GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

ceedcv CENTRE ESPECÍFIC D'EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA

DESARROLLO DE INTERFACES
EJEMPLO DE CONTROL DE USUARIO PERSONALIZADO

Departamento de Informática
Francisco Lifante Rivera

Ejemplo de User control

Creamos un nuevo elemento de tipo UserControl, lo llamaré UserControlPathButton. Se creará un fichero .xaml y un .cs para nuestro UC.

En el código xaml dividiré el grid en dos columnas y dos dilas, crearé un label, un textbos y un button:

```
<UserControl x:Class="EjemploUserControl.UserControlPathButton"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:EjemploUserControl"
             mc:Ignorable="d"
             d:DesignHeight="200" d:DesignWidth="600">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="7*"/>
            <ColumnDefinition Width="3*"/>
        </Grid.ColumnDefinitions>

        <Label x:Name="LabelPath"  Grid.Row="0" Grid.Column="0"></Label>
        <TextBox Grid.Row="1" Grid.Column="0"/>
        <Button Grid.Row="0" Grid.RowSpan="2"  Grid.Column="1"></Button>
    </Grid>
</UserControl>
```

Ilustración 1. Código Xaml de nuestro User Control

En el code behind, el fichero .cs haré lo siguiente:

1) Registraré una nueva propiedad de dependencia:

```
// Creamos una propiedad de dependencia para nuestro labeltext
public static DependencyProperty LabelTextProperty =
    DependencyProperty.Register(nameof(LabelText), typeof(string), typeof(UserControlPathButton),
        new PropertyMetadata(LabelTextPropertyChanged));
```

Ilustración 2. Creación de Propiedad de Dependencia (DP en inglés)

Las propiedades de dependencia que creemos, por convención, se suele poner al final la coletilla property, de forma que la llamaré LabelTextProperty, ya que estoy haciendo una DP (dependency property) para un LabelText. Utilizamos el método Register de la clase DependencyProperty de .net y le introducimos los siguientes parámetros:

- *Nameof(LabelText)* es para registrar el nombre de nuestra propiedad y utilizamos nameof para coger el nombre de la propiedad y convertirlo en un string.
- *Typeof(string)* es para asignar el tipo de nuestra propiedad.
- *Typeof(UserControlPathButton)* es el propietario de esta propiedad. Será nuestra clase donde estamos implementando esta DP.
- *New PropertyMetaData(LabelTextPropertyChanged)* esto sirve para que cuando nuestra propiedad se modifique, utilice el método registrado aquí. Como podéis ver en

el constructor utiliza un PropertyChangedCallback, que es un delegado que ejecuta código específico en respuesta al cambio de valor de la propiedad.

Al definir nuestra DP, tenemos que crear la propiedad asociada a la primera:

```
public string LabelText
{
    get
    {
        return (string)GetValue(LabelTextProperty);
    }
    set
    {
        SetValue(LabelTextProperty, value);
    }
}
```

Ilustración 3. Creación de la propiedad LabelText

Con el código de la imagen anterior hemos creado la propiedad para utilizar junto a la DP.

Ahora nos faltaría crear la función que va a responder al PropertyChangedCallback, es decir, el evento que ocurrirá una vez modifiquemos nuestro control. Entonces creamos el siguiente método:

```
private static void LabelTextPropertyChanged(DependencyObject d, DependencyPropertyChangedEventArgs e)
{
    var userControlPath = (UserControlPathButton)d;
    userControlPath.LabelPath.Content = e.NewValue.ToString();

}
```

Ilustración 4. Método que reacciona al cambio en una propiedad

Este método recibe dos parámetros, un objeto de dependencia que va a ser en este caso UserControlPathButton y los argumentos del evento que cambia.

Utilizamos una variable para capturar el objeto de dependencia y ya podremos acceder a los valores del control LabelText. Asignamos el contenido del control que ha cambiado y le asignamos el nuevo valor obtenido.

Ahora para utilizar este UserControl (UC) que hemos creado, habría que definifrlo en el espacio de nombres de nuestra ventana, en caso de haberlo creado en algún directorio. Pero en este caso, lo hemos creado en la raíz del proyecto y ya está incluido.

```
xmlns:local="clr-namespace:EjemploUserControl"
```

Ahora en el fichero .xaml de nuestra ventana donde queremos agregar el UserControl debemos implementarlo de la siguiente forma:

```
<local:UserControlPathButton LabelText="Introduzca ubicación de fichero"/>
```

El resultado, tras modificar un poco el .xaml del control de usuario sería:

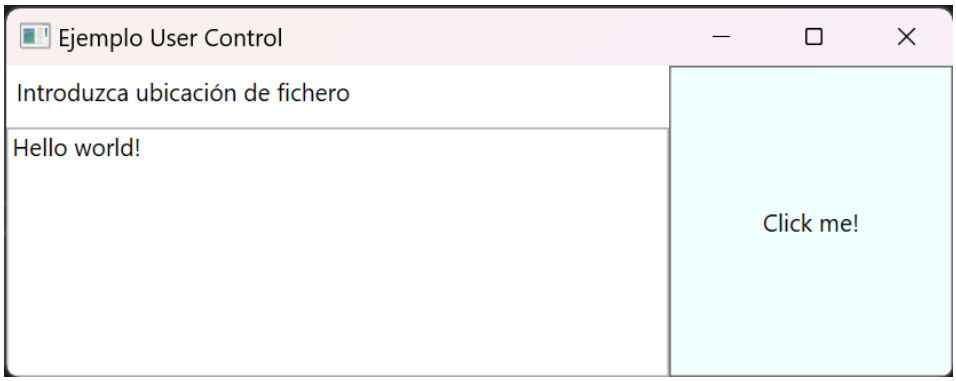

Ilustración 5. Resultado de aplicar el UC a nuestra ventana principal

Esto es un simple ejemplo de cómo crear controles de usuario personalizados para que puedan ser reutilizables en diferentes proyectos. Evidentemente se pueden realizar todo tipo de controles de usuario, mucho más avanzados y complejos que este.