

# DESARROLLO DE INTERFACES CON WPF

1	¿Qué es WPF?.....	1
2	Características de WPF.....	1
3	Controles .....	2
4	Diseño.....	2

## 1 ¿QUÉ ES WPF?

Windows Presentation Foundation (WPF) es el sucesor de Windows Forms, con sus mismas características de base y múltiples mejoras.

Concretamente WPF es una API (Applications Programming Interfaz) perteneciente al Framework .NET para crear interfaces de usuario (lo que entendemos como aplicaciones gráficas de escritorio) bajo la **plataforma Windows**. También hay una versión Core.

Los **controles** que se añadan a la interfaz se pueden modificar mediante:

- Vista Diseño.
- XAML (Extensible Application Markup Language).
- C# (code behind del MainWindow.xaml.cs).

## 2 CARACTERÍSTICAS DE WPF

### Interfaz gráfica declarativa

WPF permite crear interfaces gráficas empleando XAML (es un lenguaje de marcas, etiquetas), muy similar a HTML.

En Visual Studio tenemos la vista diseño similar al de las aplicaciones Windows Forms. La diferencia es que hay un editor donde puedes escribir el código XAML directamente.

### Binding

Es un mecanismo para enlazar los elementos de la interfaz de usuario con los objetos que contienen la información a mostrar.

El caso más típico de data binding es el enlazar un control de la interfaz de usuario con un valor o registro de una base de datos. Con **Windows Forms** se utiliza el **databinding**.

### Diseño dinámico

La organización de los componentes de una aplicación puede ser muy complicado. Además, es necesario considerar las diferentes resoluciones y características de una pantalla de un usuario.

En este sentido, las interfaces WPF generadas son **vectoriales**, es decir, se podrán escalar a diferentes tamaños sin perder definición, dependiendo de cómo se haya definido el diseño.

### Plantillas

Se pueden crear elementos reutilizables para la interfaz gráfica. Existen dos tipos de plantillas en WPF: plantillas de control y plantillas de datos.

Las **plantillas de control** (control templates) redefinen el aspecto visual de un control, permitiendo, por ejemplo, cambiar la apariencia de todos los botones de una aplicación desde un único punto.

Las **plantillas de datos** (data templates) definen la manera en que los datos son mostrados, asocian una plantilla a una entidad o clase de la aplicación. Es decir, siempre que se muestre ese tipo de dato se utilizará la misma plantilla.

### Estilos

Los estilos en WPF son muy parecidos a las hojas de estilos en cascada, CSS, para HTML. Incluso más potentes, permitiendo manejar cualquier característica gráfica que te imagines, márgenes, espacios, colores, dimensiones y muchos más.

### DirectX vs GDI

Mientras que su predecesor está basado en la interfaz de diseño de gráficos GDI (Graphics Design Interface), WPF se basa en DirectX. Esta API está enfocada al desarrollo de multimedia y particularmente a la programación de juegos.

## 3 CONTROLES

Un control es un término genérico que se aplica a una categoría de clases de WPF dentro de la ventana o página, tienen una apariencia e implementan un comportamiento determinado.

Algunos de los más empleados según su **funcionalidad**:

- **Botones:** Button.
- **Presentación de datos:** DataGrid, ListView y TreeView.
- **Presentación y selección de fechas:** Calendar y DatePicker.
- **Entrada:** TextBox, RichTextBox y PasswordBox.
- **Diseño:** Border, Canvas, DockPanel, Grid, GridSplitter, GroupBox, ScrollBar, StackPanel.
- **Multimedia:** Image, MediaElement.
- **Menús:** MenuItem y ToolBar.
- **Navegación:** Frame, TabControl.
- **Selección:** CheckBox, ComboBox, ListBox, RadioButton y Slider.
- **Información del usuario:** Label, ProgressBar, StatusBar.

## 4 DISEÑO

Al crear una interfaz de usuario, se organizan los **controles** según su ubicación y tamaño. Un requisito fundamental de cualquier diseño es adaptarse a los cambios de tamaño de la ventana y de configuración de pantalla.

WPF incluye varios controles de diseño:

- **Canvas:** los controles secundarios se colocan explícitamente mediante coordenadas relativas al área de Canvas.
- **DockPanel:** los controles secundarios se organizan de forma horizontal o vertical.
- **Grid:** los controles secundarios se colocan por filas y columnas.
- **StackPanel:** los controles secundarios se apilan vertical u horizontalmente.
- **WrapPanel:** los controles secundarios se sitúan en orden de izquierda a derecha y se ajustan a la línea siguiente cuando hay más controles de los que caben en la línea actual.