

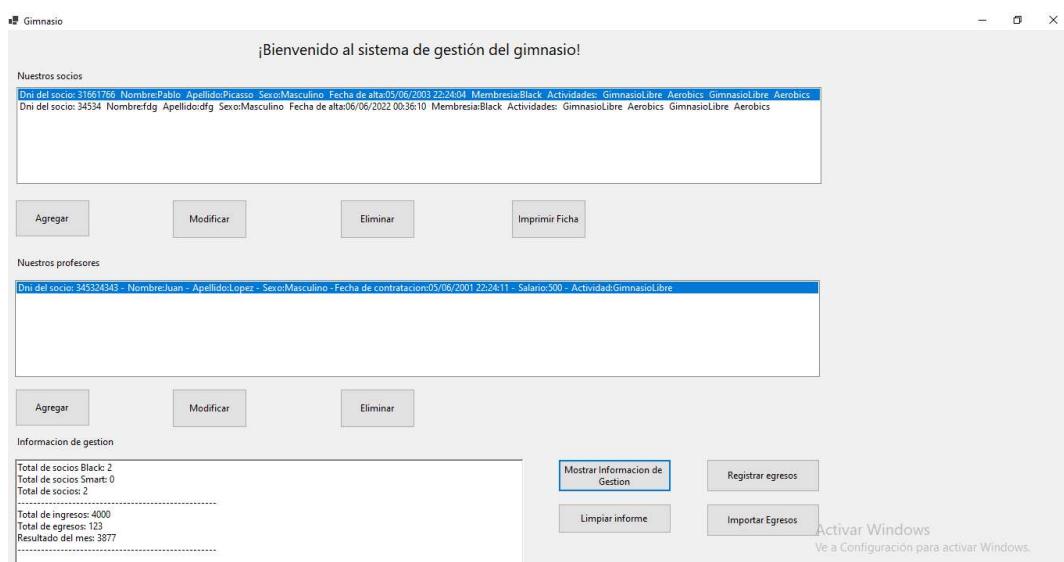
TP 3 – Laboratorio 2: “Software de gestión de gimnasios”

Aclaración importante

Para que la aplicación funcione correctamente, al usar por primera vez se debe compilar para que los archivos MockProfesores.xml y MockSocio.xml se copien en la carpeta de salida (en mi pc es la ruta “C:\Users\Marcos\Desktop\UTN
tps\TP3\Zalazar.Marcos.2E.TP3\Vista\bin\Debug\net5.0-windows\”)

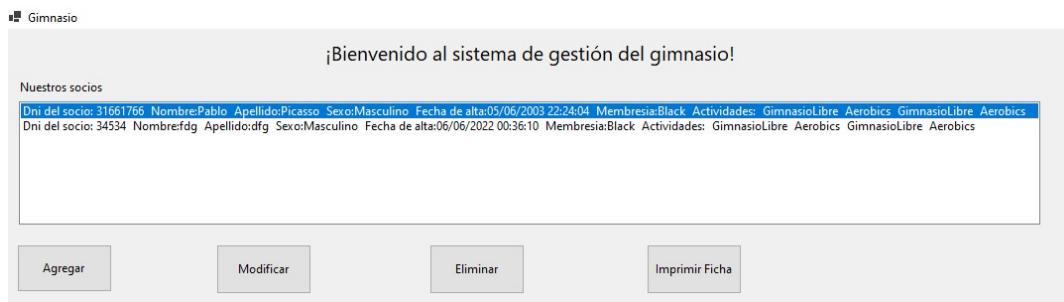
Modalidad de uso

Al ejecutar el programa, se abre la siguiente pantalla:

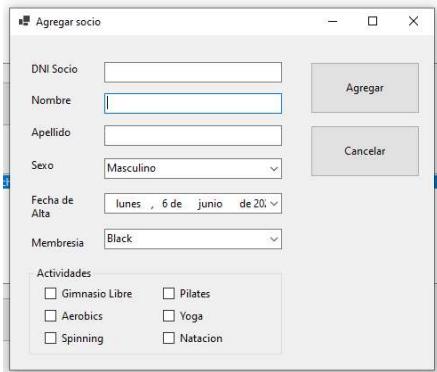


Como podemos ver, la pantalla principal está dividida en 3 secciones principales: listado de socios, listado de profesores e información de gestión.

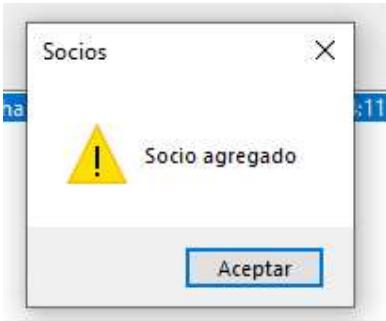
Listado de socios



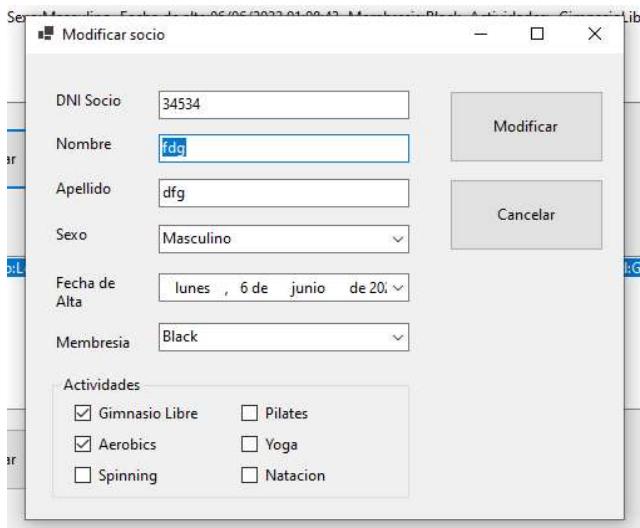
Esta sección tiene cuatro funcionalidades para gestionar los datos de los socios del club. Al desplegar la opción, se abre la siguiente pantalla:



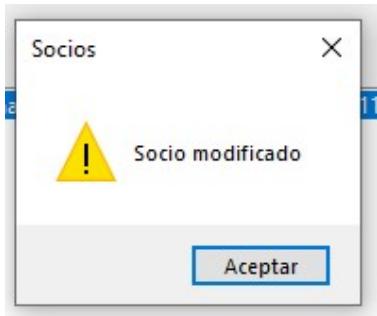
En ella, se pueden ingresar los datos necesarios para registrar al socio. Los campos que son completados por el usuario poseen validaciones que impiden que ingrese datos incorrectos, como, por ejemplo, ingresar letras en el campo DNI Socio o números en los campos de nombre y apellido. Tras oprimir el botón “aceptar”, el usuario recibirá una confirmación que el socio fue cargado con éxito. En caso de que lo desee, previo a realizar la carga puede cancelar la operación oprimiendo el botón correspondiente.



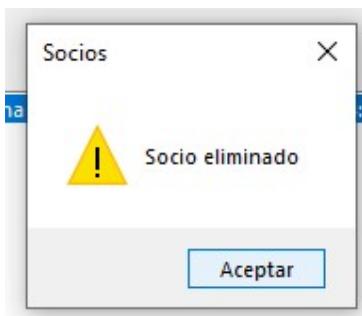
Si el usuario desea modificar los datos de un socio, debe seleccionar un socio del listado. Al hacerlo, una ventana similar a la anterior se va a desplegar, cargando los datos previamente registrados.



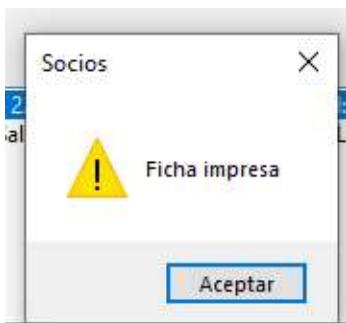
Luego de pulsar el botón “modificar”, los datos modificados podrán ser consultados en la lista.

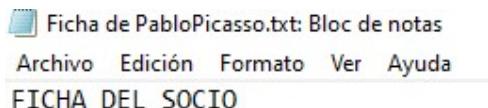


Una situación similar se da al eliminar un usuario:



Finalmente, en caso de que se deseen consultar los datos de un usuario se podrá generar un archivo de su ficha de socio.





Dni del socio: 31661766
Nombre:Pablo
Apellido:Picasso
Sexo:Masculino
Fecha de alta:05/06/2003 22:24:0
Membresia:Black
Actividades:
GimnasioLibre
Aerobics

Listado de profesores

La funcionalidad de esta sección es similar a la anterior contando con tres operaciones: alta, baja y modificación de los profesores contratados.

Nuestros profesores

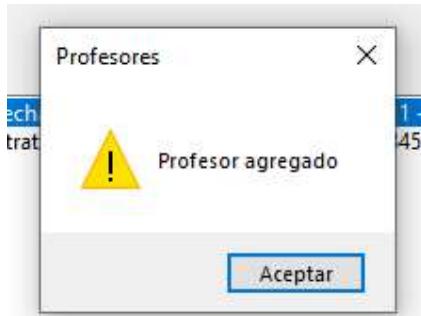
Dni del socio: 345324343 - Nombre:Juan - Apellido:Lopez - Sexo:Masculino - Fecha de contratacion:05/06/2001 22:24:11 - Salario:500 - Actividad:GimnasioLibre
Dni del socio: 345 - Nombre:dfg - Apellido:dfg - Sexo:Masculino - Fecha de contratacion:06/06/2022 01:13:21 - Salario:3455 - Actividad:GimnasioLibre

[Agregar](#) [Modificar](#) [Eliminar](#)

Nombre: Lopez Apellido:Juan Sexo:Masculino Fecha de alta:05/06/2001 22:24:11 Membresia:Black Actividad:GimnasioLibre

Agregar profesor

Dni	<input type="text"/>	Agregar
Nombre	<input type="text"/>	Cancelar
Apellido	<input type="text"/>	
Sexo	<input type="text"/> Masculino	
Fecha de contratacion	<input type="text"/> lunes , 6 de junio de 2022	
Salario	<input type="text"/>	
Actividad	<input type="text"/> GimnasioLibre	



Para evitar repeticiones, no se brindarán detalles sobre su uso, remitiendo al lector a la sección anterior.

Información de gestión

Informacion de gestion

Mostrar Informacion de Gestión Registrar egresos

Limpiar informe Importar Egresos

Esta sección informa sobre los datos de gestión más relevantes. Al oprimir el botón “mostrar información de gestión” se despliegan los principales indicadores del negocio.

Informacion de gestion

Total de socios Black: 2
Total de socios Smart: 0
Total de socios: 2

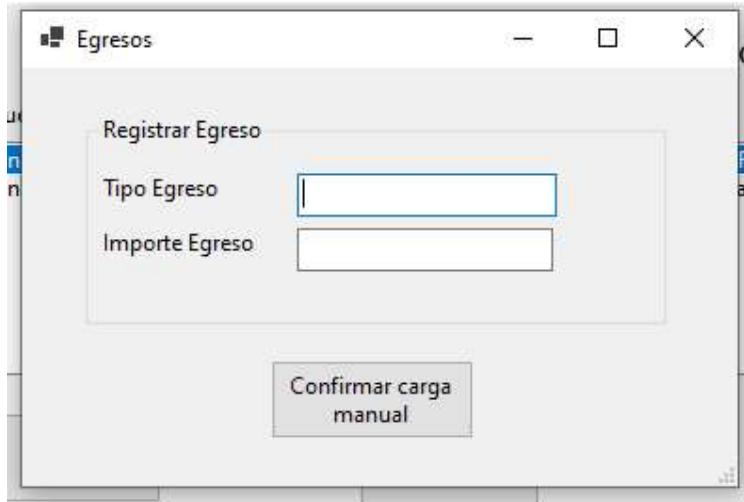
Total de ingresos: 4000
Total de egresos: 123
Resultado del mes: 3877

El sistema calcula la cantidad de socios que tiene el gimnasio, clasificados por su membresía. También informa el total de ingresos, en función de las cuotas cobradas a sus socios. En el caso de los egresos, calcula los salarios pagados a los profesores. También se podrá optar por agregar otros gastos de dos formas distintas: carga manual (botón “registrar egresos”) e importación de datos.

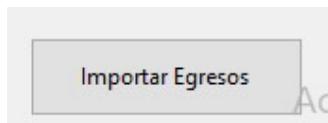
Registrar egresos

Importar Egresos

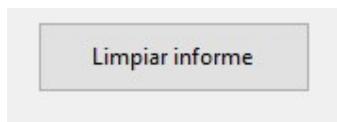
Al oprimir el primer botón, se despliega la siguiente pantalla



En el campo tipo de egreso se ingresa el concepto del egreso, por ejemplo, luz, gas, teléfono. En el otro campo disponible, el importe del egreso. Tras confirmar la carga el sistema informa que la operación fue realizada con éxito. Al hacerlo, se guarda en un archivo json esta información. En un mes posterior, el usuario podrá optar por importar estos gastos registrados en el archivo json, oprimiendo el botón “importar egresos”.



Luego de consultada la información, se podrá borrar el informe tocando el botón “Limpiar informe”.



Conceptos aplicados

- Excepciones: Se creó la clase “CargaFormException” para prevenir problemas en la carga del formulario:

```
namespace GimnasioException
{
    public class CargaFormException : Exception
    {
        public CargaFormException()
        {
        }

        public CargaFormException(string message) : base(message)
        {
        }
    }
}
```

En distintas partes del código se utilizaron bloques try-catch para atrapar posibles excepciones.

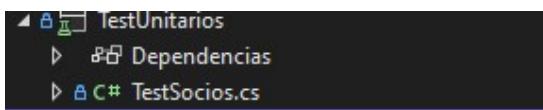
```

try
{
    if (!Directory.Exists(path))
    {
        Directory.CreateDirectory(nombreFile);
    }

    using (StreamWriter streamWriter = new StreamWriter(nombreArchivo))
    {
        XmlSerializer xmlSerializer = new XmlSerializer(typeof(T));
        xmlSerializer.Serialize(streamWriter, datos);
    }
}
catch (Exception e)
{
    throw new Exception($"Error en el archivo ubicado en {path}", e);
}

```

- Pruebas unitarias: Se realizaron algunos test unitarios de la clase socios.



- Tipos genéricos: Utilizada principalmente en las clases serializadoras de XML y Json, y también la clase "Gestión".

```

namespace Serializacion
{
    12 referencias
    public static class ClaseSerializadora<T>
    {
        static string path;
        0 referencias
    }

    3 referencias
    public class ClaseSerializadoraJson<T>
    {
        static string path;
    }

    namespace EntidadesNegocio
    {

        public class Gestion<T, U> where T : class where U : class
        {
            private List<T> egresos;
            private List<U> ingresos;
        }
    }
}

```

- Interfaces: Se creo la interface “IValidadorCampos” implementada por los formularios de alta y modificación de socios y profesores.

```

public interface IValidadorCampos
{
    public bool ElCampoNoEstaVacio(string campo);
    public bool ValidarEleccionSexo();
    public bool ValidarEleccionFecha();
    public bool ValidarEleccionActividad();
}

```

```

  ↗ referencias
public partial class FrmAgregarModificarProfesores : Form, IValidadorCampos
{
    private Profesor profesor;
}

namespace Vista
{
    ↗ referencias
    public partial class FrmAgregarModificarSocio : Form, IValidadorCampos
    {
        private Socio socio;
    }
}

```

2 referencias

- Archivos: Se crea un txt con la ficha del socio en la función imprimirFicha.

```

1 referencia
private void btnImprimirFicha_Click(object sender, EventArgs e)
{
    Socio socioSeleccionado = (Socio)this.lstSocios.SelectedItem;

    if (socioSeleccionado is not null)
    {
        string contenido = this.gimnasio.ImprimirFichaDelSocio(socioSeleccionado);
        string nombreFicha = "Ficha de "+socioSeleccionado.Nombre + socioSeleccionado.Apellido+".txt";
        ClaseSerializadora<string>.EscribirEnTxt(nombreFicha, contenido);
        MessageBox.Show("Ficha impresa", "Socios", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

```

- Serialización: se implementaron dos clases serializadoras de archivos XML y Json. Se serializan y deserealizan a XML los objetos socios y profesores. Se serializan y deserealizan a JSON los objetos egresos

```

▲ 🔍 C# Serializacion
    ▷ 🔍 Dependencias
    ▷ 🔍 C# ClaseSerializadora.cs
    ▷ 🔍 C# ClaseSerializadoraJson.cs

```