

Windows Presentation Foundation

Windows Presentation Foundation (WPF) es una [tecnología](#) de [Microsoft](#), presentada como parte de [Windows Vista](#). Permite el desarrollo de interfaces de interacción en Windows tomando características de [aplicaciones](#) Windows y de [aplicaciones web](#).

WPF ofrece una amplia infraestructura y potencia gráfica con la que es posible desarrollar aplicaciones visualmente atractivas, con facilidades de interacción que incluyen [animación](#), [vídeo](#), [audio](#), [documentos](#), navegación o gráficos [3D](#). Separa, con el lenguaje declarativo [XAML](#) y los [lenguajes de programación](#) de [.NET](#), la [interfaz](#) de interacción de la lógica del negocio, propiciando una arquitectura [Modelo Vista Controlador](#) para el desarrollo de las aplicaciones.

WPF se incluye con [Windows 7](#), [Windows Vista](#) y [Windows Server 2008](#) y también está disponible para [Windows XP](#) Service Pack 2 o posterior y [Windows Server 2003](#).

[Microsoft Silverlight](#) es un subconjunto de WPF que permite crear aplicaciones similares a [Flash](#) para web y móviles, con el mismo modelo de programación que las aplicaciones .NET. No admite funciones 3D, pero soporta [XPS](#) y planos basados en vectores.

Índice

- [1 Características](#)
 - [1.1 Direct3D](#)
 - [1.2 Data binding \(enlace de datos\)](#)
 - [1.3 Media Services](#)
- [2 Esquema](#)
 - [2.1 Plantillas](#)
 - [2.2 Animaciones](#)
 - [2.3 Imágenes](#)
 - [2.4 Efectos](#)
 - [2.5 Interoperabilidad](#)
- [3 XAML](#)
- [4 Véase también](#)
- [5 Referencias](#)
- [6 Enlaces externos](#)

Características

Direct3D

Todos los gráficos, incluyendo los elementos de escritorio mostrados en Windows, se representan usando [Direct3D](#). Esto proporciona una vía para mostrar gráficos más complejos y temas personalizados, a partir de [GDI](#) una gama más amplia de apoyo y tematización uniforme de control. También permite descargar algunas de las tareas de gráficos a la [GPU](#). Esto puede reducir la carga de trabajo de la [CPU](#), ya que las [GPUs](#) están optimizadas para cálculos de píxeles a gran velocidad lo que permite acelerar el refresco de la pantalla. Por desgracia, esta ventaja se reduce en los mercados donde las GPU no son tan rápidas, como el de los [netbooks](#). WPF pone énfasis en los gráficos vectoriales porque esto permite a la mayoría de los controles y elementos ser escalados sin pérdida de calidad o [pixelización](#), aumentando así la accesibilidad. Con la excepción de [Silverlight](#), esta integración permite el uso de una versión de Direct3D

simplificada. Además, los contenidos interactivos en 2D pueden ser superpuestos en superficies 3D nativas.

Data binding (enlace de datos)

- WPF tiene un conjunto de servicios de datos para permitir a los desarrolladores de aplicaciones enlazar y manipular los datos dentro de las aplicaciones. Existen tres tipos de enlace de datos:
 - Una vez: cuando el cliente hace caso omiso de las actualizaciones en el servidor.
 - Simplex (one way): el cliente solo tiene permiso de lectura a los datos.
 - Dúplex (two way): el cliente puede leer y escribir datos en el servidor.
- Las consultas [LINQ](#), incluyendo *LINQ to XML*, pueden actuar también como fuentes de datos para el enlace de datos.
- El enlace de datos no influye en su presentación. WPF proporciona plantillas de datos para controlar la presentación de los datos.
- Como parte de WPF se proporciona también un conjunto de controles integrados, que contiene elementos como botones, menús, rejillas, y listas.
- Un concepto muy potente en WPF es la separación lógica entre un control y su apariencia.
 - La plantilla de un control se puede sobrescribir para cambiar por completo su aspecto visual.
 - Un control puede contener cualquier otro control o diseño (*layout*), lo que permite un alto grado de control sobre la composición.
- Tiene un modo de «gráficos retenidos» (*retained graphics*): no siempre es necesario repintar la pantalla.

Media Services

WPF proporciona un sistema integrado para la creación de interfaces de usuario con elementos multimedia comunes, como imágenes vectoriales y de mapa de bits, audio y vídeo. WPF también proporciona un sistema de animación y un sistema de renderizado en 2D/3D.

- WPF proporciona primitivas de forma para gráficos en 2D junto con un conjunto integrado de pinceles, lápices, geometrías, y transformaciones.
- Las capacidades 3D de WPF son un subconjunto del conjunto completo de características proporcionado por Direct3D. Sin embargo, WPF proporciona una integración más estrecha con otras funciones como las interfaces de usuario, documentos, y multimedia. Esto hace que sea posible disponer de interfaces de usuario en 3D, documentos 3D o multimedia en 3D.
- Se soporta la mayoría de formatos de imagen más comunes: [BMP](#), [JPEG](#), [PNG](#), [TIFF](#), [Windows Media Photo](#), [GIF](#), e [iconos](#).
- WPF admite por defecto los formatos de vídeo [WMV](#), [MPEG](#) y algunos [AVI](#) archivos, pero como por debajo ejecuta Windows Media Player, WPF puede usar todos los codecs instalados para el mismo.

Esquema

Plantillas

- En WPF, puede definir el aspecto de un elemento directamente, a través de sus propiedades, o indirectamente a una plantilla o el estilo. En su forma más simple un estilo es una combinación de valores de propiedades que se pueden aplicar a un elemento de interfaz de usuario con un atributo de la propiedad individual. Las plantillas son un mecanismo alternativo para la definición de interfaz de usuario para partes de la aplicación de WPF. Hay varios tipos de plantillas disponibles en WPF.

- Detrás de todos los controles de interfaz de usuario de WPF es un modelo de nueva composición. Todos los mandos se compone de una o más "visuales". Estos sub-elementos visuales se convierten en un árbol jerárquico de Visual por WPF y eventualmente prestados por la CPU. Dado que los controles de WPF no son contenedores para controles estándar de Windows de su interfaz de usuario puede ser radicalmente modificada sin afectar el comportamiento normal del control.
- Todos los mandos de WPF tiene por defecto una plantilla que define su árbol visual. La plantilla predeterminada es creado por el autor de control y es sustituible por otros desarrolladores y diseñadores. El sustituto de interfaz de usuario se coloca dentro de una ControlTemplate.
- WPF tiene un sistema de enlace de datos flexible. elementos de la interfaz se pueden rellenar y sincronizada con los datos de un modelo de datos subyacente. En lugar de mostrar un texto sencillo de los datos enlazados WPF puede solicitar una plantilla de datos antes de representar al árbol visual.

Animaciones

- WPF admite animaciones basadas en el tiempo, en contraste con el enfoque basado en frames. Esto desacopla la velocidad de la animación de cómo el sistema está funcionando.
- WPF admite animación de bajo nivel a través de temporizadores y más alto nivel de abstracción a través de animaciones las clases de animación.
 - Cualquier propiedad elemento de WPF pueden ser animados, siempre y cuando se registra como una propiedad de dependencia.
 - Clases de animación se basan en el .NET tipo de bienes a ser animados. Por ejemplo, cambiando el color de un elemento se hace con la clase ColorAnimation y animar el ancho de un elemento (que se escribe como un doble) se hace con la clase DoubleAnimation.
- Las animaciones se pueden agrupar en los guiones gráficos.
 - Guiones gráficos son la forma principal para iniciar, detener, pausar y manipular las animaciones.
- Las animaciones pueden ser desencadenadas por sucesos externos, incluyendo la intervención del usuario.
- El redibujado de una escena se produce por el disparo de un temporizador (en inglés "timer").
- Presentación temporizadores se inicializan y gestionado por WPF.
- Los efectos de animación se pueden definir en función de cada objeto, que se puede acceder directamente desde marcado XAML.

Imágenes

- WPF puede tener acceso de forma nativa [Windows Imaging Component](#) (WIC) y el código API permite a los desarrolladores a escribir codecs de imágenes para sus formatos específicos de la imagen de archivo.

Efectos

- WPF 3.0 prevé BitmapEffects, que son efectos ráster aplicados a un proyecto Visual Studio. Estos efectos de trama están escritos en código no administrado y la fuerza de la prestación del Visual que se realizarán en la CPU y no de hardware acelerado por la GPU. BitmapEffects fueron desaprobados en .NET 3.5 SP 1.
- .NET Framework 3.5 Service Pack 1 añade la clase de efectos, que es un Pixel Shader 2.0 sentido de que se puede aplicar a un gráfico, lo que permite que toda la representación de permanecer en la GPU.
- La clase de efectos puede ampliarse para permitir la aplicación para especificar sus propios efectos de sombreado.
- .NET 3.5 SP1 con dos buques de efectos integrados, y BlurEffect DropShadowEffect.

Interoperabilidad

- WPF proporciona interoperabilidad con el [Windows API](#), se puede utilizar Windows Presentation Foundation dentro del código existente de Win32, o uno puede utilizar el código existente Win32 dentro de Windows Presentation Foundation.
- Interoperabilidad con [Formularios Windows](#).

XAML

Tras el éxito de [XML](#) para el desarrollo web, WPF introduce un nuevo lenguaje conocido como lenguaje extensible de marcado de aplicaciones , que se basa en [XML](#). XAML está diseñado como un método más eficaz de desarrollo de interfaces de aplicaciones de usuario.

La ventaja específica que XAML lleva a WPF es que XAML es un lenguaje completamente declarativo. En un lenguaje de programación declarativa, el desarrollador (o diseñador) describe el comportamiento y la integración de los componentes sin utilizar programación procedural. Aunque es raro que una aplicación completa se construya totalmente en XAML, la introducción de XAML permite a los diseñadores de aplicaciones contribuir más eficazmente al ciclo de desarrollo de aplicaciones. La utilización de XAML para desarrollar interfaces de usuario permite también la separación entre modelo y vista, lo que se considera un buen principio de arquitectura. En XAML, los elementos y atributos mapean las clases y propiedades en el API subyacente.

Al igual que en el desarrollo web, tanto en diseños y temas específicos son muy adecuadas para el marcado, pero en XAML no es necesario. De hecho, todos los elementos de WPF puede ser codificada en Microsoft C# y VB.NET. En última instancia el código XAML, se puede compilar en un ensamblado administrado por Visual Studio .NET .