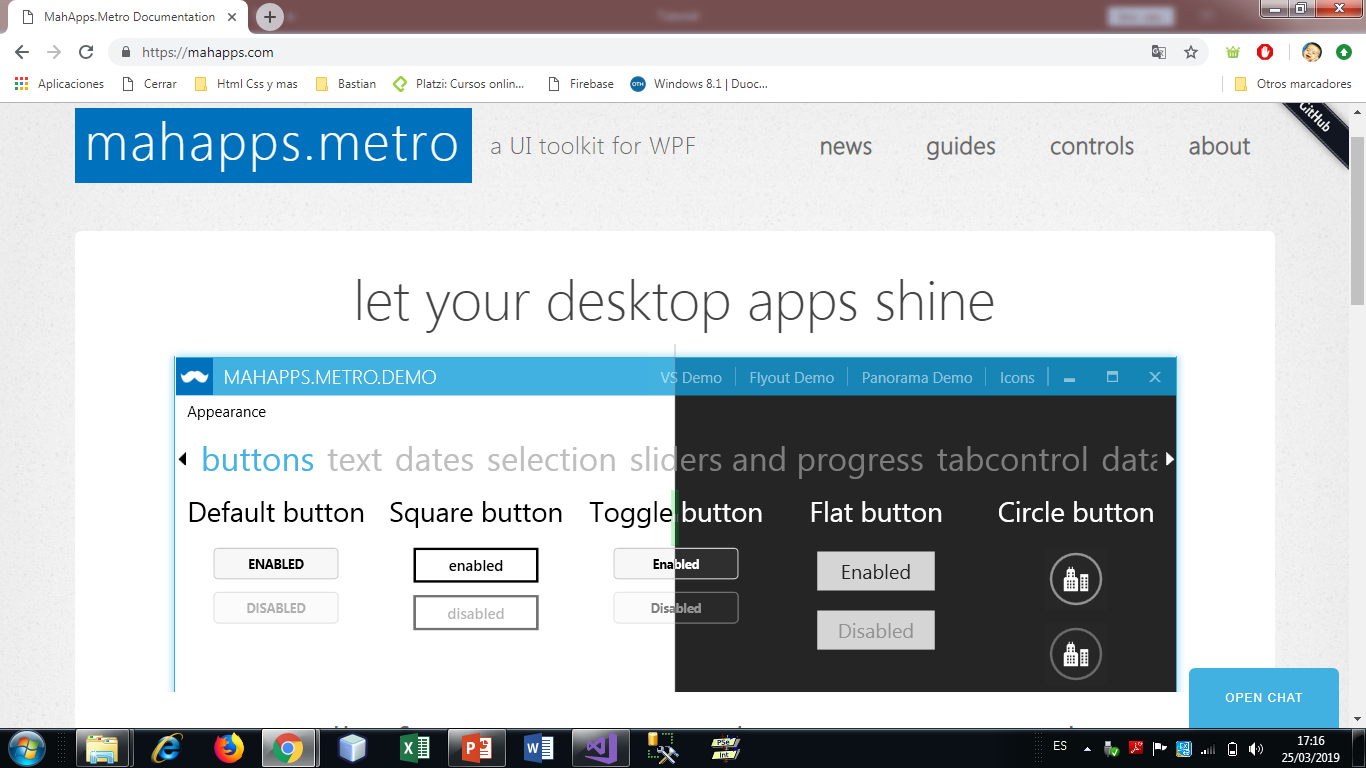
**Tutorial WPF Modern UI**

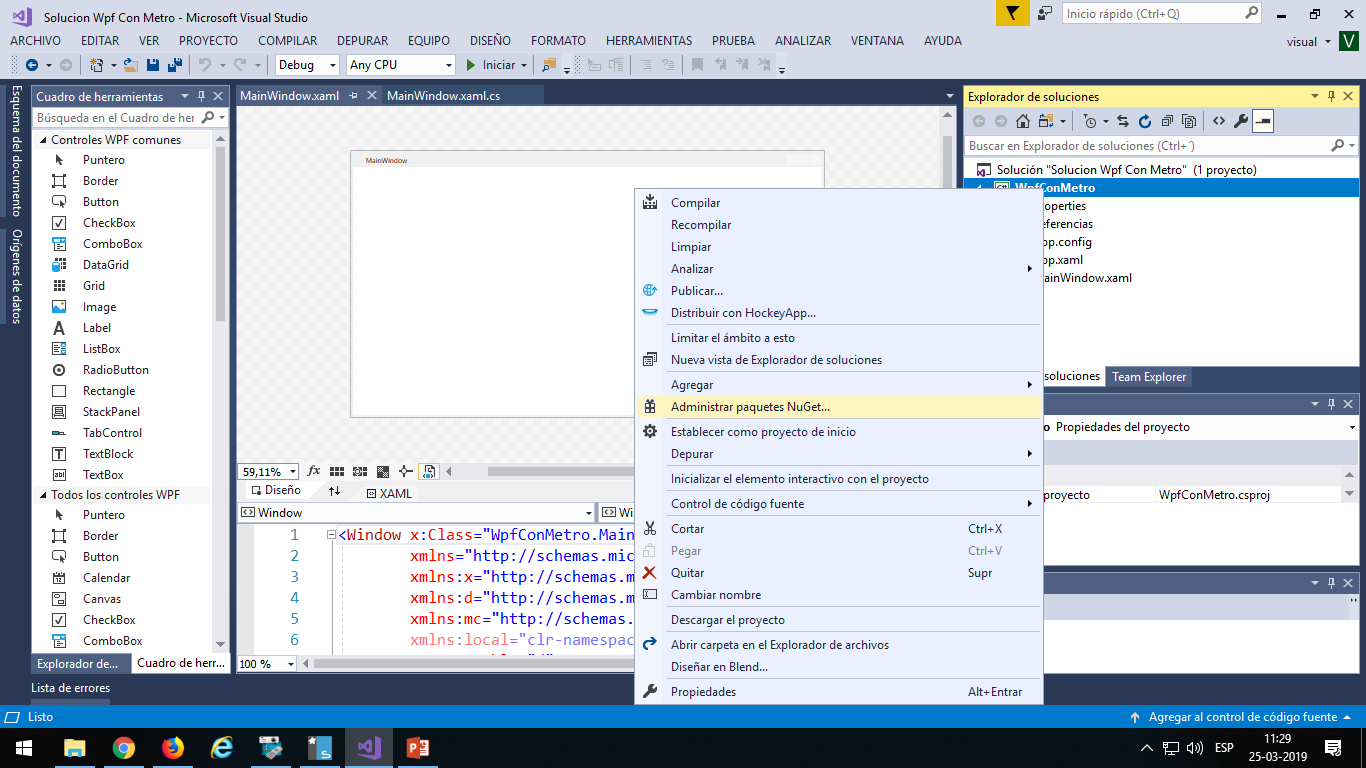
En este tutorial haremos una aplicación WPF que utilizará estilo Modern UI Style, el sitio oficial dónde puede encontrar más información del estilo es <https://mahapps.com/>



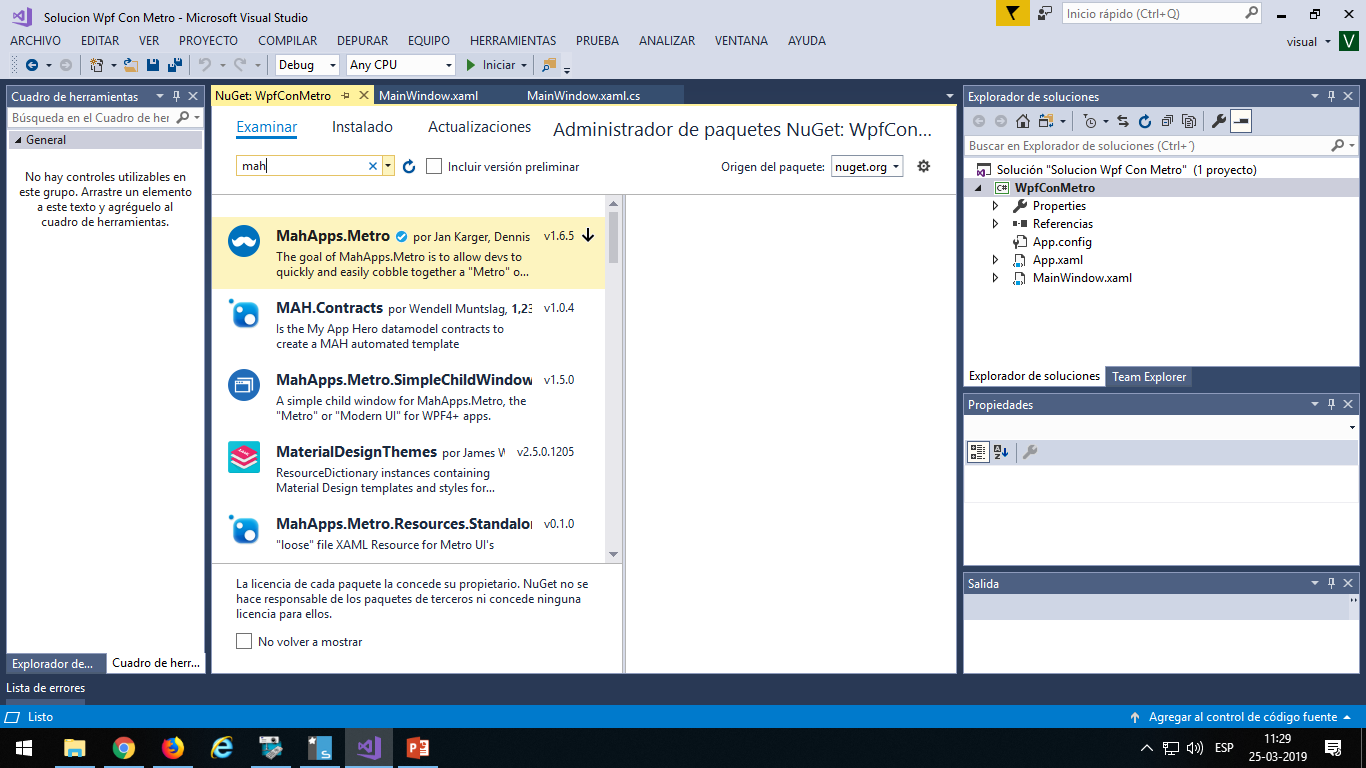
# Configuración:

Cree una solución con un proyecto WPF y luego siga los pasos:

1. Lo primero que se debe indicar, es que para utilizar este estilo se debe agregar una referencia al paquete MahApps por medio del administrador de paquetes Nuget.



Se busca en el examinador “mahapps” y se instala **MahApss.Metro**, también pueden instalarse otros paquetes más, como **MahApss.Metro.Resources, MahApss.Metro.IconPacks** (investigue los que necesita, dependiendo de las funciones que quiera usar).



1. Para que WPF integre el estilo a la plantilla hay que agregar algunas configuraciones al archivo **App.xaml** (dejaré las líneas que se agregarán para fines prácticos, recuerde que debe investigar, para qué sirven)

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.xaml" />

<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.AnimatedSingleRowTabControl.xaml" />

<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Fonts.xaml" />

<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Colors.xaml" />

<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Purple.xaml" />

<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseLight.xaml" />

<ResourceDictionary Source = "pack://application:,,,/MahApps.Metro;component/Styles/FlatButton.xaml" />

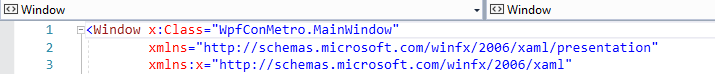
<ResourceDictionary Source = "pack://application:,,,/MahApps.Metro;component/Styles/FlatSlider.xaml" />

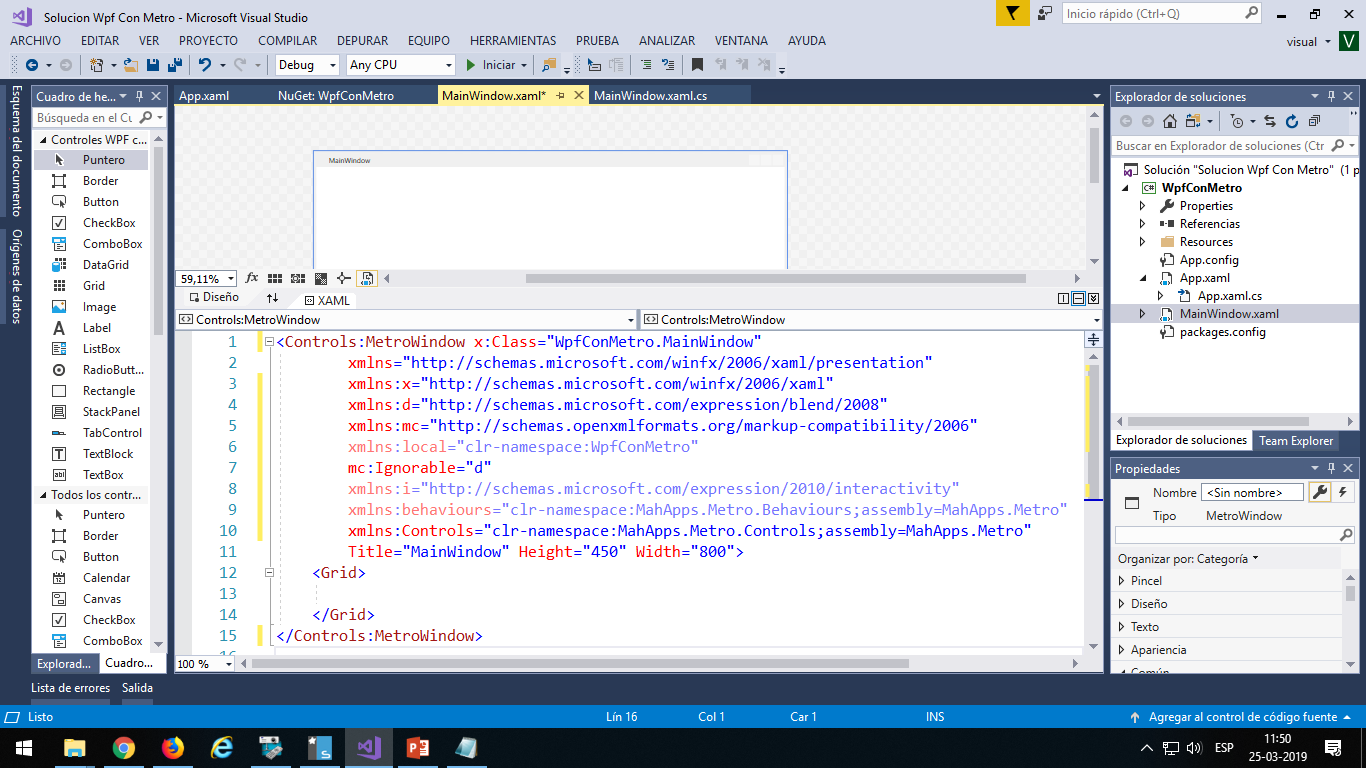
<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.AnimatedTabControl.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

1. Se debe modificar el archivo MainWindow.xaml



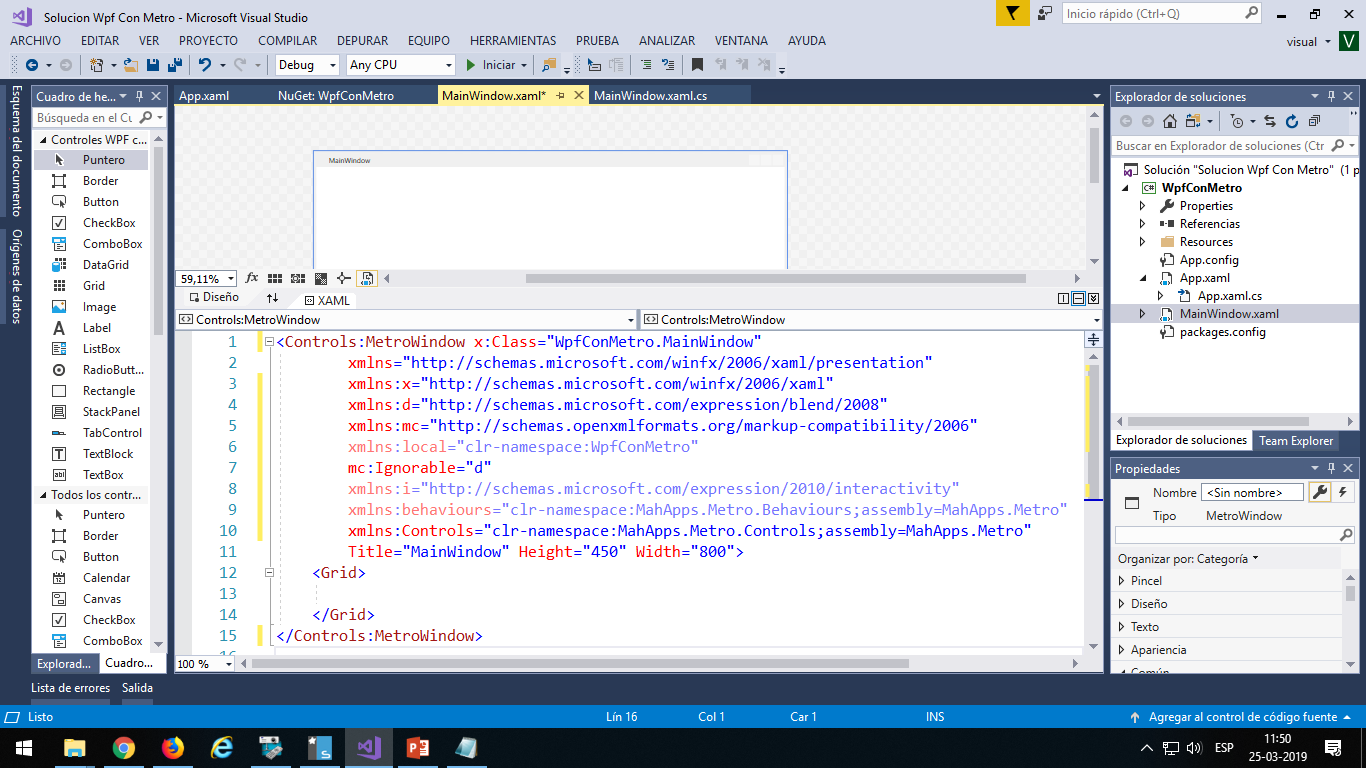


Y se agregan las siguientes líneas:

xmlns:i="http://schemas.microsoft.com/expression/2010/interactivity"

xmlns:behaviours="clr-namespace:MahApps.Metro.Behaviours;assembly=MahApps.Metro"

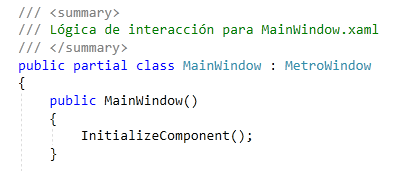
xmlns:Controls="clr-namespace:MahApps.Metro.Controls;assembly=MahApps.Metro"



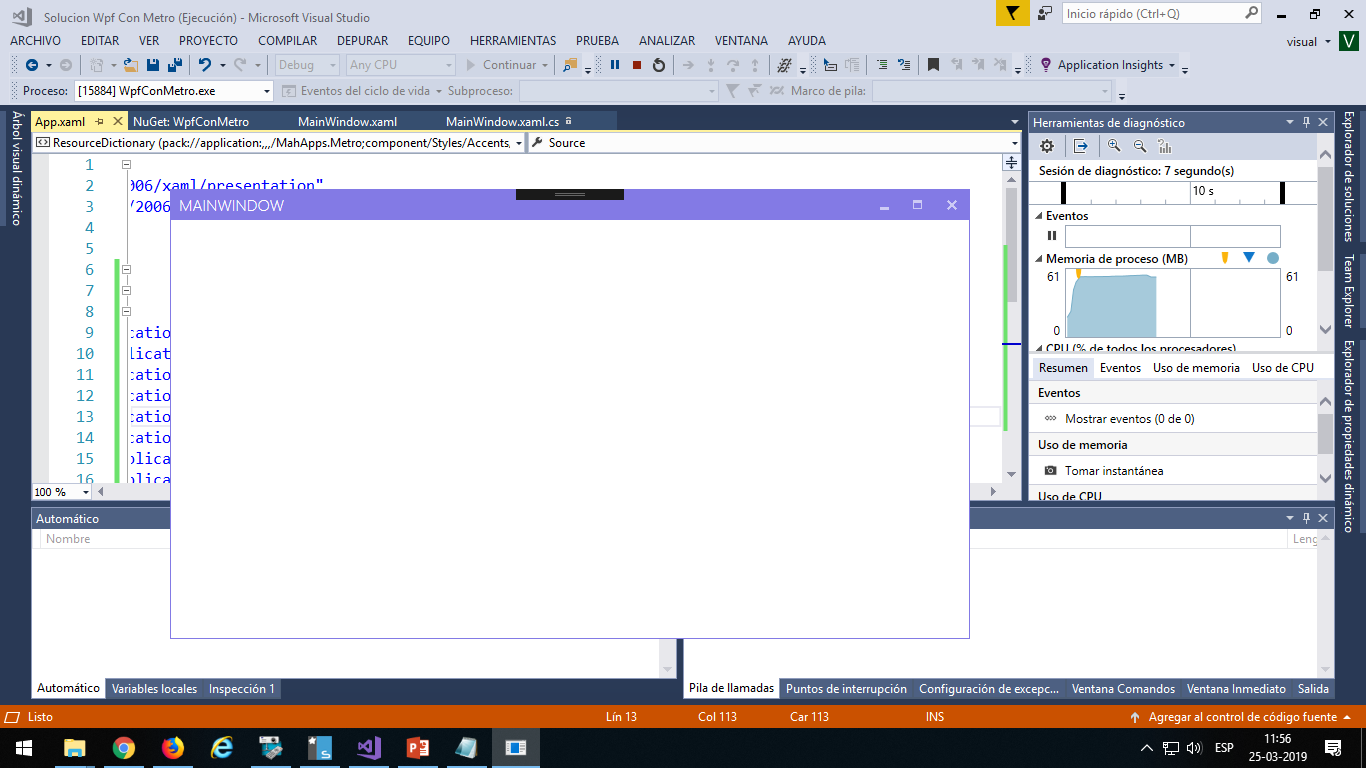
1. Con el paso anterior, ya está agregado el estilo metro a la ventana, pero si lo ejecutamos, daría un error. Por esto, también se debe agregar un using:

using MahApps.Metro.Controls;

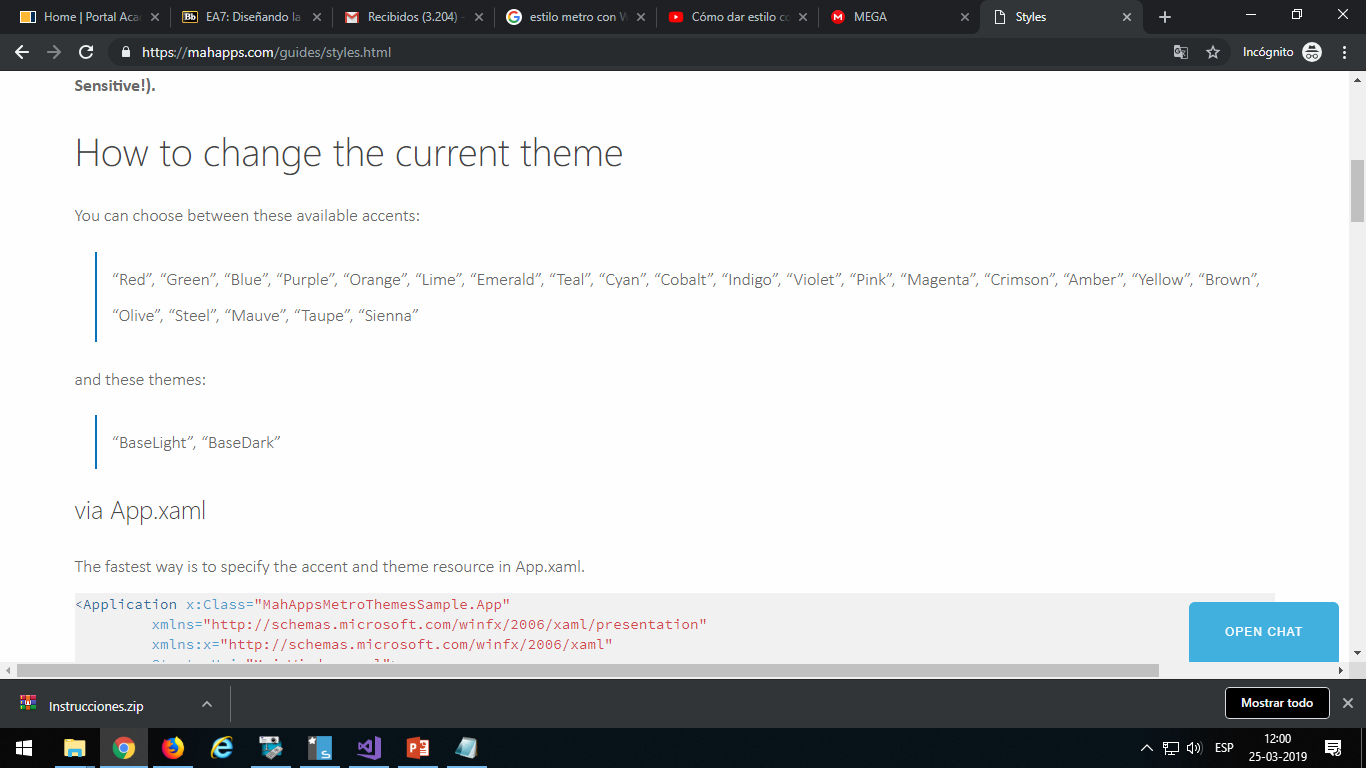
1. Finalmente se debe indicar que la ventana hereda de Metro:



Con los pasos recién descritos ya puedo visualizar la ventana con estilo Metro:

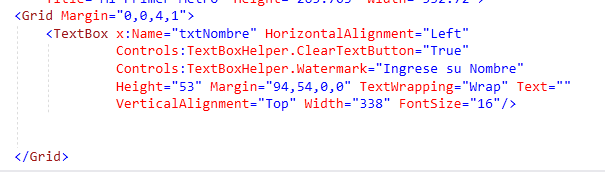


El color de la ventana está dado por la configuración del App.xaml, para ver otros colores pueden revisar la página oficial:

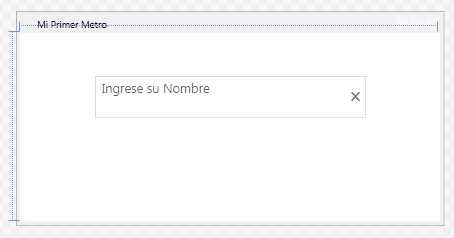


# Agregar estilo Metro a los Controles

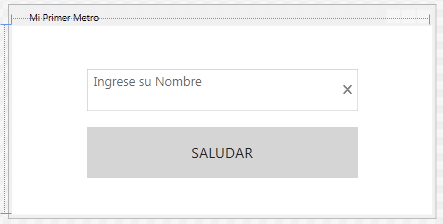
1. Agregar un control TextBox a la ventana, posicionarse en el código XAML en la declaración del TextBox y agregar propiedad Controls:TextBoxHelper.ClearTextButton="True" para que en el control aparezca una “X” que sirve para borrar el TextBox.
2. Si quiero agregar un texto al control como una marca de agua, se coloca Controls:TextBoxHelper.Watermark="Ingrese su Nombre"



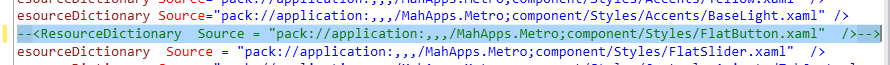
Al visualizar la ventana va quedando así:

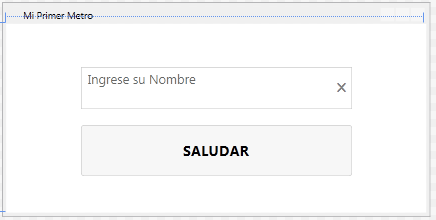


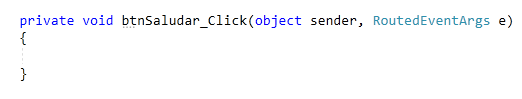
1. Agregar un botón



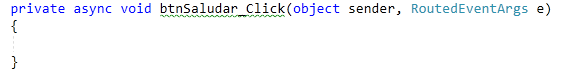
Y para cambiar su apariencia, por ejemplo, marcar como comentario en App.xaml la línea que se muestra a continuación:



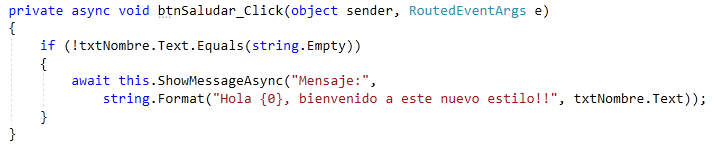


Programo el evento Click del botón

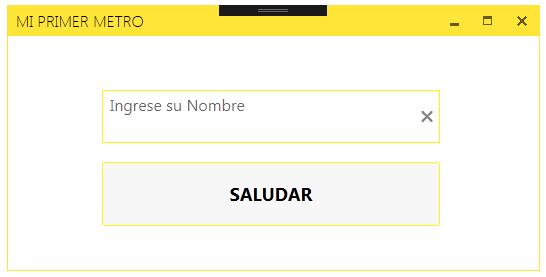
Pero lo modifico agregando “asyn” (**asyn**, investigar que significa)



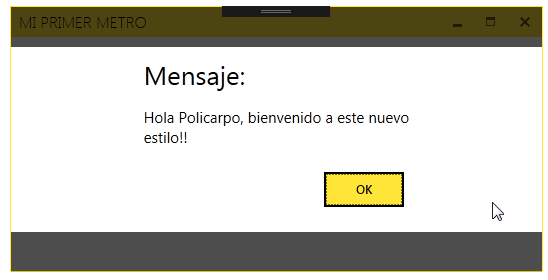
Pista: Sin ‘asyn’, NO se puede agregar mensajes que vayan con estilo Metro.



1. Ejecuto la aplicación:



Ingreso un nombre y pruebo



|  |  |
| --- | --- |
| Resultado de imagen para tips png | Para más información de cómo utilizar Mahapps en WPF y C#, te recomendamos los siguientes video tutoriales:  Parte 1 – Login: https://www.youtube.com/watch?v=4rXiDura-qk&t=584s  Parte 2 – Menú: https://www.youtube.com/watch?v=xubhMLbQO6E  Parte 3 – Flyout: https://www.youtube.com/watch?v=lNe1pcCe4Ek  Parte 4 – Flip View: https://www.youtube.com/watch?v=13ivEWwj2wk  Parte 5 – Tab Control: https://www.youtube.com/watch?v=En3V3DL1OSw |