Menús

UT5. GENERACIÓN DE INTERFACES A TRAVES DE XML

Menús

Los menús permiten organizar la información de forma jerárquica o mejorar la usabilidad adaptando la opciones al contexto actual de la aplicación.

En WPF los menús se parecen más al maquetado de una web al control autónomo de WinForms.

- Recuerda que para anclar un control en una posición de la ventana debes usar un DockPanel.
- Recuerda que si deseas añadir un icono al texto debes emplear un StackPanel con una Imagen y un TextBlock.

Barra de menú

Requiere de un DockPanel, normalmente DockPanel.Dock=”Top”

Etiqueta “**Menu**”, contenedor base, hasta que no añadas elementos casi no se ve.

Etiqueta “**MenuItem**”, entrada en el menú, se pueden encadenar con la estructura deseada.
Propiedades:

- **Header** -> atributo, texto del menú ítem.
- **Click** -> atributo, manejador del evento.
- Command -> atributo, nombre del comando a ejecutar.
- IsCheckable -> añade una caja de check al ítem.
- IsChecked -> indica si la caja de check del ítem esta activa o no.
- **MenuItem.Icon** -> permite poner un icono al ítem, el content tendrá una etiqueta “Image” con la ruta de la imagen.

Barra de menú - Ejemplo

```
<Menu DockPanel.Dock="Top">
      <MenuItem Header="Archivo">
         <MenuItem x:Name="mnArchivoNuevo" Header="Nuevo" />
         <MenuItem x:Name="mnArchivoAbrir" Header="Abrir" />
         <MenuItem x:Name="mnArchivoGuardar" Header="Guardar" />
         <Separator />
         <MenuItem x:Name="mnArchivoSalir" Header="Salir"
Click="mnArchivoSalir_Click"/>
      </MenuItem>
        ...
```

Menú contextual

Es un menú contextual que se asocia a un control padre.

Se muestra al pulsar el botón derecho del ratón sobre el control padre.

Etiqueta “ContextMenu” es el contenedor del menú.

- Los hijos igual que con “Menu”, etiqueta principal “MenuItem”.

El evento de apertura se define en el control padre con “ContextMenuOpening”.

- Permite habilitar o deshabilitar opciones en función del elemento seleccionado o de los permisos.

El menú contextual se ancla en el control padre con “EtiquetaPadre.ContextMenu”.

Menú contextual - Ejemplo

```
<Button Content="Botón con menú contextual"
ContextMenuOpening="Button_ContextMenuOpening">
        <Button.ContextMenu>
                <ContextMenu>
                        <MenuItem Header="Elemento menú 1" />
                        <MenuItem Header="Elemento menú 2" />
                        <Separator />
                        <MenuItem Header="Elemento menú 3" />
                </ContextMenu>
        </Button.ContextMenu>
</Button>
```

Barra de herramientas

Muestra las acciones de uso frecuente agrupadas por funcionalidad.

Puede contener la mayoría de los controles. Botones, Desplegables, etc…

Etiqueta “**ToolBarTray**”, bandeja sobre la que se colocan los contenedores de grupo de acciones “**ToolBar**”.

Los controles hijos pueden agregar la propiedad “ToolBar.OverflowMode” para indicar como se muestran los controles en función del tamaño de la ventana.

Requiere de un DockPanel, en este caso la barra de herramientas puede aparecer en cualquier lado.

Barra de herramientas - Ejemplo

```
<ToolBarTray DockPanel.Dock="Top">
      <ToolBar>
         <Button Command="New" Content="Nuevo" />
         <Button Command="Open" Content="Abrir" />
         <Button Command="Save" Content="Guardar" />
      </ToolBar>
      ...
</ToolBarTray>
```

Barra de estado

Pequeña barra que muestra el estado en tiempo real de la aplicación.

Requiere de un DockPanel, normalmente DockPanel.Dock=”Bottom”

Etiqueta StatusBar, contenedor, los hijos son del tipo StatusBarItem o separadores.

- StatusBar.ItemsPanel, si añadimos varios datos y deseamos formatearlos deberemos definir una plantilla.

StatusBarItem, contenedor para la información, texto, barras de progreso, iconos.

Barra de estado - Ejemplo

```
<StatusBar DockPanel.Dock="Bottom">
    <StatusBarItem>
            <TextBlock x:Name="tbEstado" Text="Estado"></TextBlock>
    </StatusBarItem>
</StatusBar>
```