

Implementació de llibreria de controls

BIEL AYMERICH

FRANCESC XAVIER S JOVÉ

POCHOCONSTRUCCIONS SA | [Dirección de la compañía]

INDEX

Objectiu de la PAC	2
Dynamic Lynk Library	2
Desenvolupament de la llibreria	2
Primers Passos	2
Exemple:	3
Implementació de la llibreria	5
Configuració	5
Implementació	6
Resultat	7
Problemes i Conclusió:	8
Problemes	8
Conclusió	8

Objectiu de la PAC

Desenvolupament de la llibreria

Primers Passos

Per començar la pràctica hem optat per seguir pas per pas la pròpia guia de l'exercici, creant la llibreria i els quatre Users Controls corresponents. Un cop creats, hem consultat amb el chat GPT per que ens ajudés a estructurar visualment el nostre TextBox, tot i que després, hem acabat implementant certs canvis, com vindrien a ser uns colors més clars i minimalistes en línia amb l'aplicació i un border amb un petit Radius gairebé imperceptible que ens agradava com quedava.

```
<StackPanel>
    <Border x:Name="border" CornerRadius="1"
Padding="1" BorderBrush="#D5D8DC" BorderThickness="1">
        <TextBox x:Name="textBox"
            BorderThickness="0"
            Text="{Binding DNI,
RelativeSource={RelativeSource
AncestorType=UserControl},
UpdateSourceTrigger=PropertyChanged}"
            VerticalContentAlignment="Center"/>
    </Border>
</StackPanel>
```

Un cop decidit el disseny, el següent pas ha estat anar cap a la configuració del **xaml.cs**, en el qual hem optat per seguir la mateixa estructura que en el exemple del email. Mentre fèiem el DNI, ens hem donat compte de que poques coses canviarien entre uns i altres així que hem posat uns noms de variables genèriques i hem decidit copiar i enganxar el mateix en cadascun, canviant tant sols les variables importants i els requisits de validació.

Exemple:

```
public partial class DNITextBox : UserControl
{
    // Definició de les DepecncyProperty DNI i Validació
    public static readonly DependencyProperty DNIProperty =
        DependencyProperty.Register(
            "DNI",
            typeof(string),
            typeof(DNITextBox),
            new FrameworkPropertyMetadata(
                string.Empty,
                FrameworkPropertyMetadataOptions.BindsTwoWayByDefault,
                OnValueChanged));

    public string DNI
    {
        get => (string)GetValue(DNIProperty);
        set => SetValue(DNIProperty, value);
    }

    public DNITextBox()
    {
        InitializeComponent();
        textBox.TextChanged += (s, e) => DNI =
textBox.Text;
    }

    // Callback quan la DependencyProperty DNI canvia
    private static void OnValueChanged(DependencyObject d,
DependencyPropertyChangedEventArgs e)
    {
        var control = (DNITextBox)d;
        var newValue = (string)e.NewValue;
        control.Validate(newValue);
    }

    // Regex de Validació
    private static bool IsValidBool(string newValue)
    {
        var dniRegex = new Regex(@"^\d{8}[A-Za-z]$");
        return dniRegex.IsMatch(newValue);
    }
}
```

```

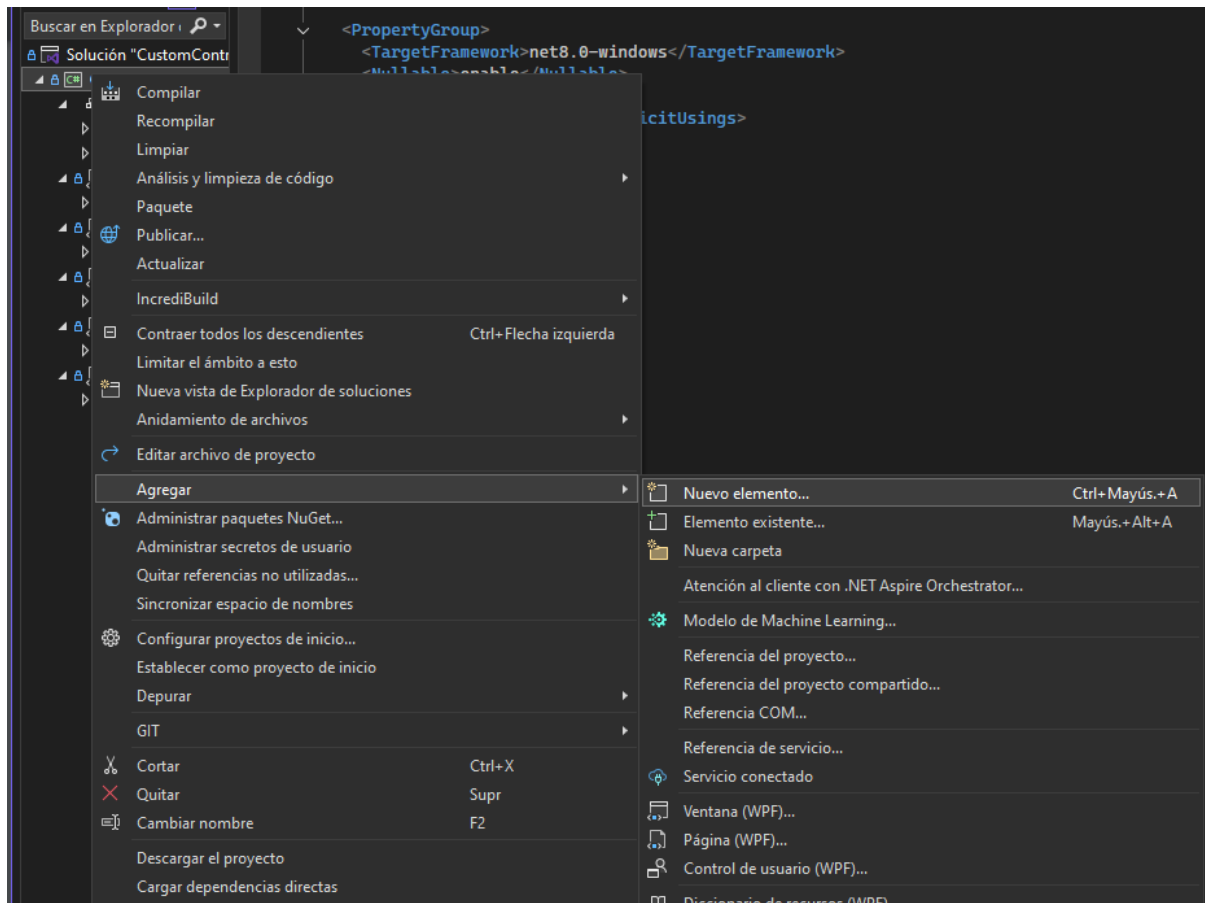
// Mètode per validar el format del DNI
private void Validate(string newValue)
{
    if (string.IsNullOrEmpty(newValue) ||
IsValidBool(newValue))
    {
        border.BorderBrush = new
SolidColorBrush((Color)ColorConverter.ConvertFromString("#D
5D8DC"));
    }
    else
    {
        border.BorderBrush = new
SolidColorBrush(Colors.DarkRed);
    }
}
}

```




Implementació de la llibreria

Configuració

Per a poder fer la integració, hem anat a Agregar Referencia del Proyecto desde la nostre aplicació :



I després hem buscat el path del document dll dintre de la carpeta Bin del projecte de la llibreria:

 CustomControlsLib.deps.json	13/11/2024 14:19	Archivo JSON	1 KB
 CustomControlsLib.dll	13/11/2024 21:02	Extensión de la ap...	19 KB
 CustomControlsLib.pdb	13/11/2024 21:02	VisualStudio.pdb....	17 KB

Fet això i clicat acceptar, ja tindríem la llibreria en l'aplicació.

Implementació

Un cop amb la llibreria ja en les nostres dependències toca implementar, és a dir, importar-la en les Views que la implementaran i començar a intentar de fer-la anar...

En resum, dir que això ha estat ón més problemes hem tingut degut a alguns fallos en el maneig de nulls, errors que hem arrossegat del copiar i enganxar, etc...

Però al final ha quedat bé i sense errors:

```
UserControl x:Class="WPF_MVVM_SPA_Template.Views.EditarUsuari"
xmlns:local="clr-namespace:CustomControlsLib;assembly=CustomControlsLib"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d"
d:DesignHeight="300" d:DesignWidth="500">

<Grid Margin="0">
    <Border BorderBrush="#FFCCCC" BorderThickness="1" CornerRadius="6" Padding="10">
        <StackPanel>
            <!-- Campo para el Nombre -->
            <TextBlock Text="Name:" FontSize="12" FontWeight="Bold" Margin="0,2" />
            <local:MinilengthTextBox MinilengthText="{Binding EditableClient.Name, UpdateSourceTrigger=PropertyChanged}"
                FontSize="10" Padding="2" Margin="0,0,0,5"/>

            <!-- Campo para el DNI/NIF -->
            <TextBlock Text="DNI/NIF:" FontSize="12" FontWeight="Bold" Margin="0,2"/>
            <local:DNITextBox DNI="{Binding EditableClient.DniNif, UpdateSourceTrigger=PropertyChanged}"
                FontSize="10" Padding="2" Margin="0,0,0,5"/>

            <!-- Campo para Profesional (CheckBox para el booleano) -->
            <TextBlock Text="Profesional:" FontSize="12" FontWeight="Bold" Margin="0,2"/>
            <CheckBox IsChecked="{Binding EditableClient.Profesional, UpdateSourceTrigger=PropertyChanged}"
                FontSize="10" Margin="0,0,0,5" />

            <!-- Campo para el Descuento -->
            <TextBlock Text="Discount (%):" FontSize="12" FontWeight="Bold" Margin="0,2" />
            <TextBox Text="{Binding EditableClient.Discount, UpdateSourceTrigger=PropertyChanged}"
                FontSize="10" Padding="2" Margin="0,0,0,5" />

            <!-- Campo para la Fecha de Registro (modificable) -->
            <TextBlock Text="Registration Date:" FontSize="12" FontWeight="Bold" Margin="0,2" />
            <DatePicker SelectedDate="{Binding EditableClient.RegistrationDate, UpdateSourceTrigger=PropertyChanged}"
                FontSize="10" Margin="0,0,0,5" />

            <!-- Campo para las Ventas Anuales -->
            <TextBlock Text="Total Annual Sales:" FontSize="12" FontWeight="Bold" Margin="0,2" />
            <local:TextBoxSimple Text="{Binding EditableClient.TotalAnnualSells, UpdateSourceTrigger=PropertyChanged}"
                FontSize="10" Padding="2" Margin="0,0,0,5" />
        </StackPanel>
    </Border>
</Grid>
```

Resultat

APP POCHA DE GESTIÓ DE CLIENTS

Start

Clients

Name: Jordi

DNI/NIF: 8765432

Professional: ☐

Discount (%): 5

Registration Date: 13/11/2024

Total Annual Sales: 1200

Tel: 65432143

Mail: jordi@uvic.cat

Save Cancel

APP POCHA DE GESTIÓ DE CLIENTS

Start

Clients

Name: Jordi

DNI/NIF: 8765432

Professional: ☐

Discount (%): 5

Registration Date: 13/11/2024

Total Annual Sales: 1200

Tel: 654321431

Mail: jordi@uvic

Save Cancel

Problemes i Conclusió:

Problemes

- No hem set capaços de tornar a fer un nou onChangeText en la app.
- Petits problemes en el codi que no donaven error fins que fallava l'aplicació.
- En un moment donat sens ha ocorregut de tocar el Date Picker, però es feia molt complicat.
- Problemes en el ordre de un OR feia que no es posés mai vermell.
- Etc....

Conclusió

En definitiva, creiem que tot i ser una llibreria molt senzilla ón només hem implementat petites i lleugeres correccions estètiques i un sistema de validació decent. Ens ha servit molt per entendre'n les possibilitats, principalment a nivell de estructura des d'ón amb una sola dependència, pots tenir implementada una complexitat de disseny molt més alta que la que porten els items de sèrie de forma ordenada i sense repetició de codi...