadDisponible
- Precio
- ID_Laboratorio (FK)

Tabla: MovimientoMedicamento

- ID_Entrada (PK)
- ID_Producto (FK)
- Cantidad
- Fecha

Tabla: TipoMovimiento

- ID_tipo (PK)
- Descripcion

Tabla: DetalleMovimiento

- ID_Salida (PK)
- ID_Producto (FK)
- Cantidad
- ID_MovMed
- Precio

Tabla: Propietarios

- ID_Propietario (PK)
- Nombre
- CorreoElectronico
- Teléfono

Tabla: Mascotas

- ID_Mascota (PK)
- ID_Propietario (FK)
- ID_Especie (FK)
- ID_Raza (FK)
- Nombre
- FechaNacimiento

Tabla: Especies

- ID_Especie (PK)
- Nombre

Tabla: Razas

- ID_Raza (PK)
- ID_Especie (FK)
- Nombre

Tabla: Proveedores

- ID_Proveedor (PK)
- Nombre
- Dirección

- Teléfono

Tabla: Laboratorios

- ID_Laboratorio (PK)
- Nombre
- Dirección
- Teléfono

Tabla: MedicamentosProveedores

- ID_Medicamento (PK)
- ID_proveedor

Tabla: Veterinarios

- ID_Veterinario (PK)
- Nombre
- CorreoElectronico
- Teléfono
- Especialidad

Consultas

Grupo A

- Crear un consulta que permita visualizar los veterinarios cuya especialidad sea Cirujano vascular.
- Listar los medicamentos que pertenezcan a el laboratorio Genfar
- Mostrar las mascotas que se encuentren registradas cuya especie sea felina.
- Listar los propietarios y sus mascotas.
- Listar los medicamentos que tenga un precio de venta mayor a 50000
- Listar las mascotas que fueron atendidas por motivo de vacunacion en el primer trimestre del 2023

Grupo B

- Listar todas las mascotas agrupadas por especie.
- Listar todos los movimientos de medicamentos y el valor total de cada movimiento.
- Listar las mascotas que fueron atendidas por un determinado veterinario.
- Listar los proveedores que me venden un determinado medicamento.
- Listar las mascotas y sus propietarios cuya raza sea Golden Retriever
- Listar la cantidad de mascotas que pertenecen a una raza a una raza. Nota: Se debe mostrar una lista de las razas y la cantidad de mascotas que pertenecen a la raza.

Rubrica de Evaluacion



Estructura adecuadamente el proyecto teniendo en cuenta el modelo de 4 o 3 tres capas.

Nada : El proyecto no contiene la estructura adecuada de acuerdo a los requerimientos del proyecto. (5ptos)

Mucho: El proyecto contiene la estructura que permite cumplir con el funcionamiento adecuado del backend. (100 ptos)



Implementa el sistema de paginacion en los endpoint de tipo get, permitiendo optimizar las respuestas del servidor.

Nada: Construye las clases para la implementacion de la paginacion pero no se aplican a ningun endpoint. (5ptos)

Poco: Construye las clases para la implementacion de la paginacion y lo aplica a 7 controladores del backend. (40ptos)

Medio: Construye las clases para la implementacion de la paginacion y lo aplica a 10 controladores del backend. (60ptos)



Genera los repositorios para cada uno de los controladores y sus endpoints.

Nada : No genera ningun repositorio que permita cumplir con cada uno de los requerimientos del proyecto. (5ptos)

Poco: Realiza el 10% de los repositorios solicitados en los requerimientos del proyecto. (20ptos)

Medio: Cumple con el 50% de los repositorios del proyecto teniendo en cuenta los requerimientos del proyecto. (60 ptos)

Mucho : Realiza todos los repositorios necesarios para cumplir con los requerimientos del proyecto y funcionan correctamente. (100ptos)



Crear los metodos de extension que permite aplicar el control de peticiones automatizadas.

Mucho : Construye las clase para la implementacion de todos los controladores del backend.(100ptos)



Implementa las consultas solicitadas en el documento de requerimientos.

Nada: No implementa ninguna consulta de las solicitadas en el documento de requerimientos (0ptos)

Poco : Implementa la mitad de las consultas y su ejecucion es correcta. (50ptos)

Mucho: Implementa todas las consultas y su ejecucion es correcta (100ptos)

Nota. Las consultas no se pueden realizar en los controladores. Se debe documentar en el readme cada una de las peticiones realizadas desde Thunder client, Insomnia o Postman. Si no hay documentacion en el readme el valor obtenido tendra una penalidad de -50%

Nada : Genera el metodo de extension sin obtener un funcionamiento correcto al momento de realizar peticiones de forma consecutiva. (5Ptos)

Mucho : Genera el metodo de extension permitiendo bloquear las peticiones consecutivas al mismo endPoint (100ptos)