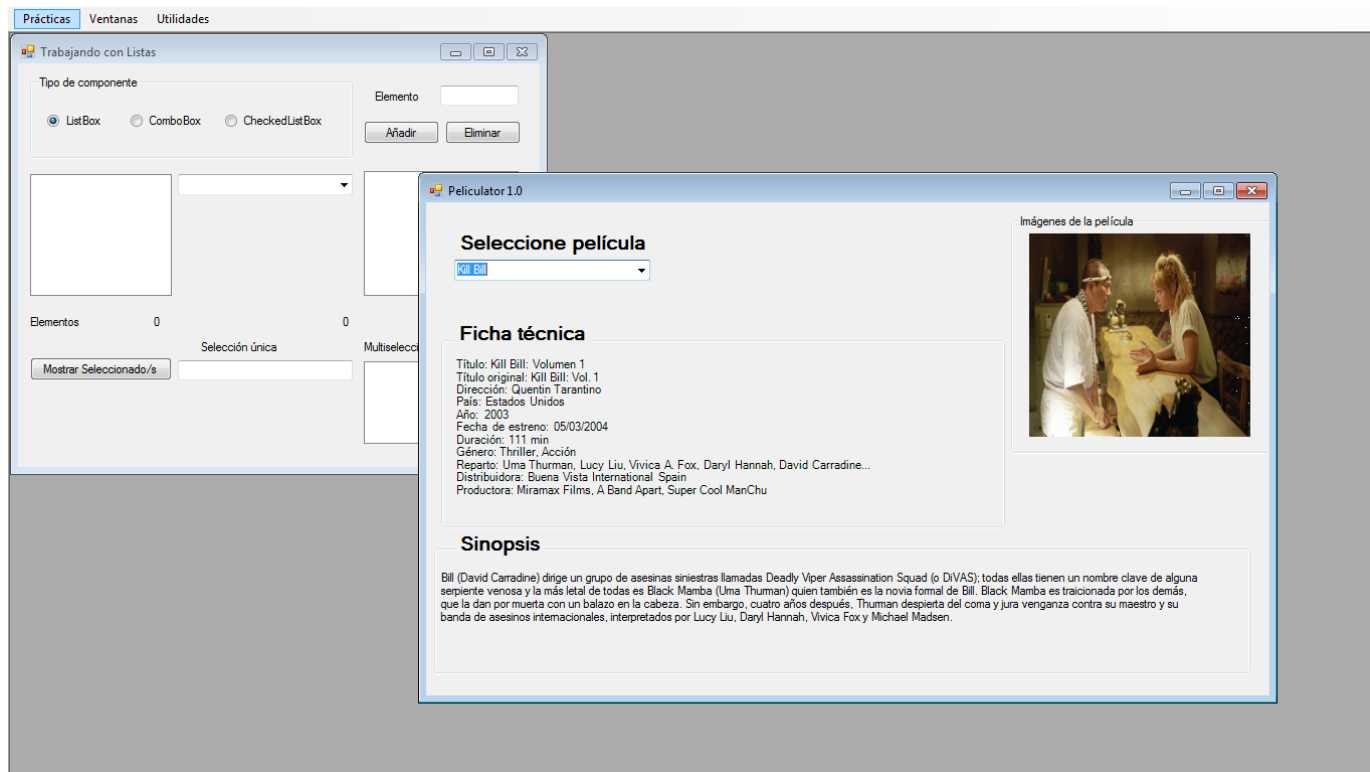


UN PROYECTO MDI QUE INTEGRA OTROS PROYECTOS

Introducción

- Ya sabemos como invocar desde un formulario MDI otros formularios.
- Ahora veremos como invocar distintos proyectos ya hechos desde el formulario MDI.
- Lo veremos mediante el siguiente ejemplo:

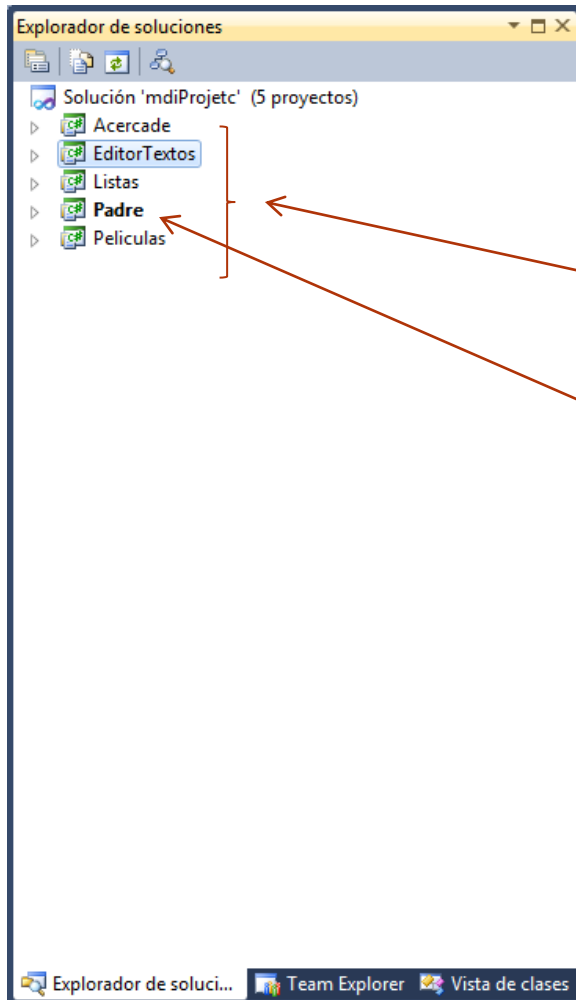


Introducción

- Como podemos observar en la imagen, en el menú del Formulario Padre hemos incluido una opción a partir de la cual accederemos a todos los ejercicios iniciales del curso:
 - Por ejemplo Listas...
- A continuación, vamos a desarrollar *—paso a paso- todo del* proceso de incorporación de un **Proyecto existente a una Solución.**
- En el ejemplo, vamos a agregar las prácticas realizadas a lo largo del curso

Empezamos...

- Vista inicial de la Solución:



Como podemos observar esta Solución la integran los siguientes proyectos:

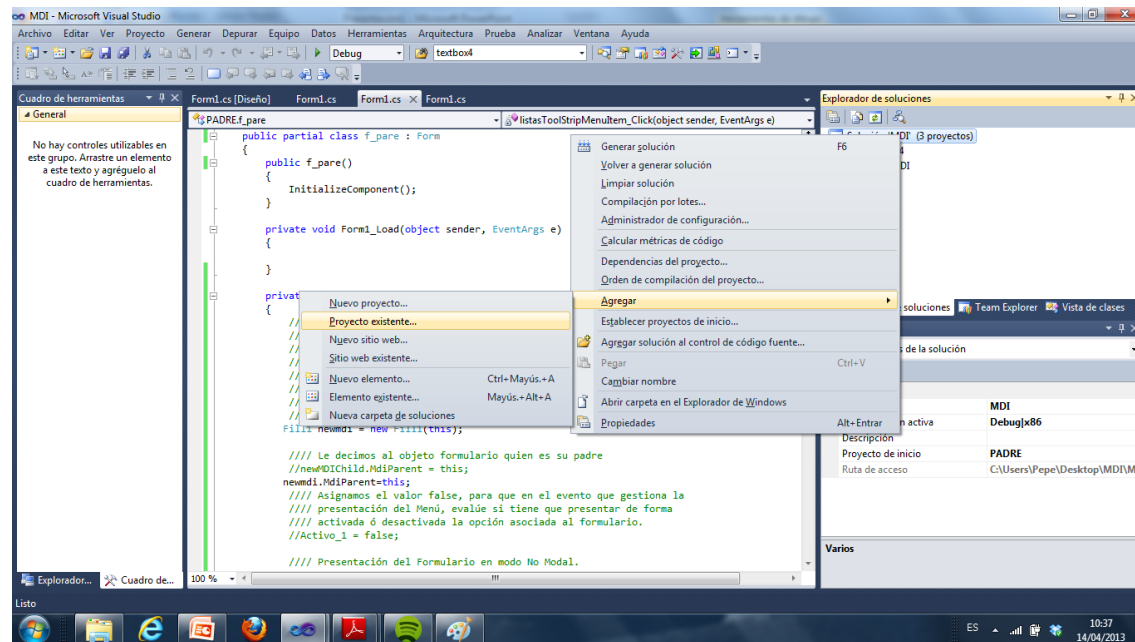
Proyectos hijos:

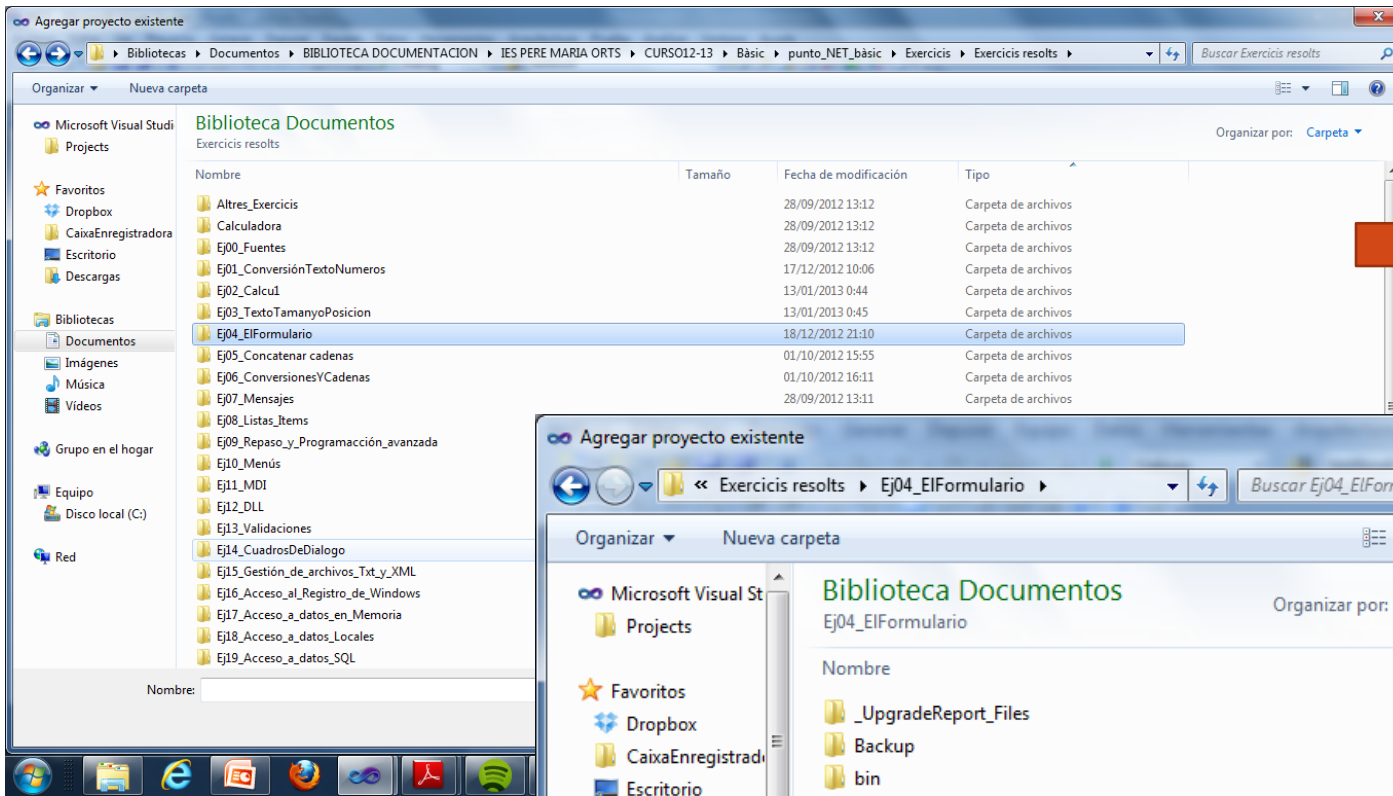
- Listas
- Acerca de
- Películas
- Editor de texto

..y el Proyecto Padre (principal)

- Padre

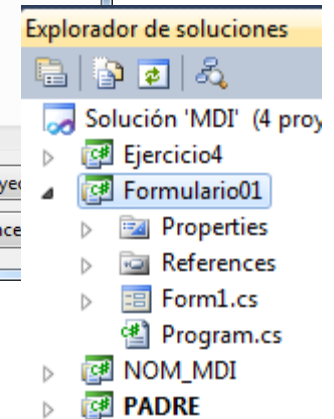
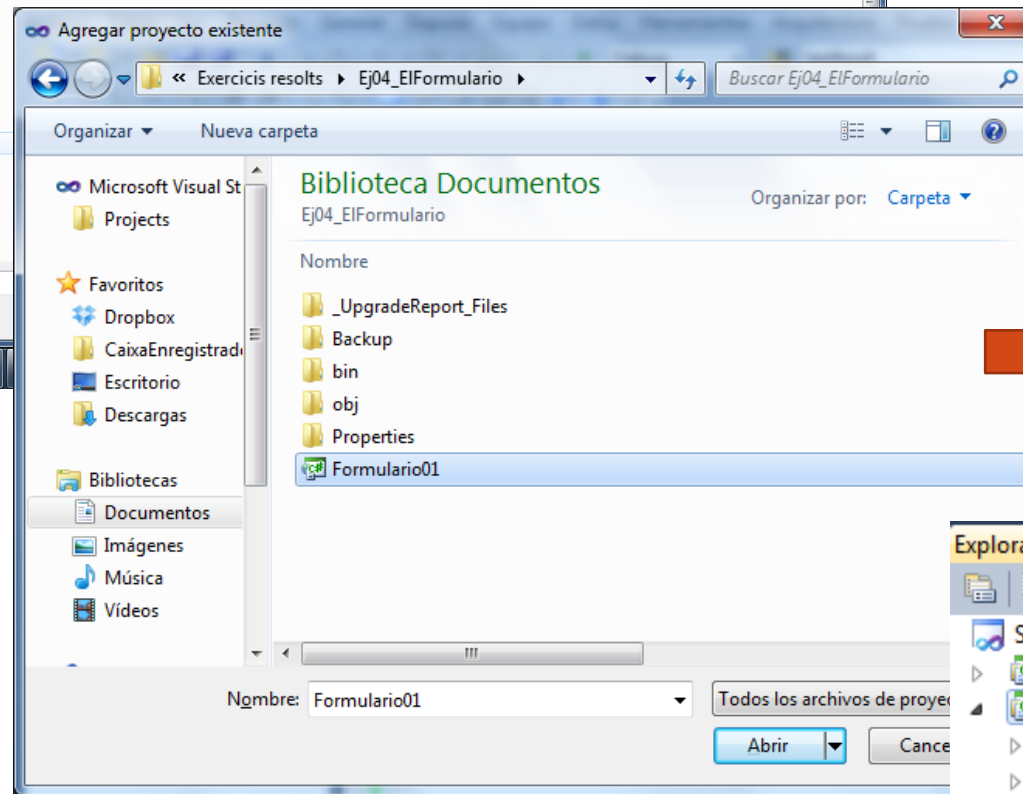
- El primer paso que tendremos que realizar será el de incorporar a la **Solución un Proyecto existente**. Para ello, nos situamos en el Explorador de Soluciones sobre el nombre de la Solución y accedemos al menú: **Archivo – Agregar ... - Proyecto existente.** (como siempre...)



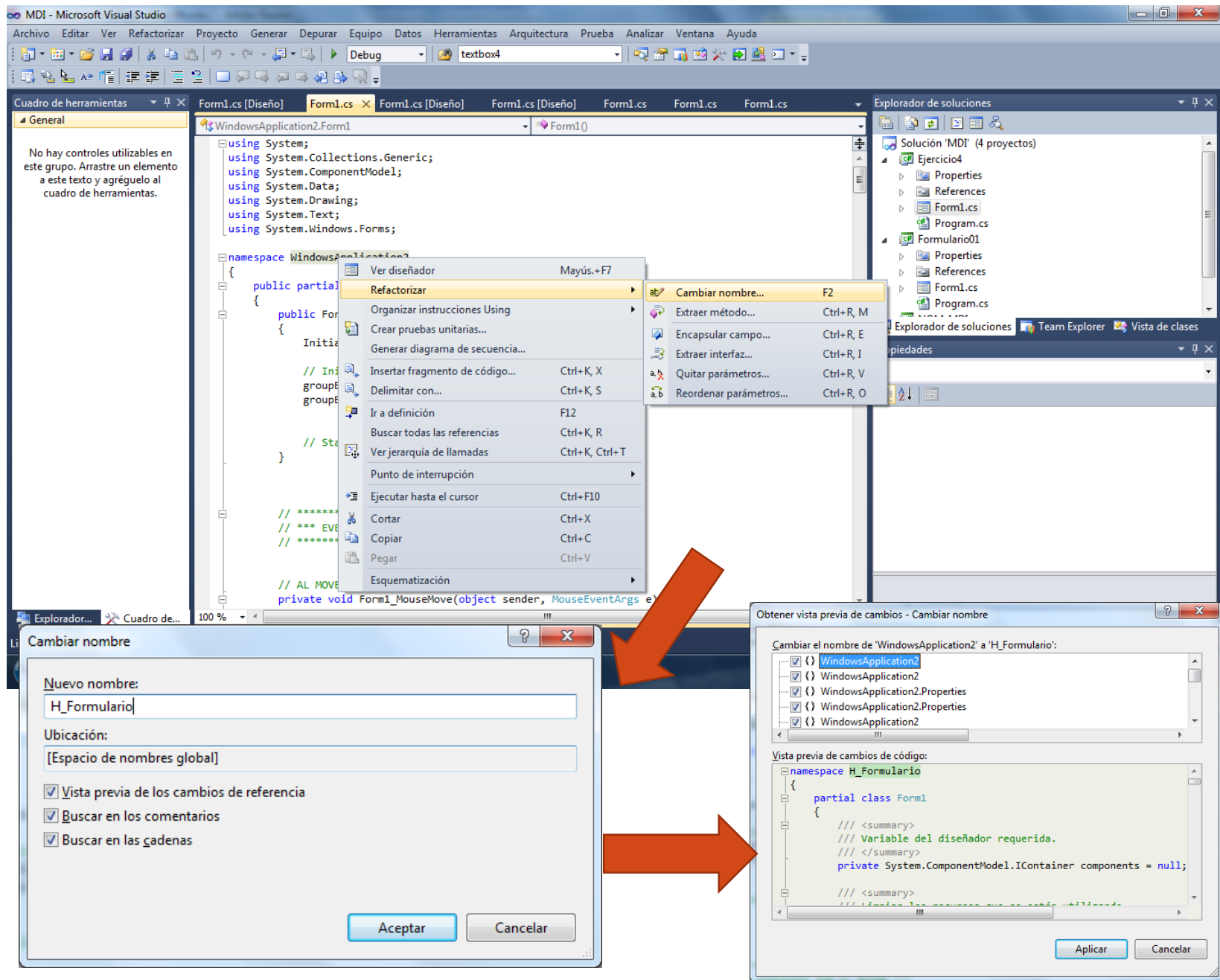


Seleccionamos el archivo correspondiente al **Proyecto** y pulsamos sobre el botón: **Abrir.**

Finalmente, observaremos como se habrá añadido la entrada



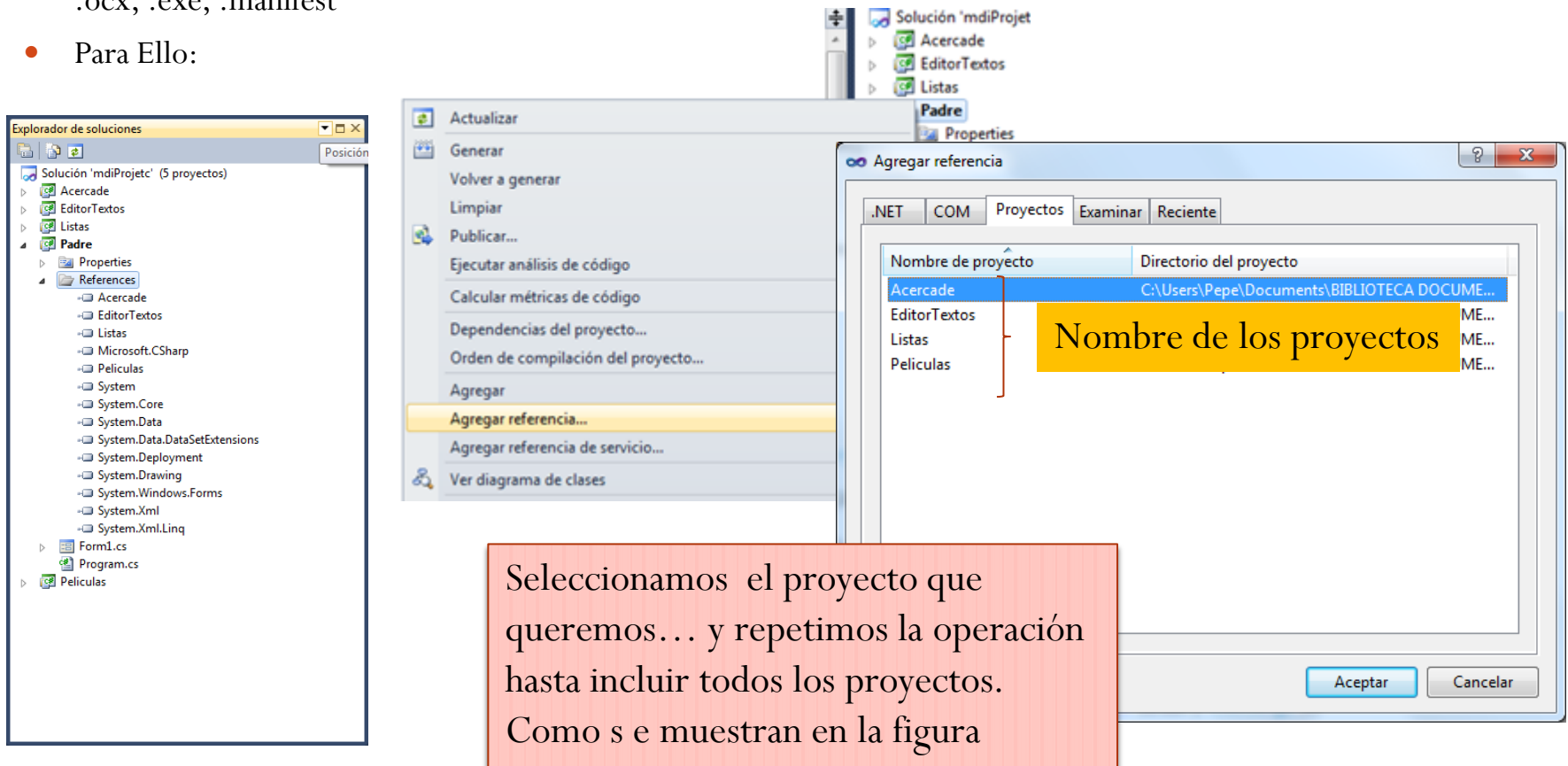
- A partir de este momento, ya podemos personalizar el contenido del proyecto para que pueda:
 - Ser **referenciado desde del proyecto Padre**.
- A continuación, podemos realizar una serie de tareas que nos facilitarán el proceso de **identificación y referencia de los recursos del Proyecto**:
 - Cambiar el nombre del **Espacio de nombre correspondiente y del Formulario** del Proyecto Hijo.
 - Para ello, podemos emplear una herramienta muy importante en todos los entornos de desarrollo actuales:
 - *‘Refactorizar’; es decir, podemos seleccionar el nombre e indicarle al entorno que realice el cambio en todos los recursos del proyecto.*
 - *Para ello, seleccionamos el nombre del espacio de nombres:*
 1. *WindowsApplication1 y pulsamos sobre el botón derecho del ratón, a continuación, seleccionamos la opción: **Refactorizar – Cambiar nombre**.*
 2. *Lo mismo con Form1*



- Este cambio que hemos realizado en el nombre del espacio de nombres, resultará interesante realizarlo cuando añadamos a la **Solución**, *Proyectos que hayan sido creados empleando los valores por defecto.*
 - *(También resultará interesante —en el caso de añadir proyectos generados con valores por defecto- modificar (refactorizando) el nombre del Formulario (su valor por defecto es: **Form1**).)*
- En este caso, el nombre del espacio de nombres generado, tal y como hemos dicho siempre será: **WindowsApplication1**.
- En el caso de que desde el **Formulario Padre se acceda a los** recursos de los Formularios Hijos y viceversa (*caso evidente*), el siguiente paso que realizaremos será: **Crear las referencias entre los proyectos.**

Seguimos...

- Creamos las referencias entre los proyectos.
- **a) El Formulario 'Padre' creamos las referencias de los hijos.**
- Podemos agregar una referencia a diferentes tipos de archivos: .dll, .tlb, .olb, .ocx, .exe, .manifest
- Para Ello:



- A continuación, empleando la cláusula *using* agregaremos las referencias al espacio de nombres del **PADRE** :

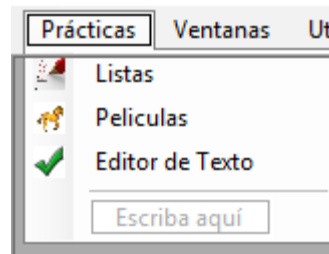
...¿De donde salen esos nombres ? Son los nombres que se le han cambiado en el proceso de refactorizar al namespace de cada proyecto

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
  
using Listas;  
using Peliculas;  
using EditorTextos;  
using Acercade;
```

```
namespace Listas  
{  
    public partial class F_Listas : Form  
    {  
    }  
}
```

...Invocar del padre a los hijos

- Partimos que tenemos nuestro menuStrip1 en nuestro formulario y que hemos creado una entrada para cada uno de los hijos como muestra la figura:



- El siguiente paso que realizaremos será generar el evento click, en nuestro ejemplo para las listas, del menú y teclear el código (fíjate en la imagen)

```
private void listasToolStripMenuItem_Click(object sender, EventArgs e)
{
    F_Listas Hijo1 = new F_Listas(); // Instanciamos el proyecto Lista
    Hijo1.MdiParent = this; //Le decimos que es hijo del padre
    Hijo1.Show(); // lo mostramos
}
```

¿Qué es F_Listas? El nombre que le he dado al formulario del proyecto **listas** en el proceso de refactorizar

```
using System.Text;
using System.Windows.Forms;

namespace Listas
{
    public partial class F_Listas : Form
    {
        public F_Listas()
        {
            InitializeComponent();
        }

        private void button1_Click(object
```

ASÍ EN TODO LOS PROYECTOS QUE QUERAMOS INCLUIR...

...Ya está... proyecto hecho!!!!!!

- ...Solo queda darle forma al menú con la gestión de ventanas... visto en las transparencias anteriores...

