Introducción a la programación con C#

Interfaz gráfica de usuario con WPF

Un poco de repaso...

¿Qué es una interfaz? ¿De qué hablamos cuando decimos interfaz de usuario?

Un poco de repaso...

¿Qué es un evento?

Un poco de repaso...

¿Por cuáles archivos está compuesta una ventana gráfica dentro del VS? ¿Qué responsabilidad hay en cada uno?

Arquitectura **M**odel-**V**iew-**V**iew**M**odel

MVVM es un patrón de arquitectura de software.

Se caracteriza por tratar de **desacoplar** lo máximo posible la interfaz de usuario de la lógica de la aplicación.

Está compuesto por 3 elementos:

- Modelo (usualmente dentro de la carpeta **Models**).
- Vistas (usualmente dentro de la carpeta **Views**).
- Modelo de la vista (usualmente dentro de la carpeta **ViewModels**).

El Modelo

Representa la capa de datos, también denominado como el objeto del dominio.

El modelo contiene la información, pero nunca las acciones o servicios que la manipulan.

En ningún caso tiene dependencia alguna con la vista. El modelo **NO** conoce a la vista.

La Vista

La misión de la vista es representar la información a través de los elementos visuales que la componen.

Las vistas en MVVM son activas, contienen comportamientos, eventos y enlaces a datos que, en cierta manera, necesitan tener conocimiento del modelo.

El Modelo de la Vista

Es un actor intermediario entre el **modelo** y la **vista**, contiene toda la lógica de presentación y se comporta como una abstracción de la interfaz.

La comunicación entre la vista y el modelo de vista se realiza por medio de los enlaces de datos **(Bindings)**.

Orden vs Caos

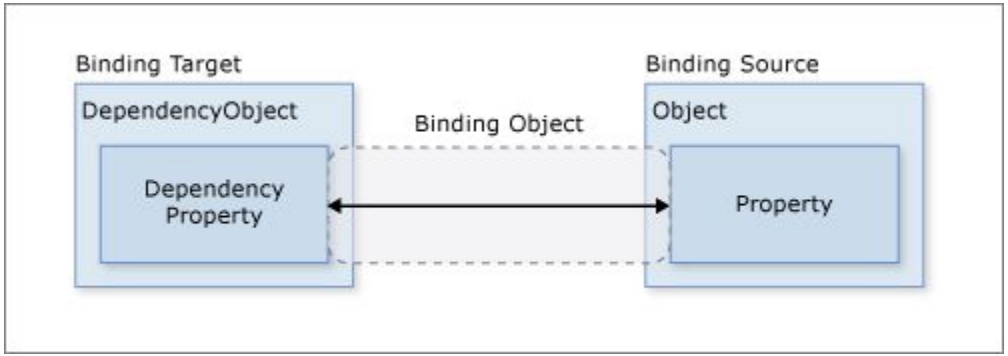
A medida que aumenta el número de funciones en un programa y el número de líneas de código. La complejidad de un sistema y los problemas recurrentes que contiene, alienta a los desarrolladores a organizar su código de tal manera que sea más fácil de **comprender**, **discutir**, **extender** y **solucionar problemas**.

Disminuimos el caos cognitivo de un sistema complejo aplicando nombres conocidos a ciertas entidades en nuestro código.

Documentación oficial

Enlace de datos con **Binding**

Es el proceso que establece una conexión entre la interfaz de usuario de la aplicación y los datos que muestra.



[Documentación oficial]

Style

```
<Window.Resources>
    <!-- .... other resources .... -->

    <!--A Style that affects all TextBlocks-->
    <Style TargetType="TextBlock">
        <Setter Property="HorizontalAlignment" Value="Center" />
        <Setter Property="FontFamily" Value="Comic Sans MS"/>
        <Setter Property="FontSize" Value="14"/>
    </Style>

    <!--A Style that extends the previous TextBlock Style with an x:Key of TitleText-->
    <Style BasedOn="{StaticResource {x:Type TextBlock}}"
           TargetType="TextBlock"
           x:Key="TitleText">
        <Setter Property="FontSize" Value="26"/>
        <Setter Property="Foreground">
            <Setter.Value>
                <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
                    <LinearGradientBrush.GradientStops>
                        <GradientStop Offset="0.0" Color="#90DDDD" />
                        <GradientStop Offset="1.0" Color="#5BFFFF" />
                    </LinearGradientBrush.GradientStops>
                </LinearGradientBrush>
            </Setter.Value>
        </Setter>
    </Style>
</Window.Resources>
```

¡¡A trabajar!!