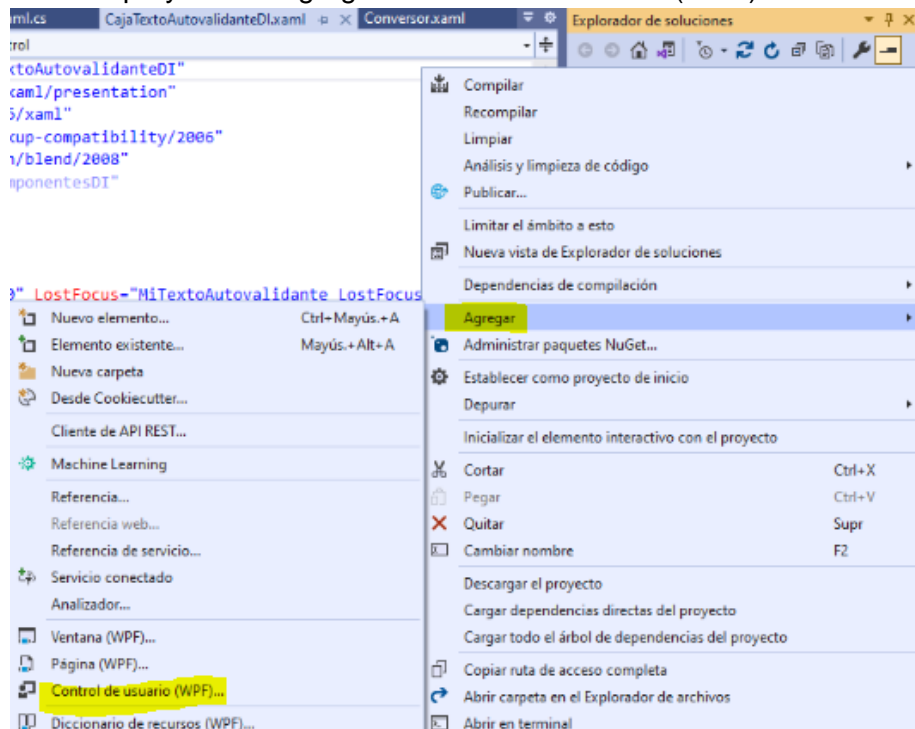


Creación de componentes visuales en Microsoft Visual Studio

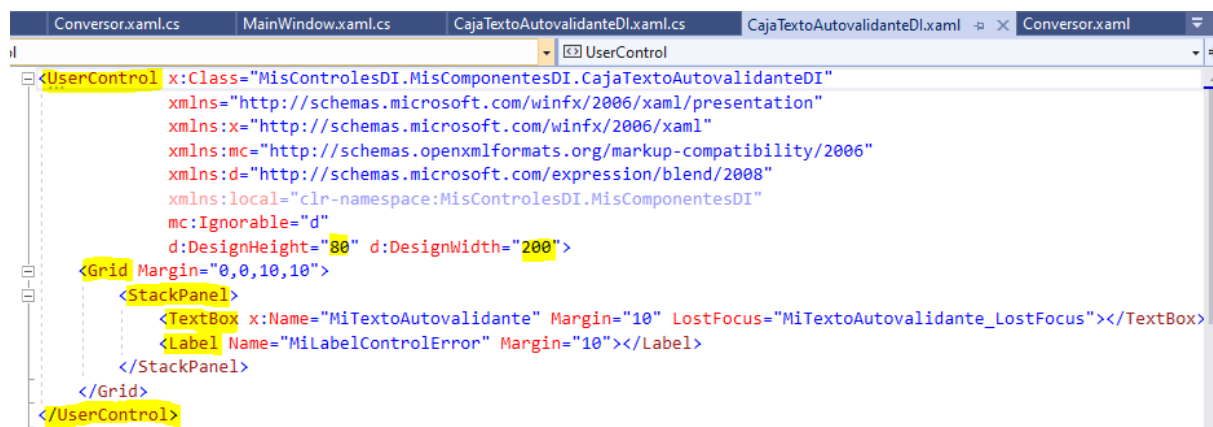
Tenemos que crear un proyecto y añadir un componente con el botón derecho del ratón sobre el proyecto -> Agregar -> Control de usuario (WPF)



Le damos nombre a nuestro componente: CajaTextoAutovalidante.

Nos generará un archivo xaml y otro asociado cs.

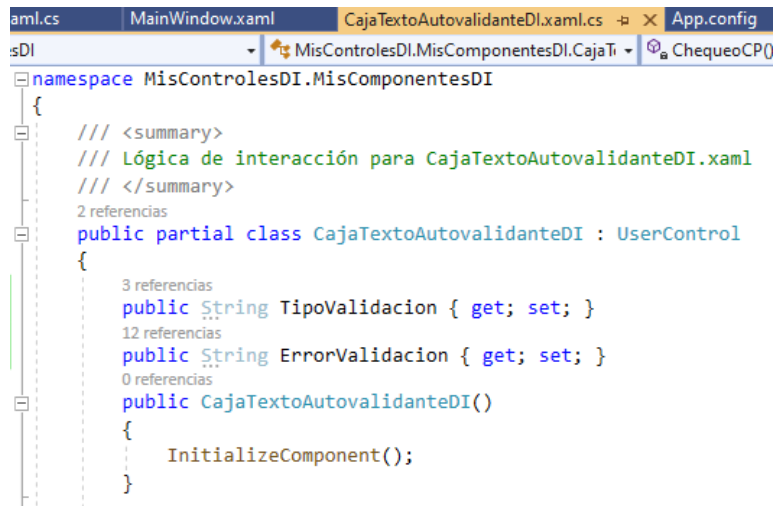
En el archivo xaml veremos la etiqueta <UserControl> como señal de que estamos creando un componente de usuario.



En nuestro caso vamos a crear un componente con dos controles: una **caja de texto** y una **etiqueta**. Para que los controles salgan apilados, los metemos dentro de un **StackPanel**. Utilizamos la propiedad Margin para separar un poco los controles.

La idea es que podamos validar el valor introducido por el usuario en la caja de texto, según el tipo de valor esperado, y el resultado de la validación se muestre en la etiqueta.

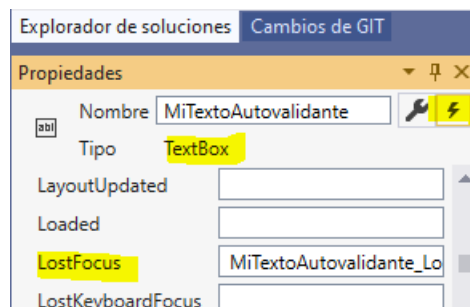
Para ello debemos crear un **atributo** en nuestro componente “**TipoValidacion**”, donde podamos indicar qué valor se va a introducir, por ejemplo: DNI, TFN, CP... y en función del valor de dicha propiedad o atributo, hagamos las validaciones correspondientes.



```
aml.cs | MainWindow.xaml | CajaTextoAutovalidanteDI.xaml.cs | App.config
sDI | MisControlesDI.MisComponentesDI.CajaTi | ChequeoCP0
namespace MisControlesDI.MisComponentesDI
{
    /// <summary>
    /// Lógica de interacción para CajaTextoAutovalidanteDI.xaml
    /// </summary>
    2 referencias
    public partial class CajaTextoAutovalidanteDI : UserControl
    {
        3 referencias
        public string TipoValidacion { get; set; }
        12 referencias
        public string ErrorValidacion { get; set; }
        0 referencias
        public CajaTextoAutovalidanteDI()
        {
            InitializeComponent();
        }
    }
}
```

También creamos una variable **ErrorValidación** para guardar el mensaje a mostrar en la etiqueta tras la validación.

El evento de la validación será cuando la caja **pierda el foco**. Este evento lo podemos generar desde la ventana de propiedades del TextBox, en la pestaña del rayo, mirando los eventos encontramos **LostFocus**, hacemos doble clic en la caja de texto y se genera el método:



En el archivo cs veremos el método creado asociado al evento:

```

App.config  Conversor.xaml.cs  MainWindow.xaml.cs  CajaTextoAutovalidanteDI.xaml.cs  CajaTextoAutovalidanteDI.xaml
MisControlesDI  MisControlesDI.MisComponentesDI.CajaTextoAutovalidanteDI  ChequeoTFN()

104  }
105  1 referencia
106  private void MiTextoAutovalidante_LostFocus(object sender, RoutedEventArgs e)
107  {
108      ErrorValidacion = "No hay Error";
109
110      if (TipoValidacion == "DNI")
111      {
112          ChequeoDNI();
113      }
114      else if (TipoValidacion == "TFN")
115      {
116          ChequeoTFN();
117      }
118      else if (TipoValidacion == "CP")
119      {
120          ChequeoCP();
121      }
122
123      MiLabelControlError.Content = ErrorValidacion;
124  }

```

y según el valor informado en TipoValidacion tendremos creados los métodos de chequeo correspondientes:

```

private void ChequeoCP()
{
    string TextoInsertado;
    TextoInsertado = MiTextoAutovalidante.Text;
    if (TextoInsertado.Length == 5)
    {
        try
        {
            int numeros = Int32.Parse(TextoInsertado);
            ErrorValidacion = "No hay Error";
        }
        catch (FormatException e)
        {
            ErrorValidacion = "El CP sólo debe tener números";
        }
    }
    else
    {
        ErrorValidacion = "El código postal debe tener 5 dígitos";
    }
}

```

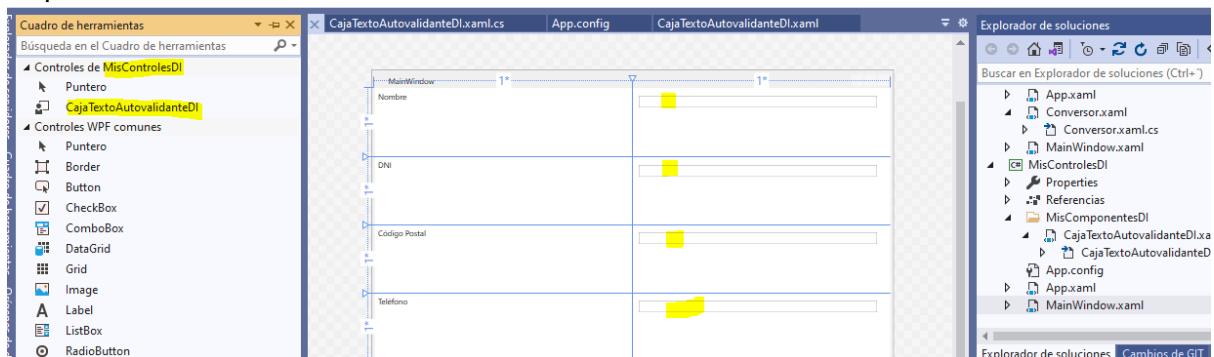
```
private void ChequeoTFN()
{
    String TextoInsertado;
    TextoInsertado = MiTextoAutovalidante.Text;
    if (TextoInsertado.Length == 9)
    {
        try
        {
            int numeros = Int32.Parse(TextoInsertado);
            ErrorValidacion = "No hay Error";
        }
        catch (FormatException e)
        {
            ErrorValidacion = "El teléfono sólo debe tener números";
        }
    }
    else
    {
        ErrorValidacion = "El teléfono debe tener 9 dígitos";
    }
}
```

```

private void ChequeoDNI()
{
    String TextoInsertado;
    TextoInsertado = MiTextoAutovalidante.Text;
    if (TextoInsertado.Length == 9)
    {
        try
        {
            int numeros = Int32.Parse(TextoInsertado.Substring(0,7));
            try
            {
                int letra = Int32.Parse(TextoInsertado.Substring(8));
                ErrorValidacion = "El último dígito debe ser una letra";
            }
            catch (FormatException e)
            {
                ErrorValidacion = "No hay errores";
            }
        }
        catch (FormatException e)
        {
            ErrorValidacion = "Los 8 primeros dígitos del dni deben ser números";
        }
    }
    else
    {
        ErrorValidacion = "El DNI debe tener 9 dígitos";
    }
}

```

Una vez creado el componente, **compilamos** el proyecto y tendríamos el componente disponible en el cuadro de herramientas:



Para organizar un poco los campos creamos un **grid** con filas y columnas:

```
ml.cs | MainWindow.xaml* | CajaTextoAutovalidanteDI.xaml.cs | App.config | CajaTextoAutovalidanteDI.xaml | Conversor.xaml
Grid
xmlns:MisComponentesDI="clr-namespace:MisControlesDI.MisComponentesDI" x:Class="MisControlesDI.MainWindow"
mc:Ignorable="d"
Title="MainWindow" Height="450" Width="800">
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*"></ColumnDefinition>
    <ColumnDefinition Width="*"></ColumnDefinition>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
    <RowDefinition Height="*"></RowDefinition>
  </Grid.RowDefinitions>
  <Label Content="Nombre"></Label>
  <Label Content="DNI" Grid.Row="1"></Label>
  <Label Content="Código Postal" Grid.Row="2"></Label>
  <Label Content="Teléfono" Grid.Row="3"></Label>
  <MisComponentesDI:CajaTextoAutovalidanteDI Grid.Column="1" Height="100"/>
  <MisComponentesDI:CajaTextoAutovalidanteDI Grid.Column="1" Grid.Row="1" TipoValidacion="DNI" Height="100"/>
  <MisComponentesDI:CajaTextoAutovalidanteDI Grid.Column="1" Grid.Row="2" TipoValidacion="CP" Height="100"/>
  <MisComponentesDI:CajaTextoAutovalidanteDI Grid.Column="1" Grid.Row="3" TipoValidacion="TFN" Height="100"/>
</Grid>
</Window>
```