

Ejercicio 1 y 2

Diseño de Interfaces : Diego Santos Bailon
Leobardo Salvador Ríos Suárez
10/10/2024
2° de DAM I

Contenido

Ejercicio 1.....	1
Maquetación.....	1
Webgrafía.....	2
Cuestión.....	3
Ejercicio 2.....	4
Maquetación.....	4
Webgrafía.....	7
Cuestión.....	7

EJERCICIO 1.

Debes tener en cuenta los siguientes puntos:

- Cambia el nombre de la ventana y archivo o proyecto a "Ejercicio 1"
- La ventana principal es de 400x400px
- El área de color azul es de tipo Canvas.
- Las áreas rosa, verde, roja y azul son de 100x100 y tipo Border.
Además, cada una está anclada a sus dos lados más cercanos de la ventana principal.
- Otras medidas que tendrás que usar (para márgenes, padding...) son 20px y 50px.

Maquetación

Titulo y tamaño de la ventana principal:

```
Title="Ejercicio 1" Height="400" Width="400">
```

PRÁCTICA 0

Borders y botones:

- Se utilizó el contenedor **Border** para los 4 elementos principales.
- Cada uno se distribuyó mediante la propiedad **Canvas.(left, top, right y bottom)** correspondientemente.
- Todos tiene una medida de **Height 100 Width 100**.
- Para que el borde se note hay que agregar dos propiedades, **BorderBrush** que da el color del borde y **BorderThickness** que determina el grosor del border.
- Para los colores tome una captura de pantalla y con Paint y su inspector de color detecte el color utilizado.
- El Boton rosa claro tiene un margin de 20 en todos sus lados, el verde de 20 solo en left y right, un margin 20 en top y bottom para el rosa y por último el azul un margin de 20 para todos sus lados. Todos los botones les puse un **BorderBrush** azul.
- Como propiedad de la etiqueta Button utilice **Content** para darle texto al boton.
- Note que contenedores **Border** y **Button** tenían una sombra, por lo que investigue y encontré la etiqueta **Border.Effect** y **Button.Effect**. que se coloca dentro de la etiqueta **Border**. **.Effect** tiene una propiedad llamada **DropShadowEffect** que permite agregar la sombra al contenedor. Como último agregué el color **LightGray** para que diera el efecto mas suave y el los botones utilice una propiedad llamada **ShadowDepth** especifica la distancia entre el objeto al que se aplica el efecto de sombra y la sombra proyectada. Esta propiedad determina qué tan lejos está la sombra del objeto en relación a su posición, lo que afecta la percepción de profundidad.

```
<Grid>
  <Canvas Background="#FFC307E8">
    <Border Width="100" Height="100" Canvas.Left="50" Canvas.Top="50" BorderBrush="#FF1570E6" BorderThickness="1" Background="#FFEDC5C5">
      <Border.Effect>
        <DropShadowEffect Color="LightGray"></DropShadowEffect>
      </Border.Effect>
      <Button Margin="20" BorderBrush="#FF1570E6" Content="Button 1">
        <Button.Effect>
          <DropShadowEffect Color="LightGray" ShadowDepth="2"></DropShadowEffect>
        </Button.Effect>
      </Button>
    </Border>
  </Canvas>
</Grid>
```

Webgrafía

- <https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.border?view=windowsdesktop-8.0>

PRÁCTICA 0

- <https://es.stackoverflow.com/questions/99022/como-lograr-efecto-de-sombra-en-wpf>
- <https://chatgpt.com>

Cuestión

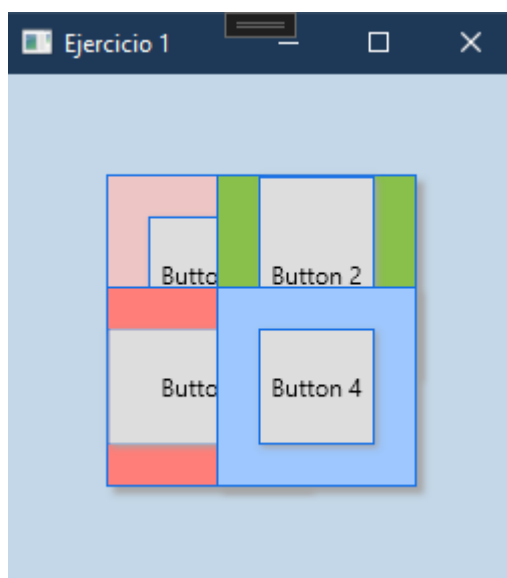
Si agrandamos la ventana, los bordes deberían seguir siempre a la misma distancia de la “paredes” de la ventana. Y si la hacemos más pequeña, unos se pondrán encima de otro.

1. ¿Cuál se pone encima de cuál y por qué?

Si se agranda la ventana los elementos no se separan del posicionamiento que se les dio con respecto a la ventana.



Si la hacemos mas pequeña los elementos se superponen uno encima de otro siendo el último elemento agregado que aparece primero.



PRÁCTICA 0

EJERCICIO 2

Ten en cuenta los siguientes puntos:

- Cambia el nombre de la ventana y archivo o proyecto a "Ejercicio 2"
- La ventana principal es de 400px x 500px.
- El área de color azul es de tipo DockPanel.
- Los Borders se deben introducir en el orden numérico que aparece.
- Las únicas medidas fijas que tienen son de 100. El resto son automáticas.
- Los márgenes miden 3.

Maquetación

Titulo y tamaño de la ventana principal:

```
Title="Ejercicio 2" Height="400" Width="500">
```

DockPanel:

- Coloque el **DockPanel** con la propiedad **LastChildFill** en true para que el último elemento ocupe todo el espacio disponible y se situé donde toca.

```
<DockPanel Background="■" #FFBED6E5" LastChildFill="True">
```

- El color lo saque apoyándome de **Paint**.

Borders:

- Para poder escribir en los dentro de los borders utilice un **TextBlock** para texto simple estático.
- **Border Top 1** tiene las siguientes propiedades
 - **DockPanel.Dock="Top"**: anclándolo arriba del DockPanel.
 - **Height** de 100.
 - **Margin** de 3.
 - **Background** que saque de una extracción con **Paint**.
 - **BorderBrush** azul.
 - **BorderThickness** de 1.
 - El **Text Block** que contiene el Border Top 1 agregue la propiedad **Foreground** en White y un **Margin** de 10 solo en Top y Left, el alineamiento Horizontal por defecto es Left.
 - Contiene un **DropShadowEffect** con la propiedad **Color** en LightGray Dentro para hacer un efecto de sombra al contenedor.

PRÁCTICA 0

```
<Border Margin="3" DockPanel.Dock="Top" Height="100" BorderBrush="#FF1570E6" BorderThickness="1" Background="#FFCCDCAB">
  <TextBlock Foreground="White" Margin="10,10,0,0" >Border Top 1</TextBlock>
  <Border.Effect>
    <DropShadowEffect Color="LightGray"></DropShadowEffect>
  </Border.Effect>
</Border>
```

- **Border Right 2** tienen las siguientes propiedades.
 - **DockPanel.Dock="Right"** para anclarlo a la derecha del Dock panel
 - **Width** de 100,
 - **BorderBrush** que saque de una extracción con **Paint**
 - **BorderThickness** de 1
 - El **Text Block** que contiene el Border Right 2 agregue la propiedad **Foreground** en White y un **HorizontalAlignment** Center, por defecto el **VerticalAlignment** está en Top.
 - Contiene un **DropShadowEffect** con la propiedad **Color** en LightGray Dentro para hacer un efecto de sombra al contenedor.

```
<Border Margin="3" DockPanel.Dock="Right" Width="100" BorderBrush="#FF1570E6" BorderThickness="1" Background="#FFC1B2D3">
  <TextBlock Foreground="White" Margin="0,10,0,0" HorizontalAlignment="Center" >Border Right 2</TextBlock>
  <Border.Effect>
    <DropShadowEffect Color="LightGray"></DropShadowEffect>
  </Border.Effect>
</Border>
```

- **Border Button 3** tiene las siguientes propiedades
 - **DockPanel.Dock="Bottom"**: anclándolo abajo del DockPanel.
 - **Height** de 100.
 - **Margin** de 3.
 - **Background** que saque de una extracción con **Paint**.
 - **BorderBrush** azul.
 - **BorderThickness** de 1.
 - El **Text Block** que contiene el Border Button 3 agregue la propiedad **Foreground** en White, un **HorizontalAlignment** Center y **VerticalAlignment** Center para centrar el **TextBlock** justo a mitad del contenedor.
 - Contiene un **DropShadowEffect** con la propiedad **Color** en LightGray Dentro para hacer un efecto de sombra al contenedor.

```
<Border Margin="3" DockPanel.Dock="Bottom" Height="100" BorderBrush="#FF1570E6" BorderThickness="1" Background="#FFFBCAA1">
  <TextBlock Foreground="White" VerticalAlignment="Center" HorizontalAlignment="Center">Border left 3</TextBlock>
  <Border.Effect>
    <DropShadowEffect Color="LightGray"></DropShadowEffect>
  </Border.Effect>
</Border>
```

PRÁCTICA 0

- **Border Left 4** tiene las siguientes propiedades
 - **DockPanel.Dock="Left"**: anclándolo a la izquierda del DockPanel.
 - **Width** de 100.
 - **Margin** de 3.
 - **Background** que saque de una extracción con **Paint**.
 - **BorderBrush** azul.
 - **BorderThickness** de 1.
 - El **Text Block** que contiene el Border Button 3 agregue la propiedad **Foreground** en White, un **HorizontalAlignment** Center y **VerticalAlignment** Center para centrar el **TextBlock** justo a mitad del contenedor.
 - Contiene un **DropShadowEffect** con la propiedad **Color** en LightGray Dentro para hacer un efecto de sombra al contenedor.

```
<Border Margin="3" Width="100" DockPanel.Dock="Left" BorderBrush="#FF1570E6" BorderThickness="1" Background="#FFFFFFC7">
  <TextBlock Foreground="White" VerticalAlignment="Center" HorizontalAlignment="Center">Border left 4</TextBlock>
  <Border.Effect>
    <DropShadowEffect Color="LightGray"></DropShadowEffect>
  </Border.Effect>
</Border>
```

- **Border Center 5** tiene las siguientes propiedades
 - No contiene posicionamiento en el **DockPanel** para que la propiedad **LastChildFill** le obligue a ocupar todo el espacio disponible.
 - **Height** de 100.
 - **Margin** de 3.
 - **Background** que saque de una extracción con **Paint**.
 - **BorderBrush** azul.
 - **BorderThickness** de 1.
 - El **Text Block** que contiene el Border Button 3 agregue la propiedad **Foreground** en White, un **HorizontalAlignment** Center y **VerticalAlignment** Center para centrar el **TextBlock** justo a mitad del contenedor.
 - Contiene un **DropShadowEffect** con la propiedad **Color** en LightGray Dentro para hacer un efecto de sombra al contenedor.

```
<Border Margin="3" Width="auto" BorderBrush="#FF1570E6" BorderThickness="1" Background="#FFF7E7">
  <TextBlock Foreground="White" VerticalAlignment="Center" HorizontalAlignment="Center">Border center 5</TextBlock>
  <Border.Effect>
    <DropShadowEffect Color="LightGray"></DropShadowEffect>
  </Border.Effect>
</Border>
```

PRÁCTICA 0

Webgrafía

- <https://www.c-sharpcorner.com/UploadFile/maresh/wpf-textblock/>
- <https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.dockpanel?view=windowsdesktop-8.0>

Cuestión

Si hiciéramos este mismo ejercicio, pero en vez de Border, introduciendo DockPanels dentro del DockPanel principal.

1. ¿Qué ventajas tendríamos luego?

No encuentro ventaja alguna ya que ambos se comparten de la misma manera, por otra parte considero que se pierde la posibilidad de decorar los borders de los DockPanels.

