

Profesor:  
Neiner Maximiliano

### Nota:

Se recomienda utilizar los prefijos para cada elemento dentro de las aplicaciones de tipo Windows Form como las convenciones al nombrar clases, métodos, atributos, etc., ya que se tendrán en cuenta al momento de evaluar este trabajo práctico.

Todas las clases intervinientes en los ejercicios de tipo WindowsForm, deben ser creadas en proyectos de tipo Biblioteca de clases.

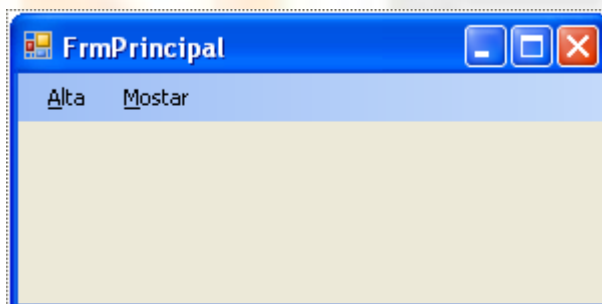
```
try{
    FrmPrincipal frmInicio = new FrmPrincipal();
    Application.Run(frmInicio);
}
catch(Exception ex)
{
    MessageBox.Show("Error: " + ex.Message, "Error desconocido");
}
```

Es obligatorio el uso del bloque `try catch finally` en todas las instrucciones que puedan generar excepciones.

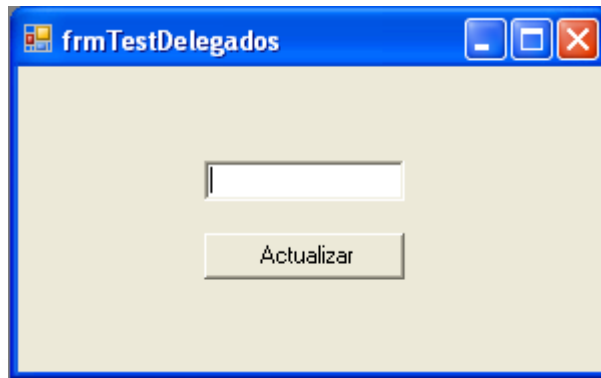
## Delegados en aplicaciones Windows Form

### Aplicación Nº 1

Se deberá construir una aplicación de tipo Windows Form, que contenga como formulario principal (*frmPrincipal*) un formulario de tipo **MDI**. Las configuraciones sobre el formulario se deberán hacer siempre en tiempo de ejecución. El formulario se mostrará maximizado. Este formulario tendrá un menú de opciones tal como se muestra en la siguiente figura:



La opción *Alta* tendrá dos *submenús* que serán: **Test Delegados** y **Alumno**. Cuando el usuario pulse el submenú *Test Delegados*, se creará una instancia de **frmTestDelegados**, que se mostrará en el centro de la pantalla y cuyo propietario (**owner**) será *frmPrincipal*. El formulario será similar al de la siguiente figura:



Una vez que el formulario se encuentre visible se deberá seleccionar la opción de menú **Mostrar**, la cual creará una instancia de **frmDatos** (cuyo propietario también será **frmPrincipal**).

El formulario contendrá, en un primer momento, un control **Label** como único control.



**frmDatos** expondrá un método público llamado **ActualizarNombre**, que recibirá por parámetro un valor de tipo **String**. Dicho método no retornará ningún valor.

La funcionalidad que posee el método anteriormente descrito será simplemente cambiar el valor de la propiedad **Text** del control Label.

Se pide que el usuario ingrese un nombre por el TextBox de la instancia de *frmTestDelegados* y que al pulsar el botón **btnActualizar**, se cambie el valor del Label de la instancia de *frmDatos*.

**Nota:** Se deberá utilizar un **delegado**, convenientemente diseñado, declarado en un archivo de código (buscar plantilla en Visual Studio) y cuya variable asociada sea un atributo público de *frmPrincipal*, nombrado como *actualizarNombrePorDelegado*.

## Aplicación N° 2

Agregar un control de tipo **PictureBox** a *frmDatos* y un control de tipo **Button** y otro de tipo **OpenFileDialog** en *frmTestDelegados*.

Se pide que se configure el control *OpenFileDialog*, a través de un método privado (**ConfigurarOpenSaveFileDialog**) invocado en el constructor de *frmTestDelegados*, para que:

Su directorio inicial sea **Mis Imágenes** (utilizar `Environment.GetFolderPath`).

- Su extensión por defecto sea .jpg.
- Que verifique que el archivo no exista.
- Que verifique que el directorio no existe.
- Que no se permita la multi-selección de archivos a abrir.
- Y que el título de la ventana de dialogo sea **"Seleccione una foto..."**.

Al pulsar el botón **btnBuscarFoto**, se le permitirá al usuario seleccionar una imagen. La ruta se deberá guardar en un atributo privado de *frmTestDelegados*.

Agregar un método público en *frmDatos*, llamado **ActualizarFoto**, con la misma firma que el método *ActualizarNombre*, que cambie el valor de la propiedad **ImageLocation** del control PictureBox.

Al pulsar el botón **btnActualizar** se deberá mostrar la imagen seleccionada en el PictureBox de *frmDatos*.

**Nota:** Generar otro atributo en *frmPrincipal* (*actualizarFotoPorDelegado*), del mismo tipo de **delegado** del ejercicio anterior, que "apunte" al método *ActualizarFoto* de la instancia de *frmDatos*.

### Aplicación Nº 3

Agregar un proyecto de tipo **Class Library** llamado **Entidades**, que posea una clase **Alumno**.

Las características de la clase serán:



Diseñar un formulario que permita el alta de un alumno (**frmAltaAlumno**). Tener en cuenta los atributos de la clase Alumno para su construcción.

The image shows a Windows form titled 'frmAltaAlumno'. It contains four text boxes for data entry, each with a label above it: 'Apellido', 'Nombre', 'Dni', and 'Foto'. At the bottom of the form, there are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

Para capturar la foto del alumno se seguirán los mismos pasos que en el ejercicio anterior, con la salvedad que el **OpenFileDialog** se abrirá al hacer **dobles clic** con el botón izquierdo del mouse sobre el TextBox *txtFoto*.

También se pide diseñar un formulario (**frmDatosAlumno**), que herede de *frmAltaAlumno* y que agregue un control de tipo PictureBox.

Además este formulario deberá exponer un método público (**ActualizarAlumno**) que no retornará ningún valor y recibirá un parámetro de tipo *Entidades.Alumno* y como segundo parámetro un objeto de tipo *EventArgs*.

En *frmPrincipal* modificar el menú Mostrar y agregarle un submenú llamado **Test**, con la misma funcionalidad del **Ejercicio 1**, y otro llamado **Alumno**, que muestre una instancia de *frmDatosAlumno* como MDIChild de *frmPrincipal*.

Lo que se debe lograr es crear un objeto de tipo *Entidades.Alumno* en *frmAltaAlumno* y que se muestren todos sus datos (incluida su foto) en *frmDatosAlumnos*.

**Nota:** Se deberá utilizar un **delegado**, convenientemente diseñado, declarado en un archivo de código (buscar plantilla en Visual Studio) y cuya variable asociada sea un atributo público de *frmPrincipal*, nombrado como *MostrarAlumnoPorDelegado*.