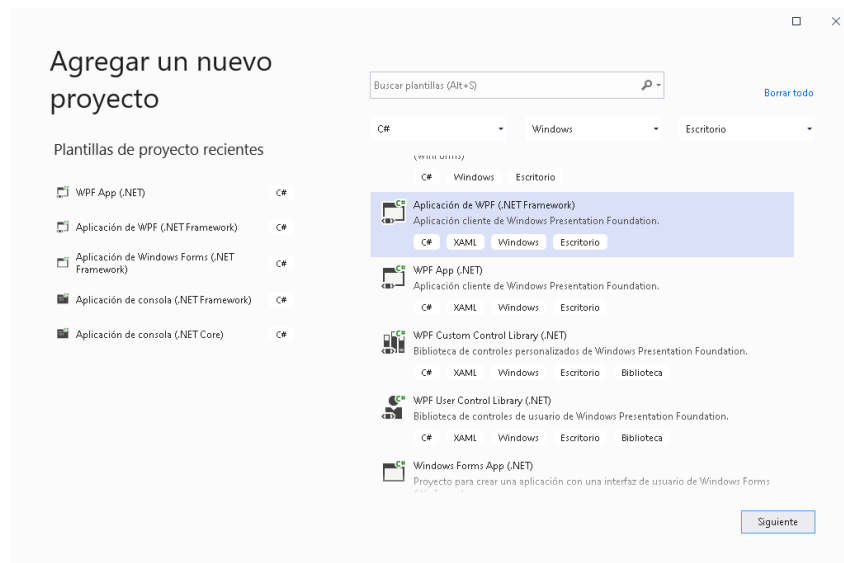


## Hola Mundo WPF

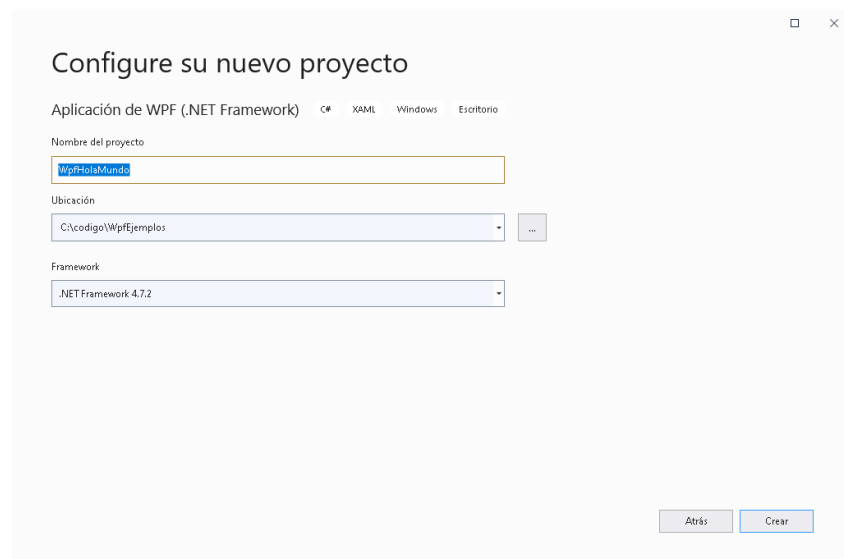
Vamos a ver como creamos una aplicación hola mundo de escritorio WPF, para ello en visual estudio creamos una nueva solución o un nuevo proyecto y filtramos por las opciones de WPF.

Seleccionamos “Aplicación de WPF (.NET Framework)” y pulsamos en siguiente. Ten cuidado en este punto porque también está disponible la versión de .NET Core en “WPF App (.NET)” y no son del todo compatibles.

NOTA: En este caso voy a agregar un nuevo proyecto a una solución existente.

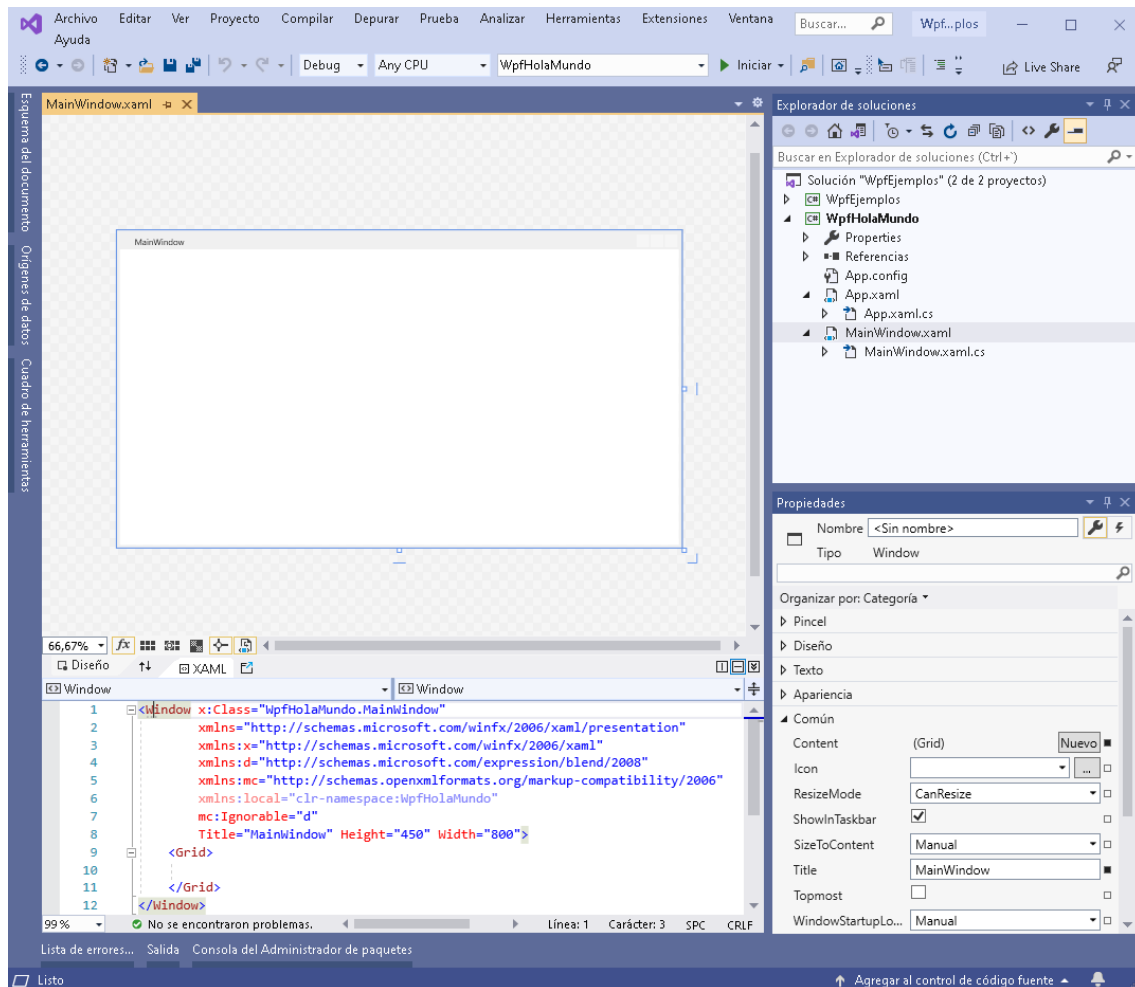


Le damos un nombre a nuestro proyecto, en este caso “WpfHolaMundo”. Indicamos la ubicación y la versión de Framework.



Y ya tenemos la estructura básica de nuestro proyecto. Por defecto se abre la ventana “MainWindow.xaml”. Fíjate que contamos con dos áreas de trabajo en el diseñador de la ventana, la superior o vista de diseño con la representación gráfica y la inferior con el código XAML. Cualquier cambio que realices en cualquiera de las dos áreas se reflejara en la otra.

Eso sí, si rompes el código XAML con por ejemplo una errata o etiqueta mal cerrada, veras que la vista de diseño muestra un mensaje indicando que se han encontrado errores.



Como es habitual, la caja de propiedades muestra en cada momento las propiedades del elemento seleccionado en cualquiera de las vistas del diseñador.

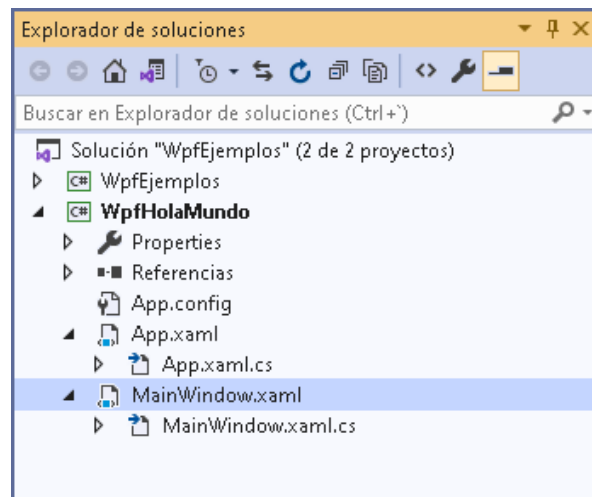
Es importante asignar la propiedad “Nombre” a los controles en especial si vas a manipularlos por código o a referenciarlos con un Binding. Fíjate que en la parte XAML las propiedades sin valor no se incluyen.

En el explorador de soluciones tenemos 3 ficheros:

- App.config, con las opciones de configuración de la aplicación.
- App.xaml, contiene la etiqueta de entrada de la aplicación.
- MainWindow.xaml, ventana principal de la aplicación.

Fíjate que además por cada fichero xaml tenemos un fichero con el mismo nombre y acabado en .cs que contiene el código C# de respaldo.

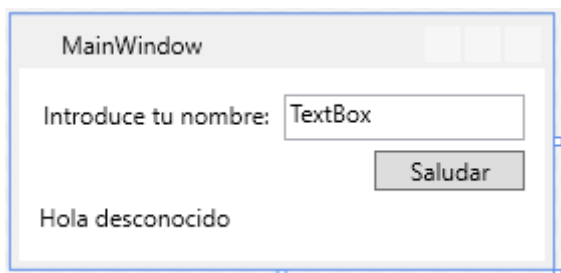
De momento vamos a trabajar con la ventana, más adelante volveremos a los otros ficheros.



Vamos a agregarle funcionalidad a la ventana. Para ello vamos a arrastrar algunos controles desde el cuadro de herramientas.

1. Un label con el Content "Introduce tu nombre:"
2. Un textbox con el Name "txtNombre"
3. Un button con el Name "btnSaludar" y el Content "Saludar"
4. Un textblock con el Name "tbSaludo" y el Text "Hola desconocido"

Deberías ver algo así.



Además, vamos a implementar el evento "Click" en el botón. Escribe en la etiqueta xaml del botón Click y pulsa el tabular para autocompletar.

```
<Grid>
    <Label Content="Introduce tu nombre: " HorizontalAlignment="Left" Margin="7,10,0,0" VerticalAlignment="Top" />
    <TextBox x:Name="txtNombre" Height="23" Margin="132,13,10,0" TextWrapping="Wrap" Text="" />
    <Button x:Name="btnSaludar" Content="Saludar" Click="btnSaludar_Click" HorizontalAlignment="Left" Margin="10,66,0,0" TextWrapping="Wrap" />
    <TextBlock x:Name="tbSaludo" HorizontalAlignment="Left" Margin="10,66,0,0" TextWrapping="Wrap" />
</Grid>
```

Échale un ojo al código XAML autogenerated, verás que hay muchas propiedades que se han establecido automáticamente, esto se debe principalmente a que estamos ubicando los controles a mano sin editar un layout apropiado.

Verás también que algunas propiedades como "Margin" se parecen mucho a HTML, y no solo se parecen, sino que también copian otros aspectos como su notación, por ejemplo, con Margin puedes indicar 1, 2 o 4 valores para indicar los valores de la misma forma que lo harías en la web.

**NOTA:** Es muy posible que hayas creado sin querer elementos de más, por ejemplo, una plantilla con el formato de las filas y columnas para el Grid, bórralos sin problemas.

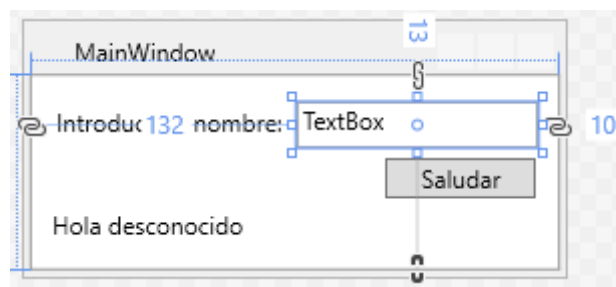
Modifica código del evento Click para que modifique el mensaje del TextBlock.

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }

    private void btnSaludar_Click(object sender, RoutedEventArgs e)
    {
        string nombre = this.txtNombre.Text;
        this.tbSaludo.Inlines.Clear();
        this.tbSaludo.Inlines.Add("Hola " + nombre);
    }
}
```

**IMPORTANTE:** Una diferencia importante con WinForms es que en WPF los controles no tienen propiedades para acoplar “Dock” y para anclar “Anchor”. Que no existan estas propiedades no significa que no podamos conseguir el mismo comportamiento.

- Para acoplar elementos debemos emplear un DockPanel.
- Para anclar elementos debemos emplear el diseñador. Fíjate en los iconos de cadena a los lados del control, cuando están unidos están anclados, si el eslabón está roto no está anclado. En código XAML se refleja en la propiedad “Margin”, valor 0 no anclado, cualquier otro valor ancla con el valor indicado.



Con el cambio anterior al redimensionar la ventana verás como la caja de texto se ajusta al tamaño de la ventana.

## Fichero App.xaml

Este fichero es el punto de entrada de la aplicación, tanto para los elementos XAML como para el código fuente.

Los principales usos para este fichero son:

- Definir la ventana de inicio, propiedad “**StartupUri**”, opcional, aunque por defecto se le da valor.
- Definir el evento de inicio, evento “**Startup**”, posibles usos:
  - Gestionar los parámetros de entrada. Parámetro “e.Args”
  - Arrancar servicios adicionales necesarios para la aplicación.
  - Arrancar una ventana de forma condicional por código.
- Definir el evento de excepción no controlada, evento “**DispatcherUnhandledException**”, posibles usos:
  - Logear o comunicar los errores no esperados. Parámetro “e.Exception”.

- Definir los recursos globales de la aplicación, etiqueta **"Application.Resources"**.  
Posibles usos:
  - Almacenar datos.
  - Definir estilos, globales para los controles XAML de la aplicación o mediante clave para aplicar explícitamente, por ejemplo, estilo para mensajes de error.
  - NOTA: Considera envolver los recursos globales con la etiqueta `ResourceDictionary`, de esta manera los datos se cachearán.

```
<Application x:Class="WpfHolaMundo.App"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:local="clr-namespace:WpfHolaMundo"
StartupUri="MainWindow.xaml">
  <Application.Resources>
    <!--
      Aquí van los recursos de la aplicación
      Pueden ser datos o lo más común estilos
    -->
  </Application.Resources>
</Application>
```

### Fichero App.config

Este fichero permite almacenar los parámetros de configuración, por defecto únicamente incluye la versión del framework que ejecuta la aplicación.

Otros usos comunes son:

- Definición de las cadenas de conexión.
- Parámetros de configuración de módulos adicionales.
- Parámetros que definamos en la propia aplicación.

El fichero tiene formato XML y debe distribuirse junto con el ejecutable de la aplicación.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0"
sku=".NETFramework,Version=v4.7.2" />
  </startup>
</configuration>
```