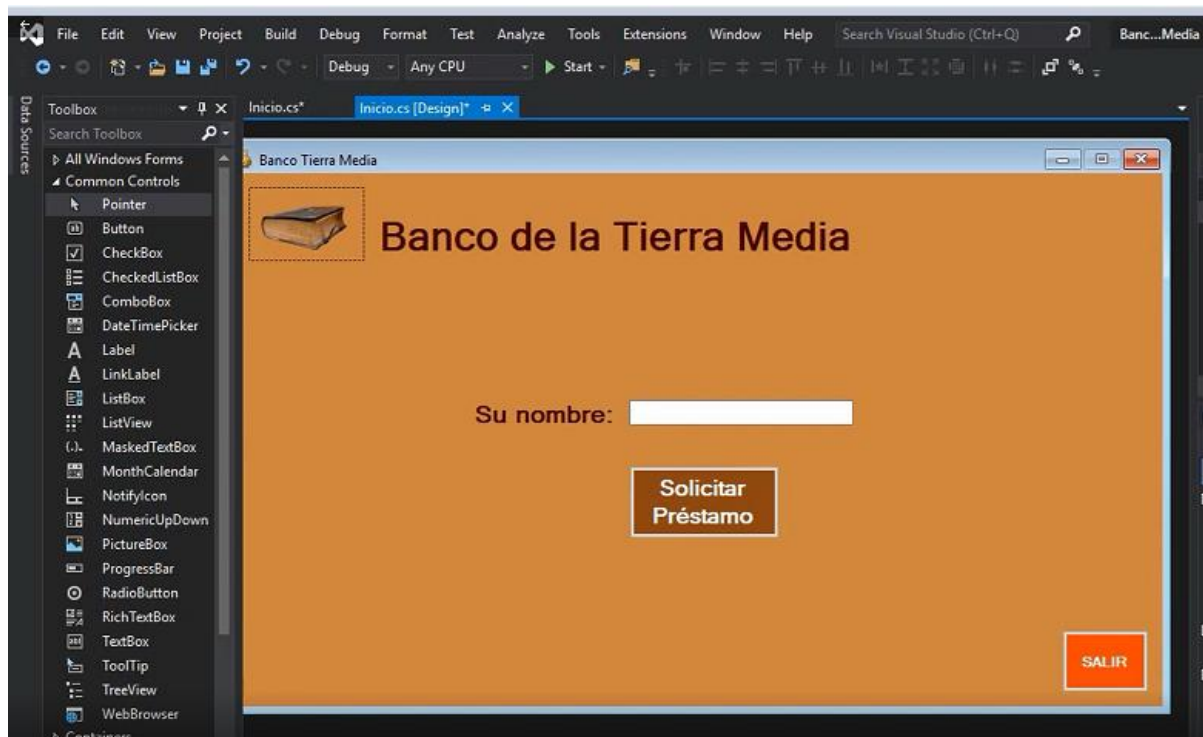


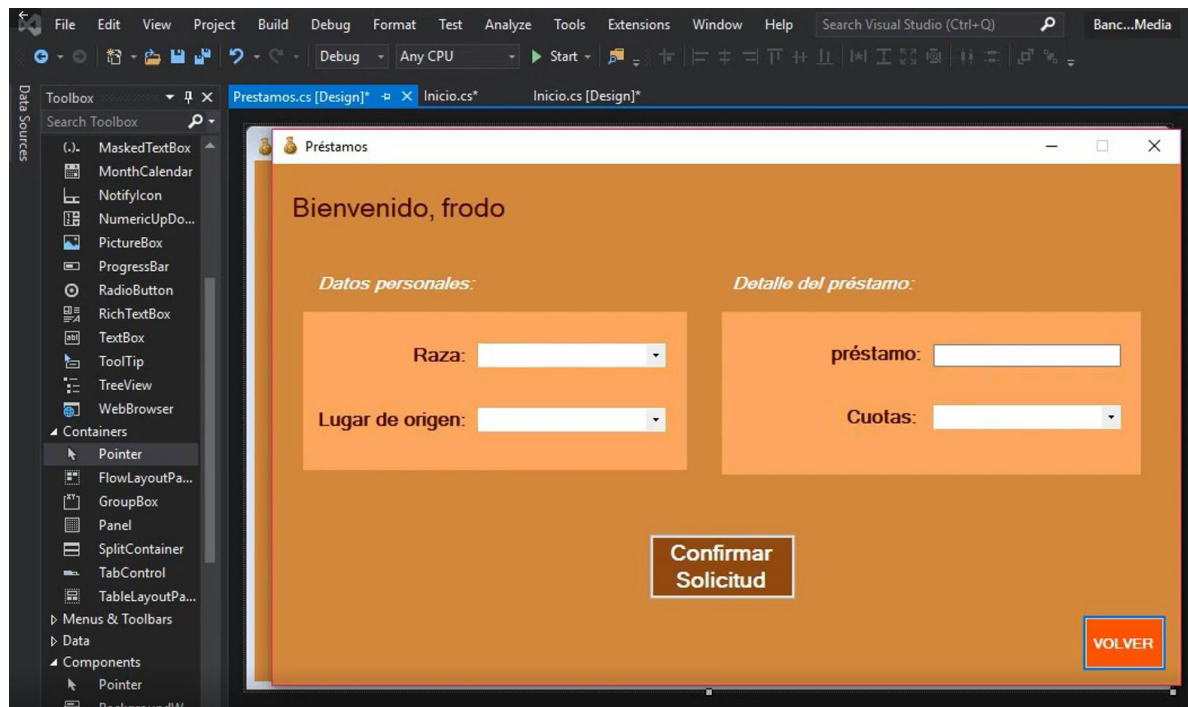
Ejercicio funcionalidades básicas

Trabajaremos con:

- etiquetas de texto
- imágenes
- cuadros de texto
- botones
- Comprobación de introducción de datos numéricos /alfanuméricos y visualización mensajes error
- Configuraciones básicas

Vamos a crear dos formularios con esta apariencia:





Paso 1: Crear un nuevo proyecto de formulario

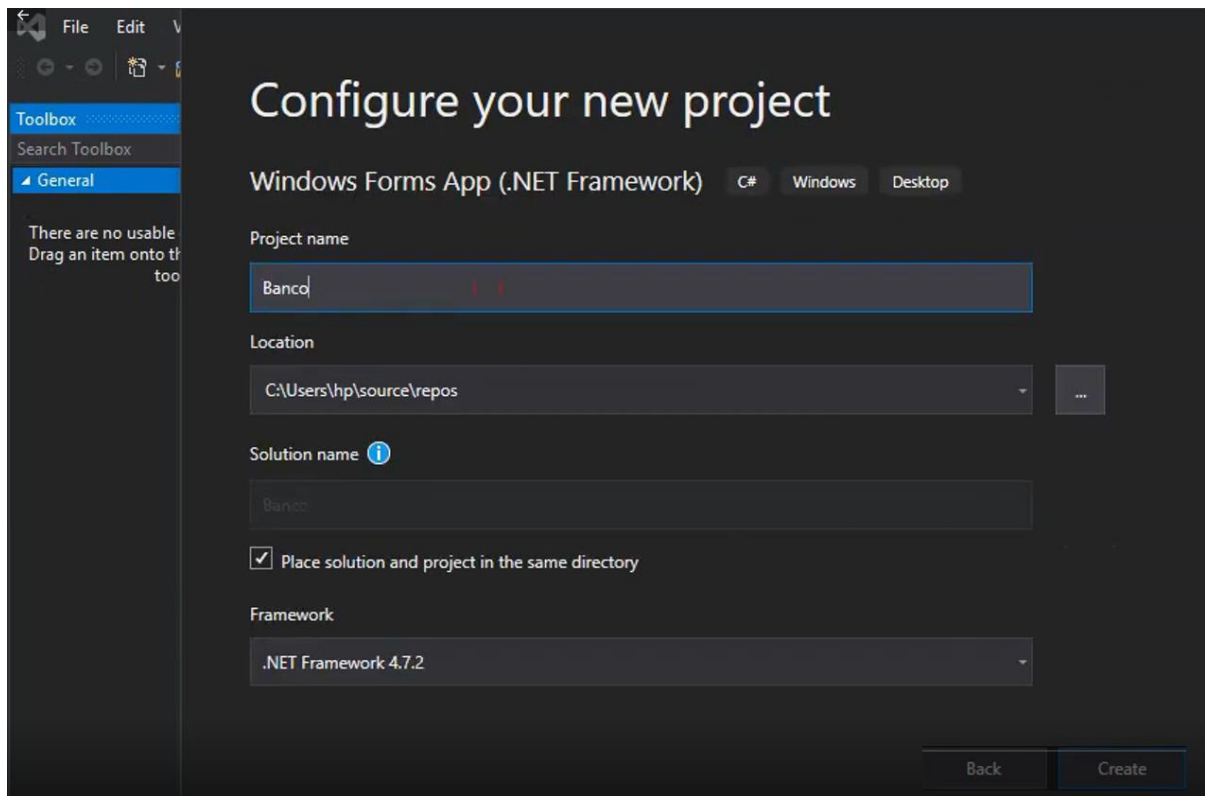
Para crear un nuevo proyecto seleccionamos File->New o también podemos pulsar las teclas CTRL + SHIFT + N

Vamos a crear una aplicación de C# Windows Forms App (.NET Framework) (compilación para Windows en entorno de escritorio)

Una vez seleccionado este tipo de aplicación en Visual Studio pinchamos en Next

Introducimos un nombre y la ubicación donde queremos almacenarla (aparece la ruta por defecto) y el Framework .NET 4.7.2 que vamos a utilizar

Pinchamos sobre el botón "Crear"



Paso 2: Configuración del formulario

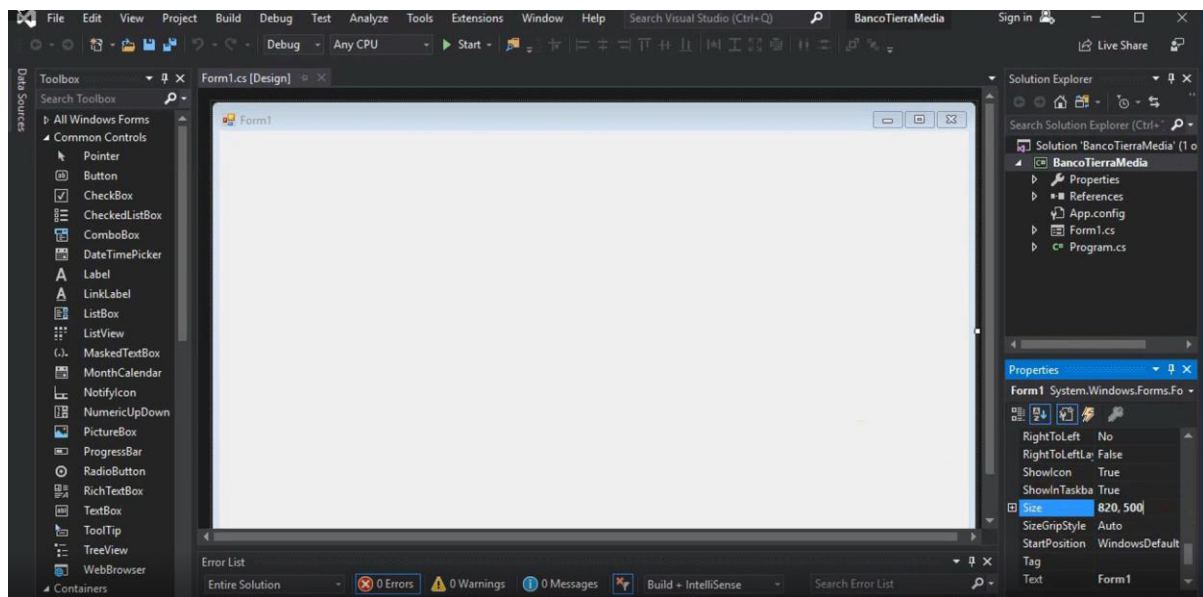
En la siguiente pantalla nos aparecerá el formulario (ventana) que posteriormente se ejecutará en Windows. En la parte izquierda de la ventana aparecen los componentes que podemos incluir en el formulario, en la parte central el formulario propiamente dicho y en la parte derecha el explorador de la solución y las propiedades del elemento seleccionado. Seleccionando cada una de las propiedades que aparecen se muestra debajo de la ventana de propiedades una breve explicación de la propiedad.

Vamos a realizar algunos cambios iniciales.

Podemos cambiar el nombre del fichero .cs dentro del explorador de la solución

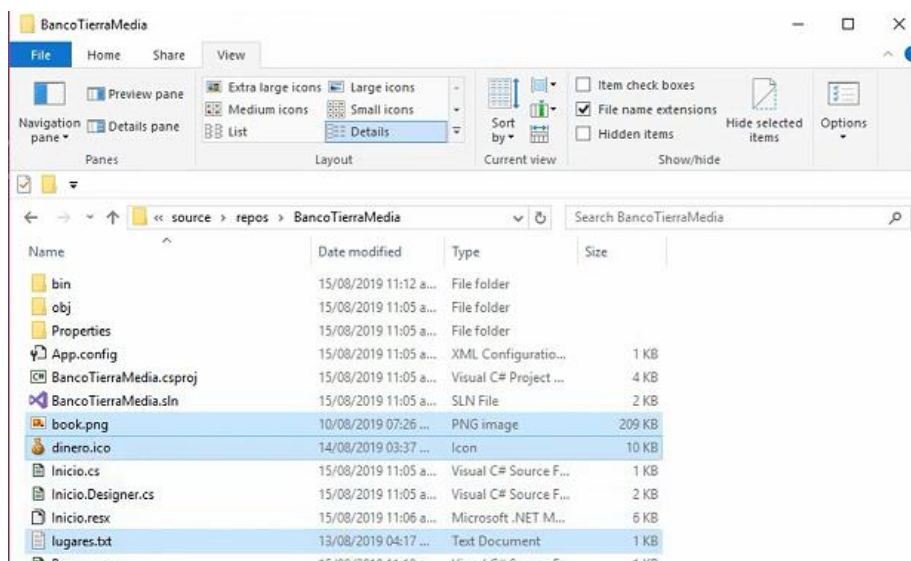
- cambiamos el nombre del formulario en DataBindings, ese va a ser el nombre de la clase a partir de la cual obtendremos objetos.
- Cambiamos el nombre que aparece en la ventana (por defecto Form1) en la categoría Font propiedad text
- Cambiamos algunas propiedades de nuestro formulario:
- indicamos un tamaño (Size) de 820x500
- indicamos dentro de la misma categoría que la ventana al ejecutarse se posicione en el centro asignado el valor CenterScreen a la propiedad StartPosition


- bloqueamos este tamaño dentro de Font propiedad FormBorderStyle aplicando el valor FixedSingle o Fixed3D
- eliminamos la opción de maximizar la ventana indicando dentro de Location la propiedad MaximizeBox=False
- Cambiamos la apariencia del color de fondo en categoría Appearance propiedad BackColor



También podemos cambiar el icono que aparece en la ventana.

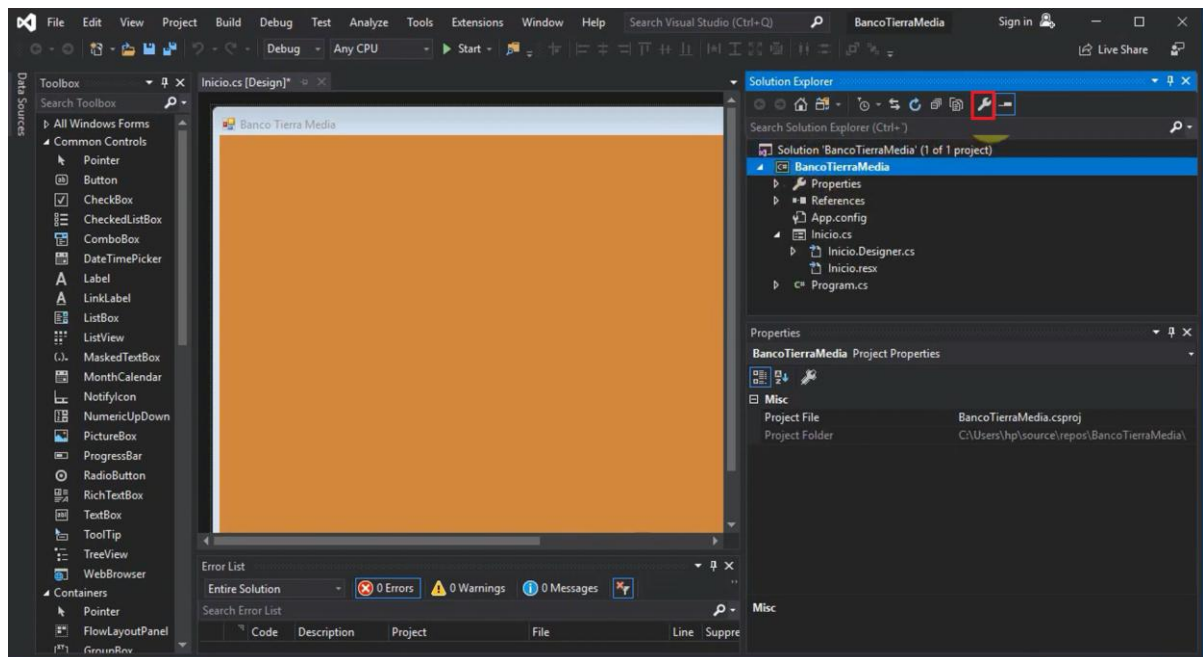
Previamente hay que colocar un icono dentro de la carpeta del proyecto. Recordad que la ubicación la habíamos seleccionado al crear el proyecto.



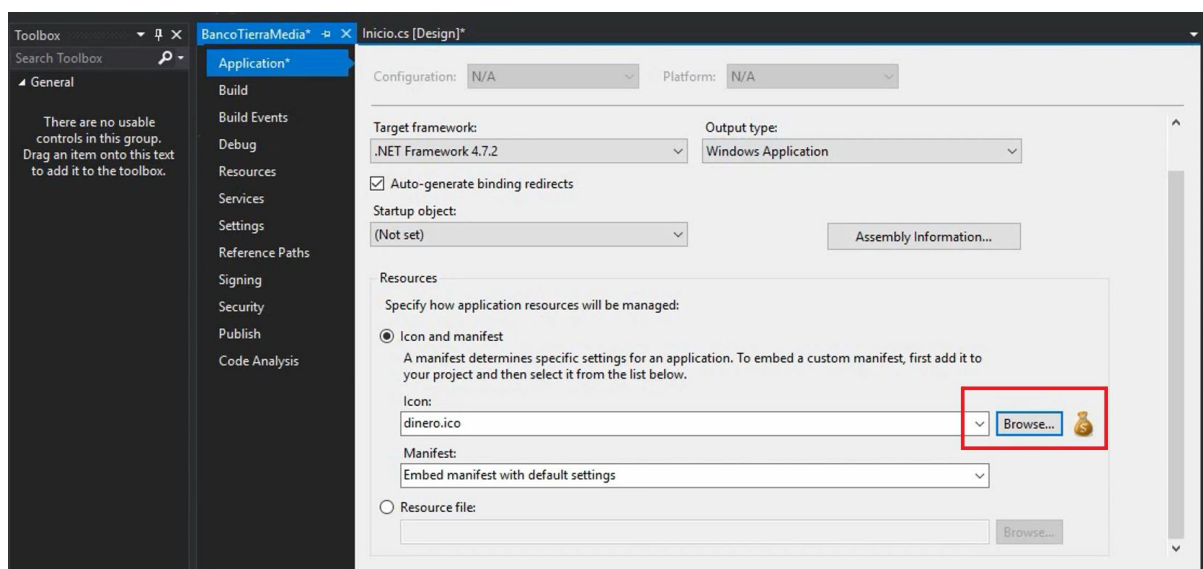
	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	---	---

Este icono se cambia en la categoría Icon donde podemos seleccionar este icono.


En el explorador de Soluciones hacemos clic sobre nuestro proyecto y accedemos a sus propiedades:



Podemos indicar un icono que es el que luego aparecerá en nuestro ejecutable:

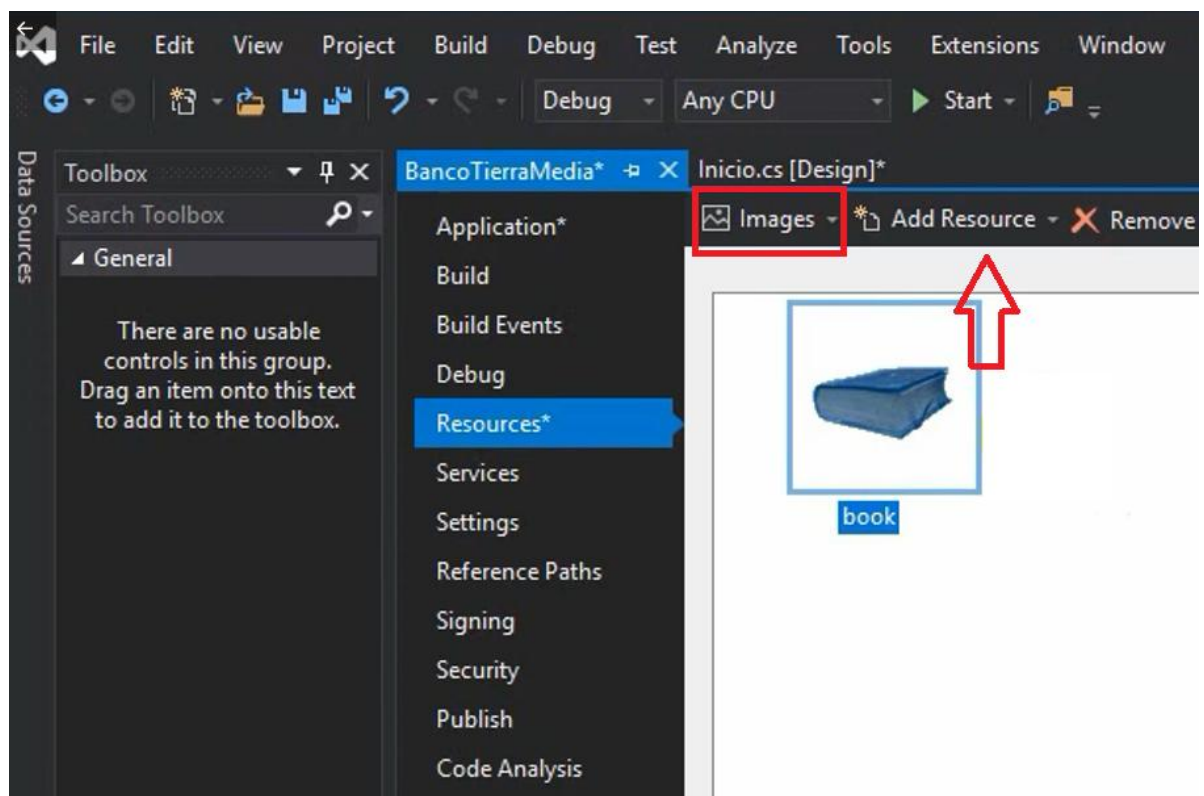
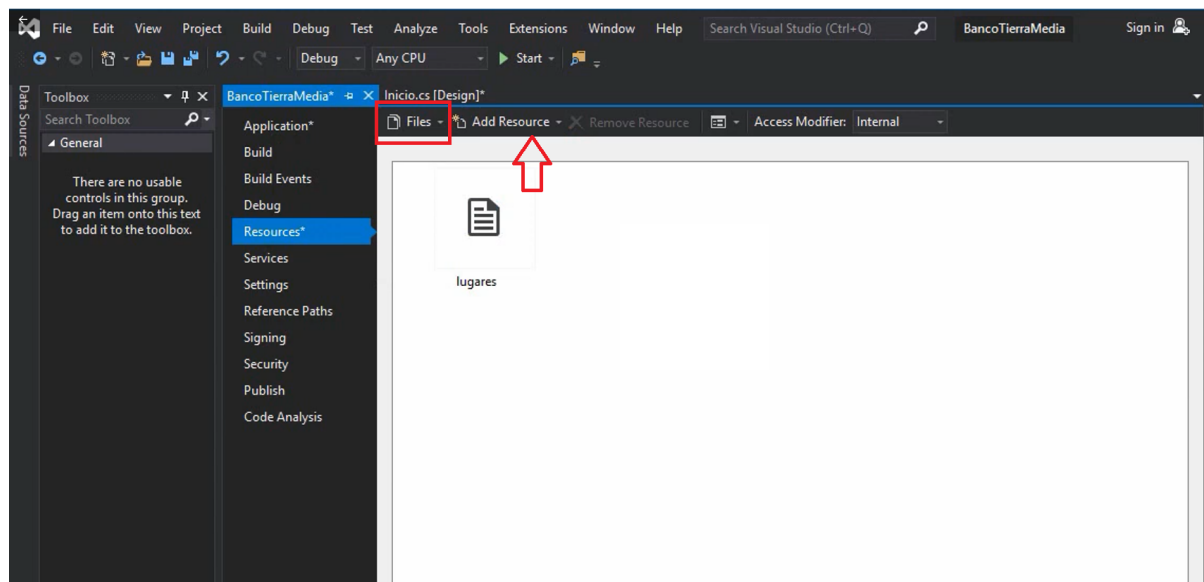


En esta pestaña de proyecto dentro de recursos podemos agregar los archivos e


	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	---	---

imágenes que vamos a utilizar en el proyecto de forma que al compilar el proyecto estos recursos queden incrustado dentro del archivo ejecutable.

En nuestro caso tenemos que incorporar un fichero de texto y una imagen:



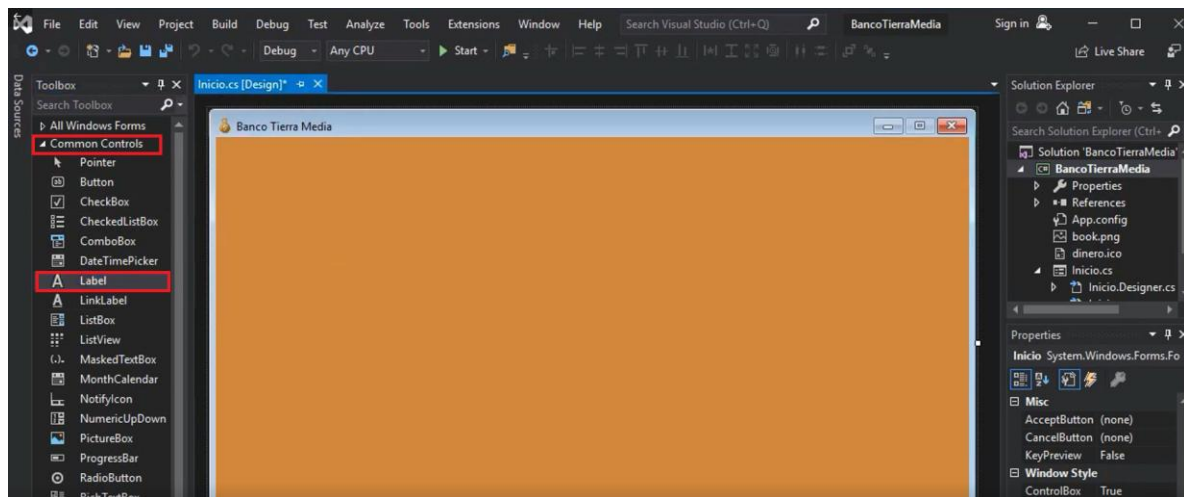
Cerramos la ventana de propiedades

	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	---	---

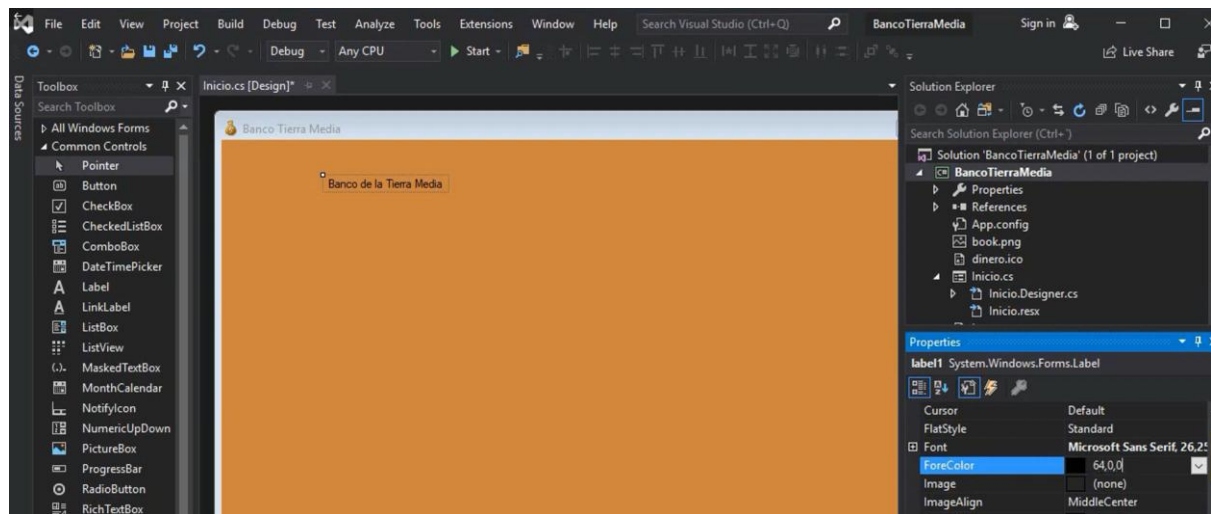
Paso 3: Introducción de componentes

Con las etiquetas de texto indicamos mensajes en pantalla que no van a tener funcionalidad dentro del código.

Incorporamos una etiqueta a nuestro formulario arrastrando el componente Label hasta la ventana. Podemos cambiarla el nombre dentro de la categoría Design ->name o dejarla con el nombre que aparece por defecto (label 1) ya que, como se ha comentado, no va a tener ninguna funcionalidad adicional.

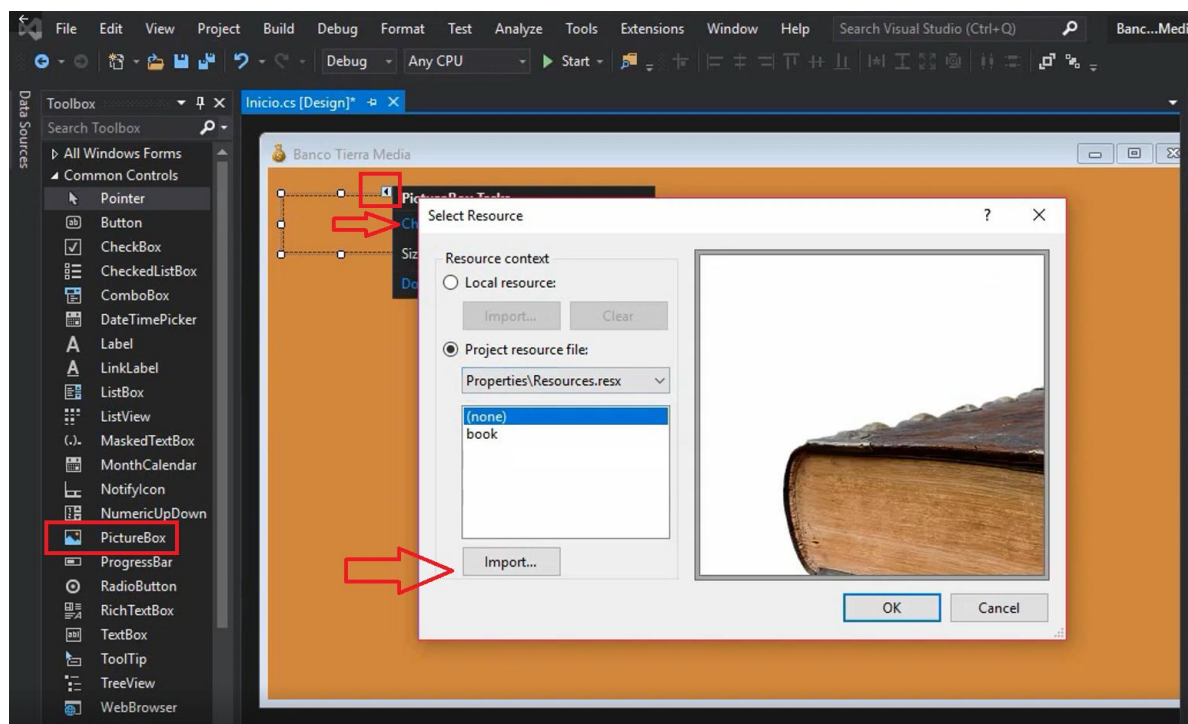



Lo que sí hay que cambiar es el texto que aparece en el formulario. Por defecto aparece label 1 y queremos poner un literal, por ejemplo "Banco de la Tierra Media" y cambiamos su tamaño, tipo de letra y color en la categoría Font (recordar que para cambiar las propiedades el elemento sobre el que se quiere actuar debe estar seleccionado)



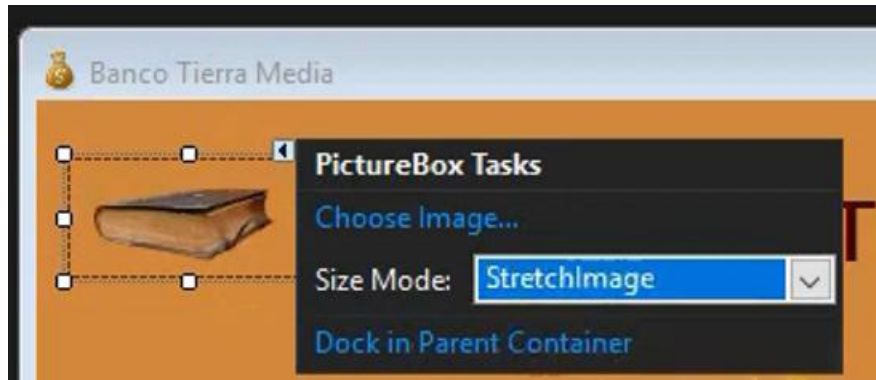
Agregamos una imagen que incorporando un componente llamado PictureBox que igual que con la etiqueta arrastramos hasta nuestro formulario. Tampoco va a tener funcionalidad por lo que podemos dejar su nombre por defecto.

Seleccionamos este PictureBox y nos aparece resaltado y en recuadro aparece una flecha. Pinchando sobre la flecha nos permite elegir la imagen que previamente hemos colocado en los recursos del proyecto o bien podemos importarla en ese momento y se guardará dentro de los recursos del proyecto.



	UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i>	CFGS Desarrollo de Aplicaciones Multiplataforma Módulo: DDI
---	---	--

Podemos indicar distintos modos de tamaño. Con el modo zoom va a mantener su proporción de aspecto ancho y alto. Podéis probar el resto de opciones que no mantienen la relación de aspecto.



Agregamos otra etiqueta "Su nombre:" cambiando igualmente tipo de letra, el color, tamaño 16 y negrita. A Continuación de esta etiqueta añadimos un TextBox. Igualmente cambiamos tamaño y tipo de letra. Este TextBox si va a tener funcionalidad ya que va a ser utilizado para acceder al siguiente formulario y se va a realizar una validación ya que el nombre no puede contener otra cosa que no sean caracteres por lo que esta vez sí cambiamos su nombre de objeto.

Con el TextBox seleccionado de nuevo accedemos a la categoría Design y dentro de esta cambiamos en Name su nombre (p.e. indicar literal nombre).

Añadimos dos botones, componente Button.


Uno va a ser el botón de salir, indicamos en nombre. Es aconsejable utilizar algún tipo de convención para identificar objetos dentro del código por lo que por ejemplo para botones nombrarles anteponiendo al nombre las iniciales btn, pe. btnsalir.

En la categoría Font indicamos como texto "SALIR" y de nuevo cambiamos la letra negrita y cambiando el tamaño y el color a blanco y el color BackColor. Colocamos el botón en la parte inferior derecha de la pantalla.

El segundo botón tiene que poner "Solicitar Préstamo" y realizamos los mismos pasos que con el anterior.

Paso 4: Funcionalidades y controles

Para poder dar funcionalidad a los botones dentro de la ventana de propiedades del

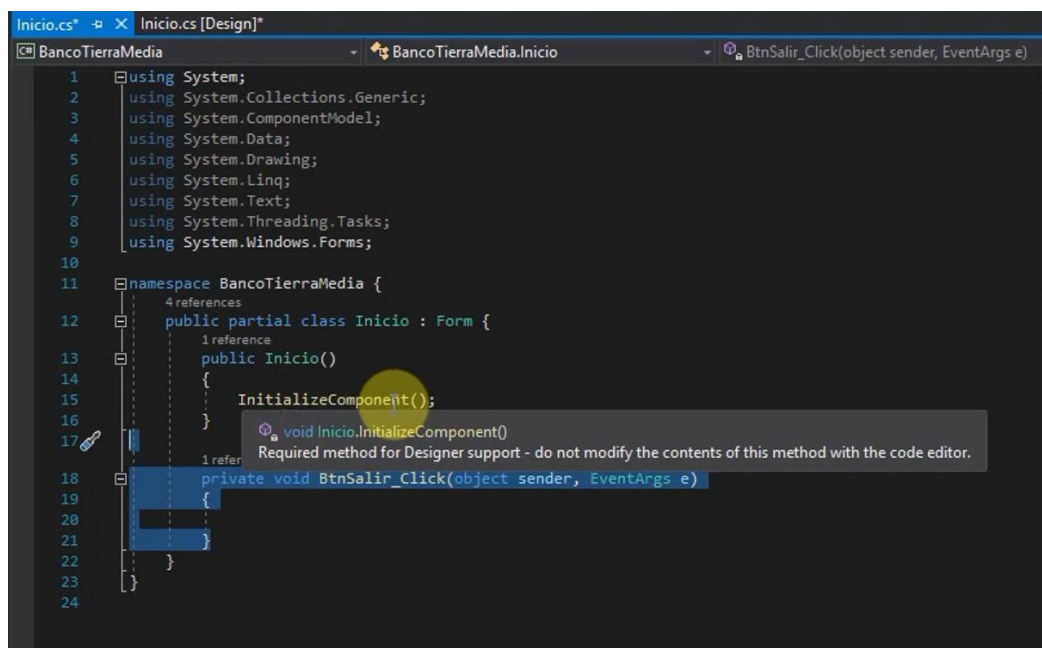
	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	---	---


elemento encontramos el icono de eventos.



Por defecto aparece marcado el evento de "Click" que es el que necesitamos. Al hacer doble clic sobre el botón nos aparece el código en que aparece el método para el botón con el evento Click.

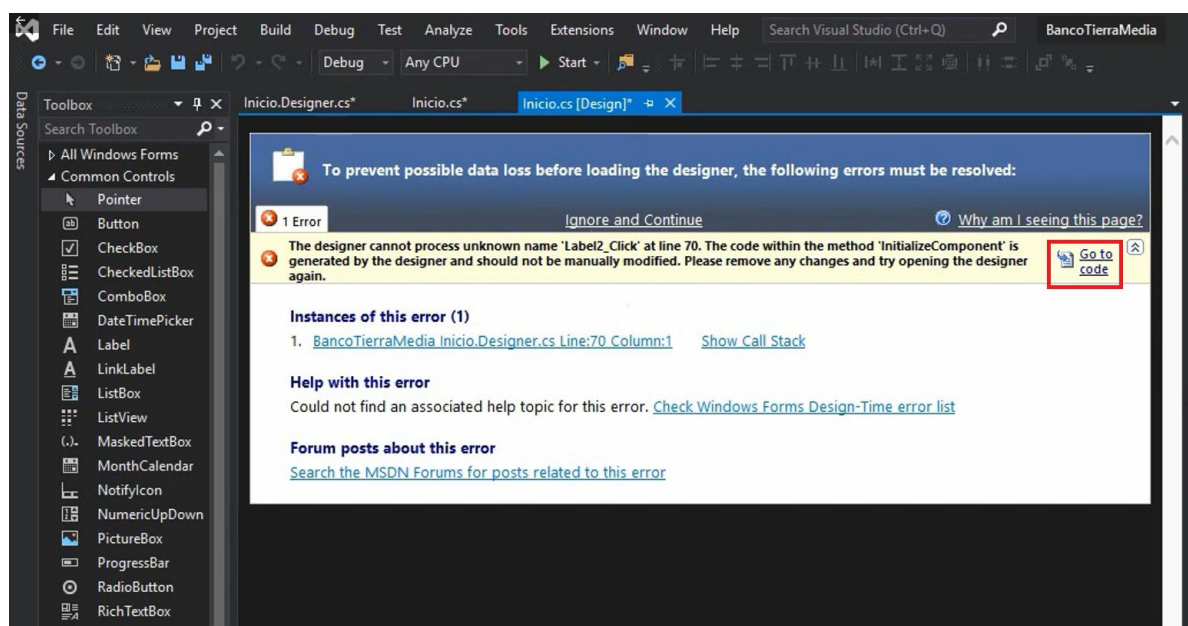
Podemos ver en el código como esta formado el formulario. Aparece un constructor con el nombre que hemos dado al objeto del formulario (Inicio) que inicializa los componentes. Al pasar el puntero del ratón sobre el código nos va indicando que función realiza esa parte del código



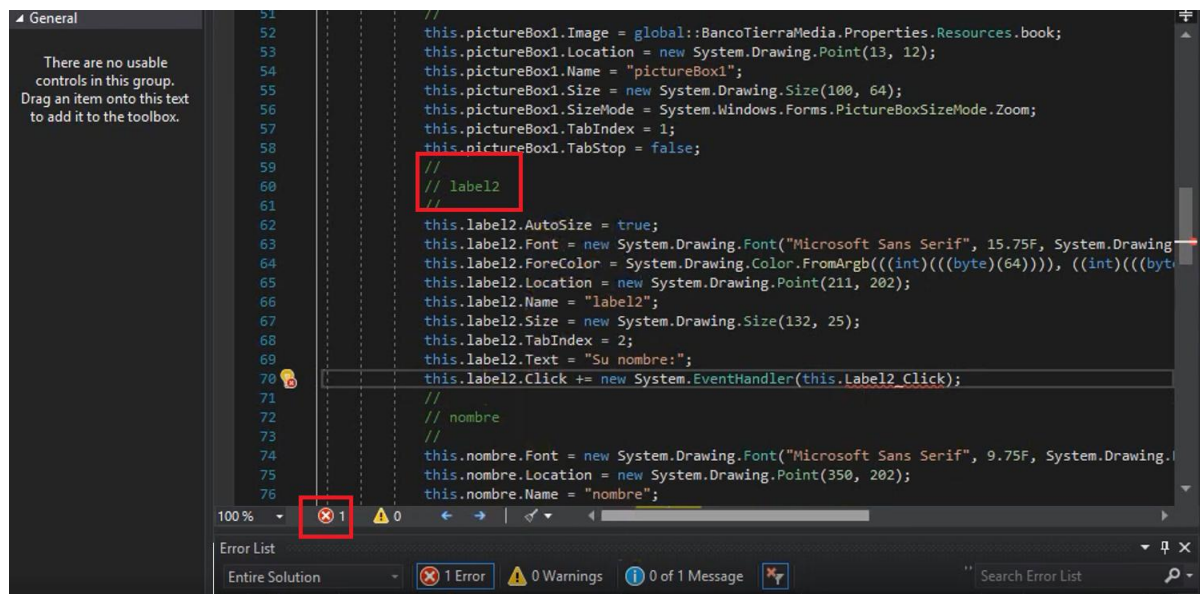
	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	---	---

Vemos que la clase Inicio hereda de la clase Form que es la clase base a partir de la cual se crean las ventanas en C#.

Al incluir el evento de click cada vez que se haga clic sobre el botón se va a ejecutar el código que se indique en ese evento y también nos dice que el método tiene 1 referencia lo que indica que está siendo utilizado. Puede ocurrir que en algún momento se pierda la correspondencia entre la referencia que se indica en código y el diseñador. Hagamos una prueba. Si pinchamos doble clic sobre la etiqueta label2 que aparece con el literal "Su nombre" Vemos que en código se nos añade un método para el evento de clic. Realmente no queremos que la etiqueta no tenga este método así que lo borramos y queramos volver a nuestro diseñador nos va a devolver un error porque la referencia se ha quedado sin su correspondencia.

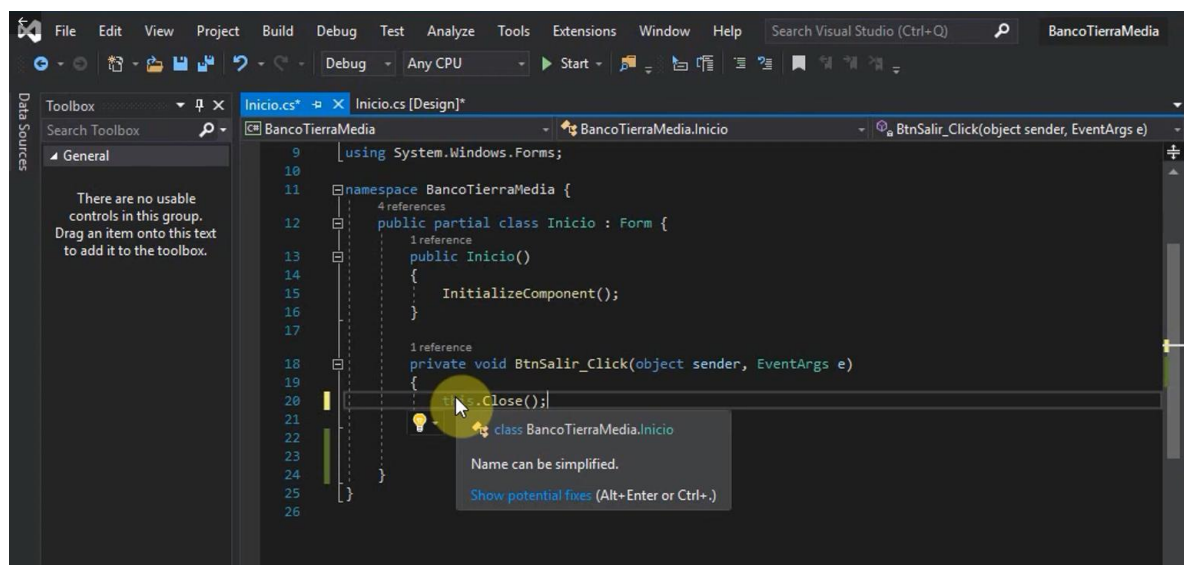


Para solucionarlo tendremos que ir al código que hay detrás del formulario y eliminar la línea errónea y cerramos esta ventana de código y volveremos a ver el diseño de nuestro formulario.




Volvemos a editar el método clic del botón, bien pinchando con doble clic de nuevo sobre él o bien seleccionándolo y doble clic dentro eventos en la ventana de propiedades en la categoría Action sobre el nombre del evento Click.

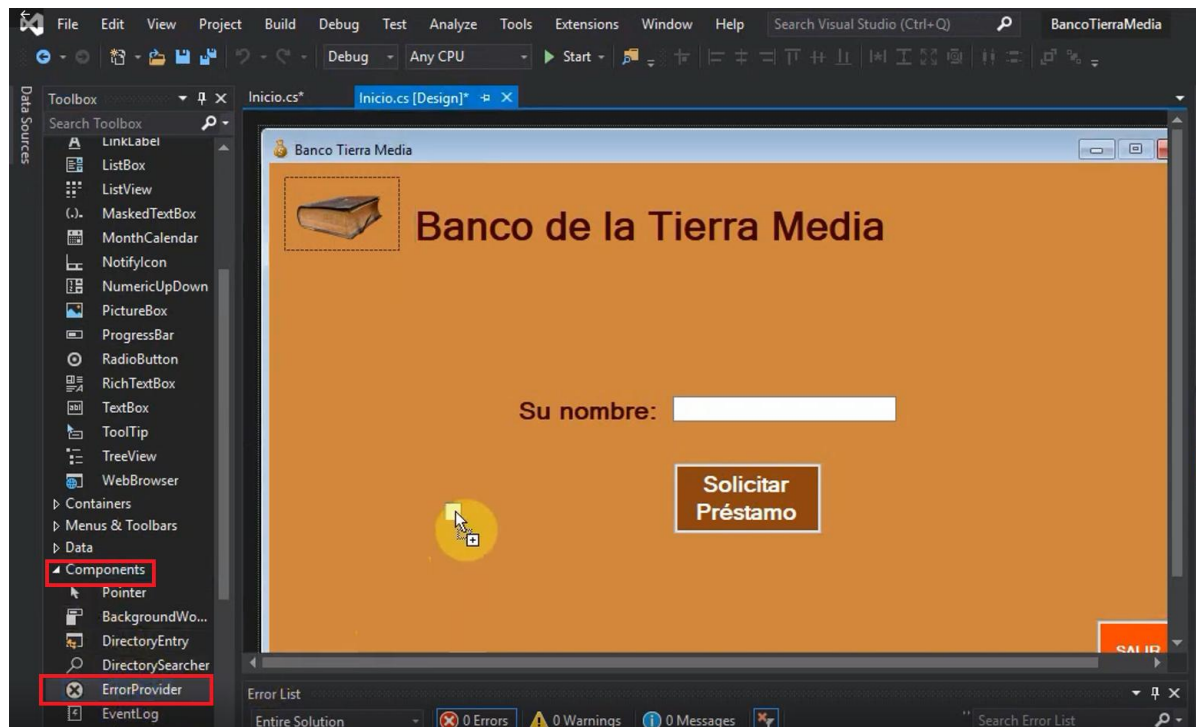
El botón de salir tiene que cerrar el formulario. Para ello utilizamos la instrucción `this.Close();` que cerrará el objeto que está siendo refenciado, en este caso es el formulario Inicio ya que `this` es una pseudovariable que hace referencia al objeto actual.



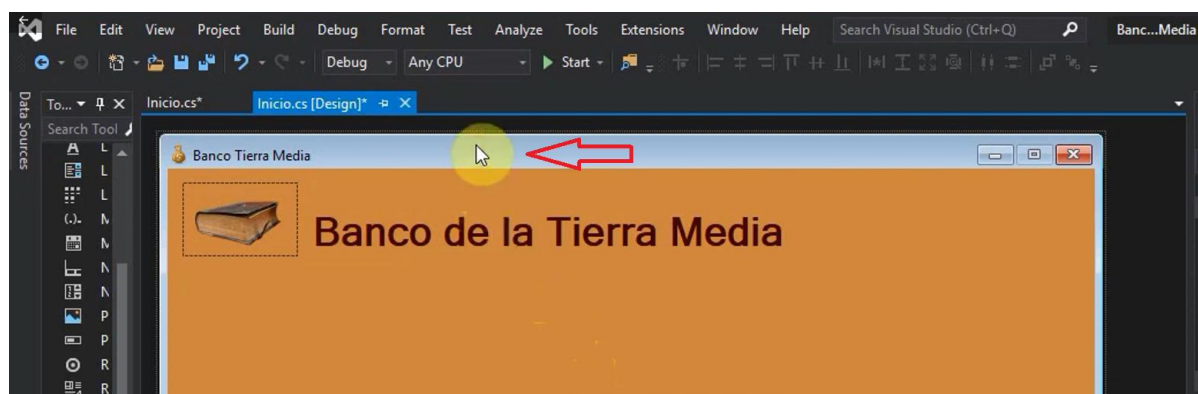
El botón "Solicitar prestamo" tiene que aparecer deshabilitado hasta que el usuario


	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	---	---

introduzca un nombre correcto que solo contenga caracteres. Para esta funcionalidad vamos a utilizar la funcionalidad ErrorProvider que encontramos dentro de componentes:

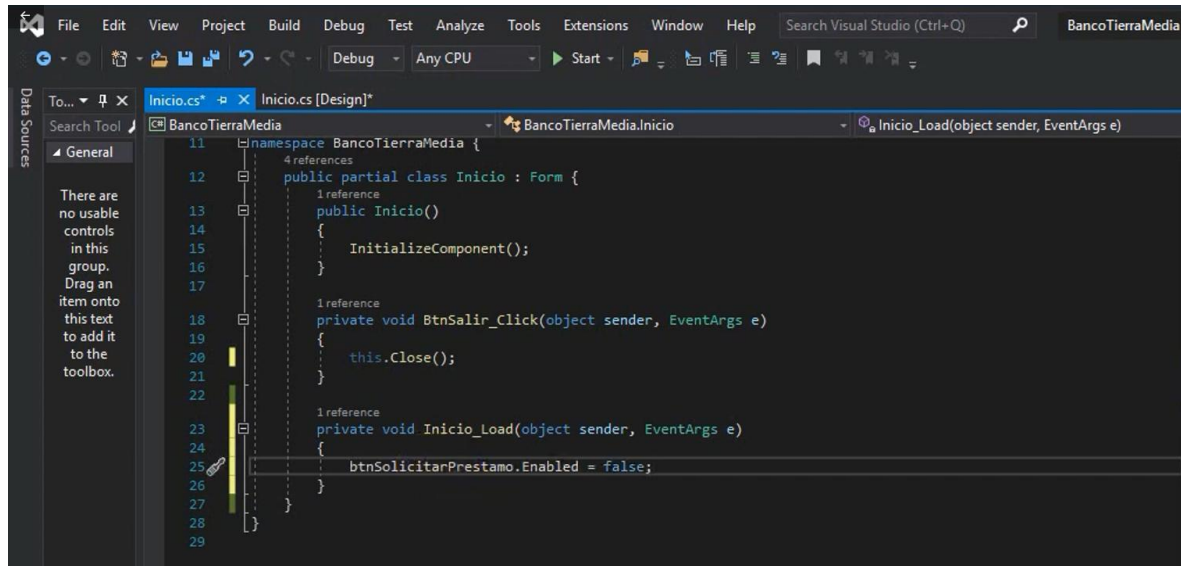


Lo arrastramos en cualquier parte del formulario. Podemos dejar su nombre por defecto. El botón debe estar deshabilitado inicialmente. Para ello podemos indicar que el formulario deshabilite el botón cuando inicia el programa y se habilitará cuando se cumplan las condiciones del TextBox. El formulario también tiene sus eventos. Para que aparezcan pinchamos con doble clic en la barra que contiene su nombre



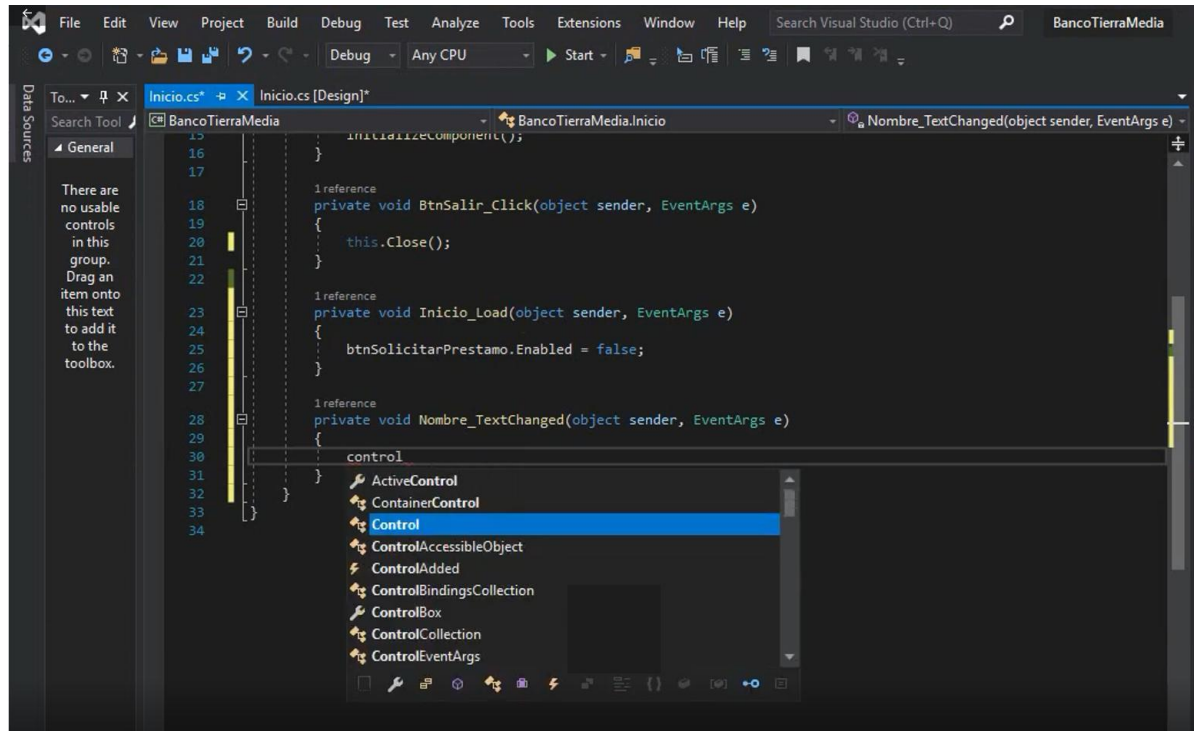
	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	---	---

Aparece el evento Load que se ejecuta al cargar el formulario e introducimos el código indicando que el botón debe estar deshabilitado cuando se inicie el formulario (propiedad enable tenga el valor falso btn.Enabled=false;)

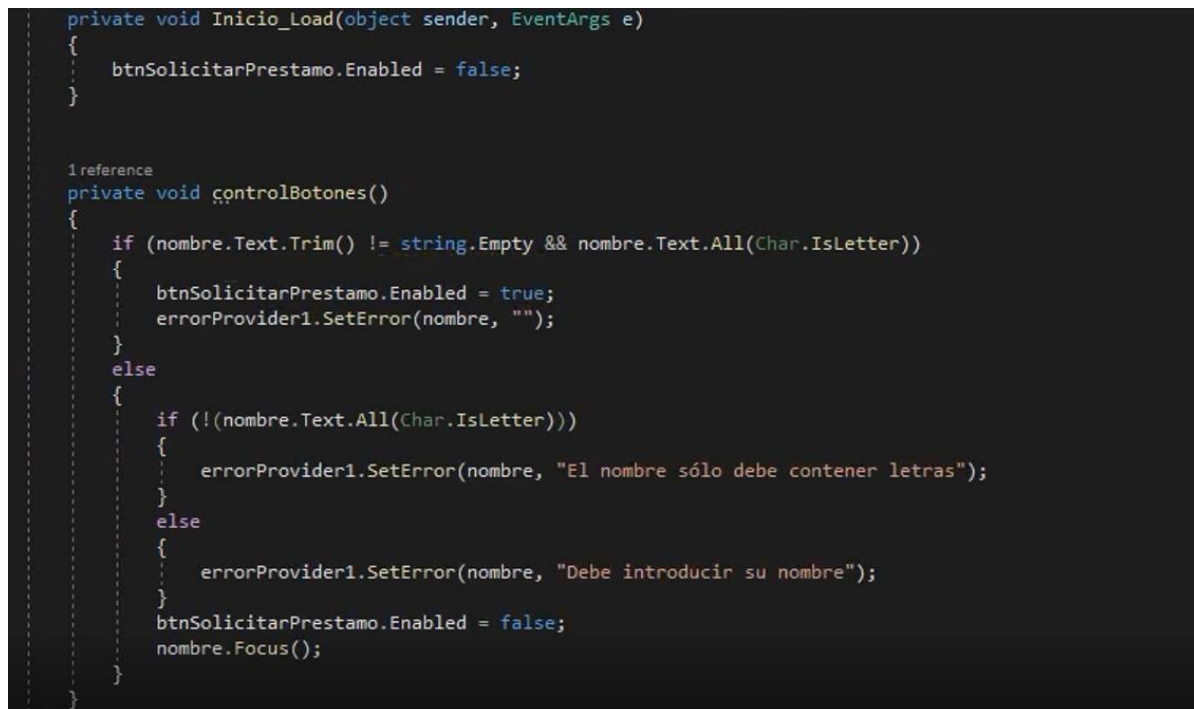


Volvemos a nuestro diseño. Tenemos que controlar que nuestro TextBox solo admita caracteres. Seleccionamos el cuadro de texto y accedemos a evento TextChanged. Dando doble clic visualizamos el código del evento del objeto. Tenemos que deshabilitarlo, controlar que solo se introduzca caracteres y entonces habilitar el botón y controlar el error con el ErrorProvider.


Vamos a utilizar este nuevo método TextChanged que va a controlar el botón. Dentro del evento vamos a llamar al método controlBotones()

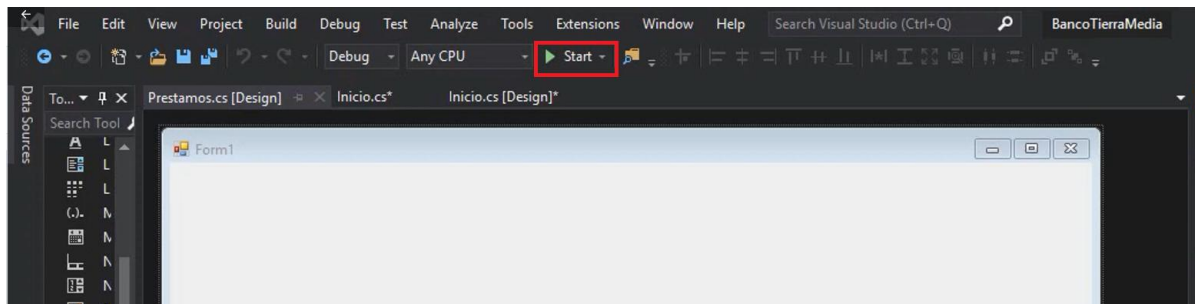


Y añadimos el código del método:



Este formulario aún no es funcional pero podemos comprobar su funcionamiento en cualquier momento a través del botón Start:

	<p>UT 2 - <i>Confección de interfaces de usuario y Creación de componentes visuales</i></p>	<p>CFGS Desarrollo de Aplicaciones Multiplataforma</p> <p>Módulo: DDI</p>
---	--	---



Es recomendable ir guardando el proyecto a medida que se avanza en él.