

UNIVERSIDAD PRIVADA DE TACNA

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *“Sistema Web de Gestión Veterinaria para Mascotas”***

Curso: *Programación Web II*

Docente: *Ing. Patrick Cuadros Quiroga*

|  |  |
| --- | --- |
| Integrantes: |  |
| ***Ccalli Chata, Joel Robert*** | ***(2017057528)*** |
| ***Apaza Mamani, Edward Hernan*** | ***(2018060915)*** |

**Tacna – Perú**

***2025***

# Estándar de Programación

1. **Introducción**

Este documento establece las normas y estándares de programación para garantizar la calidad, mantenibilidad y escalabilidad del código en los proyectos de software.

# Convenciones de Nomenclatura

* 1. **Clases y Objetos**
     + Usar PascalCase para nombres de clases y objetos.
     + Ejemplo:
     + public class ClienteService {
     + // Código
     + }

# Variables y Métodos

* + - Usar camelCase para variables y nombres de métodos.
    - Ejemplo:
    - public string obtenerNombreCompleto() {
    - string nombreCompleto = "Juan Perez";
    - return nombreCompleto;
    - }

# Constantes

* + - Usar UPPER\_CASE para constantes.
    - Ejemplo:
    - public const int MAX\_INTENTOS = 5;

# Base de Datos

* + - Usar PascalCase para nombres de tablas y columnas.
    - Ejemplo:
    - CREATE TABLE Usuarios (
    - UsuarioID INT PRIMARY KEY,
    - Nombre VARCHAR(50),
    - Apellido VARCHAR(50)
    - );

# Organización del Código

* 1. **Estructura del Proyecto**

El proyecto debe seguir la arquitectura MVC con una estructura ordenada: Proyecto

│── Proyecto.sln

│── Proyecto.Web (Capa de Presentación)

│── Proyecto.Core (Lógica de Negocio)

│── Proyecto.Infrastructure (Acceso a Datos)

│── Proyecto.Tests (Pruebas Unitarias)

# Comentarios y Documentación

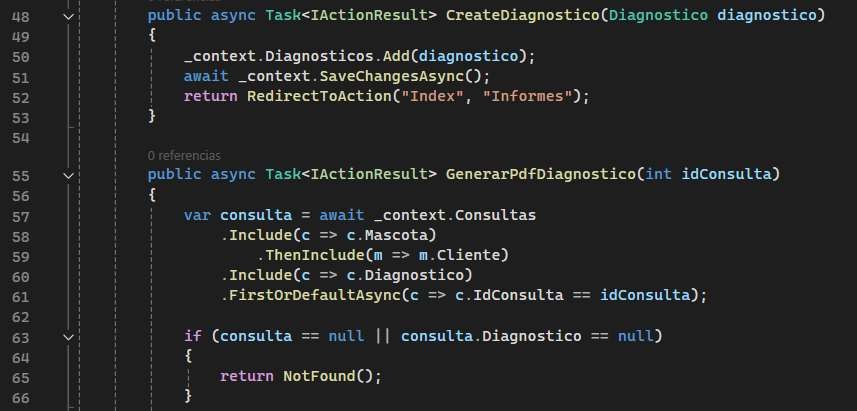
* Usar comentarios XML en C#.
* Ejemplo:
* /// <summary>
* /// Obtiene una lista de usuarios activos.
* /// </summary>
* /// <returns>Lista de usuarios</returns>
* public List<Usuario> ObtenerUsuariosActivos() {
* return usuarios.Where(u => u.Activo).ToList();
* }

# Manejo de Errores

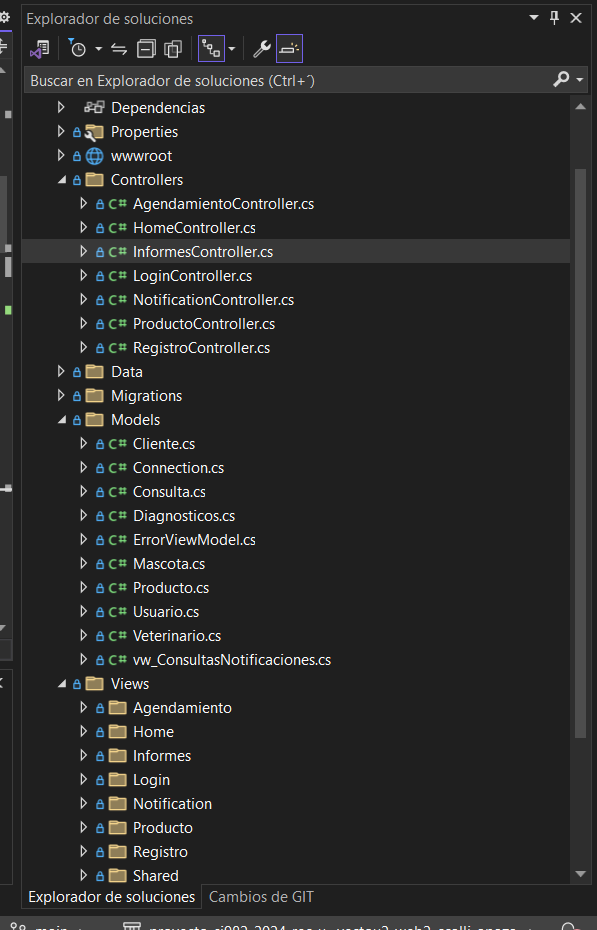
* Usar try-catch para capturar excepciones.
* Ejemplo:
* try {
* int resultado = CalcularDivision(10, 0);
* } catch (Exception ex) {
* Console.WriteLine("Error: " + ex.Message);
* }

# Seguridad

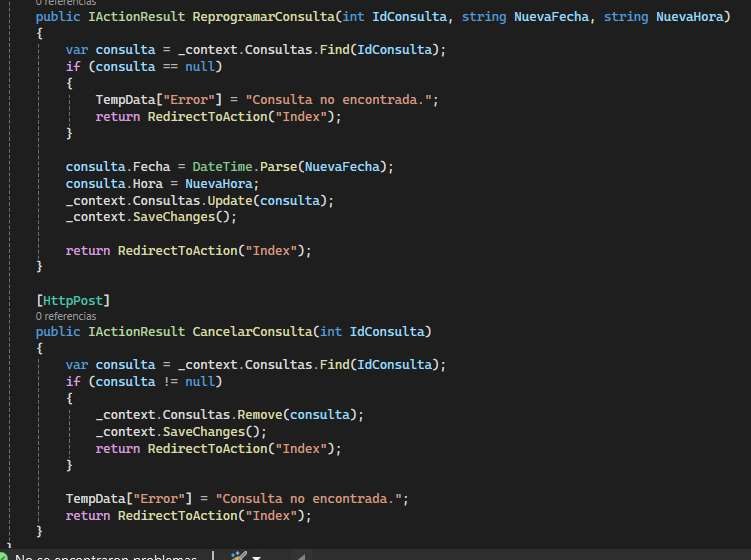
* No almacenar contraseñas en texto plano.
* Usar appsettings.json para configuraciones sensibles:
* {
* "ConnectionStrings": {
* "DefaultConnection": "Server=myserver;Database=mydb;User Id=myuser;Password=mypassword;"
* }
* }

Controlador: Informes

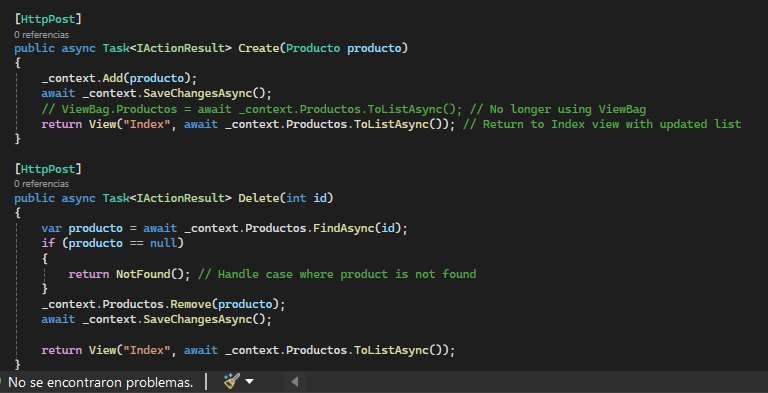
Estructura de Proyecto:



Controlador: Agendamiento



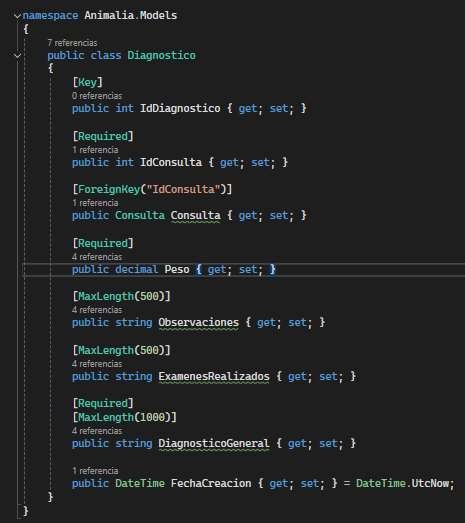
Controlador: Producto



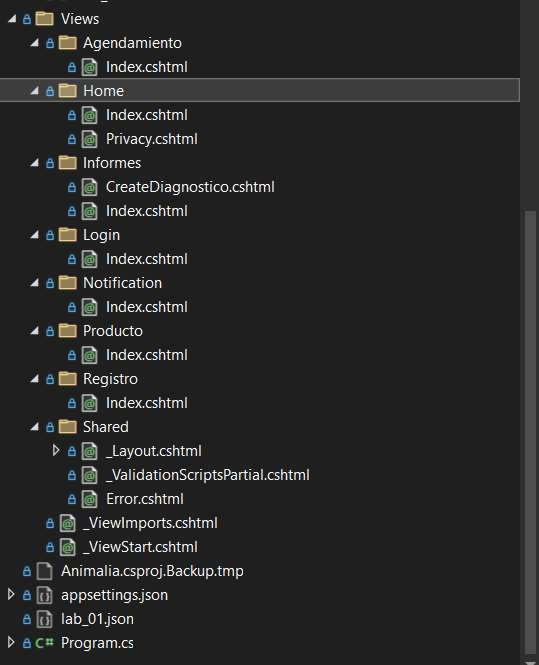
Modelo: Mascota



Modelo: Diagnostico



Vistas:



Vista General: HOME

