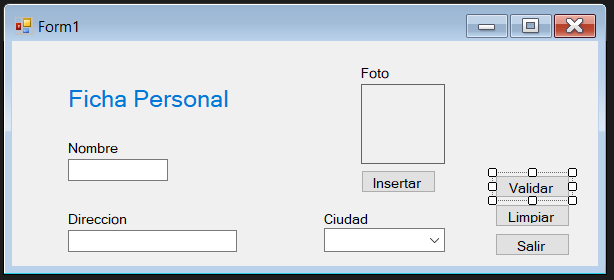
# Practica 1:

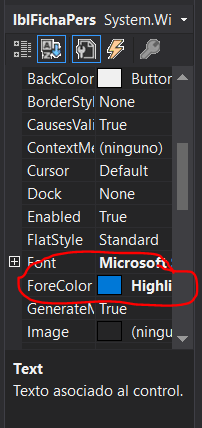
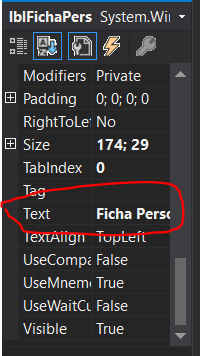
### Añadir Elementos

Se arrastran los elementos necesarios y se le cambian los nombres y colores para que concuerden como los de la práctica.

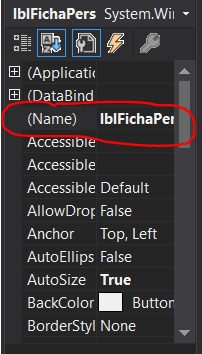


### Cambiar Texto y Color

Para cambiar el nombre se cambia la etiqueta de text y para el color de texto se cambia la etiqueta ForeColor

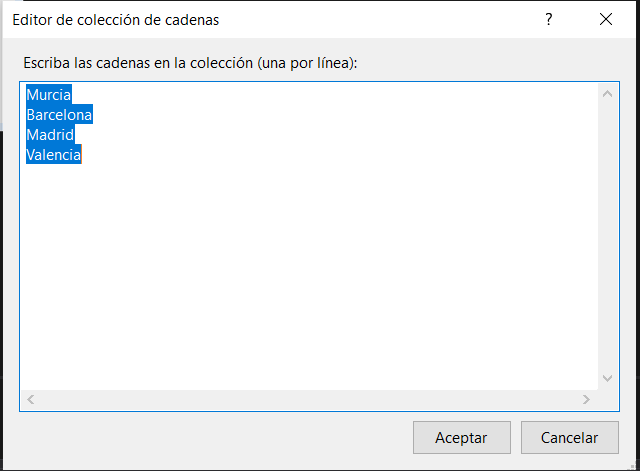
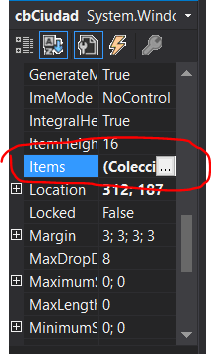


Lo ideal es cambiar el nombre a las variables en sí también para poder guiarte mejor a la hora de trabajar con ellas, para esto se cambia la etiqueta (name).

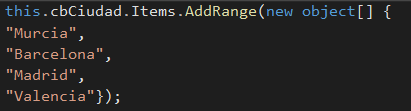


### Añadir elementos al ComboBox

Para añadir elementos al ComboBox no hace falta crear un elemento ComboBoxItems como en Java, basta con añadirlos en la etiqueta Items:

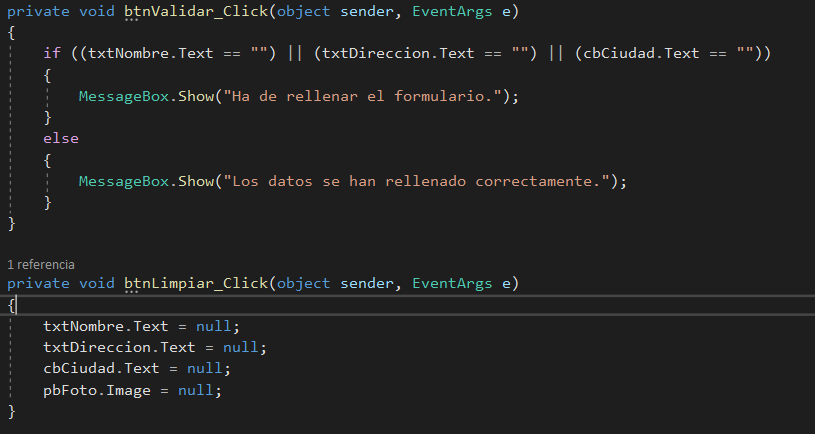


El código que genera esto es importante tenerlo en cuenta para saber cómo manejar estos ítems más adelante:



### Añadir evento click:

Para añadir un evento de click en un botón simplemente se pulsa dos veces dicho botón y te envía a el bloque de código que maneja el evento. Dentro de estos eventos se puede insertar todo lo que sea necesario. En este caso para validar y para limpiar se queda así el código:



Para añadir eventos mediante código se usa la siguiente línea:

this.btnInsertar.Click += new System.EventHandler(this.btnInsertar\_Click);

El esquema general es este:

{this}.[elemento].[nombre del evento] += System.EventHandler({this}.[función a ejecutar])

### File dialog:

Se añade al botón insertar el evento click, en la pestaña de diseño se añade un OpenFileDialog y se le cambia el nombre a uno que se pueda entender.

El insertar una imagen desde el OFD se usa el siguiente código:

private void btnInsertar\_Click(object sender, EventArgs e)

{

if (opfFoto.ShowDialog() == DialogResult.OK)

{

pbFoto.Image = Image.FromFile(opfFoto.FileName);

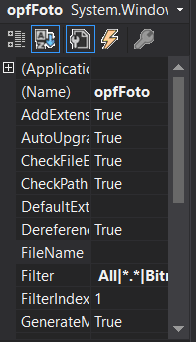
}

}

opfFoto.ShowDialog() hace la llamada al OFD y pausa el resto, al darle a ok el campo FileName se actualiza a la foto seleccionada.

### Propiedad Filter del OFD

Para añadir un filtro al OpenFileDialog hay que cambiar la etiqueta filter.



En este caso el filtro añadido es el siguiente:

All|\*.\*|Bitmap|\*.bmp|JPEG|\*.jpg

La estructura de este filtro es la siguiente:

[nombre que mostrar] | [extensión del archivo]

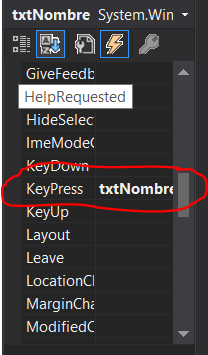
Importante ver que en la extensión del archivo se puede colocar un \* como “comodín” y mostrará todos los archivos. Se puede buscar también por nombre del archivo cambiando el primer \* -> foto.\* mostrará únicamente archivos cuyo nombre sea foto independientemente del formato.

Mediante la etiqueta filterIndex se puede cambiar la opción por la que empieza.

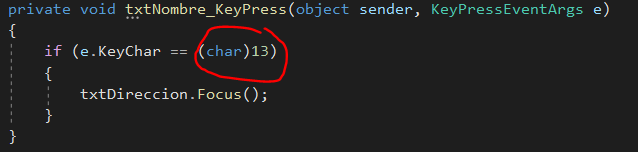
### Eventos de pulsar teclas:

Los eventos de pulsar teclas se manejan en el elemento que tengamos el foco.

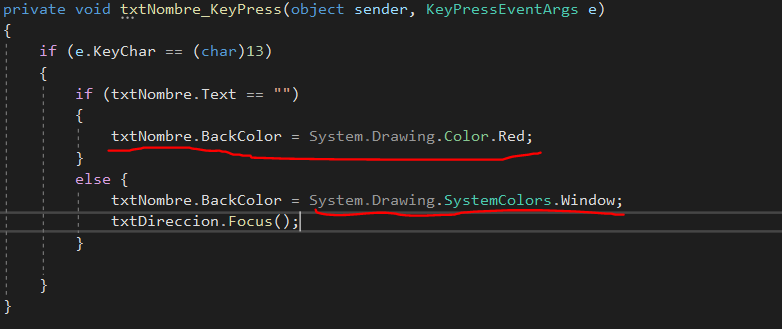
En este caso para añadir el evento en el txtNombre hay que ir a la etiqueta de KeyPress en la sección de eventos:



Este evento detecta cualquier tecla que se pulse dentro de ese txtBox por lo que el controlar la tecla pulsada hay que hacerlo dentro del código de evento.



En este caso la tecla pulsada que nos interesa es el retorno de carro que en ASCII corresponde al 13, si quisiéramos tener en cuenta otras teclas el numero que usaríamos sería otro. Tambien podemos hacer cosas mas complejas dentro del código, vamos a hacer que compruebe que hay algo escrito y si no lo hay que no te deje cambiar.



Además de no cambiar el foco pone el fondo rojo para que sepas que ha habido un error.

### Botón Salir:

Para cerrar un formulario se usa el método Close();

### Cambiar la posición del formulario:

Se cambia la propiedad Start Position

