## Control DocPanel

Arrastrar un control [[DocPanel]](http://www.joaquin.medina.name/web2008/documentos/informatica/lenguajes/puntoNET/System/Windows/Controls/Wpf_Menus/Wpf_Menus01.html#Ref_DocPanel) sobre cualquier punto del formulario (ojo fijarse que no se deja caer encima de algún control).

En la ventana propiedades modificar los siguientes campos

Listado1

HorizontalAlignment="Stretch" VerticalAlignment="Top" Width="Auto" Margin="0"

El control deberá situarse en la parte superior del formulario ocupando toda su anchura

No es necesario usar un control [DockPanel] pero facilita el mover el menú posteriormente si en necesario y merece la pena utilizarlo

Arrastrar un control [[Menu]](http://www.joaquin.medina.name/web2008/documentos/informatica/lenguajes/puntoNET/System/Windows/Controls/Wpf_Menus/Wpf_Menus01.html#Ref_Menu) y dejarlo caer en el control [DockPanel]

En la ventana propiedades modificar los siguientes campos

Listado2

HorizontalAlignment="Stretch" VerticalAlignment="Top"

Width="Auto" DockPanel.Dock="Top"

Y LastChildFill="False"

Listado3

<Grid>

<DockPanel Name="dockPanelMenu"

Height="31" Width="Auto"

HorizontalAlignment="Stretch"

VerticalAlignment="Top"

DockPanel.Dock="Top"

LastChildFill="False">

<Menu Name="MenuGeneral"

Height="23" Width="Auto"

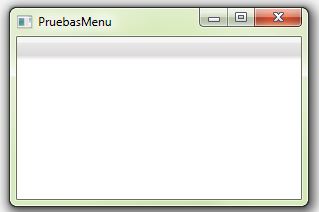
DockPanel.Dock="Top">

</Menu>

</DockPanel>

</Grid>

El control deberá aparecer como una barra gris que ocupa la parte superior del control [DockPanel]



A continuación, en la ventana propiedades, hacer clik en [Items] se abre la ventana del editor de colecciones y allí escribiremos nuestros menús.

Listado4

<Menu Name="MenuGeneral"

Height="23" Width="Auto"

DockPanel.Dock="Top">

<MenuItem Header="\_Archivo" Name=" MenuArchivo">

</MenuItem>

<MenuItem Header="\_Editar" Name=" MenuEditar">

</MenuItem>

<MenuItem Header="\_Herramientas" Name=" MenuHerramientas">

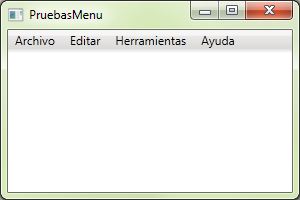
</MenuItem>

<MenuItem Header="\_Ayuda" Name=" MenuAyuda">

</MenuItem>

</Menu>

* El atributo [Header] establece la etiqueta que mostrara el control. El carácter [Guion bajo (\_)] sirve para mostrar debajo del carácter siguiente a donde se encuentra un subrayado y establece el atajo de teclado que se asigna a esta opción de menú.
* Los atajos de teclado, se muestran pulsado la tecla [Tab] y permite usar el teclado, (Pulsando la letra que aparece subrayada) para seleccionar la opción de menú.
* La etiqueta [Name] establece un nombre que identifica a ese elemento



Cada elemento del menú puede tener a su vez una colección de elementos de menú, de esta forma iremos escribiendo el árbol completo de nuestros menús

Ejemplo del menú edición

Listado5

<MenuItem Header="\_Editar" Name="MenuEditar">

<MenuItem Header="\_Cortar" Name="MenuEditarCortar"/>

<MenuItem Header="C\_opiar" Name="MenuEditarCopiar"/>

<MenuItem Header="\_Pegar" Name="MenuEditarPegar"/>

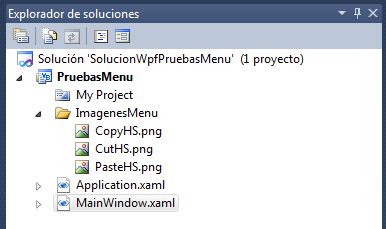
</MenuItem>

Observa que el **atributo [name]** que es el que identifica a este elemento está compuesto por el nombre del elemento padre seguido del nombre del hijo. Esta convención no está escrita por ningún sitio, pero evita problemas posteriores



## Añadir una imagen al menú

Para añadir una imagen al menú, lo primero que hay que hacer es incorporar al proyecto las imágenes, y después importar las imágenes que quiero mostrar en el menú.



A continuación tenemos que escribir el código para incluir las imágenes en el menú y se hace de la siguiente forma

Listado6

<MenuItem Header="\_Editar" Name="MenuEditar">

<MenuItem Header="\_Cortar" Name="MenuEditarCortar">

<MenuItem.Icon>

<Image Source="ImagenesMenu/CutHS.png" />

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="C\_opiar" Name="MenuEditarCopiar">

<MenuItem.Icon>

<Image Source="ImagenesMenu/CopyHS.png" />

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="\_Pegar" Name="MenuEditarPegar">

<MenuItem.Icon>

<Image Source="ImagenesMenu/PasteHS.png" />

</MenuItem.Icon>

</MenuItem>

</MenuItem>



## Añadir un separador en un menú

Para añadir un separador dentro del menú usaremos la etiqueta <Separator />,  a continuación tienes un ejemplo con el menú Ayuda.  
listado7

<MenuItem Header="\_Ayuda" Name="MenuAyuda">

<MenuItem Header="\_Contenidos" Name="MenuAyudaContenidos"/>

<Separator />

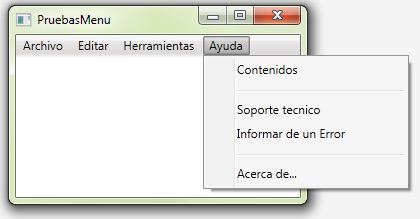
<MenuItem Header="Soporte \_tecnico" Name="MenuAyudaSoporteTecnico"/>

<MenuItem Header="Informar de un \_Error" Name="MenuAyudaInforarError"/>

<Separator />

<MenuItem Header="Acerca \_de..." Name="MenuAyudaAcercaDe"/>

</MenuItem>



## Aplicar un estilo

También podemos aplicar un estilo, definiendo y/o aplicando un estilo que ya exista

Primero se define el estilo

Listado8

<Window.Resources>

<Style x:Key="EstilosDelMenu" TargetType="Menu">

<Setter Property="Background" Value="Coral" />

<Setter Property="Foreground" Value="Navy" />

<Setter Property="FontFamily" Value="Comic Sans MS" />

</Style>

</Window.Resources>

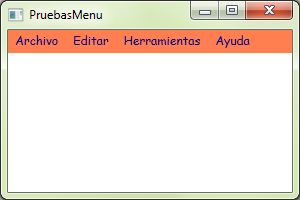
Y a continuación se aplica el estilo al menú

<Menu Name="MenuGeneral"

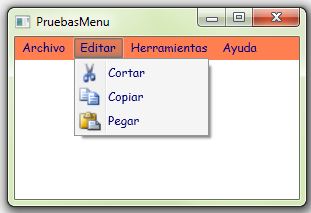
Height="23" Width="Auto"

DockPanel.Dock="Top"

Style="{StaticResource EstilosDelMenu}">



Los estilos se pueden aplicar (o no) a los menús hijos, en este caso solo se aplican al menú padre



## Para añadir un evento

Escribiremos [Click=] y en el menú emergente que se abre haremos clic sobre y aceptaremos el nombre que nos propone el diseñador para la función "Escuchadora? Del evento

Listado9

<MenuItem Header="\_Archivo" Name="Archivo">

<MenuItem Header="\_Nuevo"

Name="MenuArchivoNuevo"

Click="MenuArchivoNuevo\_Click"/>

<Separator />

<MenuItem Header="\_Terminar"

Name="MenuArchivoTerminar"

Click="MenuArchivoTerminar\_Click" />

</MenuItem>

Observa que el nombre del evento se forma usando el texto que hemos escrito en el parámetro Name y se añade [\_Click] ejemplo [MenuArchivoTerminar\_Click]

En el código del formulario se escriben automáticamente el esqueleto de la función que escucha el evento

Listado10

Private Sub MenuArchivoNuevo\_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)

' Falta escribir el codigo

End Sub

Private Sub MenuArchivoTerminar\_Click(sender As System.Object, e As System.Windows.RoutedEventArgs)

Me.Close()

End Sub

Ahora solo tenemos que escribir código para darle contenido a los eventos del menú

## Añadir un Check en un menú

Un menú puede tener la marcha Check a la izquierda del menú y se consigue con el atributo [IsChecked="True"]. Este atributo se puede modificar a través de código

Listado11

<MenuItem Header="\_Herramientas" Name="MenuHerramientas">

<MenuItem Header="Siempre\_Visible"

Name="MenuHerramientasSiempreVisible"

IsChecked="True"

Click="MenuHerramientasSiempreVisible\_Click"/>

</MenuItem>

El código del controlador de eventos será más o menos así

Private Sub MenuHerramientasSiempreVisible\_Click(

sender As System.Object, e As System.Windows.RoutedEventArgs)

' Quitar o poner el Check del menú

Me.MenuHerramientasSiempreVisible.IsChecked =

Not (Me.MenuHerramientasSiempreVisible.IsChecked)

' Establecer la ventana como [siempre visible] en

' función del valor [IsChecked] del menú

Me.Topmost = Me.MenuHerramientasSiempreVisible.IsChecked

End Sub

