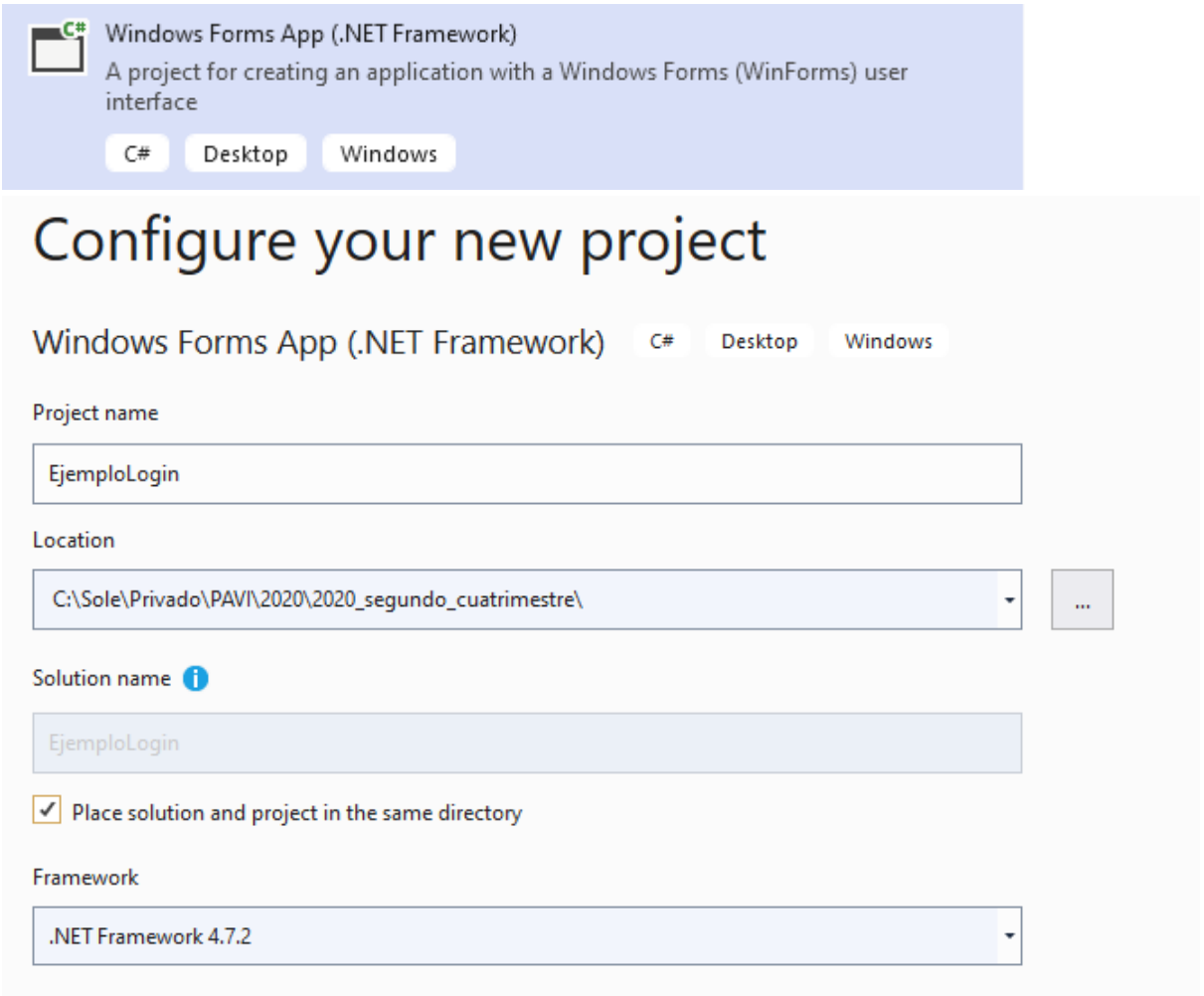
Proyecto Login WinForm C# en 5 pasos

El objetivo de este documento es que el lector diseñe y desarrolle una aplicación simple de inicio de sesión, en la cual un usuario pueda ingresar su nombre de usuario + password y la aplicación valide que las credenciales sean correctas.

1. Iniciando un nuevo Proyecto

Creamos un nuevo proyecto de tipo “Aplicación de Windows Forms”:

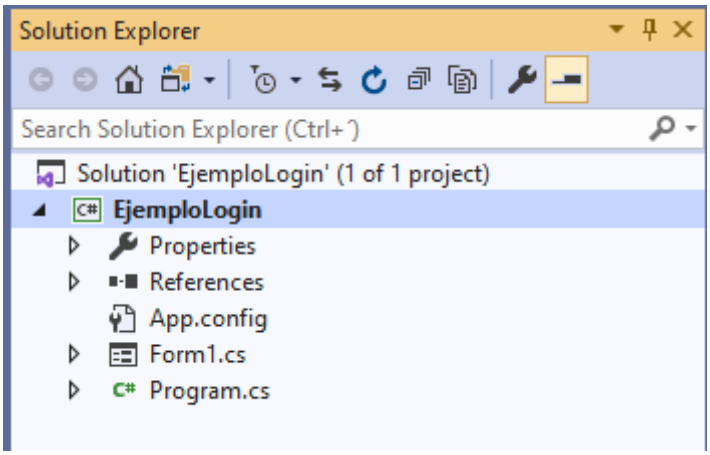


- Lo importante en este paso es elegir correctamente la **plantilla C# Desktop Windows Aplicación de Windows Form.**
- Se puede crear una carpeta para la solución y otra para el proyecto. Esto nos permite luego agregar más proyectos (nuevos o existentes) dentro de la misma solución. O bien elegir el mismo directorio para ambos.
- El proyecto al ser de tipo WinForm, se crea con un formulario llamado Form1 que podemos utilizar para comenzar con nuestro trabajo.
- Por defecto el directorio donde se crean las soluciones es la carpeta del usuario\ visual studio XX\ projects\ del usuario de Windows utilizado durante la instalación del IDE.

2. Conociendo el entorno de trabajo

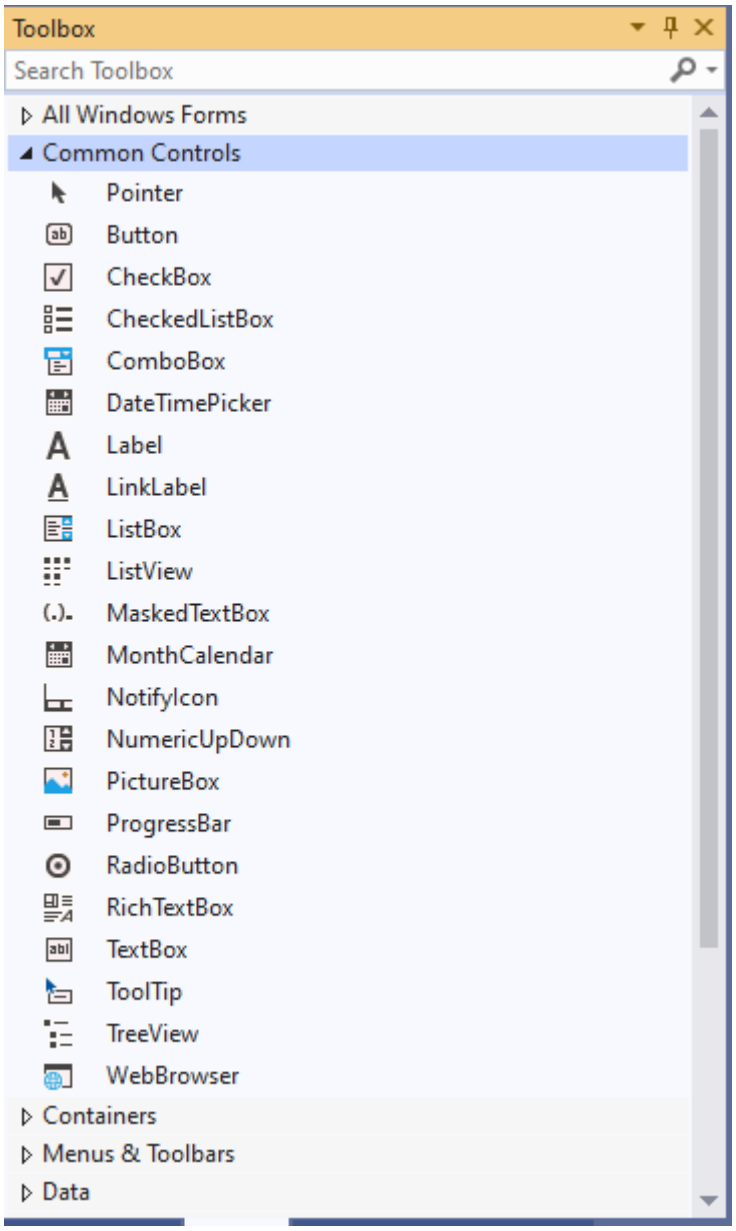
- Principales ventanas que vamos a utilizar para el desarrollo de los proyectos:

a) Explorador de soluciones: [opción: Ver>Explorador de Soluciones o Ctrl + Alt + L]



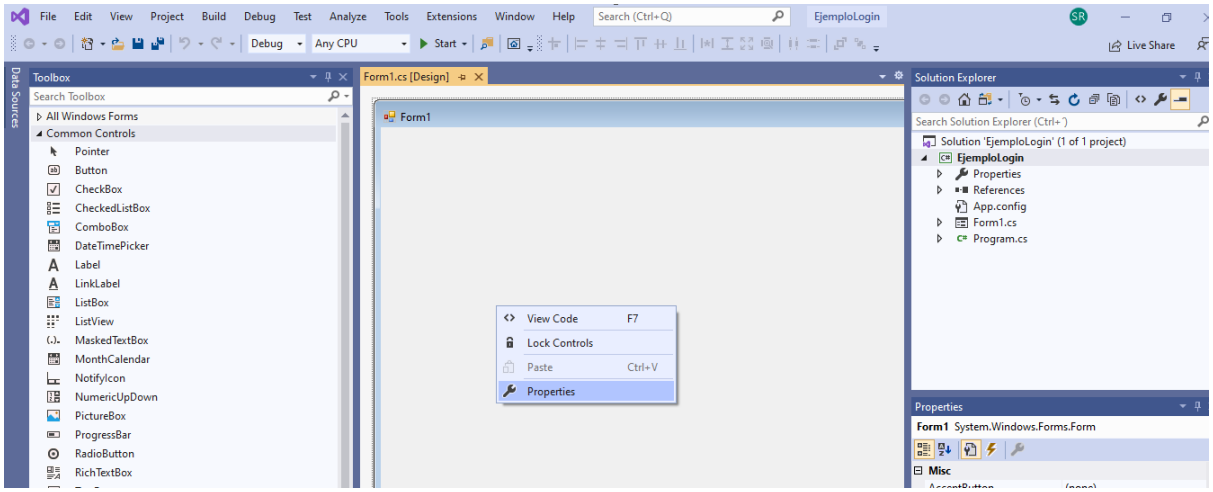
- Nos muestra una vista jerárquica de los archivos de la solución. Si hacemos click derecho sobre la solución o el proyecto, con la opción **Abrir carpeta en el explorador de archivos…** podremos navegar directamente en la organización física de carpetas donde se encuentra nuestra nuestro proyecto o solución.

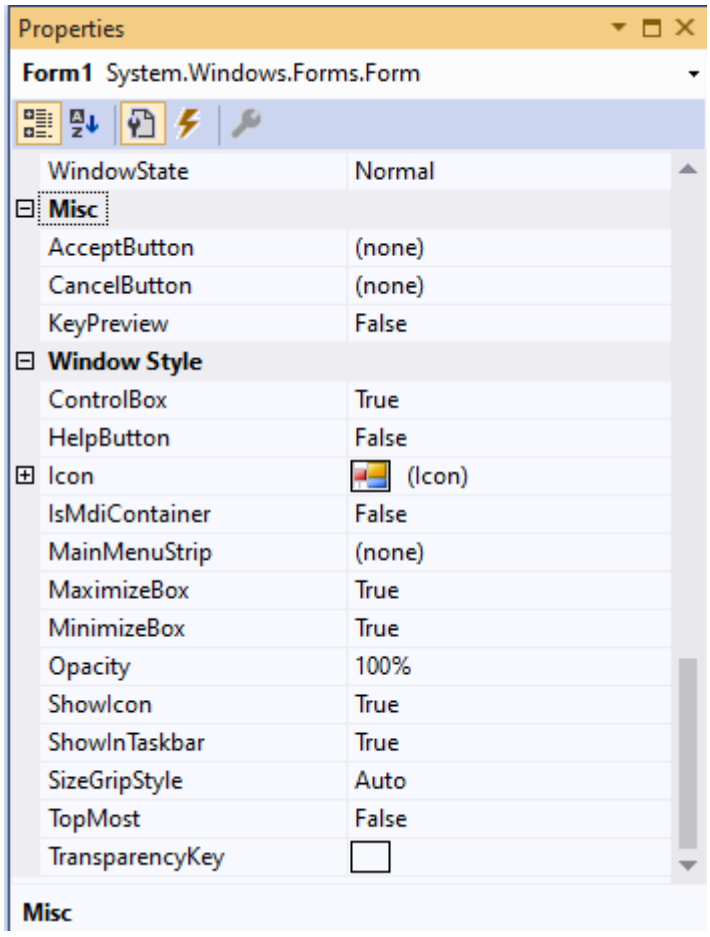
b) **Cuadro de herramientas:** [opción: Ver>Cuadro de Herramientas o Ctrl + Alt + X]



- Permite seleccionar tanto los componentes a dibujar dentro de los formularios como el resto de los objetos utilizados para el diseño de reportes y origen de datos del proyecto.

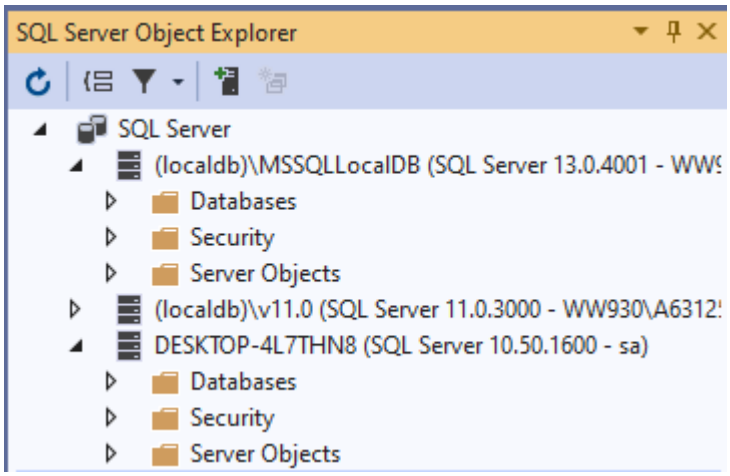
c) Propiedades: [Sobre un objeto seleccionado, Alt + Enter]


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
EjemploLogin
Toolbox
Form1.cs [Design]
Form1
Solution Explorer
Solution 'EjemploLogin' (1 of 1 project)
EjemploLogin
Properties
References
App.config
Form1.cs
Program.cs
View Code F7
Lock Controls
Paste Ctrl+V
Properties
Form1 System.Windows.Forms.Form
Misc


Properties
Form1 System.Windows.Forms.Form
WindowState Normal
Misc
AcceptButton (none)
CancelButton (none)
KeyPreview False
Window Style
ControlBox True
HelpButton False
Icon (Icon)
IsMdiContainer False
MainMenuStrip (none)
MaximizeBox True
MinimizeBox True
Opacity 100%
ShowIcon True
ShowInTaskbar True
SizeGripStyle Auto
TopMost False
TransparencyKey
Misc

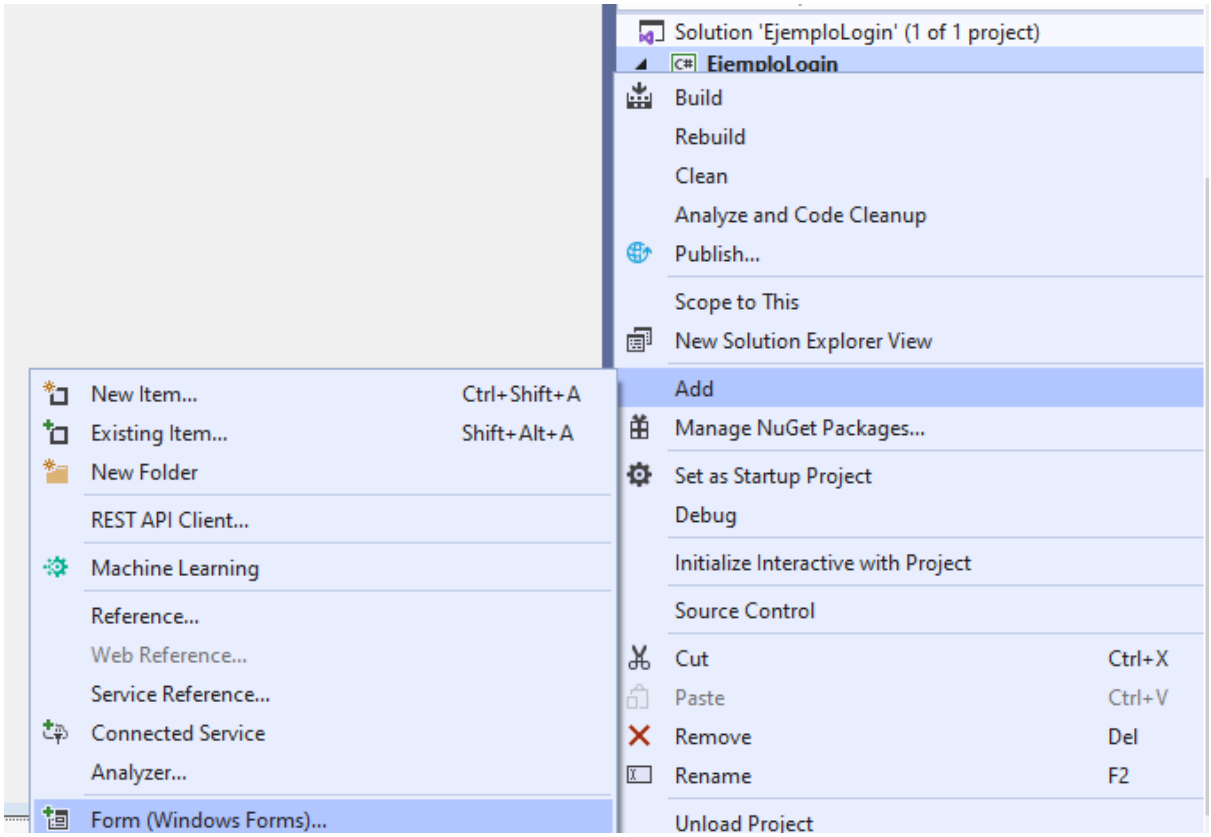
- Esta ventana permite ver y asignar valores a las propiedades de los objetos seleccionados, sean formularios, componentes o cualquier tipo de objeto incluido en el proyecto.

    d) Explorador de Objetos SqlServer: [opción: Ver>Explorador de Objetos Sql Server]



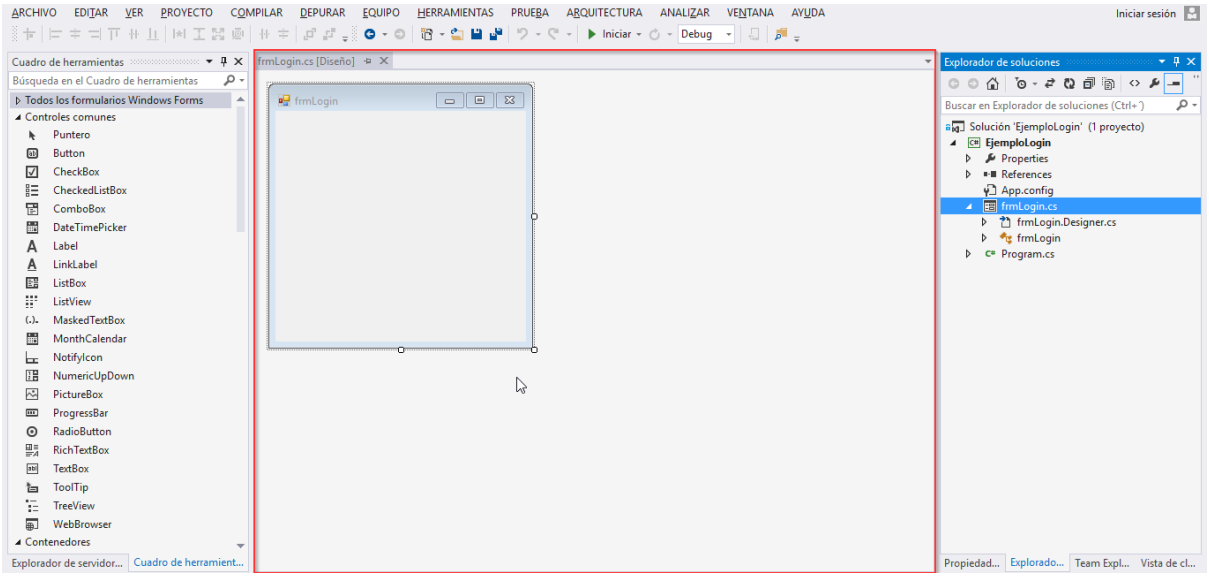
- Permite navegar por los servidores de datos instalados en el equipo y generar conexiones a distintas fuentes de datos.

3. Agregando un formulario de Login



- Como se mencionó anteriormente podemos agregar un nuevo formulario o bien renombrar el creado por defecto indicando que se actualicen todas las referencias existentes (opción **Cambiar nombre**).
- Una vez creado el formulario, tenemos 2 vistas principales sobre las que necesitamos trabajar:

1) **Vista de Diseño:** [Mayus + F7 o bien sobre el formulario click derecho opción: Ver diseñador]

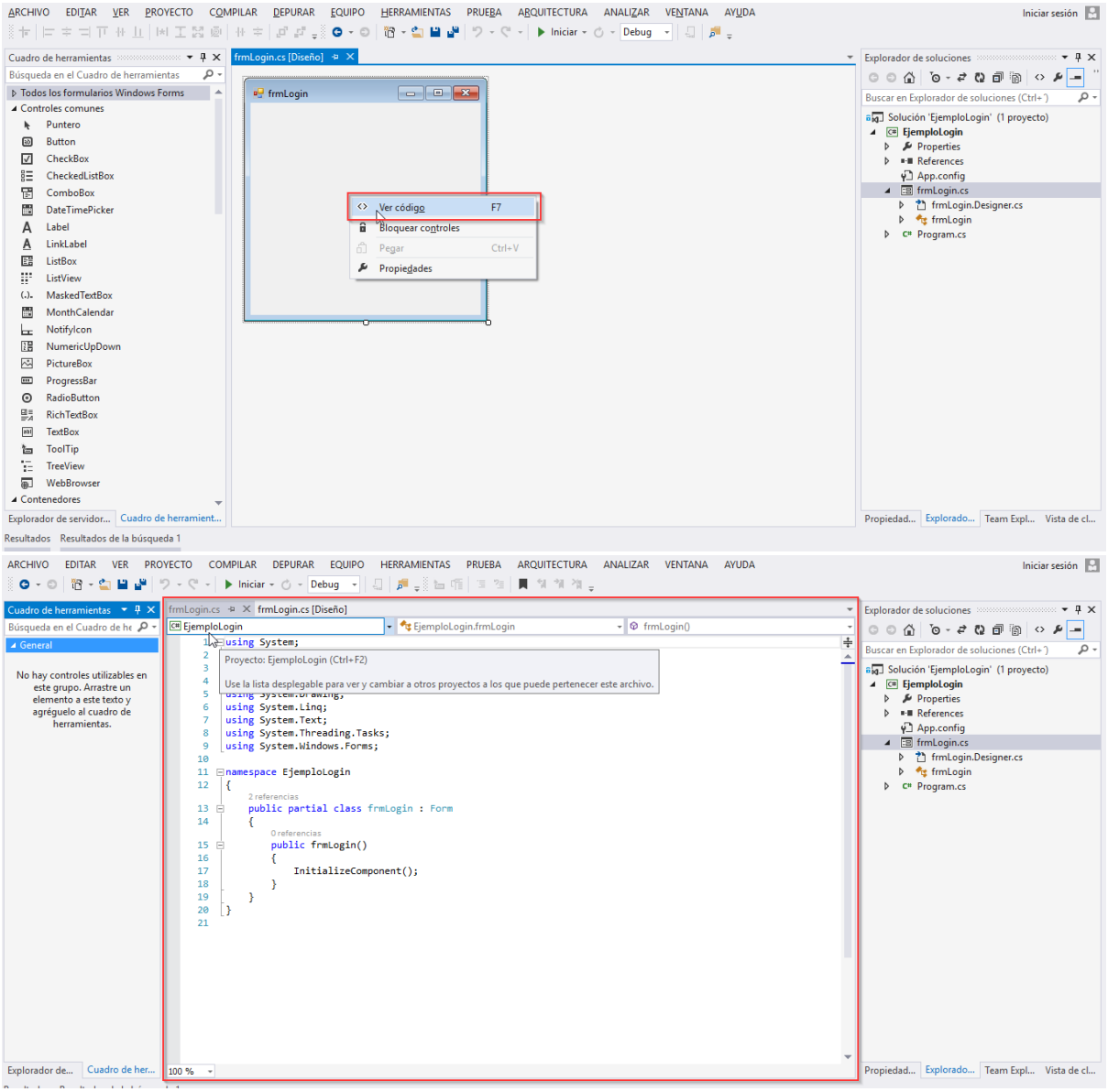


- Sobre esta vista se permite arrastrar y soltar componentes al formulario, así como modificar sus propiedades

2) **Vista de Código**: [F7 o bien sobre el formulario click derecho opción Ver código]

En la vista de código se trabaja con dos archivos diferentes correspondientes a la definición de una misma Clase FormLogin:

2.1) Clase FormLogin (lógica)

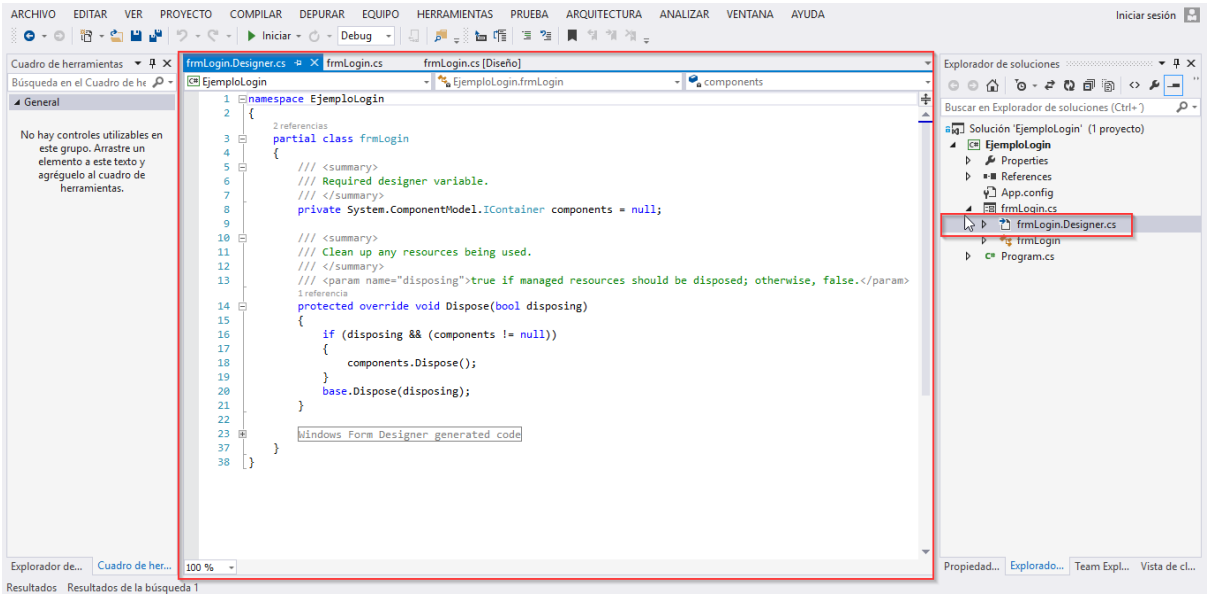


- Este es nuestro archivo fuente propiamente dicho donde se declara la clase que se utilizará para modelar nuestro formulario. Algunas consideraciones:
  - ✓ La clase hereda de **Form** (:Form) y por ende es nuestra clase **es un** Formulario.
  - ✓ La clase es definida como **parcial class**. Esto significa que su definición puede estar distribuida en más de un bloque de código. La ventaja de las clases parciales es que nos permite separar la definición de una clase a través de varios archivos. Uno de los usos más comunes es justamente la definición de interfaces gráficas, ya que existe un archivo para la interfaz y otro para la lógica, esto resulta bastante útil ya que nos evita correr el riesgo de modificar el código que define la interfaz y nos quita de la vista código que realmente nosotros nunca escribimos por nuestra cuenta.

✓ Entonces, este archivo es donde escribiremos la lógica de la ventana encerrada en métodos de eventos

.

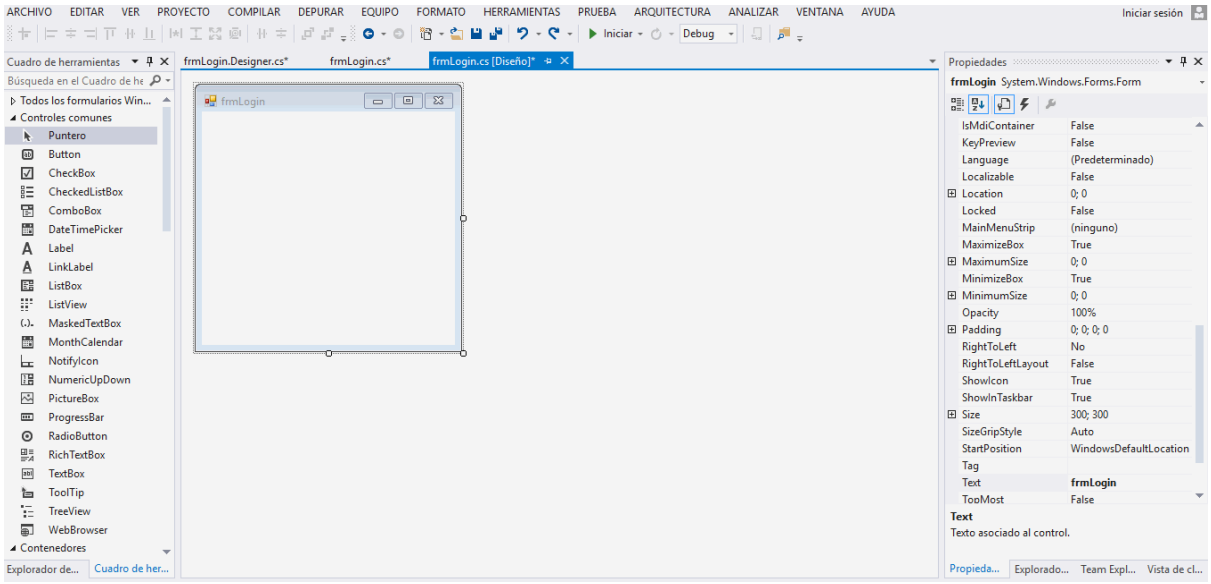
2.2) Clase formLogin.Designer.(Interfaz)



✓ Este archivo contiene las definiciones de la clase FormLogin que son creadas y mantenidas automáticamente por el entorno y que rara vez se modifica de manera manual. Básicamente las definiciones corresponden a la creación componentes junto con el seteo de sus propiedades y registro de eventos.

4. Agregando componentes (controles) al formulario

Posicionados sobre la vista de diseño, podemos mediante el **Cuadro de Herramientas** arrastrar y soltar los controles necesarios para el formulario formLogin.



En nuestro caso utilizamos:

- ✓ Labels
- ✓ TextBoxs
- ✓ Buttons

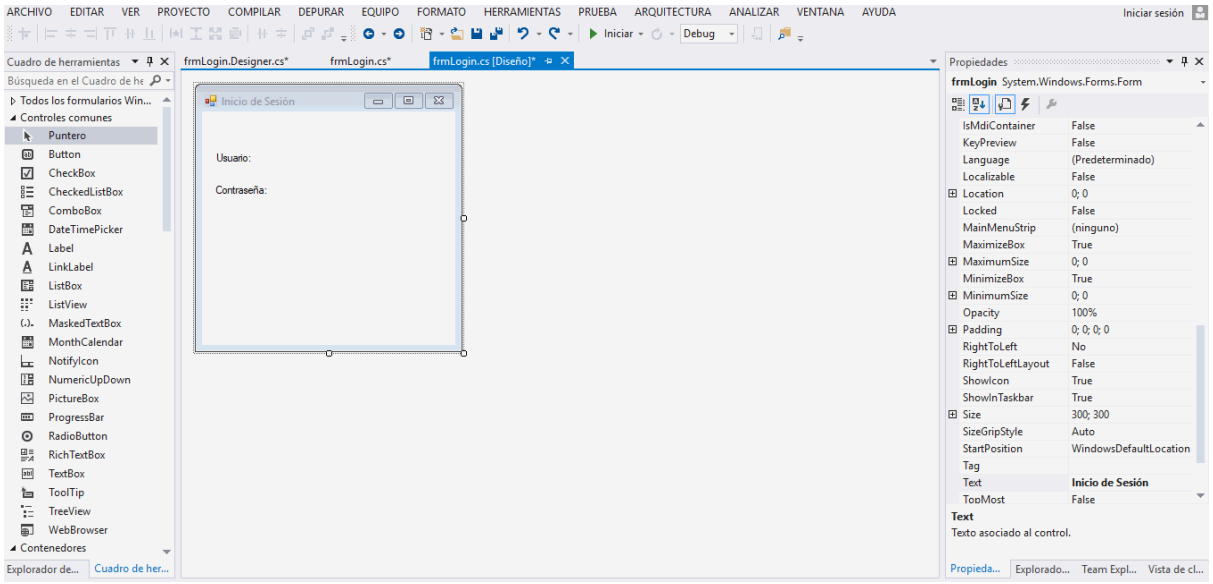
Para todos los controles es recomendable setear la propiedad **Name** anteponiendo un prefijo de tres letras que referencie al tipo de control, de modo que podamos identificar rápidamente los controles en la vista de código.

Algunas convenciones:

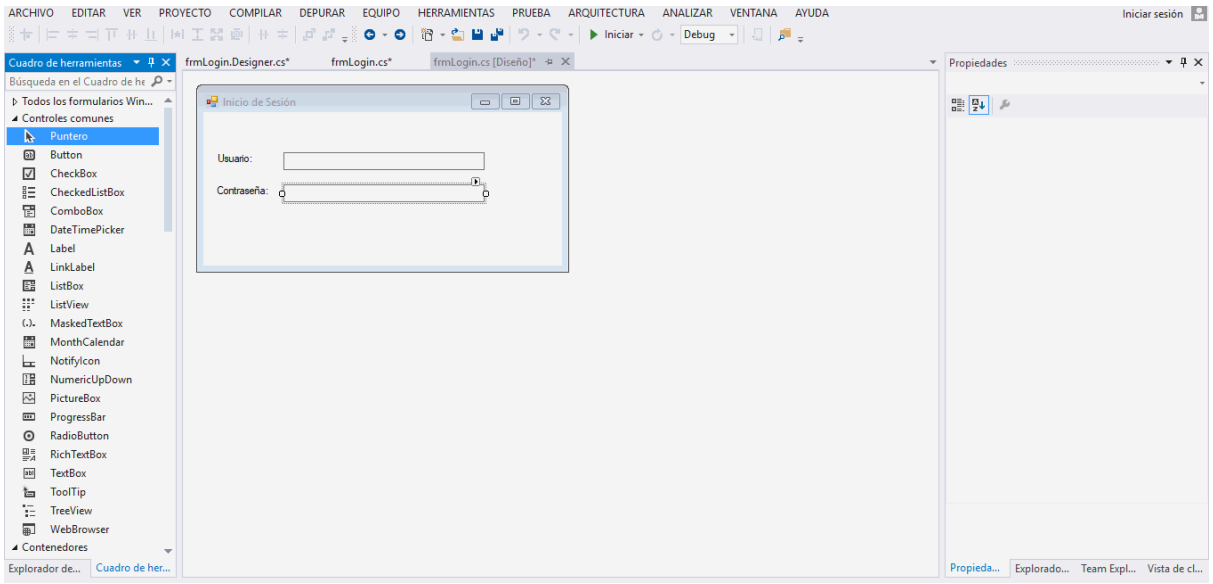
- ✓ frm (Formularios)
- ✓ lbln (Etiquetas)
- ✓ txt (Cajas de texto)
- ✓ btn (Botones)
- ✓ cbo (Cuadros combinados)
- ✓ chk (Casillas de verificación)
- ✓ op (Botones de radio)
- ✓ lst (Listas)

Resultando:

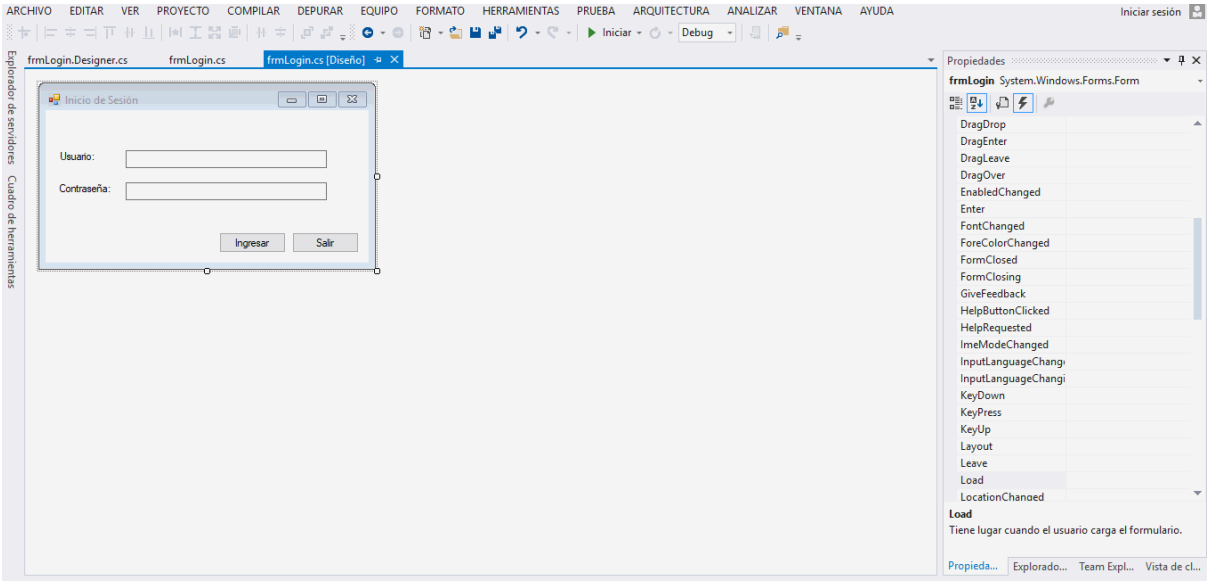
- ✓ Labels



✓ Textboxs

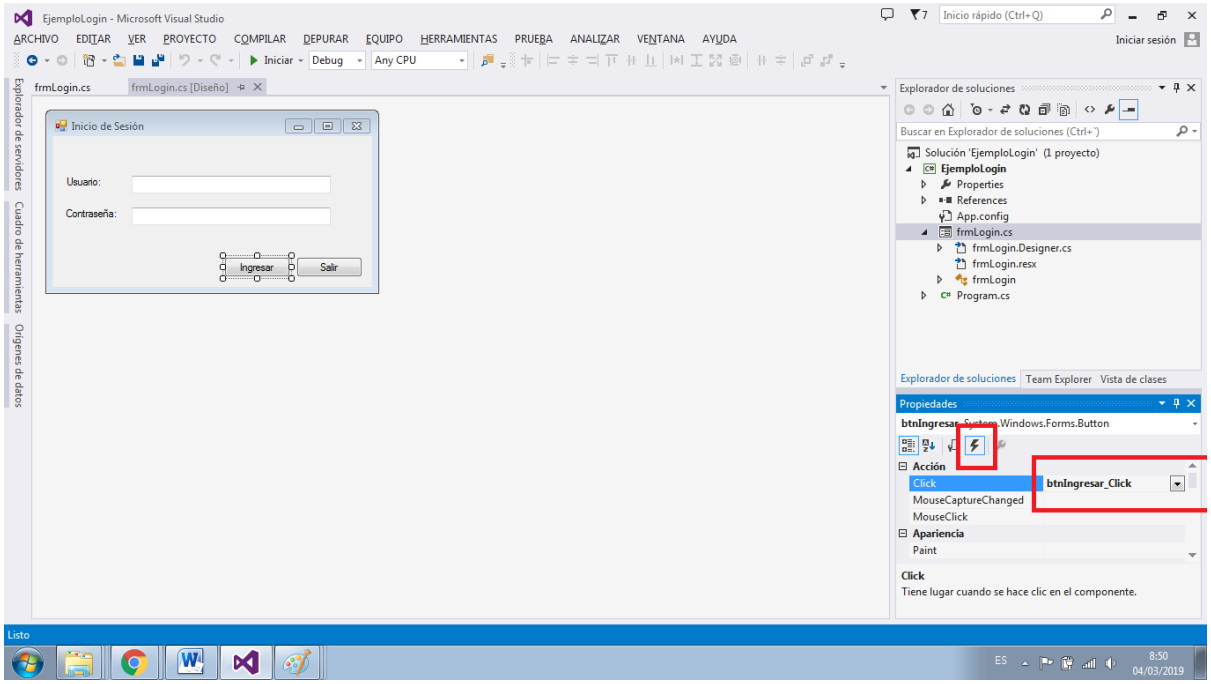


✓ Buttons

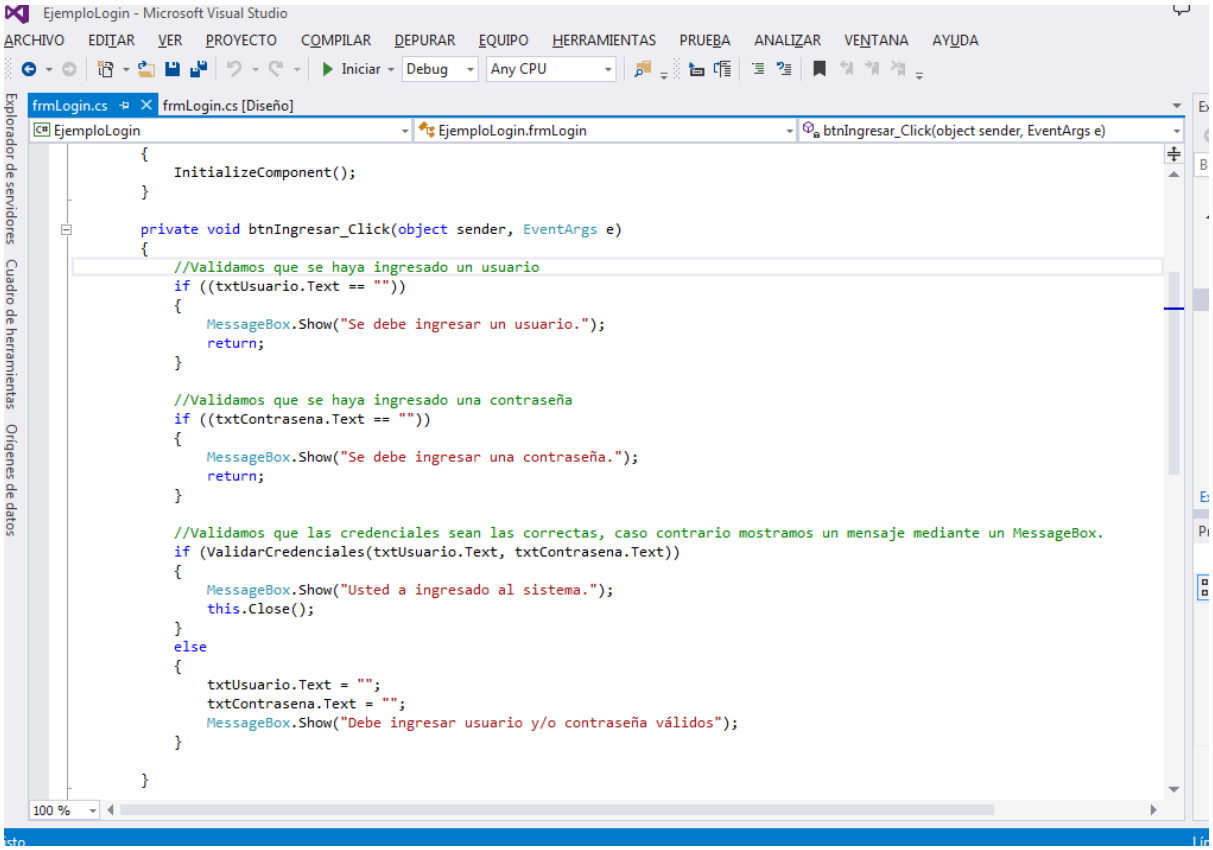


5. Por último, manejando eventos

- El último paso para dejar funcionando nuestro proyecto es gestionar los eventos asociados con los controles diseñados. Para ello vamos a necesitar gestionar, como mínimo, los eventos correspondientes a los botones **Aceptar** y **Cancelar.**
- Para registrar y generar un método de evento que responda ante la interacción con un control debemos seleccionar antes que nada el tipo de evento. Para ello nos posicionamos sobre el control y accediendo a la herramienta **Propiedades** encontramos un botón con un símbolo de rayo, que nos permite elegir el tipo de evento. Tal como se indica a continuación:



- El entorno nos sugiere automáticamente un método que tiene por nombre: nombre del control como prefijo seguido de _ y el tipo de evento a controlar. En nuestro caso btnIngresar_Click. Haciendo doble click sobre el combo accedemos directamente la vista de código de la clase frmLogin donde se nos genera automáticamente el método necesario para escribir la lógica del formulario requerida para dicho evento.



- Haciendo doble click sobre el botón llegando de igual manera, pero tener presente que el entorno genera automáticamente el método de evento asociado con el tipo de evento por defecto del control.
- Notar que cuando trabajamos con formularios, nuestra labor como desarrolladores se reduce solo a escribir lógica embebida en los métodos de evento de controles diseñados para que interactúen con los usuarios.