



La aplicación es para que una veterinaria pueda llevar registro de sus clientes con sus mascotas y guardar el historial médico por el que fueron a la veterinaria.

Puede guardar los datos del cliente (Nombre, Dirección y Teléfono) y los datos de cada una de sus mascotas (Tipo de animal, Nombre, Edad y Raza) y exportar los datos por si el cliente quiere quedarse con el registro.


#### FrmIngreso:

Entrada a la app, se puede registrar un nuevo cliente(Dueño), cargar un cliente ya ingresado o salir de la app.

Botón “Nuevo”: abre FrmDuenio

Botón “Cargar”: abre FrmCargar

Botón “Salir”: sale de la app

Botón  : Prende la música y hace mover el lbl de abajo o apaga la música y el cartel se queda estático

**FrmDuenio:**

### Nuevo Dueño

**Nombre:**

**Telefono:**

**Dirección:**

Si se eligió la opción de ingresar un nuevo cliente entra a este formulario para llenar los datos.

Botón “Agregar”: agrega dueño a la lista

### Modificar Dueño

**Nombre:**

**Telefono:**

**Dirección:**

Si se eligió la opción de modificar del formulario FrmCargar.

Botón “Modificar”: modifica al dueño

**FrmCargar:**

## Cargar Dueño

**Buscar:**

Nombre: Stephen King - Telefono: 42513659 - Dirección: Misery 4522 - Cantidad de mascotas: 1  
Nombre: Will Smith - Telefono: 458753254 - Dirección: Estados Unidos 7775 - Cantidad de mascotas: 1  
Nombre: Madonna - Telefono: 4501254 - Dirección: Londres 653 - Cantidad de mascotas: 1  
Nombre: Goku Kakarotto - Telefono: 45875621 - Dirección: Namekusei 123 - Cantidad de mascotas: 1  
Nombre: Pepe Argeto - Telefono: 41203590 - Dirección: Av. Santa Fe 5642 - Cantidad de mascotas: 1  
Nombre: Ash Ketchup - Telefono: 48885201 - Dirección: Pueblo Paleta 4452 - Cantidad de mascotas: 1  
Nombre: Steven Spielberg - Telefono: 41252120 - Dirección: Estados Unidos 1253 - Cantidad de mascotas: 2  
Nombre: Harry Potter - Telefono: 42012365 - Dirección: Hogwards 9 3/4 - Cantidad de mascotas: 1  
Nombre: Super Mario - Telefono: 42012548 - Dirección: Mushroom Kingdom 4650 - Cantidad de mascotas: 1  
Nombre: Neo el Elegido - Telefono: 42135458 - Dirección: Matrix 4567 - Cantidad de mascotas: 2

Borrar

Modificar

Cargar

Atras

En este form va a aparecer un ListBox con un listado de todos los clientes en la base de datos.

Se puede buscar un cliente específico escribiendo su nombre en el TextBox de arriba.

Seleccionando un cliente del ListBox se lo puede eliminar tocando el botón borrar, modificar con el botón modificar y cargar para ver los datos del cliente.

Botón “Borrar”: Hace una baja lógica del dueño

Botón “Modifica”: Abre FrmDuenio para modificar los datos

Botón “Cargar”: Abre FrmMenuPrincipal para ver los datos del dueño

Botón “Atrás”: vuelve al form FrmIngreso

## FrmMenuPrincipal:

### Stephen King

Dueño

**Teléfono:** 42513659**Dirección:** Misery 4522

**Mascota:**

Nombre: Cujo - Tipo: Perro - Edad: 5 - Raza: Bulldog

Nueva Mascota

Modificar

Borrar

Mascota

**Tipo de animal:** Perro**Edad:** 5

**Nombre:** Cujo**Raza:** Bulldog

Historial

Fecha: 19/6/2022

tiene rabia

Nueva Entrada

Exportar Mascota

Atras

Después de crear un nuevo cliente en el FrmIngreso, o después de seleccionar un cliente de la lista en FrmCargar se va a abrir este form automáticamente.

Se van a ver los datos del cliente y si tiene cargado al menos una mascota se mostrará por default la primera de la lista.

En el ComboBox se puede elegir una mascota y modificarla o borrarla con sus respectivos botones.

Botón "Nueva Mascota": abre el form FrmMascota

Botón "Modificar": abre el form FrmMascota

Botón "Borrar": pregunta si está de acuerdo y borra la mascota de forma lógica

Botón "Nueva Entrada": abre el form FrmHistorial

Botón "Exportar Mascota": exporta los datos del dueño y mascota en un archivo de texto donde desee el usuario.

Botón "Atrás": vuelve al form FrmIngreso

## FrmMascota:

### Nueva Mascota

**Tipo:**

**Nombre:**

**Edad:**

**Raza:**

Se puede llegar a este form desde el FrmMenuPrincipal apretando el botón “nueva mascota”.

Se pueden elegir un tipo de animal dentro de 3 opciones: gato, perro, ñandú. Y se pueden completar los datos nombre, edad y raza.

### Modificar Mascota

**Tipo:**

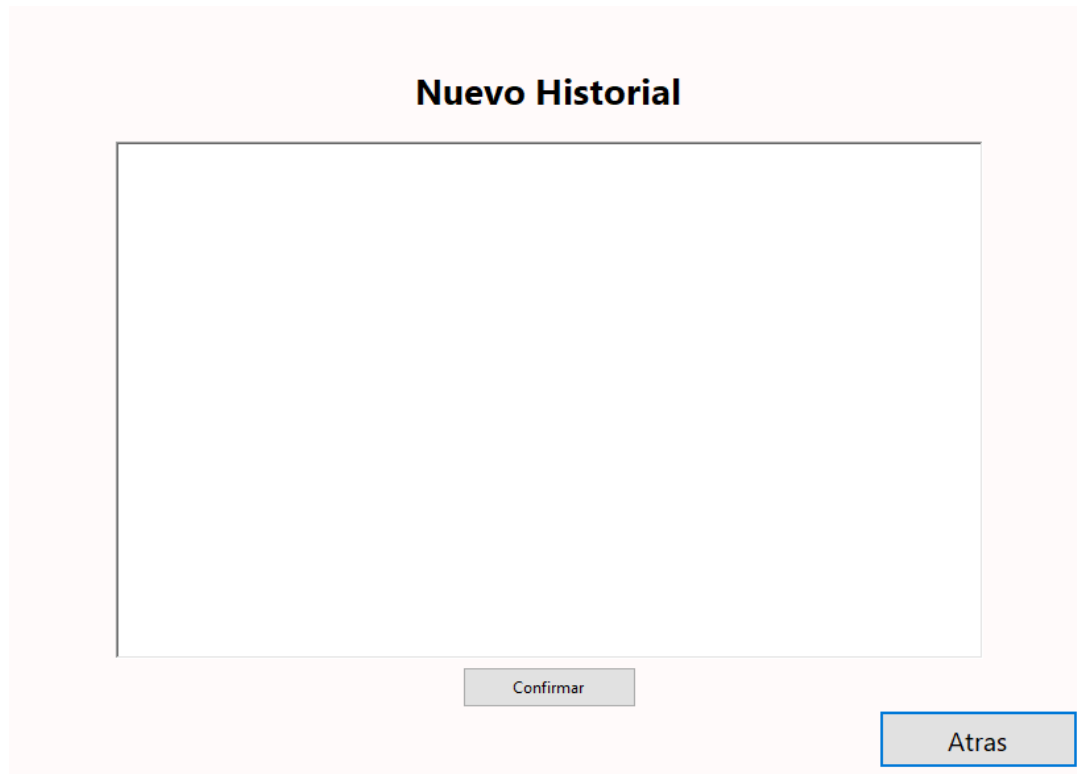
**Nombre:**

**Edad:**

**Raza:**

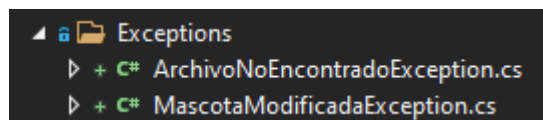
Se puede llegar a este form desde el FrmMenuPrincipal apretando el botón “modificar”.

### FrmHistorial:



Agregar historial al animal previamente seleccionado, y a ésta se le agregará la fecha del día de forma automática.

### Excepciones:



“ArchivoNoEncontradoException”: Se la arroja dentro de la clase “Serializador” en las funciones que manejan los archivos xml y json.

```
public void CargarListaJson(string path, string nombreArchivo)
{
    string pathCompleto = path + $"\\{nombreArchivo}.json";
    if (File.Exists(pathCompleto))
    {
        using (StreamReader reader = new StreamReader(pathCompleto))
        {
            string serializer = reader.ReadToEnd();
            this.lista = JsonSerializer.Deserialize<List<T>>(serializer);
        }
    }
    else
    {
        throw new ArchivoNoEncontradoException();
    }
}
```

se las ataja en el form “FrmIngreso” en el load y el formClosing

```

    catch (ArchivoNoEncontradoException)
    {
        MessageBox.Show("ERROR!!! No se pudieron encontrar los archivos");
    }
    catch (Exception ex)

```

“MascotaModificadaException”: Se la arroja dentro de la clase “MascotaDAO” en la función CambiarFormatoTipoAnimal

```

static TipoAnimal CambiarFormatoTipoAnimal(string tipo)
{
    TipoAnimal retorno;

    switch (tipo)
    {
        case "Gato":
            retorno = TipoAnimal.Gato;
            break;

        case "Perro":
            retorno = TipoAnimal.Perro;
            break;

        case "Ñandú":
            retorno = TipoAnimal.Ñandú;
            break;
        default:
            throw new MascotaModificadaException();
    }
}

```

se la atajó en el Load de FrmIngreso

```

    catch (MascotaModificadaException)
    {
        MessageBox.Show("ERROR!!! Un tipo de mascota fue modificado fuera del programa");
    }
    catch (ArchivoNoEncontradoException)

```

### Unit Testing:

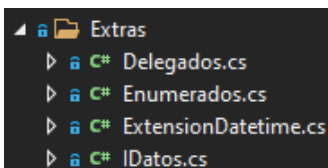
Dentro de la biblioteca Test, en la clase TestSerializador se prueban varias funciones de la clase Serializador con la excepción “ArchivoNoEncontradoException”.

### Tipos Genéricos:

La clase Serializador es una clase genérica. Esta contiene un atributo de una lista y funciones que permiten guardar dicha lista en formato xml y json.

Se implementa en el FrmIngreso en 3 formas, como Duenios, Mascotas e int.

### Interfaces:



```

Extras
├── C# Delegados.cs
├── C# Enumerados.cs
├── C# ExtensionDatetime.cs
└── C# IDatos.cs

```

se crea la clase IDatos que va a ser usado por las clases Duenio y Mascota

**Archivos:**

Se pueden guardar los datos del cliente y sus mascotas en un archivo de texto en un lugar a elegir.

Esto se implementa en el FrmMenuPrincipal mediante el botón "Exportar Mascota"

```
void ExportarArchivo(string contenido)
{
    SaveFileDialog save = new SaveFileDialog();
    save.Filter = "Archivo de texto|.txt";
    DialogResult result = save.ShowDialog();
    string path = save.FileName;

    if(result == DialogResult.OK)
    {
        using (StreamWriter writer = new StreamWriter(path))
        {
            writer.WriteLine(contenido);
        }
    }
}
```

**Serialización:**

Las funciones de serialización de xml y json están en la clase genérica Serializador.

La app abre automáticamente los archivos cuando se abre y al cerrar la app se preguntará si desea guardar los cambios.

**Conexión a bases de datos:**

Las clases "DuenioDAO" y "MascotaDAO" se encargan de cargar, guardar, agregar y modificar datos en la base de datos SwiftMedicalDB (El archivo SQL se encuentra en la misma carpeta que este PDF)

Se los llama desde el load de FrmIngreso para cargar los datos

```
DuenioDAO.CargarDueniosSql(dueniosSql, dueniosSqlOriginal);
MascotaDAO.CargarMascotasSql(mascotasSql, mascotasSqlOriginal);
MenuPrincipal();
```

Se lo llama desde el FormClosing de FrmIngreso para guardar los datos

```
DuenioDAO.GuardarDueniosSql(dueniosSql, dueniosSqlOriginal);
MascotaDAO.GuardarMascotasSql(mascotasSql, mascotasSqlOriginal);
```



## Delegados:

Extras  
Delegados.cs

```
public delegate void DelegadoCartel(int ejeX);  
public delegate int DelegadoCantidadMascotas(Duenio duenio);  
public delegate void DelegadoCargarMascotaEnLista(Mascota mascota);
```

## DelegadoCartel en FrmIngreso

```
void IniciarCartel(int ejeX)  
{  
    if (this.InvokeRequired)  
    {  
        DelegadoCartel delegadoCartel = new DelegadoCartel(IniciarCartel);  
        this.BeginInvoke(delegadoCartel, new object[] { ejeX });  
    }  
    else  
    {  
        this.lblAviso.Left = ejeX;  
    }  
}
```

## DelegadoCantidadMascotas en clase Duenio

```
public event DelegadoCantidadMascotas ContarMascotasPorDuenio;
```

## DelegadoCargarMascotaEnLista en FrmMascota

```
1 referencia | TomasBustamante92, Hace 18 horas | 1 autor, 1 cambio  
public FrmMascota(DelegadoCargarMascotaEnLista delegadoCargarMascotaEnLista,  
    int idDuenio, int ultimoId) : this()  
{  
    this.delegadoCargarMascotaEnLista = delegadoCargarMascotaEnLista;  
    this.idDuenio = idDuenio;  
    this.ultimoId = ultimoId;  
    MascotaModificada = false;  
}
```

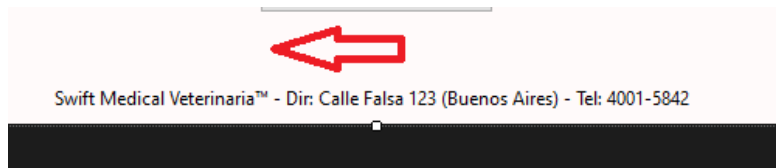
```
1 referencia | TomasBustamante92, Hace 18 horas | 1 autor, 1 cambio  
void AgregarMascotaALista(int edad)  
{  
    TipoAnimal tipoAux = (TipoAnimal)this.cmbTipoAnimal.SelectedItem;  
    this.ultimoId++;  
    this.mascota = new Mascota(this.ultimoId, tipoAux, txtNombre.Text, edad, txtRaza.Text, this.idDuenio, true);  
    this.delegadoCargarMascotaEnLista.Invoke(this.mascota);  
}
```

## Hilos:

en FrmIngreso

```
2 Referencias | TomasBustamante92, Hace 18 horas | 1 Auto!, 1 Cambio  
private void MoverCartel()  
{  
    try  
    {  
        this.cancellationTokenSource = new CancellationTokenSource();  
        this.cancellationToken = this.cancellationTokenSource.Token;  
        Task.Run(LooperCartel, this.cancellationToken);  
    }  
}
```

permite mover el lblAviso de derecha a izquierda mientras la música está sonando



Si se presiona este botón el lbl dejará de moverse, y si se vuelve a presionar seguirá su recorrido.

## Eventos:

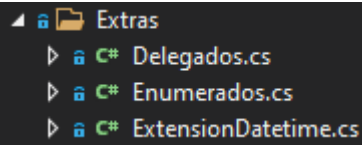
```
public string Mostrar()  
{  
    int cantidadMascotas = 0;  
    string retorno = $"Nombre: {this.nombre} - Telefono: {this.telefono} - Dirección: {this.direccion}";  
  
    if (ContarMascotasPorDuenio is not null)  
    {  
        cantidadMascotas = ContarMascotasPorDuenio.Invoke(this);  
        retorno += $" - Cantidad de mascotas: {cantidadMascotas}";  
    }  
  
    return retorno;  
}
```

evento ContarMascotasPorDuenio en clase Duenio

```
3 Referencias | TomasBustamante92, Hace 18 horas | 1 Auto!, 1 Cambio  
void CargarListbox()  
{  
    this.lbListaDuenios.Items.Clear();  
  
    foreach (Duenio duenio in this.dueniosJson.Lista)  
    {  
        if(duenio.Activo)  
        {  
            duenio.ContarMascotasPorDuenio += ContarMascotasPorDuenio;  
            this.lbListaDuenios.Items.Add(duenio);  
        }  
    }  
}
```

Se lo asocia en FrmCargar

## Métodos de extensión:



```
Extras
├── Delegados.cs
├── Enumerados.cs
└── ExtensionDatetime.cs
```

1 referencia | TomasBustamante92, Hace 18 horas | 1 autor, 1 cambio

```
public static string AgregarFecha(this DateTime d)
{
    return DateTime.Now.ToShortDateString().ToString();
}
```

1 referencia | TomasBustamante92, Hace 18 horas | 1 autor, 1 cambio

```
void CargarHistorial(string h, Mascota a)
{
    DateTime d = new DateTime();

    this.historial.Clear();
    this.historial.Append(a.Historial);
    this.historial.Insert(0, "----- \n\n");
    this.historial.Insert(0, h + "\n\n");
    this.historial.Insert(0, $"Fecha: {d.AgregarFecha()} \n\n");
    a.Historial = historial.ToString();
    this.rtbHistorial.Text = a.Historial;
}
```

Agrega método AgregarFecha y se usa en FrmMenuPrincipal