|  |
| --- |
|  |
|  |

Ejercicio 1 y 2

|  |
| --- |
|  |
| Diseño de Interfaces : Diego Santos Bailon  Leobardo Salvador Ríos Suárez  **10/10/2024**  **2º de DAM I** |

Contenido

[**Ejercicio 1.** 1](#_Toc179548139)

[**Maquetación** 1](#_Toc179548140)

[**Webgrafía** 2](#_Toc179548141)

[**Cuestión** 3](#_Toc179548142)

[**Ejercicio 2** 4](#_Toc179548143)

[**Maquetación** 4](#_Toc179548144)

[**Webgrafía** 7](#_Toc179548145)

[**Cuestión** 7](#_Toc179548146)

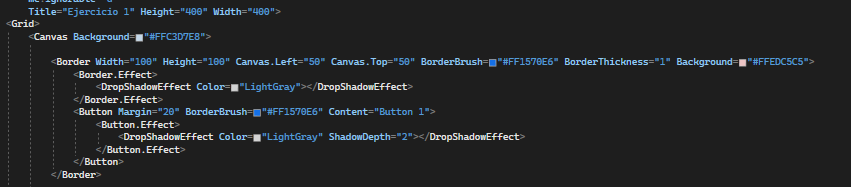
# **Ejercicio 1.**

## **Maquetación**

**Titulo y tamaño de la ventana principal:**

**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 **botton**) 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 botton 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 sombre 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.



## **Webgrafía**

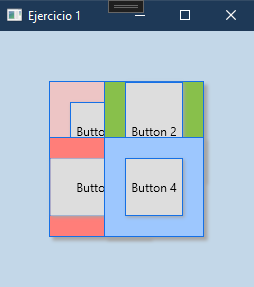
* [**https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.border?view=windowsdesktop-8.0**](https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.border?view=windowsdesktop-8.0)
* [**https://es.stackoverflow.com/questions/99022/como-lograr-efecto-de-sombra-en-wpf**](https://es.stackoverflow.com/questions/99022/como-lograr-efecto-de-sombra-en-wpf)
* [**https://chatgpt.com**](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.**

# **Ejercicio 2**

## **Maquetación**

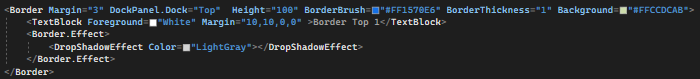
**Titulo y tamaño de la ventana principal:**

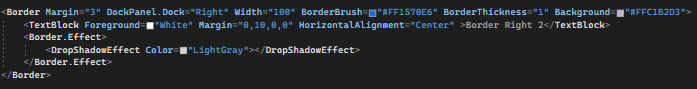
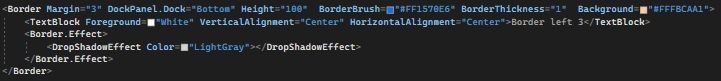
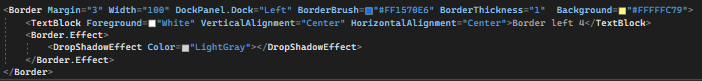
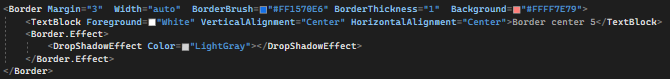
**DockPanel:**

* Coloque el **DockPanel** con la propiedad **LastChildFill** en true para que el último elemento ocupe todo el espacio disponible y se situé donde toca.
* El color lo saque apoyándome de **Paint**.

**Borders:**

* Paro poder escribir en los dentro de los borders utilice un **TextBlock** para texto simple estático.
* **Border Top 1** tiene las siguites 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 sombre al contenedor.



* **Border Right 2** tienen las siguientes propiedades.
  + **DockPanel.Dock=”Right”** para anclarlo a la derecha del Dock panel
  + **Width** de100,
  + **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 **VerticalAlgnment** está en Top.
  + Contiene un **DropShadowEffect** con la propiedad **Color** en LightGray Dentro para hacer un efecto de sombra al contenedor.
* **Border Button 3** tiene las siguites 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 Left 4** tiene las siguites propiedades
  + **DockPanel.Dock=”Lett”:** anclándolo ala 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 Center 5** tiene las siguites propiedades
  + No contiene posicionamiento en el **DockPanel** paraque 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.

## **Webgrafía**

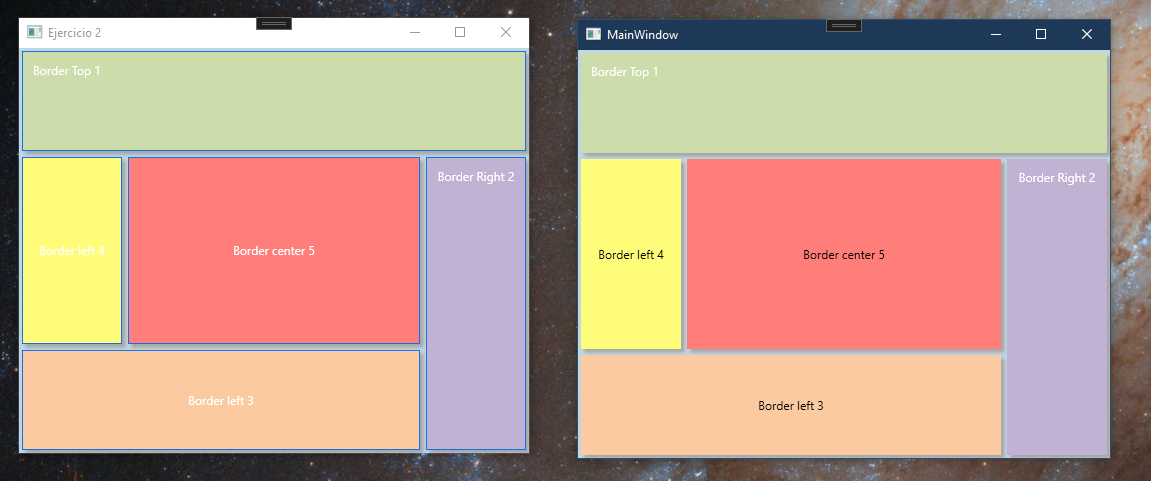
* [**https://www.c-sharpcorner.com/UploadFile/mahesh/wpf-textblock/**](https://www.c-sharpcorner.com/UploadFile/mahesh/wpf-textblock/)
* [**https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.dockpanel?view=windowsdesktop-8.0**](https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.dockpanel?view=windowsdesktop-8.0)

## **Cuestión**

Si hiciaramos este mismo ejejrcicio, pero en vez de Borde, introduciendo DockPanels dentro del DockPanel principal.

1. ¿Qué ventajas tendríamos luego?

**No encuentro ventaja alguna ya que ambos se compartan de la misma manera, por otro lada considero que se pierde la posibilidad de decorar los borders del los DocksPanels.**

****