

Introducción a VETERINARIA CATDOG

Veterinaria Catdog es una aplicación desarrollada en C# que ofrece soluciones integrales para la gestión de información relacionada con veterinarias y sus clientes. Diseñado para facilitar las operaciones diarias, el programa abarca una variedad de características clave:

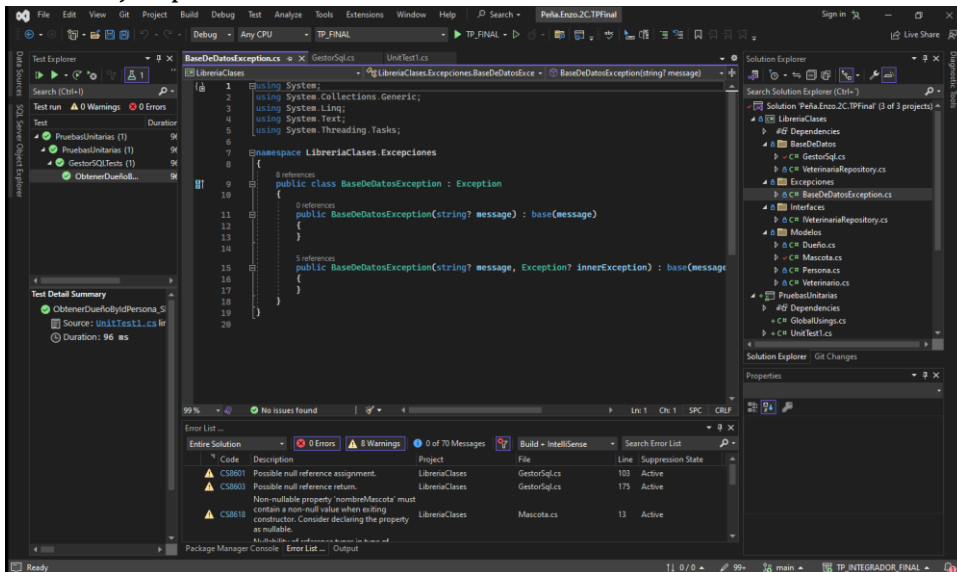
Principales Características:

- **Gestión de Mascotas:**
 - Registro detallado de mascotas, incluyendo nombre, fecha de nacimiento, historial médico y vacunas aplicadas.
- **Clientes y Usuarios:**
 - Administración de la información de los clientes y sus mascotas asociadas.
- **Interfaz Amigable:**
 - Una interfaz de usuario intuitiva que facilita la navegación y el acceso rápido a las funciones esenciales.
- **Conexión a Base de Datos:**
 - Integración con bases de datos SQL Server para un almacenamiento seguro y eficiente de la información.
- **Funcionalidades Avanzadas:**
 - Uso de conceptos avanzados de programación, como excepciones, pruebas unitarias, generics, interfaces y eventos para garantizar un rendimiento robusto y una experiencia de usuario optimizada.
- **Pruebas Unitarias:**
 - Implementación de pruebas unitarias para verificar y validar la funcionalidad del código, garantizando la calidad del software.
- **Seguridad y Confiabilidad:**
 - Enfoque en la seguridad y la confiabilidad de los datos, respaldado por el manejo adecuado de excepciones y la conexión segura a la base de datos.

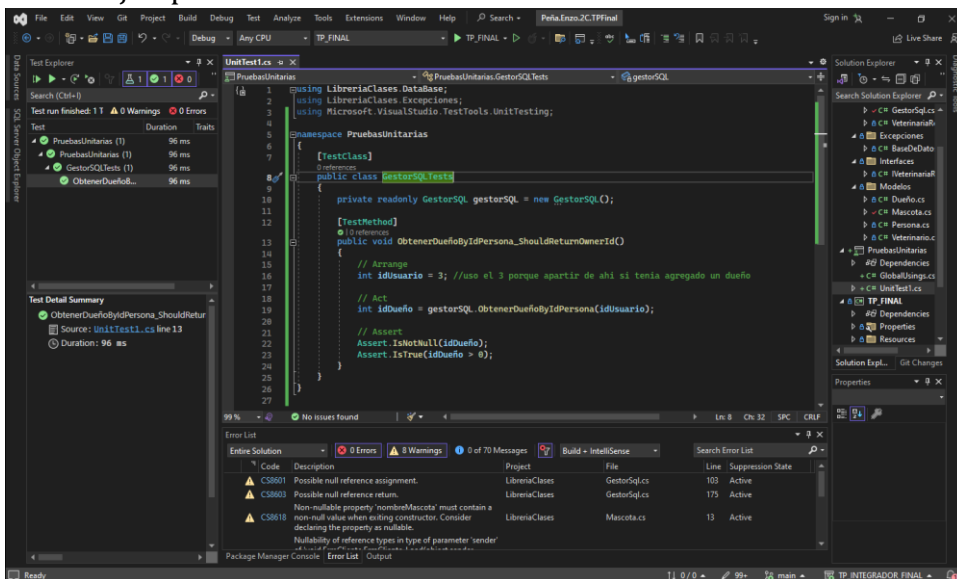
Temas

- **Excepciones**
- Definición
- Manejo de excepciones

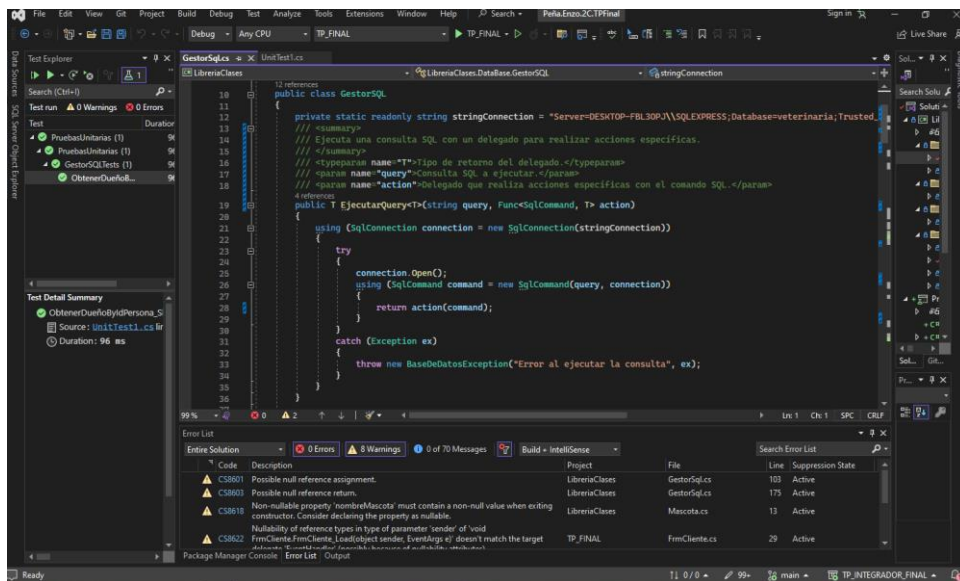
- Ejemplo



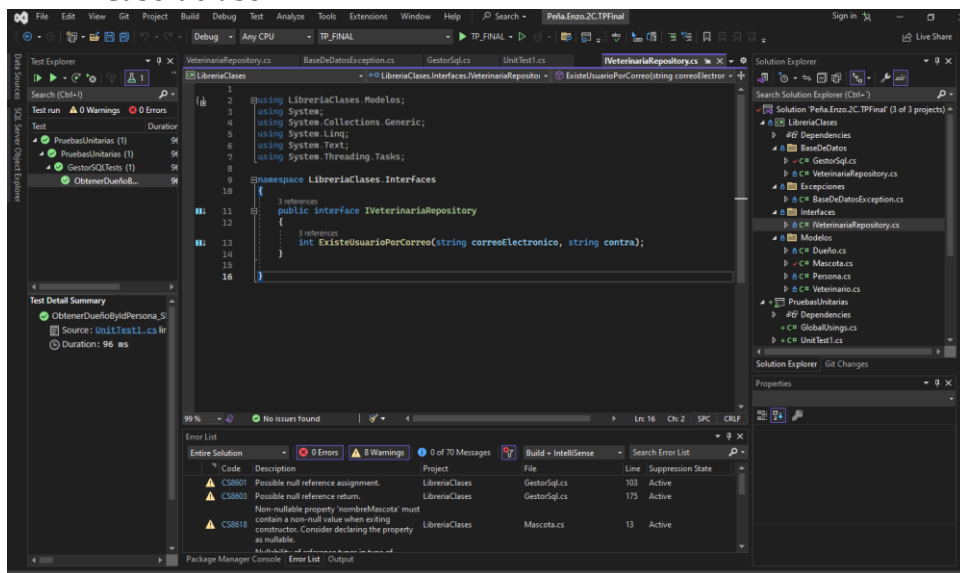
- Pruebas Unitarias
- Importancia de las pruebas unitarias
- Herramientas populares
- Ejemplo



- Generics
- Concepto de generics
- Ventajas y uso
- Ejemplo



- Interfaces
- Caso de uso



- Archivos y Serialización
- Manipulación de archivos
- Serialización y deserialización
- Ejemplo

```

1 reference
private void button1_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Archivos XML (*.xml)|*.xml";
    saveFileDialog.Title = "Guardar archivo XML";

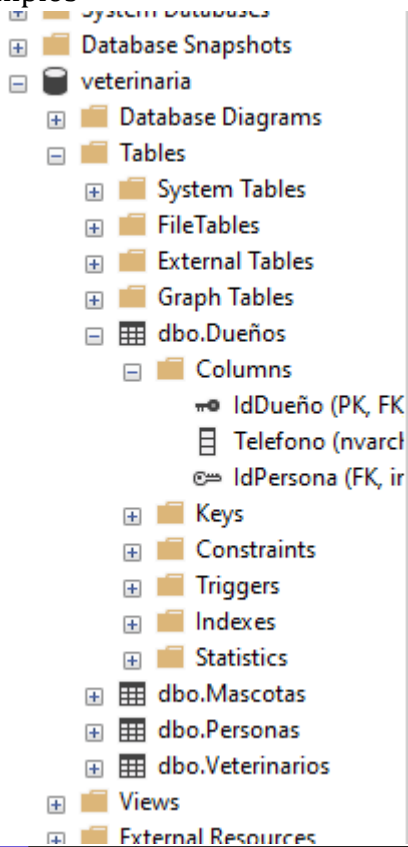
    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        string rutaArchivo = saveFileDialog.FileName;

        XmlSerializer serializer = new XmlSerializer(typeof(List<Mascota>));
        using (TextWriter writer = new StreamWriter(rutaArchivo))
        {
            //SERIALIZACION XML PARA GUARDAR EL OBJETO
            serializer.Serialize(writer, mascotas);
        }

        MessageBox.Show("Archivo XML guardado exitosamente.", "Éxito", MessageBoxButtons.OK);
    }
}

```

- **Introducción a SQL y Conexión a Bases de Datos**
- Conceptos básicos de SQL
- Conexión a bases de datos desde C#
- Ejemplos



```

// LibreriaClases
// FrmCliente.cs
// FrmCliente.cs [Design]
// LibreriaClases.DataBase.GestorSQL
// stringConnection

4 using System.Collections.Generic;
5 using System.Data.SqlClient;
6 using System.Diagnostics.Contracts;
7
8 namespace LibreriaClases.DataBase
9 {
10     12 references
11     public class GestorSQL
12     {
13         private static readonly string stringConnection = "Server=DESKTOP-FBL30PJ\\SQLEXPRESS;Database=veterinaria;Trusted_Conne
14         /// <summary>
15         /// Ejecuta una consulta SQL con un delegado para realizar acciones especificas.
16         /// </summary>
17         /// <typeparam name="T">Tipo de retorno del delegado.</typeparam>
18         /// <param name="query">Consulta SQL a ejecutar.</param>
19     }
20 }

```

```

FrmCliente.cs | FrmCliente.cs [Design]
LibreriaClases.DataBase.GestorSQL
stringConnection

/// <param name="id">ID de persona.</param>
/// <returns>ID del dueño.</returns>
4 references | 1/1 passing
public int ObtenerDueñoByIdPersona(int id)
{
    using (SqlConnection connection = new SqlConnection(stringConnection))
    {
        string query = $"SELECT IdDueño FROM Dueños WHERE IdPersona = {id}";

        try
        {
            connection.Open();
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                using (SqlDataReader reader = command.ExecuteReader())
            }
        }
    }
}

```

- Delegados y Expresiones Lambda
- Uso de delegados
- Sintaxis de expresiones lambda
- Ejemplos

```

if (comboBox1.SelectedItem is Mascota mascotaSeleccionada)
{
    ObjetoDelegado delegado = new ObjetoDelegado(Mascota.EspecieDescripcion);
    string informacionMascota = ObtenerInformacionMascota(mascotaSeleccionada);
    bool calendarioVacunas = mascotaSeleccionada.estadoCalendario;
    bool enfermo = mascotaSeleccionada.estadoEnfermo;

    string queryInsertUsuario = $"INSERT INTO Mascotas (Nombre, Especie, FechaNacimiento,
gestorSQL.EjecutarQuery(queryInsertUsuario, command => command.ExecuteNonQuery());

    MessageBox.Show("Mascota agregada correctamente con veterinario asignado", "Éxito",

    bool calendarioVacunas = mascotaSeleccionada.estadoCalendario;
    bool enfermo = mascotaSeleccionada.estadoEnfermo;
    bool altaMedica = mascotaSeleccionada.altaMedica;
    string buscarSectorVeterinario = "select Especialidad from Veterinarios where IdVeterinario ="+
    string sectorVeterinario = gestorSQL.EjecutarQuery(buscarSectorVeterinario, command =>
    {
        using (SqlDataReader reader = command.ExecuteReader())
        {
            if (reader.Read())
            {
                return reader["Especialidad"].ToString();
            }
            return null;
        }
    });
    lblCalendario.Text = $"Calendario de vacunas: {(calendarioVacunas ? "Completo" : "Incompleto")}";
    lblEnfermo.Text = $"Estado de salud: {(enfermo ? "Enfermo" : "Sano")}";
    lblAlta.Text = $"Estado de alta médica: {(altaMedica ? "Dado de alta" : "No dado de alta")}";
    lblSectorVeterinario.Text = $"Sector Veterinario: {sectorVeterinario}";
}

```

- Programación Multi-hilo y Concurrencia
- Conceptos básicos de multi-hilo
- Concurrencia en C#
- Ejemplos

```

1 reference
private async Task CargarMascotasAsync()
{
    try
    {
        await Task.Run(() =>
        {
            int idDueño = gestorSQL.ObtenerDueñoByIdPersona(idUsuario);
            mascotas = gestorSQL.ObtenerMascotasByIdPersona(idDueño).ToList();
        });
    }
    catch (BaseDatosException ex)
    {
        MessageBox.Show($"Error al cargar mascotas desde la base de datos: {ex.Message}", "Error", MessageBoxButtons.OK);
    }
}

1 reference
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)

```

- **Eventos**
- Uso de eventos en C#
- Implementación
- Ejemplo

```

    }

    1 reference
    private void btnAgregarMascotaNueva_Click(object sender, EventArgs e)
    {
        try
        {
            Random random = new Random();
            string nombre = txtBoxNombreMascotaNueva.Text;
            string especie = txtBoxEspecieMascotaNueva.Text;
            DateTime fNacimiento = dateTimePicker1.Value;
            bool altaMedica = false;
            bool vacunasAplicadas = false;
            bool estadoEnfermo = true;
            //ya que no tengo el tiempo para generar una logica especifica para asignar un doctor
            //prefiero simplemente asignarle uno de los dos veterinarios y listo
            int idVeterinario = random.Next(1, 3);
        }
    }

```

- **Métodos de Extensión**
- Definición y ventajas
- Sintaxis en C#
- Casos prácticos
- Ejemplo

```

    RellenarComboBoxMascotas();
}

// Método de extensión que obtiene la información de la mascota
1 reference
public static string ObtenerInformacionMascota(Mascota mascota)
{
    return $"Nombre de mascota seleccionada: {mascota.nombreMascota}";
}

1 reference
private void button2_Click(object sender, EventArgs e)
{
    FrmMascota frmMascota = new FrmMascota(idUsuario, this);
}

```