

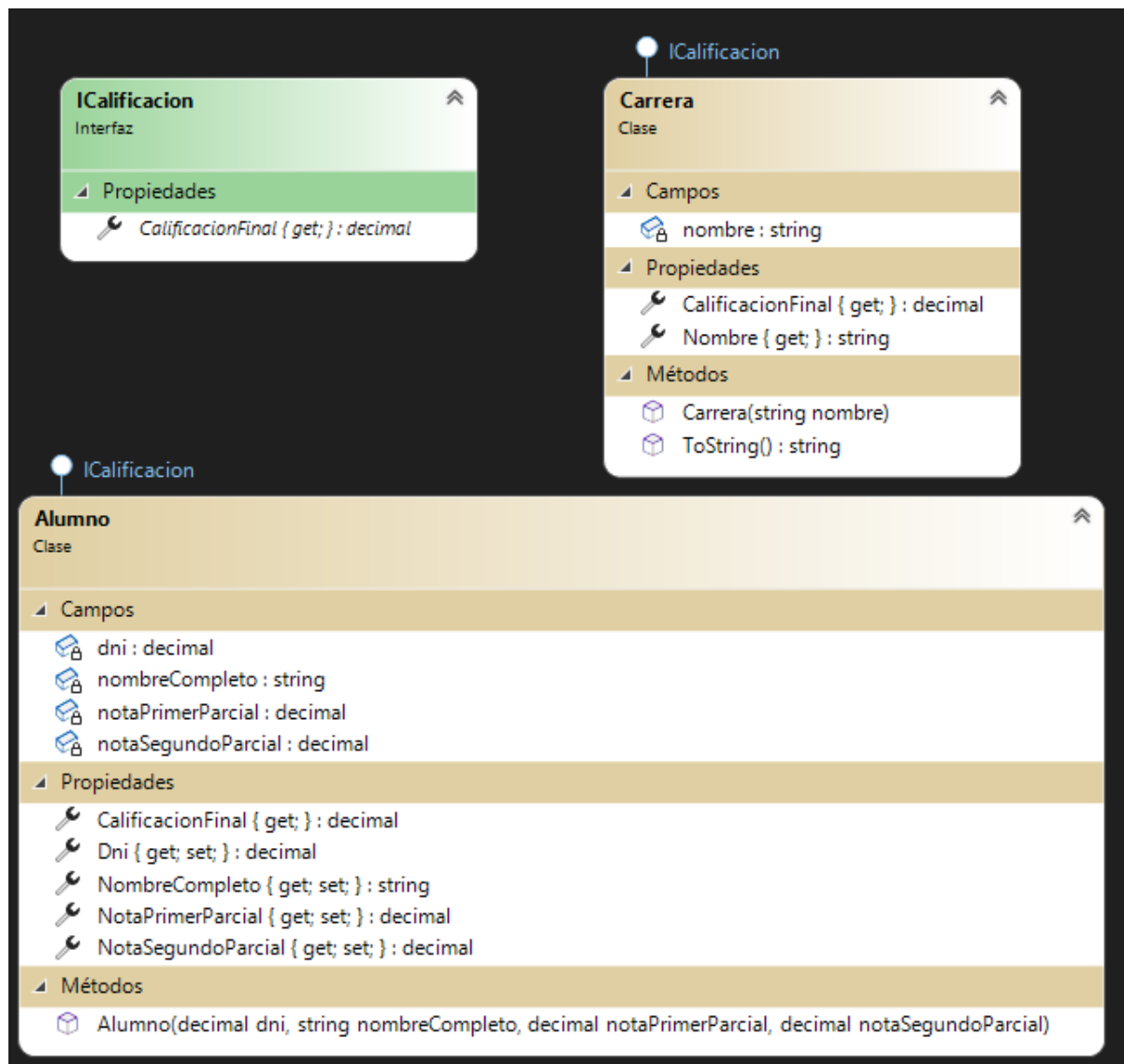
## 2do Parcial Lab II - 2°D - 1°Cuatri 2022

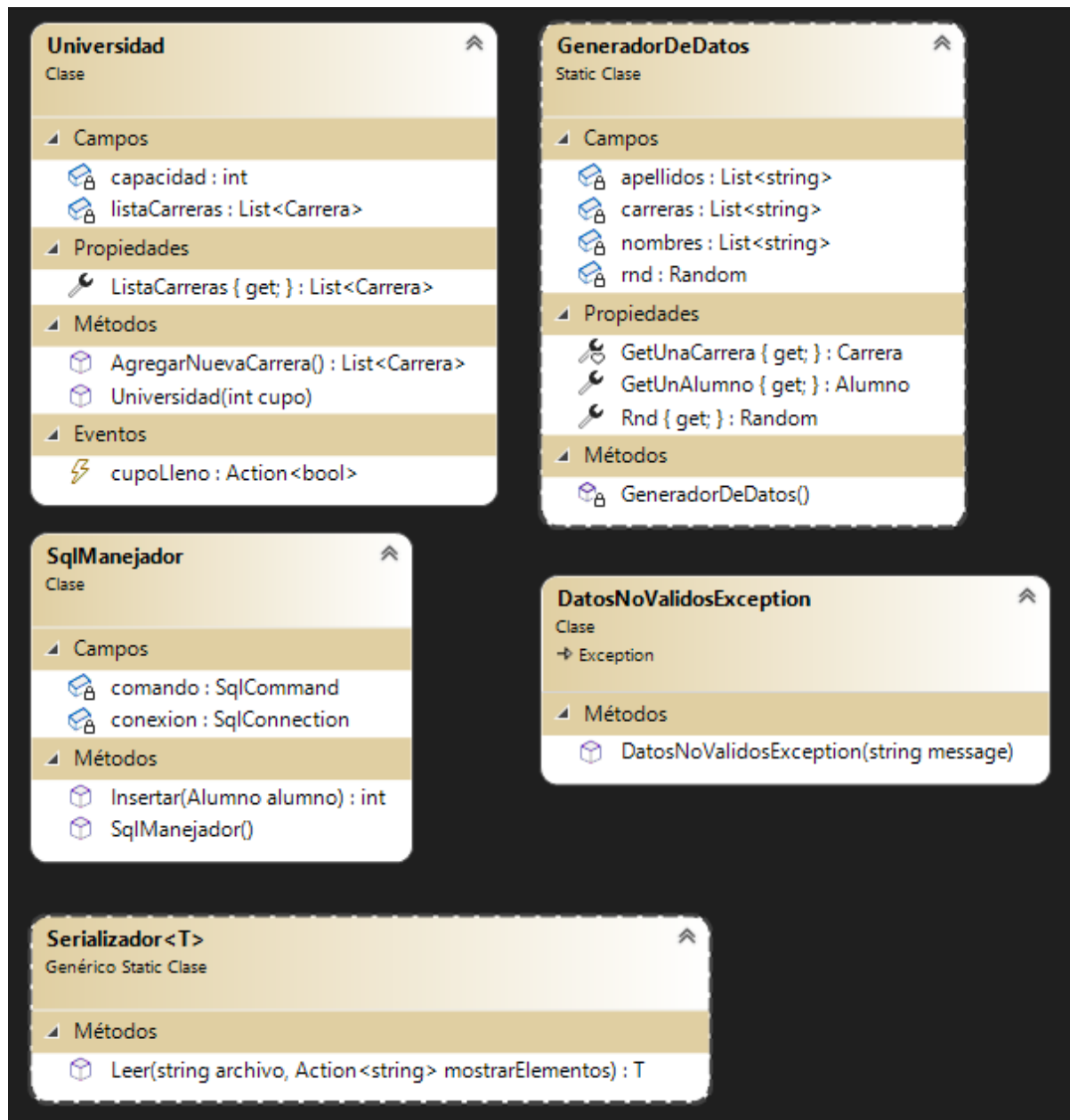
### Introducción

Junto con este documento, recibieron una aplicación a la cual hay que terminar las funcionalidades detalladas más adelante. Tener en consideración que hay funcionalidades ya desarrolladas que no se deben cambiar. Todo aquello que NO sea necesario modificar, aparecerá informado en el código. De todas formas, si algo les genera duda, pregunten.

### Secciones de la aplicación

## Biblioteca de clases





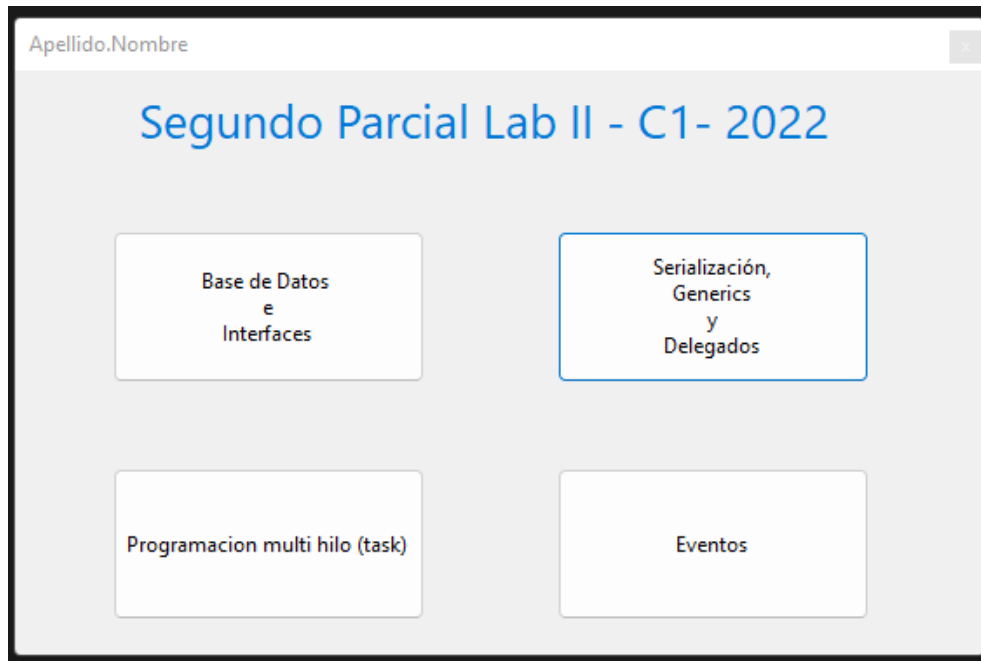
## Interfaz Calificación:

La interfaz **ICalificación** deberá implementarse en las clases Alumno y Carrera.

- En **Alumno**, deberá retornar el promedio entre los parciales. Acortar el valor a 2 decimales (Utilizar `Decimal.Round`).
- En **Carrera**, deberá retornar un número aleatorio entre 1 y 10 inclusive. Para esto deberán utilizar la clase **GeneradoraDeDatos**, la propiedad **Rnd**. Esta propiedad retorna un objeto random, del cual deberán utilizar su método `Next`. (Si buscan donde se utiliza la propiedad, verán un ejemplo en el código ya).

## Menú principal

En el menú principal de la aplicación solo deben reemplazar el “Apellido.Nombre” por sus propios datos.



## Base de datos e Interfaces



The screenshot shows a window titled "FrmBaseDeDatos" with a close button. The main content area has a title "Base de datos e Interfaces" in blue. Below the title, there are four input fields with labels: "Dni", "Nombre completo", "Nota 1", and "Nota 2". At the bottom, there is a button labeled "Aceptar".

Este módulo deberá insertar un alumno en la base de datos ( Al final de este documento encontrarán el script para crear la tabla en la base de datos).

A tener en cuenta:

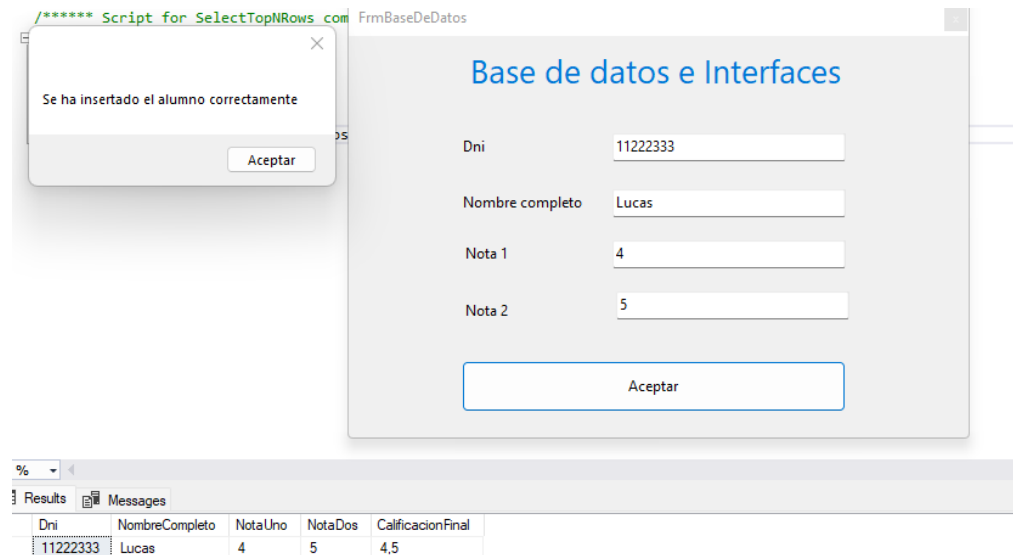
- Deberán validar que los datos sean correctos. Es decir:
  - Dni: Deberá tener un valor entre 10.000.000 y 45.000.000
  - Nombre: No puede ser ni nulo ni vacío.
  - Nota 1 / Nota 2: debe ser un valor entre 1 y 10 pero no decimal. Es decir, números redondos.
- La validación no se hará en el form. Será en el método de **instancia** Insertar de la clase **SqlManejador** ( también será de instancia). Si algun valor es incorrecto, se arrojará una excepcion de tipo DatosNoValidosException, que será capturada en formulario mostrando en un messagebox el mensaje.

Ejemplo:

- Error:

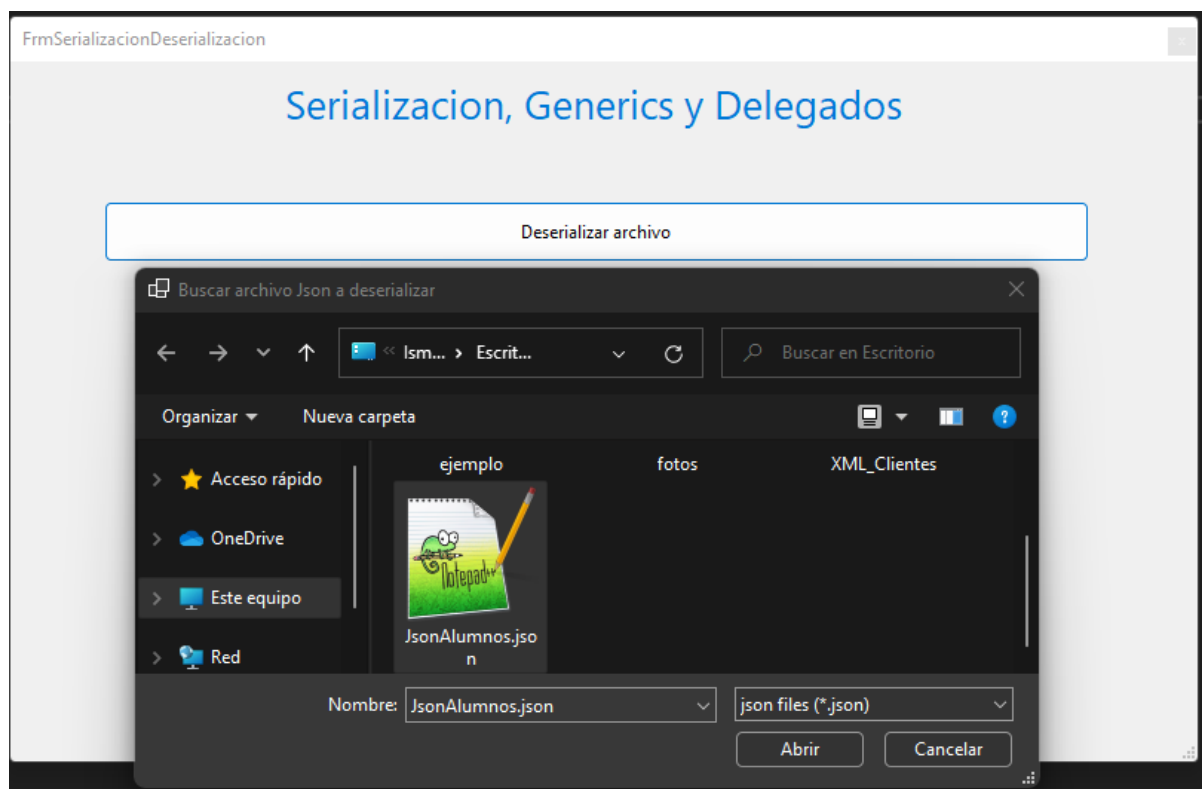
The screenshot shows a Windows application window titled "FrmBaseDeDatos". Inside the window, there is a form titled "Base de datos e Interfaces". The form contains four input fields with the following labels and values: "Dni" with value "-1", "Nombre completo" with value "pepe", "Nota 1" with value "11", and "Nota 2" with value "22". At the bottom of the form is a large button labeled "Aceptar". Overlaid on the right side of the form is a modal error message box. The message box has a close button (X) in the top right corner and contains the text "Error, uno de los parametros no es valido". At the bottom of the message box is a button labeled "Aceptar".

- Correcto



## Serialización, Generics y Delegados

Este módulo lo que hará es deserializar una lista de Alumnos, mostrarlo en el datagrid, desactivar el botón para deserializar otro archivo y mostrar en pantalla la leyenda “Documento serializado con éxito”.



FrmSerializacionDeserializacion

## Serializacion, Generics y Delegados

Deserializar archivo

Dni	NombreCompleto	NotaPrimerParcial	NotaSegundoParcial	CalificacionFinal
26482111	Eudora Kingsly	6	10	8
23117607	Egan Easlea	6	5	5,5
41454931	Neall Burlay	1	5	3
36347385	Ezmeralda Keeri	4	7	5,5
16944741	Buddie Sheldon	8	7	7,5
37561893	Shoshanna Balwin	4	6	5
41898940	Chip Kissell	6	7	6,5
19130301	Jerome Giorgi	1	9	5
32831918	Dorothee Lea	9	1	5
38550800	Cornelius Yurbie	0	1	5

Documento deserializado con exito

## Programación MultiHilos ( Task )

El botón comenzar carga simulará una carga de alumnos mediante una task.

Al presionar Cancelar Carga Alumnos se deberá cancelar esa tarea.

Dicho método hará el proceso de::

- Agregar un alumno a la lista
- Actualizar el datagrid con la lista
- Repetirá este proceso cada 2 segundos hasta que se cancele la tarea presionando el botón "Cancelar Carga..."

FrmProgramacionMultiHilo

Programación multihilos ( Task )

Dni	NombreCompleto	NotaPrimerParcial	NotaSegundoParcial	CalificacionFinal
18970973	Sudlow,Deeann	4	1	2,5
10510979	Tumber,Gabie	8	7	7,5
24661605	Henriques,Sadie	10	5	7,5
18475930	Sudlow,Riane	8	4	6
24714648	Wattins,Gage	10	5	7,5
24288187	Beazley,Sammie	10	1	5,5

Comenzar carga Alumnos

Cancelar Carga Alumnos

## Eventos

Se presionará el botón “Agregar curso a la carrera” hasta que el cupo de carreras de la universidad esté lleno.

Al llenarse, se deshabilitarán los componentes y se mostrará por pantalla el siguiente mensaje en un messagebox “La carrera se encuentra completa. Agradecemos su participación. Los esperamos la próxima “.

Deberán modificar el método toString() de carrera para que también muestre la clasificación de la carrera y no solo el nombre.

*Tip: La propiedad **GetUnaCarrera** de la clase estática **GeneradorDeDatos** devuelve una carrera aleatoria.*



FrmEventos



## Eventos

A large, empty rectangular box with a thin gray border, likely intended for displaying a list of events or a detailed view of an event.

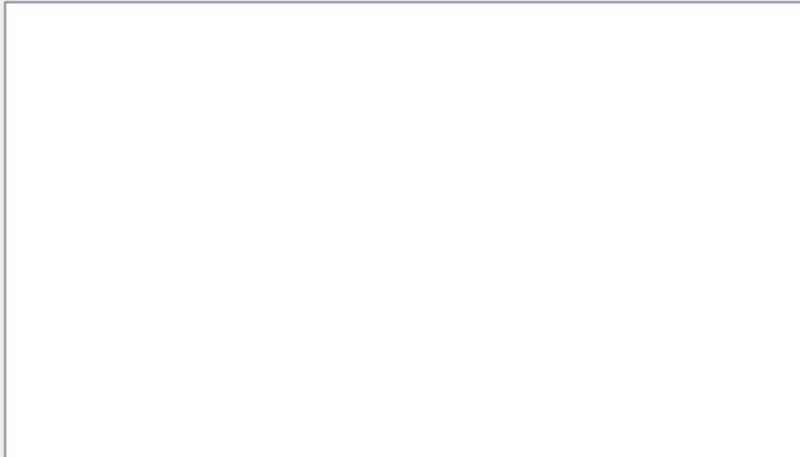
Agregar Carrera

## Eventos

Biodiversidad - Puntaje del curso: 9  
Programador de Sistemas - Puntaje del curso: 2  
Guionista de Radio y TV - Puntaje del curso: 9  
Actividad física y deportiva - Puntaje del curso: 4  
Ciencia Política - Puntaje del curso: 4

Agregar Carrera

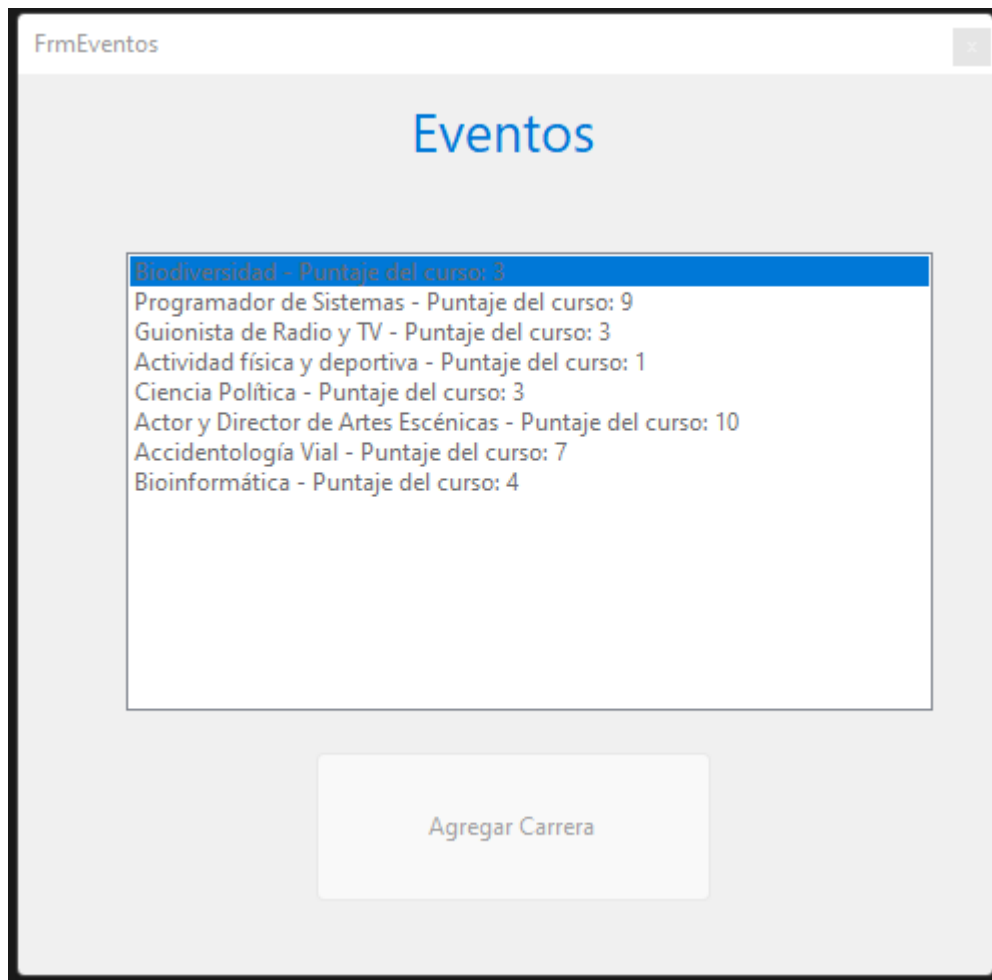
## Eventos



Agregar Carrera

La carrera se encuentra completa. Agradecemos su participacion.  
Los esperamos la proxima

Aceptar



## Test Unitarios

Se piden **dos** de los siguientes casos:

- Testear que al insertar un alumno con valores incorrectos, se arroja correctamente la excepción de tipo `DatosNoValidosException`.
- Testear que un alumno se insertó correctamente. (Tip: El método `ExecuteNonQuery()` devuelve el número de rows insertadas, por lo tanto si es mayor a 0, quiere decir que se insertó correctamente).
- Testear que los métodos **GetUnAlumno** y **GetUnaCarrera** devuelven datos válidos.

## Script Base de datos:

- 1) Crear una base de datos llamada **SP\_2D\_2022\_C1**
- 2) Correr el siguiente script

```
USE [SP_2D_2022_C1]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Alumnos](
    [Dni] [int] NOT NULL,
    [NombreCompleto] [nvarchar](50) NOT NULL,
    [NotaUno] [float] NOT NULL,
    [NotaDos] [float] NOT NULL,
    [CalificacionFinal] [float] NOT NULL
) ON [PRIMARY]
```