ListBox

Algo así es lo pretendido.

Comienza con esto:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="200" ></RowDefinition>

</Grid.RowDefinitions>

</Grid>

Establece 2 filas. La primera va a adaptar su altura a los 200px de la segunda. En la primera van las 4 lineas, en la segunda va un botonazo.

<ListBox Name="listaPoblaciones" HorizontalContentAlignment= "Stretch" Grid.Row="0">

Así mete la listbox en la primera fila. En realidad, en este caso quedó implícita, no puso nada. En el caso del botón si que puso Grid.Row=”11>

HorizontalContentAlignment= "Stretch" Eso implica que se estire hasta ocupar todo el ancho.

ItemTemplate Con esa etiqueta se pueden empezar a meter elementos

DataTemplate Plantilla con datos. Ahí se pone el tipo de grid.

Refiere que es necesario hacer un binding para especificar los datos que van a entrar en cada una de las 4 columnas que hizo. Eso, con C#

ItemsSource lo usa para indicarle de donde vienen los datos que la lista va a tomar.

public MainWindow()

{

InitializeComponent();

List<Poblaciones> ListaPob = new List<Poblaciones>(); //Acá crea una lista

//Como se meten objetos (cada población) en la lista

ListaPob.Add(new Poblaciones() { Poblacion1 = "Madrid", Temperatura1 = 15, Poblacion2 = "Barcelona", Temperatura2 = 17 });

ListaPob.Add(new Poblaciones() { Poblacion1 = "Lanús", Temperatura1 = 25, Poblacion2 = "Banfield", Temperatura2 = 2 });

ListaPob.Add(new Poblaciones() { Poblacion1 = "Haedo", Temperatura1 = 18, Poblacion2 = "Morón", Temperatura2 = 19 });

ListaPob.Add(new Poblaciones() { Poblacion1 = "Bariloche", Temperatura1 = 9, Poblacion2 = "Mendoza", Temperatura2 = 12 });

listaPoblaciones.ItemsSource = ListaPob;

//Ahí mismo instancia, o crea, los objetos que va a meter en la lista. Lo que después irá al listbox.

}

}

public class Poblaciones

{

public string Poblacion1 { get; set; }

public int Temperatura1 { get; set; }

public string Poblacion2 { get; set; }

public int Temperatura2 { get; set; }

}

Grid Margin="2">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock Grid.Column="0" Text="{Binding Poblacion1}"></TextBlock>

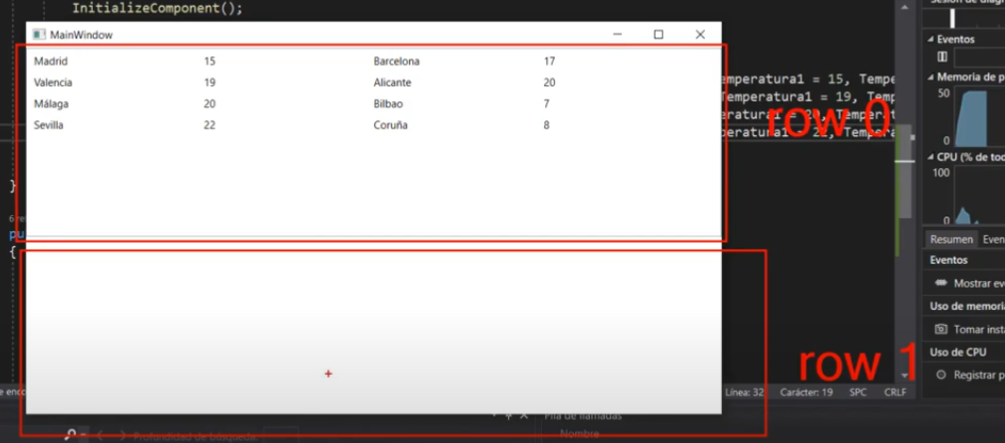
<TextBlock Grid.Column="1" Text="{Binding Poblacion1}"></TextBlock>

<TextBlock Grid.Column="2" Text="{Binding Poblacion1}"></TextBlock>

<TextBlock Grid.Column="3" Text="{Binding Poblacion1}"></TextBlock>

</Grid>

Luego, esto se ve así. 0, 1, 2, 3 refiere a las columnas de arriba, por mas que a esas no se les haya puesto número.



Una vez mas, para hacer que un botón, al tocarlo pase algo, se escribe Click, y va a aparecer un cartelito ”Nuevo controlador de evento”, si se toca Tab, se va a autocompletar Click=”Button\_Click” y se crea la clase en C#.

Así se hace eso:

private void Button\_Click(object sender, RoutedEventArgs e)

{

MessageBox.Show((listaPoblaciones.SelectedItem as Poblaciones).Poblacion1 + " " +

(listaPoblaciones.SelectedItem as Poblaciones).Temperatura1 + " °C " +

(listaPoblaciones.SelectedItem as Poblaciones).Poblacion2 + " " +

(listaPoblaciones.SelectedItem as Poblaciones).Temperatura2 + " °C ");

}

Para que cuando se toque el botón, no tire el error:::

private void Button\_Click(object sender, RoutedEventArgs e)

{

if (listaPoblaciones.SelectedItem != null)

{

MessageBox.Show((listaPoblaciones.SelectedItem as Poblaciones).Poblacion1 + " " +

(listaPoblaciones.SelectedItem as Poblaciones).Temperatura1 + " °C " +

(listaPoblaciones.SelectedItem as Poblaciones).Poblacion2 + " " +

(listaPoblaciones.SelectedItem as Poblaciones).Temperatura2 + " °C ");

}

else

{

MessageBox.Show("Debes seleccionar un elemento");

}

}

Esa es una forma. Lo del else es optativo, recomendable.

Agregar progress bar::: (que muestre la diferencia de temperaturas entre las 2 poblaciones de una fila).

Agrega esto en la clase Poblaciones:

public int Diferencia { get; set; }

También agrega el 5o dato. Haciendo el cálculo de la diferencia:

ListaPob.Add(new Poblaciones() { Poblacion1 = "Madrid", Temperatura1 = 15, Poblacion2 = "Barcelona", Temperatura2 = 17, Diferencia = 2 });

ListaPob.Add(new Poblaciones() { Poblacion1 = "Lanús", Temperatura1 = 25, Poblacion2 = "Banfield", Temperatura2 = 2, Diferencia=23 });

ListaPob.Add(new Poblaciones() { Poblacion1 = "Haedo", Temperatura1 = 18, Poblacion2 = "Morón", Temperatura2 = 19, Diferencia=1 });

ListaPob.Add(new Poblaciones() { Poblacion1 = "Bariloche", Temperatura1 = 9, Poblacion2 = "Mendoza", Temperatura2 = 12, Diferencia=3 });

También hay que tocar el XAML, agregando una columna mas:

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" ></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

<ColumnDefinition Width="\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

Y también agrega el progressbar Value= “{Binding Diferencia}”. El Diferencia hay que agregarlo uno, no te ayuda para hacerlo. Es la variable recién creada.

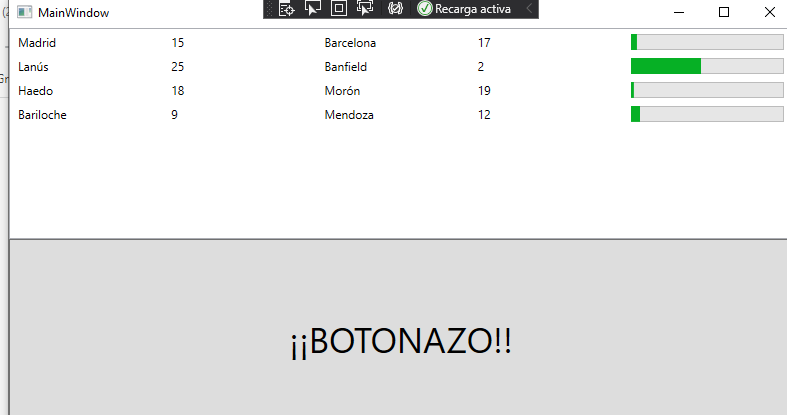
<TextBlock Grid.Column="0" Text="{Binding Poblacion1}"></TextBlock>

<TextBlock Grid.Column="1" Text="{Binding Temperatura1}"></TextBlock>

<TextBlock Grid.Column="2" Text="{Binding Poblacion2}"></TextBlock>

<TextBlock Grid.Column="3" Text="{Binding Temperatura2}"></TextBlock>

<ProgressBar Grid.Column="4" Minimum="0" Maximum="50" Value="{Binding Diferencia}"></ProgressBar>



Por ultimo, quiere hacer que tocando un elemento de la lista, también despliegue el mbox

El textbox no tiene el evento click, hay que usar previewmousedown:

<TextBlock Grid.Column="0" PreviewMouseDown="TextBlock\_PreviewMouseDown" Text="{Binding Poblacion1}" ></TextBlock>

<TextBlock Grid.Column="1" PreviewMouseDown="TextBlock\_PreviewMouseDown" Text="{Binding Temperatura1}"></TextBlock>

<TextBlock Grid.Column="2" PreviewMouseDown="TextBlock\_PreviewMouseDown" Text="{Binding Poblacion2}"></TextBlock>

<TextBlock Grid.Column="3" PreviewMouseDown="TextBlock\_PreviewMouseDown" Text="{Binding Temperatura2}"></TextBlock>

<ProgressBar Grid.Column="4" Minimum="0" Maximum="50" Value="{Binding Diferencia}"></ProgressBar>

Así se hace. Al progressbar no se le asigna evento. Se crea un solo método en C#, que se aplica a los 4 elementos, que a su vez es idéntico al buttonClick

private void TextBlock\_PreviewMouseDown(object sender, MouseButtonEventArgs e)

{

if (listaPoblaciones.SelectedItem != null)

{

MessageBox.Show((listaPoblaciones.SelectedItem as Poblaciones).Poblacion1 + " " +

(listaPoblaciones.SelectedItem as Poblaciones).Temperatura1 + " °C " +

(listaPoblaciones.SelectedItem as Poblaciones).Poblacion2 + " " +

(listaPoblaciones.SelectedItem as Poblaciones).Temperatura2 + " °C ");

}

else

{

MessageBox.Show("Debes seleccionar un elemento");

}

}

[siamese 12](https://www.youtube.com/channel/UCAUwHGqsxasy9MMMLoAOYfw)

[hace 2 años](https://www.youtube.com/watch?v=LiXli1RJt1M&lc=Ugxacy-mj7aGoqWT9894AaABAg)

Hola Juan, para el ejercicio de seleccionar el elemento yo utilice el evento SelectionChanged de la listBox y el programa se simplifica bastante. Muchas gracias por este pedazo de curso, eres un gran profesor.

17

Responder

2 respuestas

[Eliú Castillo](https://www.youtube.com/channel/UCP4_WmgHPI2zIiYGXWDuTVw)

[Eliú Castillo](https://www.youtube.com/channel/UCP4_WmgHPI2zIiYGXWDuTVw)

[hace 2 años](https://www.youtube.com/watch?v=LiXli1RJt1M&lc=Ugxacy-mj7aGoqWT9894AaABAg.976K7hnbA9y98SP_KZD1sY)

Hice casi lo mismo pero con el evento DoubleClick en el list box, de esta manera te deja seleccionar un elemento, saludos

3

Responder

[TonicoBonico](https://www.youtube.com/channel/UCOZ48Jnb5MA_JFxYJy1lZUA)

[TonicoBonico](https://www.youtube.com/channel/UCOZ48Jnb5MA_JFxYJy1lZUA)

[hace 6 meses](https://www.youtube.com/watch?v=LiXli1RJt1M&lc=Ugxacy-mj7aGoqWT9894AaABAg.976K7hnbA9y9dDr_ojNmOo)

[@Eliú Castillo](https://www.youtube.com/channel/UCP4_WmgHPI2zIiYGXWDuTVw) Eso estaba pensando al ver el vídeo, mucho mejor que responda al evento el ListBox que cada uno de los textblocks. Y si es con doble click, más chachi aún.