Paneles

UT5. GENERACIÓN DE INTERFACES A TRAVES DE XML

Paneles

Siguen la filosofía de la web, permiten posicionar los controles de una forma relativa (salvo que queramos lo contrario).

Son necesarios para realizar algunas operaciones que en el caso WinForms se realizaban con controles especializados o mediante propiedades. Por ejemplo.

- Acoplar controles
- Menús
- Presentar un texto con un icono asociado
- Dividir la ventana (splitcontainer de winforms)
- Simular comportamiento MDI

Aunque el posicionamiento absoluto es tentador, piensa que al emplear posicionamiento relativo tus aplicaciones se adaptaran a distintas pantallas sin tener que tocar el código.

Panel – Canvas

Posicionamos los elementos libremente a través de coordenadas.

Las propiedades de posicionamiento se indican en los controles hijos.

- Canvas.Left, Canvas.Top, Canvas.Right, Canvas.Bottom, Panel.Zindex

Se emplea para dibujar dentro de la aplicación.

```
<Canvas>
<Ellipse Panel.ZIndex="2" Fill="Gainsboro" Height="216" Canvas.Left="47"
Canvas.Top="41" Width="216" />
</Canvas>
```

Panel - WrapPanel

Coloca los elementos uno tras otro y de izquierda a derecha, salta de línea al completar el espacio disponible.

Se puede cambiar la orientación. Por defecto horizontal.

```
<WrapPanel Orientation="Horizontal">

        <Button>Botón de pruebas 1</Button>
        ...

        <Button>Botón de pruebas 6</Button>

</WrapPanel>
```

Panel - StackPanel

Como el WrapPanel pero elemento añadido ocupa todo el espacio de la fila.

Se puede cambiar la orientación. Por defecto vertical.

Ajustando la alineación el control se consigue que el control no se estire aunque sigue ocupando toda la filas.

Se emplea para dar formato al texto de algunos controles, por ejemplo al mostrar un icono más un texto.

```
<StackPanel Orientation="Vertical">
        <Button HorizontalAlignment="Left">Botón de pruebas 1</Button>
        ...
        <Button HorizontalAlignment="Right">Botón de pruebas 6</Button>
</StackPanel>
```

Panel - DockPanel

Permite acoplar contrales en orden, el último elemento ocupa todo el espacio que quede disponible.

LastChildFill="false", el último elemento no ocupa todo el espacio disponible.

Las propiedades de posicionamiento se indican en los controles hijos.

- DockPanel, valores Left, Top, Right, Bottom.

Se emplea para implementar menús.

```
<DockPanel LastChildFill="False">

        <Button DockPanel.Dock="Top" Height="50">Arriba</Button>

        ...

</DockPanel>
```

Panel - Grid

Panel por defecto, permite posicionar elementos de forma libre ajustando la propiedad Margin. No confundir con Canvas.

Permite definir columnas (**ColumnDefinitions**), filas (**RowDefinitions**) o ambas.

Varias formas de indicar el tamaño.

- Número -> tamaño absoluto en pixeles.
- Proporcional (*) -> **valor por defecto 1***, * equivale a el tamaño de una columna teniendo en cuenta el total de columnas que asignemos.
- Automático (auto) -> espacio que necesite el control.

Panel – Grid II

Posicionamos los controles hijos en el grid con las propiedades.

- Grid.Column
- Grid.Row

La numeración de columnas y filas comienza por el “0”.

Si no se indica la columna o la fila se suponen los valores “0”.

Podemos combinar varias celdas con las propiedades.

- Grid.ColumnSpan
- Grid.RowSpan

Panel – Grid III

Ejemplo con 2 columnas con ancho por defecto.

```
<Grid>
        <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*" />
                <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Button>Botón 1</Button>
        <Button Grid.Column="1" VerticalAlignment="Center"
HorizontalAlignment="Right">Botón 2</Button>
</Grid>
```

Panel – GridSplitter

Es un control permite dividir un grid en dos partes con una barra.

La configuración habitual es de 3 columnas, el GridSplitter ocupa la columna central.

- Si cambiamos las columnas por filas dividimos horizontalmente.

Recomendación, definir tanto el ancho de la columna como el ancho del GridSplitter con el mismo valor absoluto.