

Sistema Veterinaria San Miguel

Contexto

La **Veterinaria San Miguel** necesita un sistema de consola en **C#** que permita gestionar sus **clientes, mascotas, veterinarios y las atenciones médicas** realizadas.

El sistema debe usar **Entity Framework Core** con **MySQL** como base de datos.

Toda la información debe almacenarse y consultarse directamente desde la base de datos.

El objetivo es simular el proceso completo de registro, consulta y atención médica de las mascotas, aplicando los conceptos de **POO** vistos en clase y aprendiendo cómo integrar una base de datos con **EF Core**.

Menú Principal

Gestión de Clientes

- Registrar cliente
- Listar clientes
- Editar cliente
- Eliminar cliente

Gestión de Mascotas

- Registrar mascota
- Listar mascotas
- Editar mascota
- Eliminar mascota

Gestión de Veterinarios

- Registrar veterinario
- Listar veterinarios
- Editar veterinario
- Eliminar veterinario

Gestión de Atenciones Médicas

- Registrar atención médica (fecha, diagnóstico, mascota y veterinario)
- Listar atenciones médicas

- Editar atención médica
- Eliminar atención médica

Historial Médico

- Mostrar todas las atenciones médicas realizadas a una mascota determinada.

Consultas Avanzadas (LINQ sobre EF)

- Consultar todas las mascotas de un cliente.
- Consultar el veterinario con más atenciones realizadas.
- Consultar la especie de mascota más atendida en la clínica.
- Consultar el cliente con más mascotas registradas.

Salir

Espacio para aplicar POO

Los estudiantes deberán identificar las entidades principales del sistema (**Cliente, Mascota, Veterinario, Atención**) y representarlas como clases con atributos y relaciones.

El sistema debe construirse manipulando **objetos y colecciones de EF**, no datos aislados.

Cada clase se reflejará en una tabla en la base de datos mediante **Entity Framework**.

Además, deberán aplicar el concepto de **sobrecarga de métodos**, permitiendo que una misma operación pueda ejecutarse de diferentes formas.

Espacio para diagramas UML

Antes de escribir código, los equipos deben diseñar:

1. **Diagrama de Casos de Uso:** mostrar cómo interactúa el usuario con el sistema (registrar cliente, registrar mascota, registrar atención, etc.).
2. **Diagrama de Clases:** representar las clases necesarias con sus atributos, métodos y relaciones (Cliente tiene Mascotas, Mascota tiene Atenciones, Atenciones son realizadas por un Veterinario).

Flujo esperado

1. El sistema inicia mostrando un mensaje de bienvenida.
2. El usuario interactúa con el menú principal.
3. Puede acceder a los diferentes submenús (**Cientes, Mascotas, Veterinarios, Atenciones, Consultas**) para realizar operaciones.

4. El sistema se mantiene en ejecución hasta que el usuario seleccione la opción **Salir**.
-

Requisitos técnicos

- Implementar en **C# consola**.
 - Usar **Entity Framework Core con MySQL** para almacenar y consultar información.
 - Aplicar ciclos (`while`, `foreach`) en la navegación de menús.
 - Usar **LINQ sobre EF** para consultas avanzadas.
 - Aplicar **POO** para la definición y relación de entidades.
 - Implementar al menos **una sobrecarga de métodos** en alguna de las clases del sistema.
-

Entregables

- Implementación del sistema **Veterinaria San Miguel** en **C# consola**.
- Base de datos creada con **migraciones EF Core**.
- Justificación de cómo se aplicó **POO**.
- Ejemplo de **sobrecarga de métodos** dentro de alguna clase.
- Diagrama de Clases UML.
- Diagrama de Casos de Uso UML.