

Guía de Ejercicios Prácticos: WPF y XAML

Ejercicio 1: El Apilador (StackPanel)

Objetivo: Comprender cómo se alinean los elementos automáticamente.

- **Tarea:** Crea una ventana con 3 botones de diferentes colores (Rojo, Verde, Azul) apilados verticalmente.

- **Código XAML sugerido:**

XML

```
<StackPanel Margin="20">
    <Button Content="Botón 1" Background="Red" Margin="5"/>
    <Button Content="Botón 2" Background="Green" Margin="5"/>
    <Button Content="Botón 3" Background="Blue" Foreground="White"
Margin="5"/>
</StackPanel>
```

Ejercicio 2: El Tablero (Grid Básico)

Objetivo: Aprender a dividir el espacio en filas y columnas.

- **Tarea:** Divide la ventana en 2 filas. La de arriba debe medir 100 píxeles y la de abajo debe ocupar el resto del espacio (*). Coloca un texto en cada una.

- **Código XAML sugerido:**

XML

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="100"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>
    <TextBlock Text="Cabecera Fija" Grid.Row="0"
Background="LightGray"/>
    <TextBlock Text="Cuerpo Flexible" Grid.Row="1" Background="White"/>
</Grid>
```

Ejercicio 3: Interacción Básica (Code-Behind)

Objetivo: Conectar el diseño con la lógica de C#.

- **Tarea:** Crea un botón que, al ser pulsado, cambie su propio texto a "¡Me has clicado!".

- **XAML:** `<Button x:Name="btnMagico" Content="Haz clic" Click="btnMagico_Click"/>`

- **C# (.xaml.cs):**

C#

```
private void btnMagico_Click(object sender, RoutedEventArgs e) {
    btnMagico.Content = "¡Me has clicado!";
}
```

Ejercicio 4: Entrada de Datos (TextBox)

Objetivo: Capturar información del usuario.

- **Tarea:** Diseña un formulario con un `TextBox` y un `Button`. Al pulsar el botón, muestra un `MessageBox` con el texto que el usuario escribió.

- **C# (.xaml.cs):**

```
C#
private void btnEnviar_Click(object sender, RoutedEventArgs e) {
    string mensaje = txtEntrada.Text;
    MessageBox.Show("Escribiste: " + mensaje);
}
```

Ejercicio 5: Estilos Básicos (Resources)

Objetivo: Reutilizar diseños para no repetir código.

- **Tarea:** Crea un estilo dentro de `<Window.Resources>` que ponga el texto en negrita y color azul, y aplícalo a dos etiquetas diferentes.

- **Código XAML sugerido:**

XML

```
<Window.Resources>
    <Style x:Key="MiEstiloPersonal" TargetType="TextBlock">
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Foreground" Value="Blue"/>
        <Setter Property="FontSize" Value="20"/>
    </Style>
</Window.Resources>

<StackPanel>
    <TextBlock Text="Texto 1" Style="{StaticResource
MiEstiloPersonal}"/>
    <TextBlock Text="Texto 2" Style="{StaticResource
MiEstiloPersonal}"/>
</StackPanel>
```