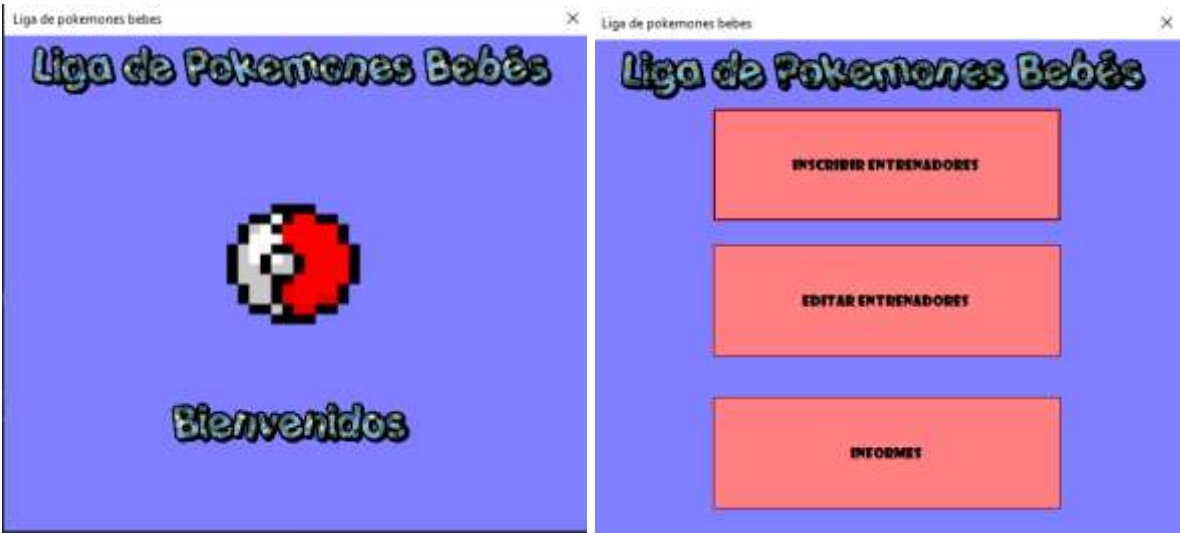


# LIGA DE POKEMONES BEBÉS

Este programa está destinado a ser el formulario de inscripción y edición de entrenadores de pokemones a la. Este empieza con un evento de simulación de carga y muestre tres botones,



## Primer Botón, Inscribir entrenadores

Llenas los datos de requeridos, seleccionas la cantidad de pokebolas que quieras tener y luego vas agregando uno a uno cada pokemon. Solo puedes obtener 1 de cada especie. Al agregarlo se actualiza la base de datos.

Formulario de inscripción a la liga Pokemon

DNI

NOMBRE

APELLIDO

EDAD

☐ ¿ERES CAMPEÓN?

ISLA

Prima

CANTIDAD DE POKEBOLAS

1

POKEMONES

Pichu

AGREGAR

QUITAR

GUARDAR

**Sedundo boton, formulario de modificacion,**  
editas los campos o eliminas un entrenador, depenpendiando del tipo de edicion a realizar que hayas elegido.

Al hacer cualquier cambio se actualiza la base de datos.

Formulario de modificacion de la liga Pokemon

**TIPO DE EDICION A REALIZAR**

Modificar

**ENTRENADOR**

Lucas Millenio

**DNI**

18495623

**CANTIDAD DE POKEBOLAS**

4

**NOMBRE**

Lucas

**POKEMONES**

Pichu

**APELLIDO**

Millenio

AGREGAR

QUITAR

**EDAD**

22

☐ ¿ERES CAMPEÓN?

**ISLA**

Prima

|   | Id  | Hp | Especie   | Tipo    | Ataque |
|---|-----|----|-----------|---------|--------|
| ▶ | 433 | 45 | Chingling | Psychic | 30     |
|   | 439 | 20 | Mime Jr.  | Psychic | 25     |
|   | 240 | 45 | Magby     | Fire    | 75     |
|   |     |    |           |         |        |

GUARDAR MODIFICACION

Tercer boton, Informes

Luego de 5 segundos de espera se muestra los datos de las estadísticas realizadas.

Al darle al botón de exportar se exporta la información mostrada en el data grid view.



## Excepciones

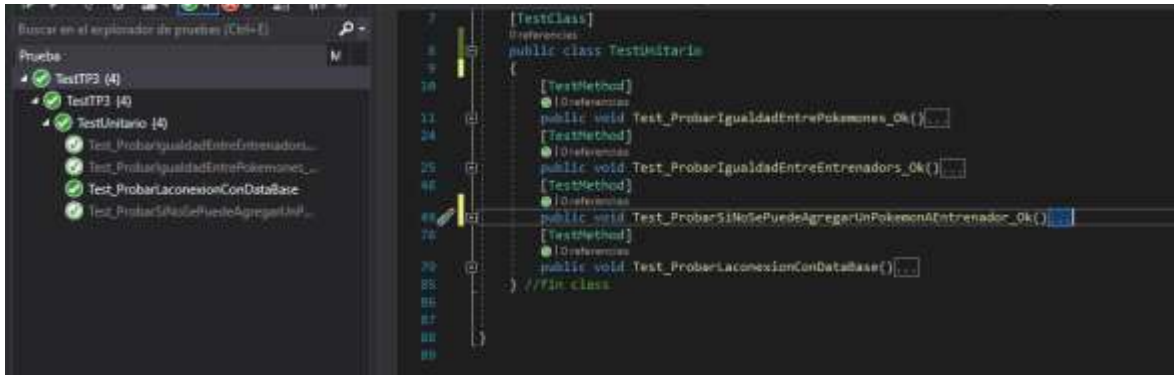
En la biblioteca entidades hay una clase excepción propia que se llama cuando hubo un problema con la serialización ya que estan agregadas en los metodos de serealizacion.

```
namespace Entidades
{
    7 referencias
    public class SerealizacionesException : Exception
    {
    }
```

```
▲ Entidades\SerealizacionArchivoJson.cs (2)
    55: throw new SerealizacionesException(ex.Message);
    78: throw new SerealizacionesException(ex.Message);
▲ Entidades\SerealizacionesException.cs (5)
    13: static SerealizacionesException()
    19: public SerealizacionesException()
    23: public SerealizacionesException(string message) : base(mensajeDeError)
    27: public SerealizacionesException(string message, Exception innerException) : base(SerealizacionesExcep...
    27: public SerealizacionesException(string message, Exception innerException) : base(SerealizacionesExcep...
```

## Pruebas unitarias

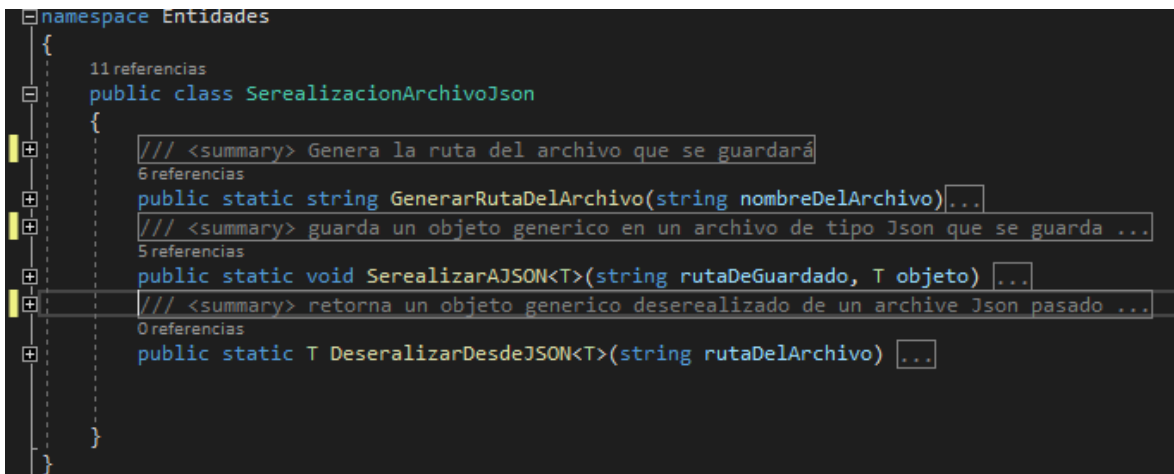
### Carpeta TestUnitario



## Generics

### Carpeta Entidades / SerealizacionArchivoJson.cs

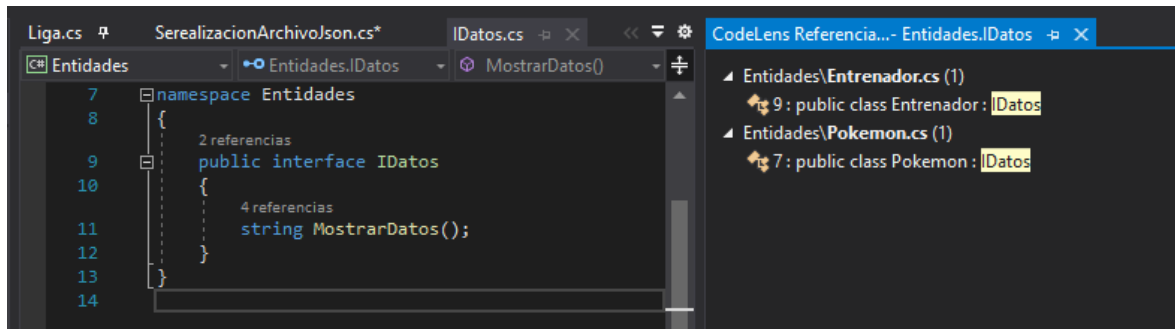
Usado en los metodos de seralizacion para serealizar y deserealizar cualquier tipo de objeto



## Interfaces

### Carpeta Entidades \ IDatos.cs

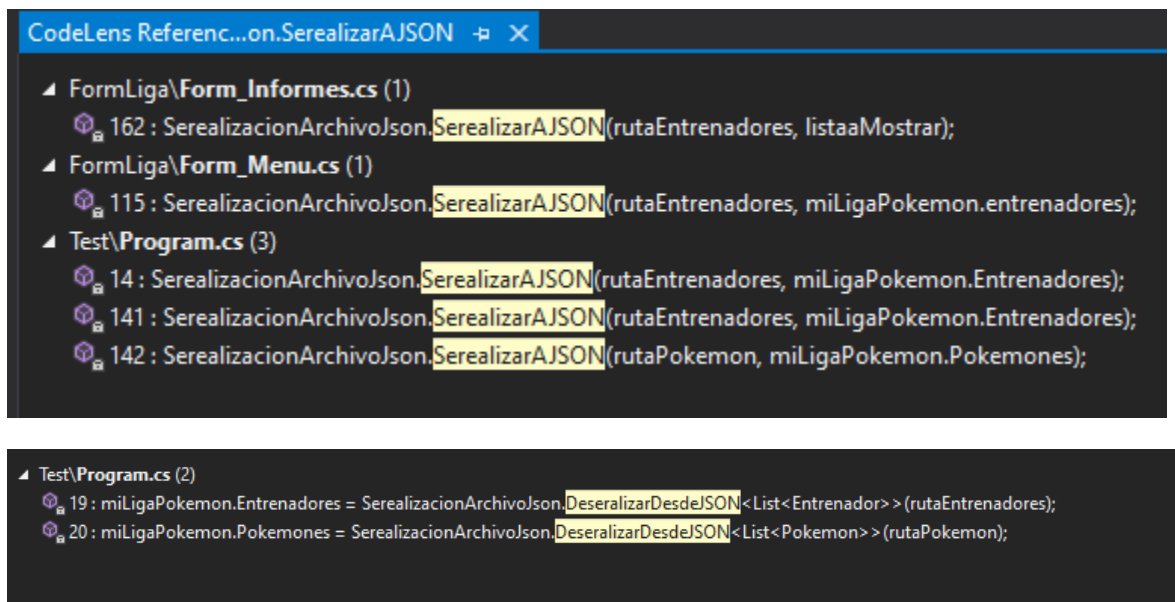
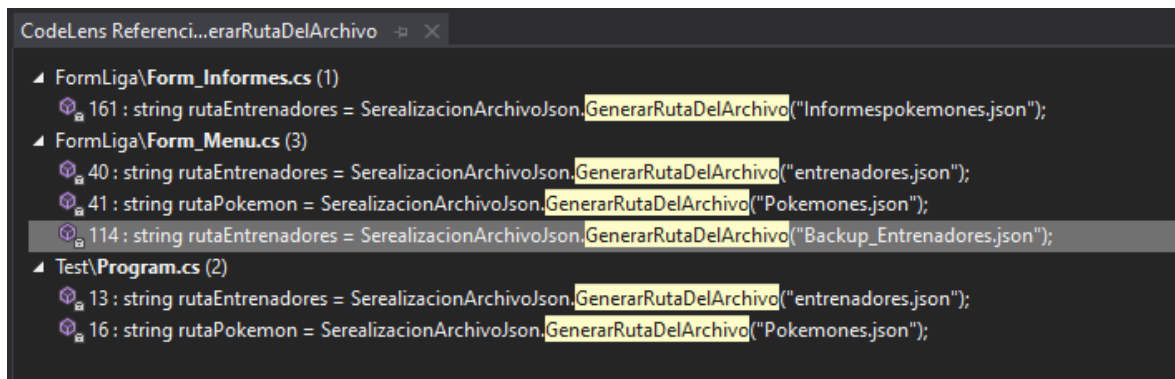
En ambas clases Entrenador y Pokemon implemento la interfaz IDatos para mostrar sus datos



## Archivos y Serialización

En Form\_Menu se guarda cada 10 segundos un backup de la lista de entrenadores en tiempo real.

EN Form\_Informes se exporta la lista de entrenadores filtrada.



## Base de datos

Carpeta Entidades \ ManejoBD

```

1 referencia | 0/1 pasando
public bool ProbarConexion()...

1 referencia
public bool AgregarPokemon(Pokemon pokemon)...

3 referencias
public bool AgregarEntrenador(Entrenador entrenador)...

2 referencias
public bool EliminarEntrenador(int DniEntrenador)...

1 referencia
public bool ModificarEntrenador(Entrenador entrenador)...

0 referencias
public bool agregarLaListaDePokemonesALaBD(List<Pokemon> pokemones)...

0 referencias
public bool EliminarPokemon(int idPokemon)...

0 referencias
public bool AgregarEntrenadorConPokemon(Entrenador entrenador, int idPokemon)...

0 referencias
public bool agregarLaListaDeEntrenadoresALaBD(List<Entrenador> entrenadores)...

```

```

1 referencia
public List<Entrenador> ObtenerListaEntrenador()...

1 referencia
public List<Pokemon> ObtenerListaDePokemones()...

```

```

FormLiga\Form_Menu.cs (1)
83 : miLigaPokemon.Pokemones = BD.ObtenerListaDePokemones();

```

```

FormLiga\Form_Menu.cs (1)
82 : miLigaPokemon.Entrenadores = BD.ObtenerListaEntrenador();

```

```

Entidades\ManejoBD.cs (2)
183 : if (!AgregarEntrenador(entrenador))
321 : if (!AgregarEntrenador(item))

FormLiga\Form_ManejoEntrenadores.cs (1)
81 : BD.AgregarEntrenador(entrenador);

```

## Hilos

Carpeta FormLiga \ Form\_Menu

Se serializa en un hilo secundario la lista de entrenadores y la liga actual cada vez que que pasan 10 segundos, para guardar una copia de seguridad Actual.

```

1 referencia
private void actualizarArchivoSeguridad(CancellationToken cancellationToken)
{
    while (!cancellationToken.IsCancellationRequested)
    {
        string rutaEntrenadores = SerealizacionArchivoJson.GenerarRutaDelArchivo("Backup_Entrenadores.json");
        SerealizacionArchivoJson.SerealizarAJSON(rutaEntrenadores, miligaPokemon.entrenadores);
        string rutaLiga = SerealizacionArchivoJson.GenerarRutaDelArchivo("Backup_Liga.json");
        SerealizacionArchivoJson.SerealizarAJSON(rutaLiga, miligaPokemon);
        Thread.Sleep(10000);
    }
}

```

#### FormLiga\Form\_Menu.cs (1)

```

86: this.BackupDeEntrenadores = new Task(() => this.actualizarArchivoSeguridad(tokenParaCancelarHilo));

```

## Hilos y eventos

Carpeta FormLiga \ Form\_Menu

Simula carga del programa al iniciar el formulario. Luego de 3 segundos se muestran los botones del menú.

```

124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
1 referencia
private void cargarBotones(CancellationToken cancellationToken)
{
    int cuenta = 0;
    while (!cancellationToken.IsCancellationRequested)
    {
        if (this.pic_pokebola.InvokeRequired)
        {
            if (cuenta == 3)
            {
                this.pic_pokebola.BeginInvoke((MethodInvoker)delegate
                {
                    EventoCarga.Invoke();
                });
                break;
            }
            cuenta++;
            Thread.Sleep(1000);
        }
        else
        {
            pic_pokebola.Visible = true;
            pic_Bienvenidos.Visible = true;
        }
    }
}

1 referencia
private void MostrarBotones()
{
    btn_inscripcion.Visible = true;
    btn_EditarEntrenadores.Visible = true;
    btn_Informes.Visible = true;
    pic_pokebola.Visible = false;
    pic_Bienvenidos.Visible = false;
}

```

FormLiga\Form\_Menu.cs (1)

49: this.simularCarga = new Task(() => this.cargarBotones(tokenParaCancelarHilo));

Carpeta FormLiga \ Form\_Menu

En este hilo se realizan las estadísticas de la liga y luego de 5 segundos se muestran.

```
1 referencia
private void cargarDatosDeEstadistica(CancellationToken cancellationToken)
{
    int cuenta = 0;
    while (!cancellationToken.IsCancellationRequested)
    {
        if (this.pic_pichu.InvokeRequired)
        {
            cuenta++;
            if(cuenta== 5)
            {
                this.pic_pichu.BeginInvoke((MethodInvoker)delegate ()
                {
                    pic_Cargando.Visible = false;
                    pic_pichu.Visible = false;
                    CuentasDePromedioListas.Invoke();
                });
                break;
            }
            Thread.Sleep(1000);
        }
        else
        {
            pic_pichu.Visible = true;
            pic_Cargando.Visible = true;
        }
    }
}
2 informacion
```

FormLiga\Form\_Informes.cs (1)

47: CuentasDePromedioListas += RealizarEstadisticas;

**Métodos de extensión.**

Carpeta FormLiga \ ExtensionForm



## ▲ FormLiga\Form\_Informes.cs (1)

155 : Dtg\_Entrenadores.ActualizarDataGridV(listaaMostrar);

## ▲ FormLiga\Form\_ManejoEntrenadores.cs (5)

173 : dtg\_pokemones.ActualizarDataGridV(entrenadorAuxiliar.Pokemones);

230 : dtg\_pokemones.ActualizarDataGridV(entrenador.Pokemones);

249 : dtg\_pokemones.ActualizarDataGridV(pokemonesSeleccionados);

304 : dtg\_pokemones.ActualizarDataGridV(pokemonesSeleccionados);

330 : dtg\_pokemones.ActualizarDataGridV(pokemonesSeleccionados);

namespace FormLiga

{

0 referencias

public static class ExtensionForm

{

/// <summary>

/// Actualiza un dataSource del data grid view con una lista pasada por parametro

/// </summary>

/// <typeparam name="T"></typeparam>

/// <param name="data">data grid view a usar</param>

/// <param name="listaAMostrar">lista para mostrar en el data grid view</param>

6 referencias

public static void ActualizarDataGridV<T>(this DataGridView data, List<T> listaAMostrar)

{

data.DataSource = null;

if (listaAMostrar.Count > 0 && listaAMostrar is not null)

{

data.DataSource = listaAMostrar;

}

}