

Trabajo integrador n° 2 : Administración de atletas para un box de Crossfit.

Con las consignas dadas del trabajo, es un programa para gestionar atletas que quieran entrenar en un box de crossfit.

Chronos Crossfit

Cargar Atleta

Modificar Atleta

Pago Cuota

Estado de cuenta

Eliminar Atleta

Exportar Informacion

Cargar Base de Datos (sql)

Base de datos cargada

Nombre: Agneta Apellido: Buer Dni: 63629527 Fecha de nacimiento: 12/5/2021 Edad: 1 Pase: Una Clase

Nombre: Ardenia Apellido: Loy Dni: 56099390 Fecha de nacimiento: 9/5/2021 Edad: 2 Pase: Una Clase

Nombre: Dannie Apellido: Morrilly Dni: 10077647 Fecha de nacimiento: 9/16/2021 Edad: 2 Pase: Libre

Nombre: Darin Apellido: Cowndley Dni: 16675451 Fecha de nacimiento: 6/26/2021 Edad: 2 Pase: Libre

Nombre: Eustacia Apellido: Matyas Dni: 43609717 Fecha de nacimiento: 1/27/2022 Edad: 1 Pase: Libre

Nombre: Evania Apellido: Robers Dni: 49094246 Fecha de nacimiento: 3/9/2022 Edad: 1 Pase: Libre

Nombre: Jess Apellido: Clemmensen Dni: 43258440 Fecha de nacimiento: 3/9/2022 Edad: 1 Pase: Libre

Nombre: Karlan Apellido: Henden Dni: 44266700 Fecha de nacimiento: 5/11/2022 Edad: 1 Pase: Libre

Nombre: Kermy Apellido: Gonthier Dni: 62304520 Fecha de nacimiento: 2/18/2022 Edad: 1 Pase: Libre

Nombre: Kingsly Apellido: Rotlauf Dni: 52975132 Fecha de nacimiento: 6/21/2021 Edad: 2 Pase: Libre

Nombre: lucas Apellido: pepe Dni: 12312312 Fecha de nacimiento: 11/22/2023 Edad: 0 Pase: Libre

Nombre: Martha Apellido: Eliasson Dni: 75727366 Fecha de nacimiento: 3/22/2022 Edad: 1 Pase: Una Clase

Nombre: Naomi Apellido: Abrahamer Dni: 45578223 Fecha de nacimiento: 7/26/2021 Edad: 2 Pase: Una Clase

Nombre: r Apellido: IDni: 34786816 Fecha de nacimiento: 11/22/2023 Edad: 0 Pase: Libre

Nombre: Ruben Apellido: Lopez Dni: 34786815 Fecha de nacimiento: 11/22/2023 Edad: 0 Pase: Libre

Nombre: Tate Apellido: Wickie Dni: 97977736 Fecha de nacimiento: 4/4/2022 Edad: 1 Pase: Una Clase

Nombre: Willem Apellido: Major Dni: 38808854 Fecha de nacimiento: 5/28/2022 Edad: 1 Pase: Libre

La app inicia con una ventana donde se puede acceder a siete acciones predeterminadas, si no hay un atleta cargado o si no se cargó la base de datos.

CargarAtleta:

Cargar Atleta

Nombre

Apellido

DNI

Fecha de nacimiento

miércoles, 22 de noviembre

Pase

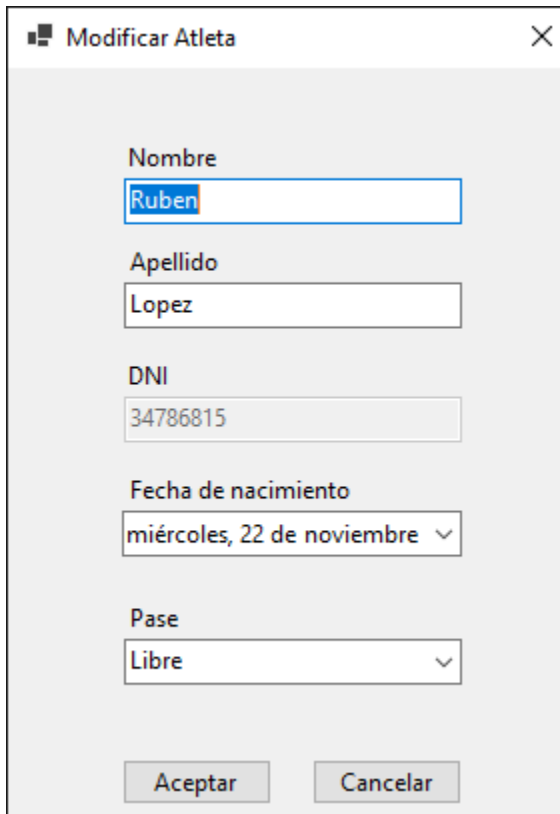
Libre

Aceptar

Cancelar

mediante esta esta ventana podemos cargar un atleta llenando los campos requeridos con el formato establecido, donde el dni solo se puede ingresar números entre 0 a 99999999, y con 7-8 dígitos. Donde dni sera un dato único e irrepitable, siendo el identificador de cada atleta.

Modificar Atleta:



The image shows a software dialog box titled "Modificar Atleta" with a close button (X) in the top right corner. The dialog contains several input fields for athlete information:

- Nombre:** A text input field containing the name "Ruben".
- Apellido:** A text input field containing the surname "Lopez".
- DNI:** A text input field containing the number "34786815".
- Fecha de nacimiento:** A date picker field showing "miércoles, 22 de noviembre" with a downward arrow.
- Pase:** A dropdown menu showing the option "Libre" with a downward arrow.

At the bottom of the dialog are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Seleccionando un atleta de la lista se puede modificar sus datos, excepto el dni, para ello debe eliminar dicho atleta y crear uno de nuevo.

Pagar cuenta:

The image shows a Windows-style dialog box titled "frmPagarCuota" with a close button (X) in the top right corner. The form contains the following fields:

- Nombre:** Text box containing "Ruben".
- Apellido:** Text box containing "Lopez".
- DNI:** Text box containing "34786815".
- Pase:** Dropdown menu with "Libre" selected.
- Importe:** Text box containing "15000".
- Fecha:** Date picker showing "miércoles, 22 de noviembre".
- Metodo de pago:** Dropdown menu with "Efectivo" selected.

At the bottom of the form are two buttons: "Aceptar" and "Cancelar".

Seleccionando un atleta de la lista podemos pagar la cuota por mes, ingresando el importe, fecha, pase y metodo de pago. Al aceptar el registro se agraga automaticamente al historial de pagos del atleta (estado de cuenta). No se puede pagar el mismo mes.

Estado de cuenta:

Seleccionando un atleta de la lista podemos ver su estado de cuenta, su historial de pagos y ver si debe o si esta al dia. Si no pago la cuota del ultimo mes es un deudor. Podemos imprimir estos datos mediante un archivo de texto con los datos del atleta, historial y estado.

Estado de cuenta

Nombre
Ruben

Apellido
Lopez

DNI
34786815

Estado de cuenta
Esta al día

Imprimir

Año: 2023, Mes: 10, Metodo de Pago: Efectivo, Importe: \$14000
Año: 2023, Mes: 11, Metodo de Pago: Efectivo, Importe: \$15000

Eliminar atleta:

Seleccionando un atleta de la lista, podemos eliminar dicho atleta del box.

Exportar Informacion:

Informacion

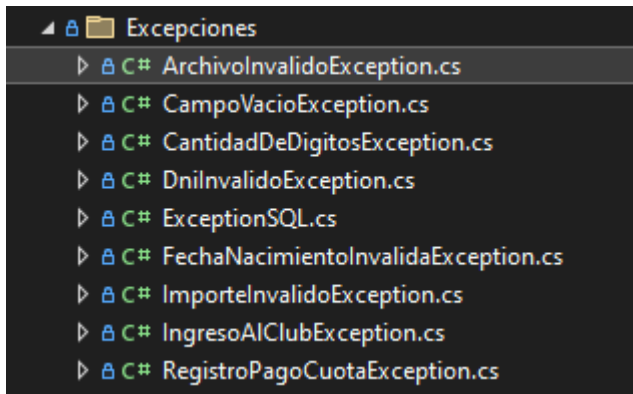
Datos exportados con éxito.
El archivo dataAtletas.xml se encuentra en:
C:\Users\ruben\OneDrive\Escritorio\RNL\UTN\Programacion y
Laboratorio II\Ejercicio Integrador n2
lab2\Lopez.Ruben.2C.TPFinal\Formularios\bin\Debug\net6.0-
windows\ListaAtletasBackUp

Exportar los datos del box en un archivo xml como backup.

Carga de base de datos (sql):

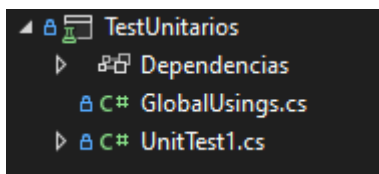
Para comenzar a gestionar la carga de datos en el programa debemos cargar la base de la datos, abajo de su boton nos indicara si se cargo o no dicha información.

Tema 10 - Excepciones:



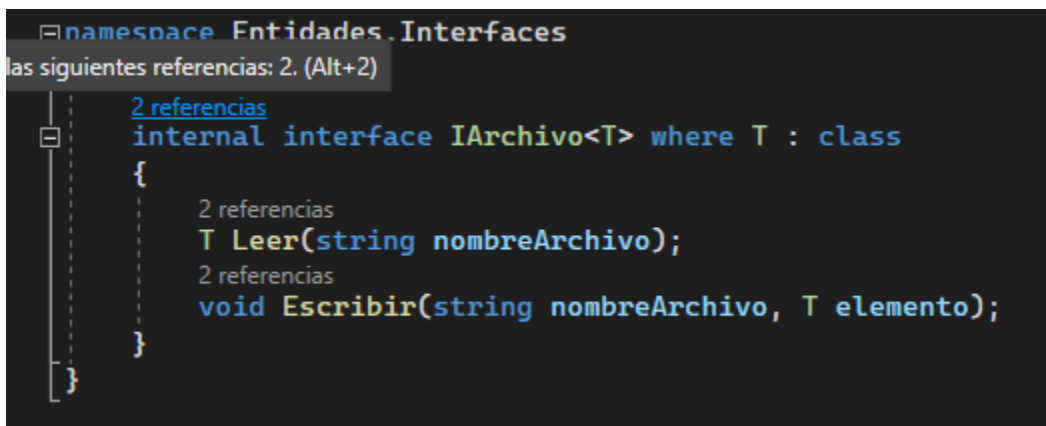
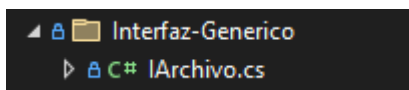
Se crearon distintas excepciones implementadas en diferentes secciones del proyecto.

Tema 11 - Test Unitarios:



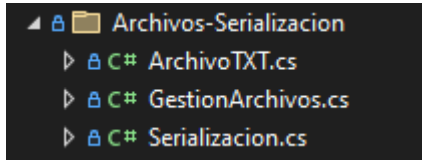
Ponemos a pruebas dos funcionalidades del proyecto, el metodo ElimnarAtleta(Atleta a) y RegistroPagos(Atleta a, Cuota c).

Tema 12 - Generics y Tema 13 - Interfaces:



De manera practica ambos se utilizan en la interfaz que usa el gestor archivos.

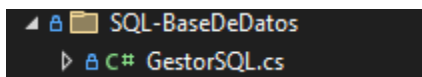
Tema 14 - Archivos y Serializacion:



Archivos se utiliza en frmEstadoDeCuenta para imprimir un comprobante del estado de cuenta de un atleta. Usando serializacion en frmInicio para exportar los datos de box con un archivo xml.

```
1 referencia
private void btnImprimir_Click(object sender, EventArgs e)
{
    try
    {
        ArchivoTXT archivo = new ArchivoTXT();
        string nombreDeArchivo;
        archivo.Escribir($"{atleta.Nombre}{atleta.Apellido}.txt", atleta.Imprimir());
        nombreDeArchivo = $"{archivo.PathEscritura()} {atleta.Nombre}{atleta.Apellido}.txt";
        ProcessStartInfo proceso = new ProcessStartInfo
        {
            FileName = nombreDeArchivo,
            UseShellExecute = true
        };
        Process.Start(proceso);
        MessageBox.Show($"Impresión exitosa.\nRecibo en: {archivo.PathEscritura()}", "Informacion", Mes
    }
    catch (ArchivoInvalidoException ex)
    {
        MessageBox.Show(ex.Message, "Error al escribir el archivo de texto", MessageBoxButtons.OK, Mess
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error al escribir el archivo de texto", MessageBoxButtons.OK, Mess
    }
}
```

Tema 15 - Introduccion a SQL y Conexión a base de datos:



En gestorSQL.cs hay siete metodos para leer y escribir en la base de datos SQL del programa, con ella se maneja el alta, modificacion o eliminacion de atletas, como el registro de sus cuotas de sus cuotas pagas. Despues de cada interacción en el programa la base de datos es actualizada para no perder información.

```

4 referencias
private void ActualizaListaSocios()
{
    try
    {
        this.Box.ListaAtletas = GestorSQL.LeerDatosAtleta();
        this.Box.ListaAtletas.Sort((Atleta a1, Atleta a2) => s
        this.lstPersonas.DataSource = null;
        this.lstPersonas.DataSource = Box.ListaAtletas;
    }
}

```

Tema 16 - Delegados y expresiones lambda , Tema 17 - Programación multi-hilo y concurrencia y Tema 18 - Eventos:

Usando expresiones lambda como parametro del método Sort() se ordena las listas de Atletas y de cuotas.

```

private void ActualizaListaSocios()
{
    try
    {
        this.Box.ListaAtletas = GestorSQL.LeerDatosAtleta();
        this.Box.ListaAtletas.Sort((Atleta a1, Atleta a2) => string.Compare(a1.Nombre, a2.Nombre));
        this.lstPersonas.DataSource = null;
        this.lstPersonas.DataSource = Box.ListaAtletas;
    }
}

```

Y usando delegados, eventos e hilos se realiza la carga de la base de datos, todo se puede ver en el código de frmInicio.

en frmInicio declaro mis delegados y creo 2 eventos, uno que informara el proceso de la carga de la base y el otro que informara cuando ya se hay cargado la base de datos.

```

public delegate void CargandoBaseDeDatos(int tiempo);
public delegate void FinCargarBaseDeDatos();

public event CargandoBaseDeDatos InformeCargando;
public event FinCargarBaseDeDatos FinDeCarga;

```

En el evento click del boton Leer base de datos(sql) se subscriben los métodos a los eventos, y en un hilo separado se lanza el evento para iniciar la carga de la base de datos.

```
private void btnLeerBaseDeDatosSQL_Click(object sender, EventArgs e)
{
    this.InformeCargando += CargandoBase;
    this.FinDeCarga += FinDeCargaBase;

    Task task = Task.Run(IniciarCarga);
}
```

Mas adelante se pueden ver los métodos mencionados e invocados. Primero el de IniciarCarga() que define el tiempo de espera, y verificando que informeCargando() tenga algun subscriptor, lo invoca pasando el tiempo como parámetro. Aqui el metodo CargandoBase() es el que actualiza el texto del label informando el tiempo faltante, y se va actualizando hasta que el tiempo se acabe. Ahi de vuelta en iniciarCarga() FinDeCarga invoca al metodo que informa que la base de datos fue cargada actualizano el label.

```
1 referencia
public void IniciarCarga()...
```

```
1 referencia
private void CargandoBase(int tiempo)...
```

```
1 referencia
private void FinDeCargaBase()...
```

Tema 20 - Métodos de extension:

Con el metodo CantidadDeDigitos() de la clase extendida, puedo obtener la cantidad de digitos de un entero, eso se usa en la validación del ingreso del dni de los atletas.

```
public static class ClassExtendida
{
    2 referencias
    public static int CantidadDeDigitos(this int numero)
    {
        return numero.ToString().Length;
    }
}
```