Controles comunes

UT5. GENERACIÓN DE INTERFACES A TRAVES DE XML

Uso de controles

La propiedad Name define el nombre de variable asociada al control. Útil para:

- Referencias en el código XAML.
- Referencias en el código de respaldo.

Podemos tener 2 propiedades de datos:

- Content, cuando heredamos de ContentControl.
  - Esta propiedad puede presentarse “inline” o desarrollada con controles hijo, en este último caso no hace falta indicar la etiqueta en el maquetado.
- Text

El modelo de eventos es más reducido que en WinForms y en algunos casos puede que se definan contenedor en vez de en el control generador.

Es posible que en vez de un evento empleemos un comando.

- Un comando es una acción reutilizable, por ejemplo copiar, pegar al portapapeles.

Label

Permite mostrar textos sencillos, normalmente se emplea como descripción de un campo de datos.

Propiedades

- Content, establece el texto a mostrar

```
<Label Content="Texto de la etiqueta" />
```

TextBox

Permite al usuario editar textos

Propiedades

- **Text**, texto editable, no hereda de ContentControl
- **AcceptsReturn**, multilínea.
- TextWrapping, distribuye la línea en la siguiente línea al alcanzar el borde del control.
- SpellCheck.IsEnabled="True", corrección ortográfica.

```
<TextBox AcceptsReturn="True" TextWrapping="Wrap"
SpellCheck.IsEnabled="True" />
```

Button

Botón, permite disparar una acción.

Propiedades

- **Content**, texto a mostrar, se puede desarrollar para darle formato.
- IsDefault, al pulsar “intro” se dispara el botón.
- IsCancel, al pulsar “esc” se dispara el botón.

```
<Button x:Name="btnFormato" Margin="10,209,0,176">
        <StackPanel Orientation="Horizontal">
                <Image Source="procedure_16xMD.png" Height="16"/>
                <TextBlock Padding="10,5">Botón formateado</TextBlock>
        </StackPanel>
</Button>
```

Imagen

Permite mostrar imágenes.

Lo vamos emplear para incluir iconos en botones, menús, etiquetas, etc…

Propiedades

- **Source**, para indicar una fuente.
- Stretch, manera en la que se ajusta la imagen al espacio disponible.

En código debemos envolver la fuente con bitmap y una URI

```
Uri archivoUri = new Uri("ruta imagen o recurso");

this.iVisor.Source = new BitmapImage(archivoUri);
```

TextBlock

Permite mostrar textos con formato.

Normalmente se emplea para mostrar textos dentro de otros controles.

Propiedades

- Text, texto a mostrar. NORMALMENTE NO SE USA, se añaden etiquetas hijas.
- **Inlines**, sólo en código, elementos hijo con texto formateado.
- TextTrimming, forma en la que se recorta en el texto sobrante.
- TextWrapping, forma de ajustar el texto al espacio disponible.

TextBlock - Inlines

Colección de elemento hijos con texto formateado.

Etiquetas de formato.

- LineBreak, salto de línea.
- Bold, negrita.
- Italic, cursiva.
- Underline, subrayado.
- **Run**, texto con formato.
- Span, texto y elementos con formato.
- Hyperlink, hipervínculo.
- AnchoredBlock, bloque anclado. Similar a un “DIV” en HTML.
- InlineUIContainer, contenedor de controles, podríamos por ejemplo añadir un combo.

TextBlock - Ejemplo

XAML

```
<TextBlock x:Name="tbSegundo" TextTrimming="CharacterEllipsis">
        <Run>Hola</Run>
        <LineBreak />
        <Span Foreground="Red" TextDecorations="Underline"
FontStyle="Italic">Hola esto <Bold>va en negrita</Bold></Span>
</TextBlock>
```

Código

```
this.tbCodigo.Inlines.Clear();
Span span = new Span();
span.Inlines.Add(new Run("Esto es un span y no admite texto sin elemento en línea"));
this.tbCodigo.Inlines.Add(span);
```

CheckBox

Representa un control activable y desactivable.

OJO: 3 estados: Checked, Unchecked y null.

IsChecked, obtiene o establece si el control esta activo.

Eventos: **Click**, Checked, Unchecked. Los dos últimos no saltan con el valor null.

```
<CheckBox Name="chkOpcion" IsChecked="True"
Click="chkOpcion_Click">Habilitar opción</CheckBox>
```

RadioButton

Conjunto de controles en los que el usuario sólo puede seleccionar pero no deseleccionarlo.

Por defecto un único radiobutton activo por contenedor (ventana, grupo, etc…)=> podemos crear grupos de radiobuttons con una opción seleccionada por grupo.

Propiedades

- IsChecked, obtiene si el control está seleccionado.
- GroupName, nombre del grupo.

Eventos: **Click**, Checked, Unchecked.

RadioButton - Ejemplo

XAML

```
<StackPanel>
        <Label FontWeight="Bold">¿Asistirás?</Label>
        <RadioButton GroupName="grupo2" Click="RadioButton_Click" Tag="si">Sí</RadioButton>
        <RadioButton GroupName="grupo2" Click="RadioButton_Click" Tag="no">No</RadioButton>
        <RadioButton GroupName="grupo2" IsChecked="True" Click="RadioButton_Click"
Tag="quizas">Quizás</RadioButton>
</StackPanel>
```

Código

```
private void RadioButton_Click(object sender, RoutedEventArgs e){
        string valor = (string)((RadioButton)e.Source).Tag;
        MessageBox.Show("Ha pulsado: " + valor);
}
```

GroupBox

Permite agrupar controles dentro recuadro con borde al que se le puede dar un título.

Etiqueta “**GroupBox**”

- **Header**, texto a mostrar, se puede desarrollar para darle formato.

```
<GroupBox Header="Título del contenedor">
 ... Contenido, panel.
</GroupBox>
```

ComboBox

Permite seleccionar un valor entre los disponibles en una lista.

Cada elemento de la lista es un **ComboBoxItem**.

Propiedades

- **SelectedIndex**, índice del elemento seleccionado.
- SelectedItem, elemento seleccionado.
- **IsEditable**, permite al usuario escribir en el desplegable.
- IsTextSearchEnabled, permite escribir mostrando sugerencias entre los valores disponibles.
  - IsTextSearchCaseSensitive, permite diferenciar entre mayúsculas y minúsculas.
- **Items**, colección de elementos, por si tenemos que editarla por código.

Eventos

- SelectionChanged, cuando cambia el elemento seleccionado.

ComboBox - II

Propiedades ComboBoxItem

- **IsSelected**, indica si el elemento de la lista esta seleccionado.

NOTA: Si quieres rellenar el control mediante un enlace a datos debes indicar la plantilla en ComboBox.ItemTemplate

```
<ComboBox>
        <ComboBoxItem>Elemento #1</ComboBoxItem>
        <ComboBoxItem IsSelected="True">Elemento #2</ComboBoxItem>
        <ComboBoxItem>Elemento #3</ComboBoxItem>
</ComboBox>
```

DatePicker

Permite seleccionar una fecha sin desajustar la vista. El control incluye un pequeño calendario desplegable además de la caja de texto.

Propiedades

- DisplayDate, fecha por defecto, sino se muestra el día actual.
- **SelectedDate**, fecha seleccionada.
- **SelectedDateFormat**, formato de la fecha.
  - **Short**, fecha corta
  - **Long**, fecha larga
- Calendar.BlackoutDates, permite definir rangos de fechas no seleccionables, los datos van en etiquetas “CalendarDateRange”.

DatePicker - II

IMPORTANTE: los selectores de fecha de WPF ya no permiten seleccionar horas.

```
<DatePicker SelectedDate="2020-12-31" SelectedDateFormat="Long">
</DatePicker>
```

La opción para trabajar con fechas es emplear un TextBox con una máscara de entrada.

```
<TextBlock Text="{Binding TuPropiedadHora, StringFormat={}{0:HH:mm tt}}"
/>
```

INFORMACIÓN ÚTIL: en nuget existe está el paquete “Extended WPF Toolkit” que incluye un control “**DateTimePicker**” con una funcionalidad similar al control de WinForms. Cuidado con licencias.

Conceptos comunes – Orden de tabulación

Permite navegar por los distintos controles de la ventana a través de la tecla TAB.

Propiedades

- **TabIndex**, indica el orden de tabulación.
- **IsTabStop**, indica que no se debe parar en el control. Normalmente no queremos parar en los controles que no podemos editar.

```
<TextBox TabIndex="5" IsReadOnly="True" IsTabStop="False" />
```

Conceptos comunes - ToolTip

Consejos de herramienta o ToolTips muestran información relacionada cuando el usuario deja inmóvil el puntero del ratón sobre el control.

Propiedades

- **ToolTip**, texto de información o ayuda. El contenido se puede desarrollar <ETIQUETA.ToolTip>…
- ToolTipService.ShowDuration, duración del mensaje. ToolTipService incluye múltiples opciones de comportamiento.

```
<Button ToolTip="Crea un nuevo archivo"
ToolTipService.ShowDuration="5000" Content="Abrir" />
```