SENA

Servicio Nacional de Aprendizaje Sena

Proyecto Vet site

Juan David Castro Marín

Evelyn Manuela Mosquera Ramírez

Luna Luzero Osorio Marín

Instructora:

Juan Carlos

**Introducción**

Este documento describe la implementación de un programa de software que se centra en la gestión de recursos humanos y la gestión de registros de pacientes en una clínica veterinaria. Este proyecto nació del deseo de optimizar y centralizar los procedimientos veterinarios para preparar y evaluar mejor los problemas de salud animal. Para este desarrollo se utilizó el diseño MVC de .NET 8, basado en su robustez para la creación de aplicaciones escalables y flexibles. El proyecto está organizado en módulos que separan las responsabilidades de front-end y back-end, lo que proporciona una mayor flexibilidad en el desarrollo y el mantenimiento.  
  
El sistema se desplegará en la nube a través de Azure App Service y la implementación de bases de datos en Azure SQL. La solución combina diferentes tecnologías como Bootstrap para proporcionar una interfaz moderna y fácil de usar y GitHub para administrar el código fuente. El proyecto no sólo se centra en proporcionar buena información a los usuarios finales, sino también en garantizar la integridad y seguridad de los datos almacenados mediante una alta autenticación y control de acceso.  
  
**Objetivo General**

Desarrollar una aplicación web modular en .NET 8 que permita la gestión de citas y el registro del historial clínico de pacientes en un centro veterinario, optimizando la administración de procesos y facilitando el acceso a información crítica de manera segura y eficiente.

**Objetivos Específicos**

1. Implementar un sistema de gestión de citas que permita a los usuarios programar, modificar y cancelar citas para sus mascotas a través de una interfaz intuitiva y responsive.
2. Desarrollar un módulo para la administración del historial clínico de pacientes, permitiendo el registro de diagnósticos, tratamientos, vacunación y desparasitaciones por parte del personal veterinario.
3. Integrar un sistema de notificación automática por correo electrónico para recordar citas programadas y eventos importantes, mejorando la experiencia del usuario y reduciendo la tasa de ausencias en las citas.

**1. Funcionalidades del Producto**

El software desarrollado tiene como objetivo gestionar el ciclo completo de atención veterinaria, incluyendo la programación de citas, el registro de historial clínico, y la notificación de eventos importantes a través de correo electrónico. A continuación, se detallan las funcionalidades principales:

* **Gestión de citas**: Los usuarios podrán agendar, modificar y cancelar citas para sus mascotas de manera eficiente.
* **Notificación por correo electrónico**: El sistema enviará recordatorios automáticos a los dueños de las mascotas con detalles sobre las citas programadas.
* **Historial clínico**: Se mantendrá un registro detallado del historial clínico de cada paciente, permitiendo a los veterinarios acceder y actualizar información sobre tratamientos, vacunas, cirugías, y desparasitaciones.
* **Control de usuarios y roles**: Habrá un sistema de autenticación y autorización que permitirá el acceso al sistema de manera segura, asignando roles a los usuarios para limitar las acciones según sus permisos.
* **Gestión de pacientes**: La aplicación permitirá el registro y la consulta de información detallada sobre las mascotas (pacientes), incluyendo datos personales, historial de vacunación y otros tratamientos.
* **Modularidad del sistema**: La aplicación se dividirá en dos módulos principales: uno para la gestión de citas y otro para la historia clínica de los pacientes.

**2. Requerimientos Funcionales**

Los requerimientos funcionales definen las acciones que los usuarios pueden realizar en el sistema:

* **Registro y autenticación de usuarios**: Los usuarios deben poder registrarse y autenticarse en el sistema de forma segura mediante contraseñas encriptadas.
* **Agendamiento de citas**: Los dueños de mascotas deben poder ver las citas disponibles y agendar una nueva, seleccionar la fecha y hora, así como el veterinario asignado.
* **Registro del historial clínico**: Los veterinarios deben tener acceso al historial de cada paciente y poder agregar información sobre tratamientos, medicamentos y resultados de exámenes.
* **Notificaciones automáticas**: El sistema debe enviar correos electrónicos automáticos a los dueños de mascotas antes de las citas o cuando se complete un tratamiento importante.
* **CRUD de usuarios y roles**: Los administradores deben poder gestionar usuarios, creando, editando y eliminando cuentas, asignando roles de acuerdo con los permisos requeridos.
* **Respaldo de datos**: El sistema debe ofrecer una copia de seguridad periódica de la información almacenada, garantizando que no se pierdan datos críticos.

**3. Requerimientos No Funcionales**

* **Seguridad**: Se utilizará autenticación con JWT y contraseñas encriptadas para garantizar la seguridad en el acceso al sistema.
* **Rendimiento**: El sistema debe ser capaz de manejar múltiples usuarios simultáneamente sin afectar el tiempo de respuesta.
* **Escalabilidad**: El software se desarrollará con la posibilidad de escalar en el futuro a medida que aumente la cantidad de usuarios.
* **Compatibilidad**: Se utilizará Bootstrap para garantizar que la interfaz sea responsive y compatible con diferentes tamaños de pantallas.

**4. Entorno Operativo**

El entorno en el que se desarrollará y desplegará el sistema es el siguiente:

* **Framework**: .NET 8 será utilizado para el desarrollo de la aplicación web en MVC (Model-View-Controller).
* **Frontend y Backend en Proyectos Separados**: Se organizarán dos proyectos separados para gestionar el frontend y el backend.
* **Lenguajes**: Se utilizará C# para el backend y Bootstrap para el diseño del frontend.
* **Servicios en la nube**: La aplicación será publicada en **Azure App Service**, mientras que la base de datos estará alojada en **Azure SQL Database**.
* **Control de versiones**: Se utilizará **GitHub** como repositorio para almacenar el código fuente, garantizando un control de versiones adecuado durante el desarrollo del proyecto.
* **Herramientas de desarrollo**: **Visual Studio 2022** será la herramienta principal utilizada para el desarrollo de la aplicación.
* **Base de datos**: Se utilizará **SQL Express** para pruebas locales y **SQL Management Studio** para gestionar y monitorear la base de datos tanto local como en la nube.

**Conclusiones**

1. El desarrollo de un sistema modular utilizando .NET 8 y la arquitectura MVC permitió la separación clara de las responsabilidades entre el frontend y el backend, lo que facilita la escalabilidad y el mantenimiento del software a largo plazo.
2. El uso de tecnologías en la nube como Azure App Service y Azure SQL Database garantiza una alta disponibilidad y seguridad para la información sensible almacenada en el sistema, cumpliendo con los requerimientos operativos de la aplicación.
3. La implementación de notificaciones automáticas por correo electrónico no solo mejora la experiencia de los usuarios, sino que también optimiza la eficiencia operativa del centro veterinario al reducir las ausencias en las citas programadas.
4. La integración de herramientas como Bootstrap y GitHub favoreció un desarrollo más ágil y colaborativo, permitiendo una implementación flexible y adaptable a futuros requerimientos