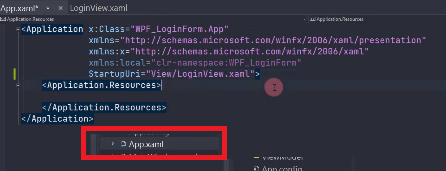
<https://www.youtube.com/watch?v=ebRTeKld4Nc&t=0s>

Lo primero que hace es crear una carpeta donde pondrá las imágenes del login.

Crea también carpeta view y ahí crea ventana (WPF).

Luego va al archivo principal (App.xaml) y ahí modifica la ventana de arranque de la app.



Recomienda editar las ventanas desde el editor de código y no arrastrando las herramienta como con WF. Dice que es mas “cómodo” y que se puede personalizar mas.

XAML es muy similar a HTML

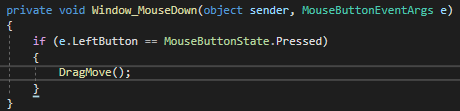
Quitar borde de ventana

WindowStyle="None"

Mouse desplaza ventana

MouseDown="Window\_MouseDown"

Codea eso y luego a ese evento “Window MouseDown le da código en C#.



5 tipos principales de paneles:

1-StackPanel. Organiza los elementos en una sola línea, vertical u horizontal.

2-DockPanel. Los agrupa hacia cualquier dirección: izquierda, derecha, arriba o abajo.

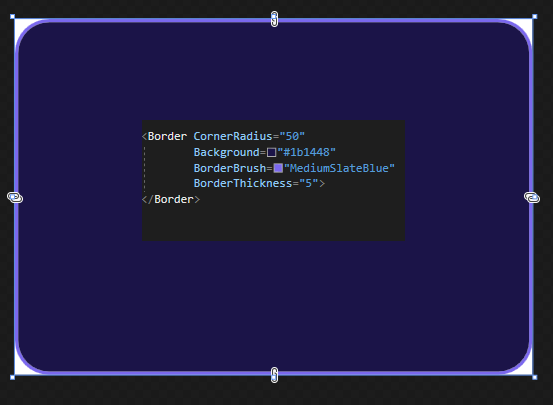
3- Grid. Es el mas poderoso. Define una cuadrícula.

4-WrapPanel.

5-Canvas. El mas sencillo. Coloca mediante coordenadas relativas.

A estos les suma la etiqueta Border. Que permita personalizar el estilo, que ninguno de los panels lo hace muy bien.

Border



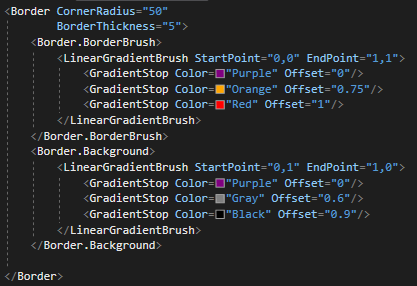
Para que el fondo no se vea blanca, hay que darle atributos a la ventana (no al border) siguientes:

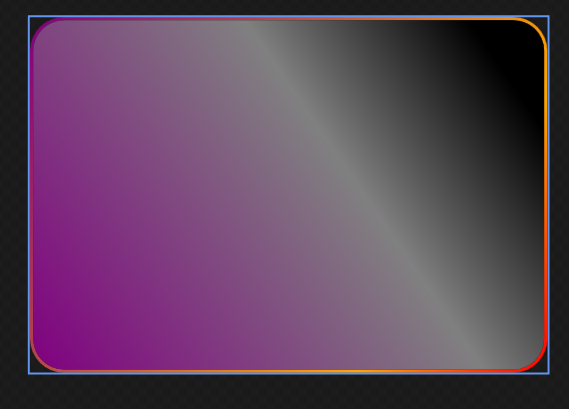
Background="Transparent"

AllowsTransparency="True">

Degradados al borde y al relleno

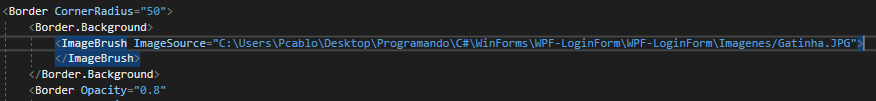
Esto es un poco avanzado:





Agregar Imagen de fondo

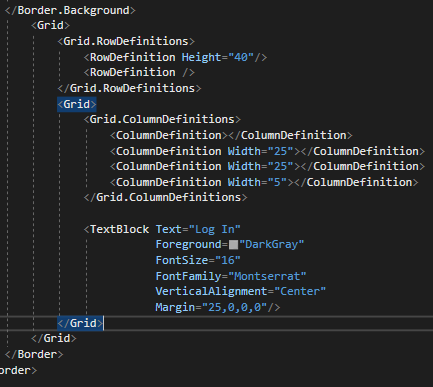
Se anida el borde previo a uno nuevo, que lo envuelve



Recordar bajarle la opacidad a el otro border



Grid+agregar texto

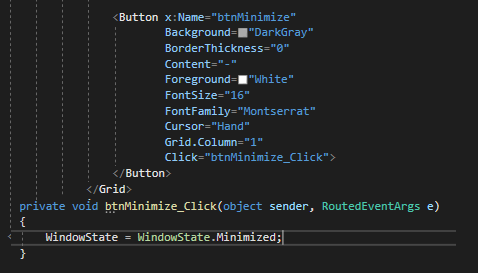


Ahí se crean 2 filas. La primera con altura de 30, la otra adquiere el restante del alto. Se crean 4 columnas, a 3 les defino el ancho, la otra es el sobrante.

Grid.Column="0" Ese 0 refiere al índice de la columna. Hay 4 y esta es la primera, por eso el 0.

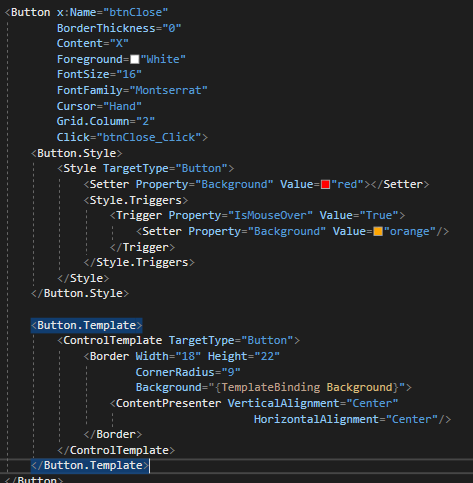
Margin="25,0,0,0"/> izquierda, superior, derecha e inferior.

Agregar botones





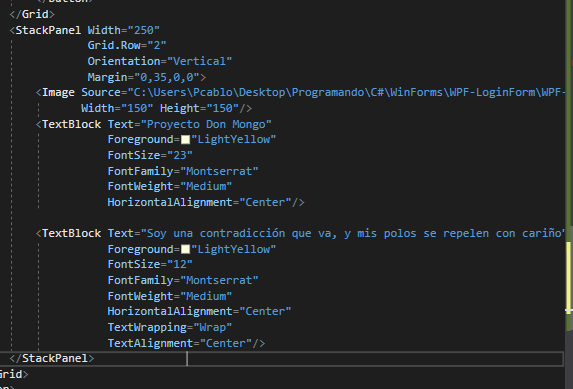
Así quedan los botones, no son feos. Al pasar el mouse encima el cursor se hace manito y los botones cambian de color. Sin embargo, no hay forma sencilla de personalizar mas, hay que tocar mucho código.





StackPanel+ imagen y texto

Se agrega dentro de la segunda columna. Es decir tras cerrar el primer grid y antes de cerrar el segundo.



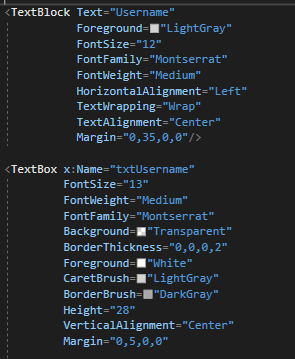
TextWrapping="Wrap"

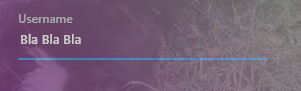
TextAlignment="Center"/>

Esos 2 son para que las líneas de texto no se salgan de la columna y para centrarlo dentro de la columna:

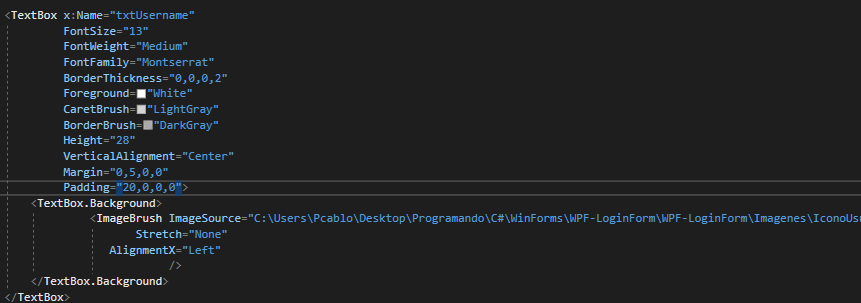


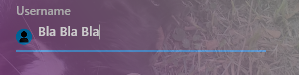
Labels+textboxs (de WF)





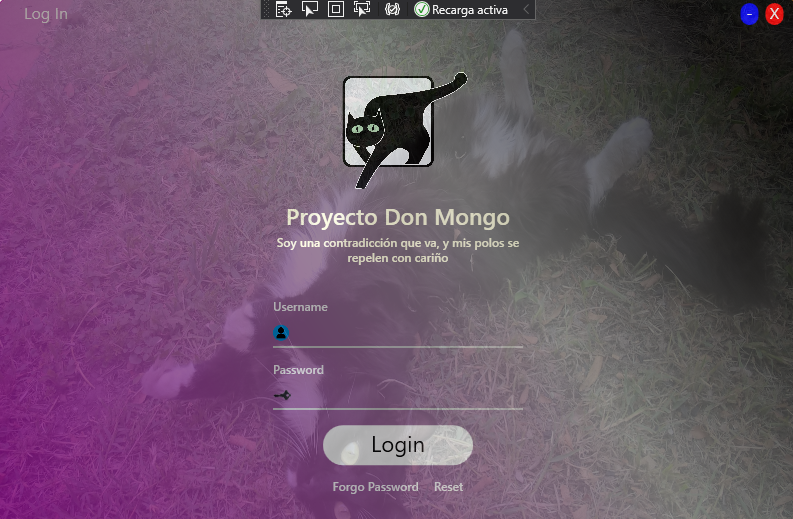
Queda bien. Para mí está muy arriba las letras de la línea. Como quiere agregar un ícono al inicio del textbox tiene que complejizar y eliminar algunas cosas.





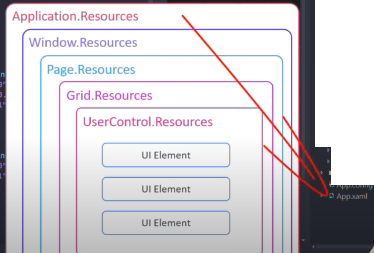
Se borra el background previo. Se agrega padding ya que si no se escribiría encima del logo y luego se agrega el logo. Yo tuve problemas con el tamaño de la imagen y tuve que subir una imagen editada con Photoshop para que sea mas pequeña.

PasswordBox se llama para que se escondan los caracteres.



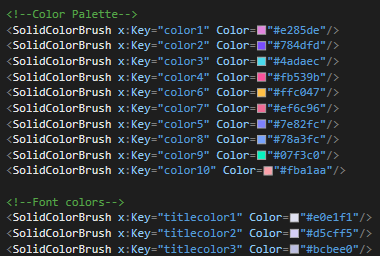
Diccionario de recursos

Habla de definir un estilo en un diccionario de recursos para reutilizar el código.



Cuanto mas afuera mas general eso. Desde la App.xaml se afectaría toda la app.

Agrega una carpeta para almacenarlos.



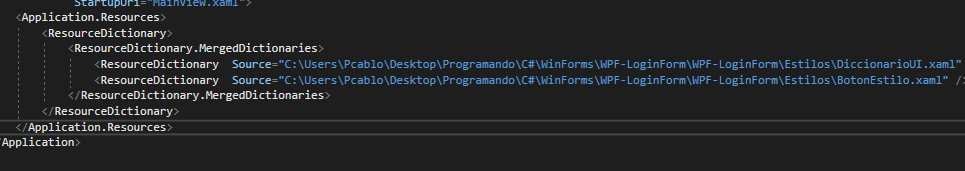
Ahí los mete y luego va a app.xaml y escribe: <ResourceDictionary Source="C:\Users\Pcablo\Desktop\Programando\C#\WinForms\WPF-LoginForm\WPF-LoginForm\Estilos\DiccionarioUI.xaml"/>

También ahí mismo define un estilo de botón.

Descarga librería de Fontawesome.WPF o Fontawesome.Sharp (este también sirve para WF) desde nuget.

Una vez hecho eso hay que agregar la referencia en el archivo en que se está definiendo el estilo:

xmlns:fa ="http://schemas.awesome.incremented/wpf/xaml/fontawesome.sharp"



Cuando se va a definir mas de una “hoja de estilos” hay que hacerlo así, agregando el merged.

También hay que agregar la referencia en cada una de las hojas en que se vaya a usar:

xmlns:fa ="http://schemas.awesome.incremented/wpf/xaml/fontawesome.sharp"

Arquitectura:

Dice que WPF se adapta mejor al modelo Modelo-Vista-Vista-Modelo