



Laboratorio II - TP3

ECO CENTER - School of English

Bianca Casetta

Introducción

Este programa refleja un programa utilizado por un instituto privado de inglés, donde se pueden cargar nuevos estudiantes, modificarlos, darlos de baja del sistema, listarlos y acceder a la información de los cursos dados.

Aspecto/Navegación

Al ejecutar el programa, solo se ve el logo y el menú horizontal como “pestañas” con las opciones Estudiantes, Cursos y Salir.

Según la opción horizontal que se seleccione, se despliega un menú vertical en el caso de Estudiantes y Cursos. Si se clickea el logo, se vuelve al estado original en el que empieza a correr el programa. Si se clickea Salir, se reconfirma la salida y se cierra el programa.

Según la opción vertical que se seleccione, se despliega un formulario acorde a la acción seleccionada a la derecha del menú vertical.

Aclaraciones sobre clases de Entidades

Instituto - Clase que contiene una lista de los diferentes tipos de estudiantes.

EstudianteParticular - Aquel que solo va al instituto a repasar algo específico y de manera individual. Paga por hora y no cuenta con una nivelación por ser personalizada.

EstudianteCurso - Aquel que cursa en el instituto durante el ciclo lectivo y de manera grupal. Paga por mes y tiene un Curso asociado.

EstudianteExterno - Aquel que no cursa en el instituto, solo asiste en época de exámenes para rendir y no tiene Curso, solo Nivel.

Temas Aplicados

10. Excepciones

Creé dos Exceptions propias:

CamposIncompletosException - Si aunque sea UN campo textBox queda sin completar al dar de Alta o Modificar un Estudiante, se lanza la Exception porque todos los campos son obligatorios. Luego con el mensaje de la Exception se muestra un mensaje de error por MessageBox.

The image shows a web application interface for 'ALTA DE ESTUDIANTE' (Student Registration). At the top, there are four red circular icons: a person with a plus sign (ALTA), a person with a minus sign (BAJA), a person with a checkmark (MODIFICACION), and a folder (LISTADO). Below these are four more red circular icons: a graduation cap (ESTUDIANTES), a book (CURSOS), an exit sign (SALIR), and a checkmark (CONFIRMAR). The main form is titled 'ALTA DE ESTUDIANTE' and contains several text boxes for data entry: NOMBRE(S), APELLIDO(S), DNI, FECHA DE NAC, GENERO, TIPO DE ESTU, NIVEL (with a dropdown menu showing 'Primer_Año'), DOMICILIO, CALLE, and ALTURA. There is also a 'DATOS DE CONTACTO' section with 'TELEFONO' and 'E-MAIL' fields. A modal dialog box titled 'Campos incompletos' is displayed over the form, showing a red 'X' icon and the message 'Hay campos incompletos. Todos los campos son obligatorios.' with an 'OK' button. At the bottom right, there are three red circular icons: a broom (LIMPIAR), a checkmark (CONFIRMAR), and a red 'X' (CANCELAR).

EstudianteYaExistenteException - Si se intenta dar de alta a un estudiante cuyo DNI ya se encuentra registrado en el sistema, se lanza la Exception porque no se admiten dos Estudiantes con el mismo DNI. Luego con el mensaje de la Exception se muestra un mensaje de error por MessageBox.



ALTA DE ESTUDIANTE

NOMBRE(S)	<input type="text" value="Bianca"/>	DOMICILIO	<input type="text" value="lalala"/>	ALTURA	<input type="text" value="123"/>
APELLIDO(S)	<input type="text" value="Casetta"/>	CALLE	<input type="text" value="Buenos Aires"/>		
DNI	<input type="text" value="3964"/>		<input type="text" value="asd"/>		
FECHA DE NACIMIENTO	<input type="text" value="30/08/2000"/>		<input type="text" value="1234"/>		
GENERO	<input type="text" value="Femenino"/>				
TIPO DE ESTUDIANTE	<input type="text" value="Estudiante"/>				
NIVEL	<input type="text" value="Primer Año"/>				
DATOS DE CONTACTO					
TELEFONO	<input type="text" value="1122223333"/>				
E-MAIL	<input type="text" value="bianca@gmail.com"/>				



11. Pruebas Unitarias

Se prueba si las sobrecargas de igualdad entre Instituto/Estudiante y Estudiante/Estudiante retornan True.

```
[TestMethod]
public void EstudianteSobrecargaIgualdad_CuandoElEstudianteTieneElMismoDni_DeberiaRetonarTrue()
{
    //Arrange
    Domicilio dom = new Domicilio("Calle Falsa", 123, "Springfield", "Missouri", 1234);
    Estudiante e1 = new EstudianteParticular("Juan", "Perez", 12345678, new DateTime(1990, 08, 25), "Masculino", 1122223333, dom,
    Estudiante e2 = new EstudianteExterno("Maria", "Perez", 12345678, new DateTime(2000, 12, 31), "Femenino", 5491177778888, dom,

    //Act
    bool estudianteYaIngresado = e1 == e2;

    //Assert
    Assert.IsTrue(estudianteYaIngresado);
}

[TestMethod]
public void InstitutoSobrecargaIgualdad_CuandoElEstudianteEstaEnElInstituto_DeberiaRetonarTrue()
{
    //Arrange
    Instituto instituto = new Instituto();
    Domicilio dom = new Domicilio("Calle Falsa", 123, "Springfield", "Missouri", 1234);
    Estudiante estudiante = new EstudianteParticular("Juan", "Perez", 12345678, DateTime.Now, "Masculino", 1122223333, dom, ETipoE

    //Act
    instituto.Estudiantes.Add(estudiante);
    bool estudianteExistente = instituto == estudiante;

    //Assert
    Assert.IsTrue(estudianteExistente);
}
```

12, 14 y 15. Generics, Archivos y Serialización

Serializer - Es una clase que (de)serializa tipos de referencia genéricos. En implementación, el tipo T puede ser List<Estudiante> o List<Curso>. Al iniciar el programa, si existe un archivo de datos de Estudiantes, se carga y se puede ver directamente en la opción Listado. Si se da de alta, se modifica o se da de baja a un Estudiante, se crea (si no existe) o se sobrescribe el archivo. Los Estudiantes están en un archivo .xml.

En el caso de la información de Cursos, no son datos que puedan modificarse, por lo que a partir del Nivel de un curso se obtiene toda la información que lo acompaña (cuota, horarios, docente). En caso de querer exportar esta información, se crea un archivo serializado en .json con los datos. No se serializan los Estudiantes que pertenecen a cada curso (que sí se muestran en rtbInfoCurso) porque no son un atributo de la clase Curso.

Tanto el .xml como el .json se encuentran en la carpeta `\Casetta.Bianca.2A.TPFinal\TP3\bin\Debug\net5.0-windows\`

13. Interfaces

ISubForm - Dado que varios Forms distintos comparten la funcionalidad de instanciar un subForm para mostrar, creé esta interfaz que contiene un método AbrirSubForm, que recibe el Form a instanciar como primer parámetro y el Panel donde se mostrará dicho Form como segundo parámetro.

```
/// <summary>
/// A partir de este FrmHome, se llama a otro Form para abrirlo en el Panel correspondiente
/// </summary>
/// <param name="subForm"></param>
/// <param name="panel"></param>
8 references
void ISubForm.AbrirSubForm(Form subForm, Panel panel)
{
    if(subFormMostrado is not null)
    {
        subFormMostrado.Close();
    }

    subFormMostrado = subForm;
    subForm.TopLevel = false;
    subFormMostrado.Dock = DockStyle.Fill;
    panel.Controls.Add(subForm);
    subForm.BringToFront();
    subForm.Show();
}
```