INTERFACES GRÁFICAS DE USUARIO

WPF: Windows Presentation Foundation

Introducción

.NET

WPF

Windows

Aplicación Básica

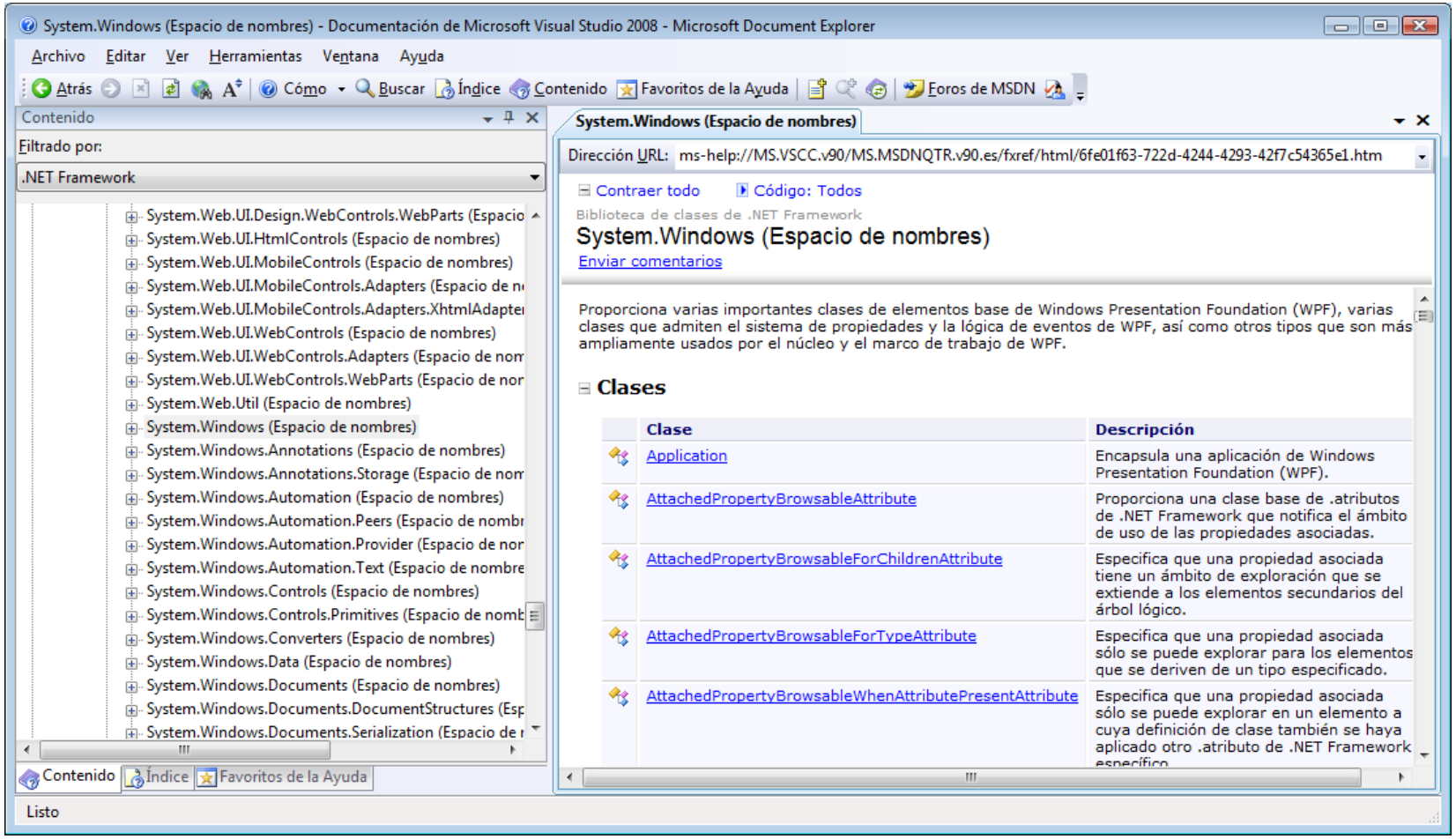
Introducción a XAML

Creando una aplicación

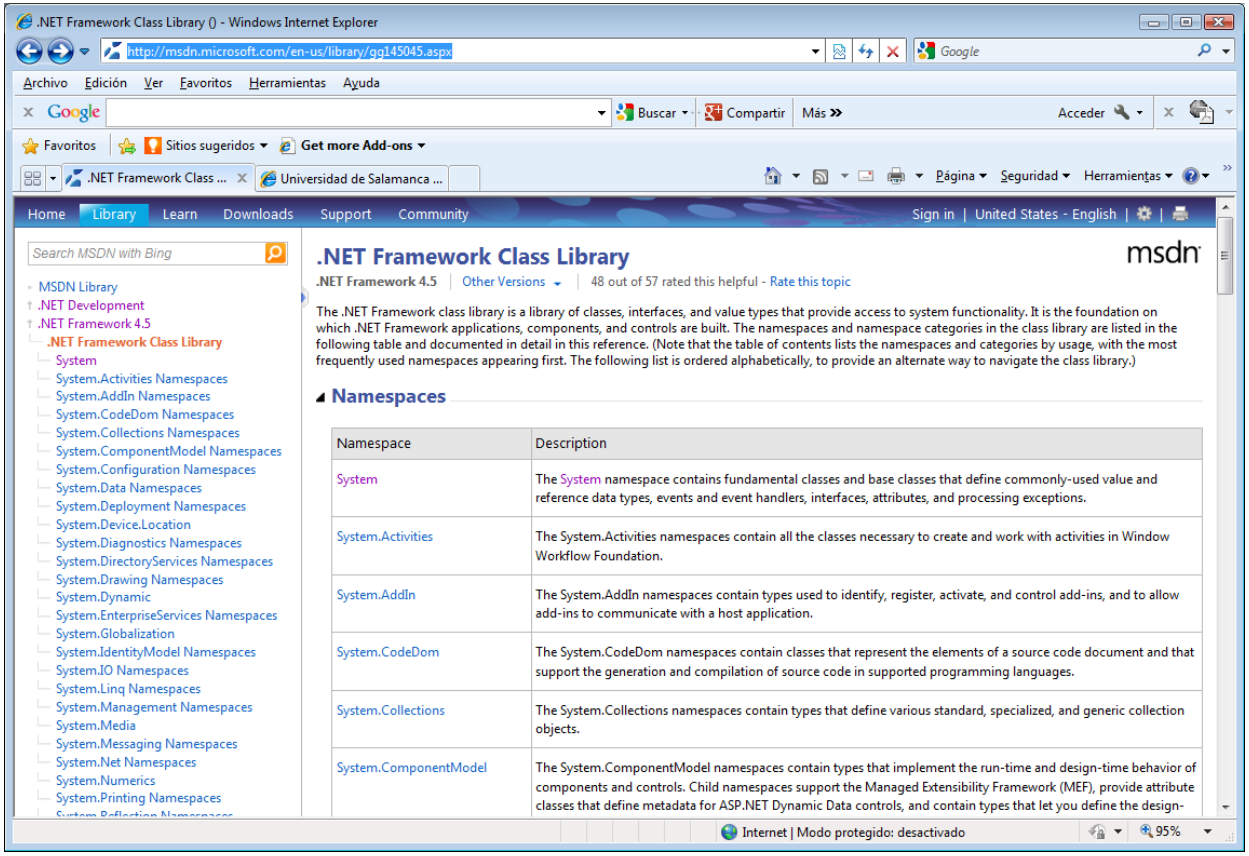
.NET

- **¿Qué es .NET?**
  - Microsoft .NET Framework es un conjunto de tecnologías de software de Microsoft
    - Define un entorno compatible con el desarrollo y ejecución de aplicaciones basadas en componentes
  - Para un usuario .NET es básicamente una colección de bibliotecas de vínculos dinámicos (DDL)
  - Para el programador, .NET es una gran biblioteca de clases que contiene todo lo necesario para escribir aplicaciones, tanto web como aplicaciones de cliente

.NET


System.Windows (Espacio de nombres) - Documentación de Microsoft Visual Studio 2008 - Microsoft Document Explorer
Archivo Editar Ver Herramientas Ventana Ayuda
Atrás Cómo Buscar Índice Contenido Favoritos de la Ayuda Foros de MSDN
Contenido
Filtrado por:
.NET Framework
System.Web.UI.Design.WebControls.WebParts (Espacio
System.Web.UI.HtmlControls (Espacio de nombres)
System.Web.UI.MobileControls (Espacio de nombres)
System.Web.UI.MobileControls.Adapters (Espacio de n
System.Web.UI.MobileControls.Adapters.XhtmlAdapte
System.Web.UI.WebControls (Espacio de nombres)
System.Web.UI.WebControls.Adapters (Espacio de nom
System.Web.UI.WebControls.WebParts (Espacio de nor
System.Web.Util (Espacio de nombres)
System.Windows (Espacio de nombres)
System.Windows.Annotations (Espacio de nombres)
System.Windows.Annotations.Storage (Espacio de nom
System.Windows.Automation (Espacio de nombres)
System.Windows.Automation.Peers (Espacio de nombr
System.Windows.Automation.Provider (Espacio de nor
System.Windows.Automation.Text (Espacio de nombre
System.Windows.Controls (Espacio de nombres)
System.Windows.Controls.Primitives (Espacio de nomb
System.Windows.Converters (Espacio de nombres)
System.Windows.Data (Espacio de nombres)
System.Windows.Documents (Espacio de nombres)
System.Windows.Documents.DocumentStructures (Esp
System.Windows.Documents.Serialization (Espacio de
Contenido Índice Favoritos de la Ayuda
System.Windows (Espacio de nombres)
Dirección URL: ms-help://MS.VSCC.v90/MS.MSDNQTR.v90.es/fxref/html/6fe01f63-722d-4244-4293-42f7c54365e1.htm
Contraer todo Código: Todos
Biblioteca de clases de .NET Framework
System.Windows (Espacio de nombres)
Enviar comentarios
Proporciona varias importantes clases de elementos base de Windows Presentation Foundation (WPF), varias clases que admiten el sistema de propiedades y la lógica de eventos de WPF, así como otros tipos que son más ampliamente usados por el núcleo y el marco de trabajo de WPF.
Clases
Clase
Descripción
Application
Encapsula una aplicación de Windows Presentation Foundation (WPF).
AttachedPropertyBrowsableAttribute
Proporciona una clase base de .atributos de .NET Framework que notifica el ámbito de uso de las propiedades asociadas.
AttachedPropertyBrowsableForChildrenAttribute
Especifica que una propiedad asociada tiene un ámbito de exploración que se extiende a los elementos secundarios del árbol lógico.
AttachedPropertyBrowsableForTypeAttribute
Especifica que una propiedad asociada sólo se puede explorar para los elementos que se deriven de un tipo especificado.
AttachedPropertyBrowsableWhenAttributePresentAttribute
Especifica que una propiedad asociada sólo se puede explorar en un elemento a cuya definición de clase también se haya aplicado otro .atributo de .NET Framework
Listo

.NET



http://msdn.microsoft.com/en-us/library/gg145045.aspx

.NET

- **Lenguajes de programación**
  - .NET soporta varios lenguajes de programación
  - Los lenguajes deben cumplir unos requerimientos mínimos
    - CLI: Common Language Infrastructure
      - Estándar ISO que incluye CLS y CTS
      - CLS: .NET Common Language Specification
        - Describe los requisitos mínimos de un lenguaje para poder utilizar las bibliotecas de clases
      - CTS: .NET Common Type System
        - Define los tipos de datos básicos que deben soportar los lenguajes en .NET

.NET

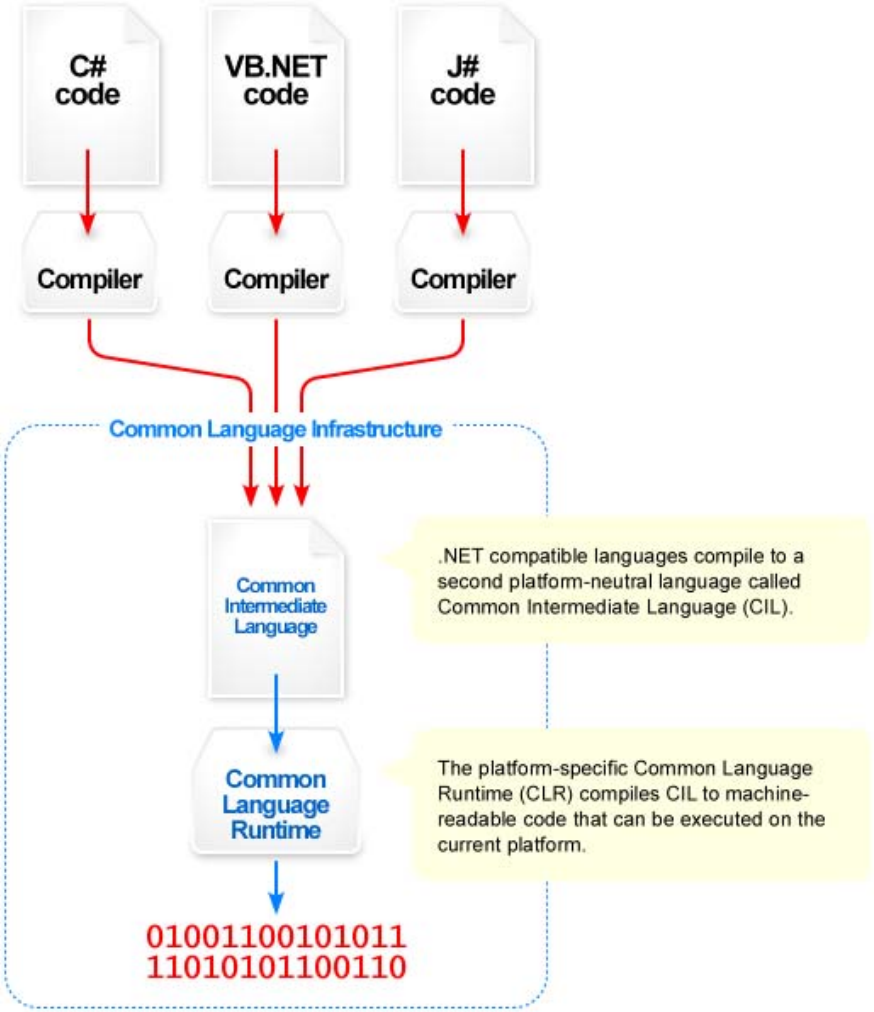


Imagen tomada de http://blog.tangcs.com/2008/01/20/dotnet-clr-overview/

.NET

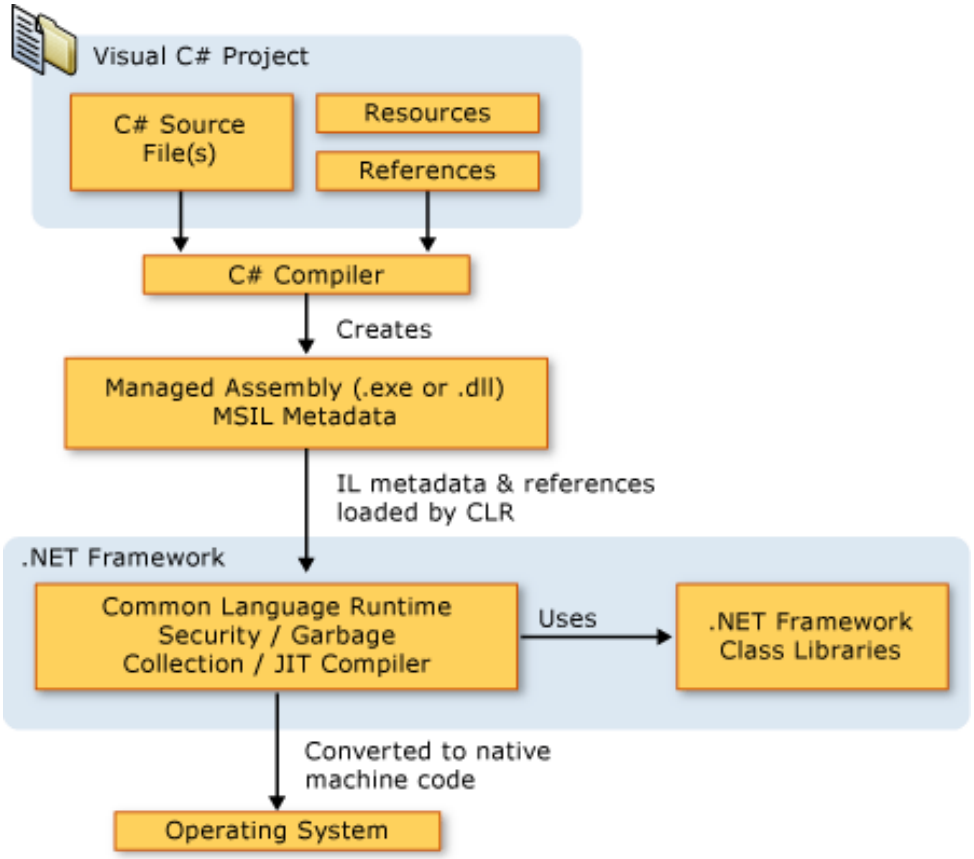
- Compilación
  - La compilación de los fuentes genera archivos ejecutables que contienen código CIL (Common Intermediate Language)
    - Conjunto de instrucciones portable e independiente de procesador y sistema operativo
    - CIL es un lenguaje intermedio
    - También llamado MSIL o IL
  - También genera metadatos
    - Permiten la interacción con otros módulos
    - Se almacenan junto con el CIL

.NET

- Ejecución
  - CLR : Common Language Runtime
    - Administra la ejecución de programas
    - Hace posible la ejecución del código CIL (código administrado)
    - El compilador JIT (Just in time) convierte el código CIL al código máquina apropiado (nativo)
    - Permite la portabilidad del código con un bajo coste por la compilación final a código nativo

.NET

- **Compilación y Ejecución**



(msdn.microsoft.com/en-us/library/z1zx9t92.aspx)

WPF

- Windows Presentation Foundation (WPF)
- Es un conjunto de bibliotecas de clases que dan soporte al desarrollo de aplicaciones para Windows
- Es un subconjunto de .NET Framework
- Separación de la implementación de la apariencia de una aplicación de la implementación de la funcionalidad

WPF

- Implementación de la apariencia
  - Usa un lenguaje de marcado
    - XAML (Extensible Application Markup Language)
  - Herramientas específicas de diseño
    - Expression (http://www.microsoft.com/expression/)
- Implementación de la funcionalidad
  - Lenguaje de programación administrado (subyacente)
    - C#, VB.NET, …

Windows

- **Características de Windows**
  - Multitarea
  - Gestión de Memoria
    - Memoria Virtual
    - Varias instancias de un programa comparten el código en memoria
    - Los programas comparten librerías de enlace dinámico (DLL)
  - Independiente de dispositivo
  - Debe asumirse en su totalidad

Windows

- Programación Tradicional
  - Leer Datos de Entrada
  - Procesar Datos
  - Presentar Salida
- Programación por eventos
  - Presentar elementos de interfaz
  - Esperar evento
    - Procesar evento

Windows

- **Evento**
  - Es un hecho cuya aparición implica un cierto proceso por parte de la aplicación
    - Se ha pulsado un botón del ratón
    - Se ha seleccionado una opción de un menú
    - Se ha pulsado una tecla
    - Se ha seleccionado una opción de una lista
    - Se ha movido el ratón
    - ...

Windows

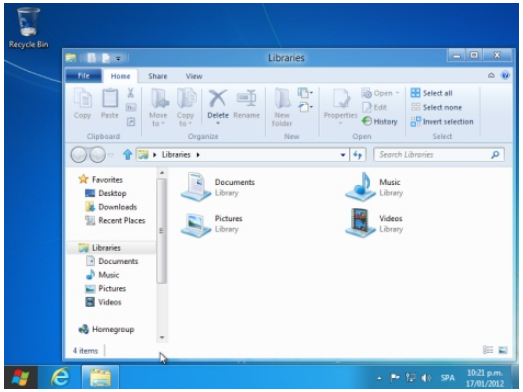
- **Programación por eventos**
  - Los eventos son comunicados a la aplicación a través de mensajes
  - Windows crea una ***cola de mensajes*** para cada aplicación
  - Windows envía mensajes a la cola de mensajes de la aplicación
  - Cada mensaje va dirigido a una de las ventanas de la aplicación
  - La aplicación recupera y procesa los mensajes de su cola en el ***bucle de mensajes***

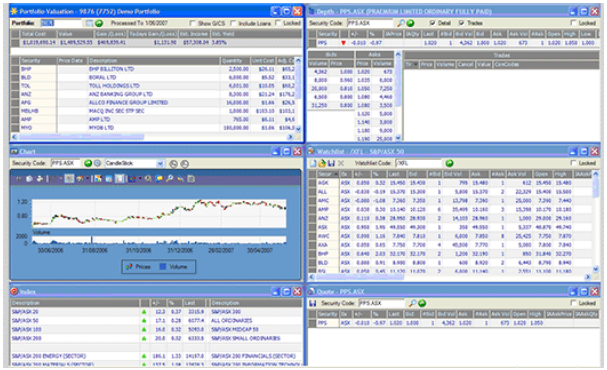
Windows

- Desarrollo en Windows
  - API nativa
  - MFC
  - Windows Forms (.NET)
  - WPF (.NET)

Windows

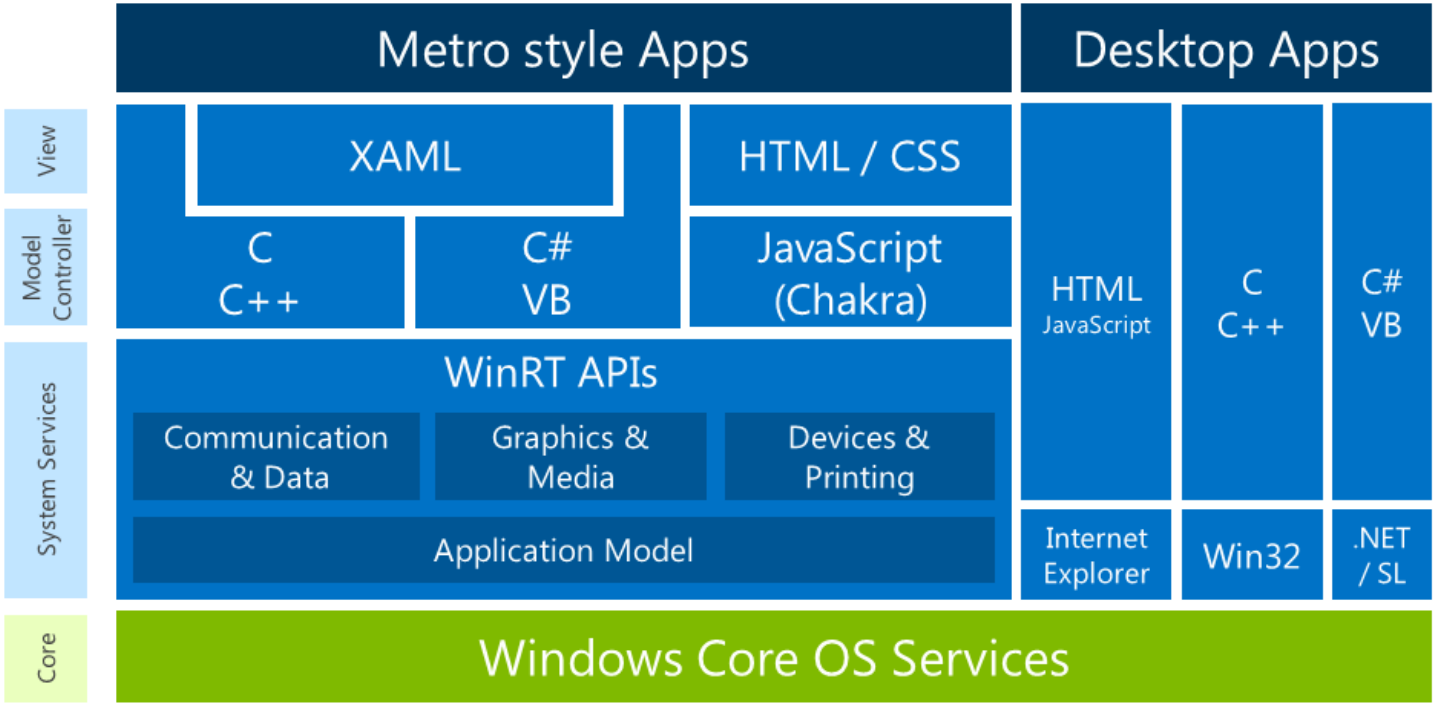
- Windows 8
  - Modern UI Style (Metro Style)
  - Escritorio





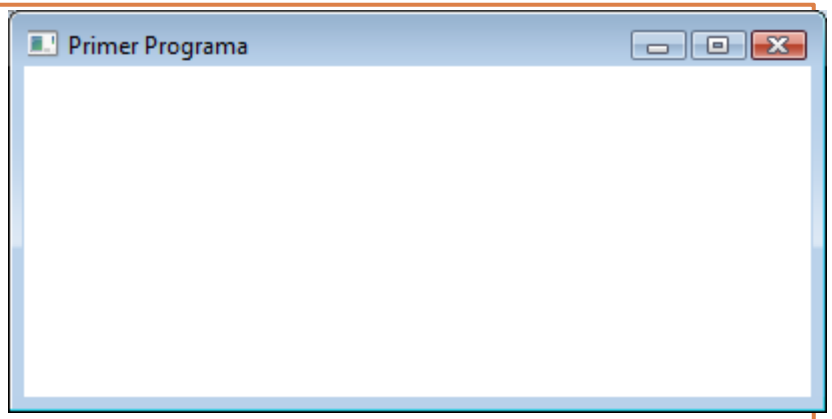
Windows

- Windows 8



WPF. Aplicación básica (1)

```
using System;
using System.Windows;
namespace Proyectos.ElPrimero
{
    class Elemental
    {
        [STAThread]
        public static void Main()
        {
            Window win = new Window();
            win.Title = "Primer Programa";
            win.Show();
            Application app = new Application();
            app.Run();
        }
    }
}
```



Aplicación Básica

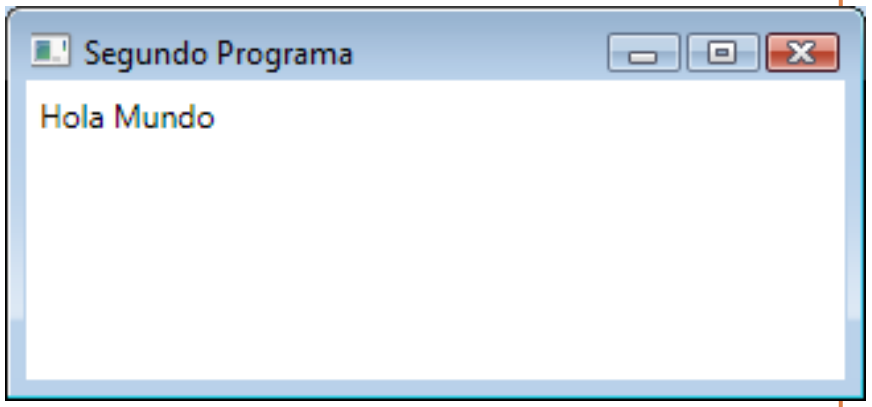
- Clase Window
  - Definida en el namespace System.Windows
  - Abstracción para una ventana de Win32
  - Métodos:
    - Show(), Hide(), Close(), …
  - Propiedades
    - Title, Icon, WindowStyle, Left, Top, Topmost, ShowInTaskbar, …
  - Eventos
    - Activated, Deactivated, Initialized, Closing, Closed, …

Aplicación Básica

- Clase Application
  - Debe instanciarse un objeto de la clase Application o de una clase derivada de ella
  - Entre otras cosas, es la encargada de recuperar los mensajes de la cola de mensajes de la aplicación y gestionar su procesamiento.
    - El método Run() implementa el bucle de mensajes
  - Propiedades
    - MainWindow, ShutdownMode, Properties
  - Eventos
    - Startup, Exit, Activated, Deactivated, …

Aplicación básica (2)

```
using System;
using System.Windows;
using System.Windows.Controls;
namespace Proyectos.ElSegundo
{
    class Basica
    {
        [STAThread]
        public static void Main()
        {
            Window win = new Window();
            win.Title = "Segundo Programa";
            win.Show();
            Label etiqueta = new Label();
            etiqueta.Content = "Hola Mundo";
            win.Content = etiqueta;
            Application app = new Application();
            app.Run();
        }
    }
}
```



XAML

- Es un lenguaje declarativo de propósito general basado en XML
- En WPF se utiliza para construir e inicializar objetos
- El uso de XAML en WPF es opcional
  - En las aplicaciones básicas anteriores no se ha utilizado
- Un archivo XAML contiene la definición de un elemento
- Sintáxis:

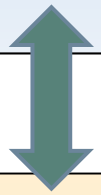
<elemento [propiedad="valor" …]> contenido </elemento>

<elemento [propiedad="valor" …]/>

XAML

```
<Window x:Class="WpfApplication1.Window1”
   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Primero con XAML“>
    <Label>
        Hola Mundo
    </Label>
</Window>
```

XAML



```
Window1 win = new Window1();
 win.Title = "Primero con XAML";
 Label etiqueta = new Label();
 etiqueta.Content = "Hola Mundo";
 win.Content = etiqueta;
```

C#

XAML

- Espacio de nombres
  - Contienen los elementos y atributos que se pueden utilizar
  - La mayoría de los elementos y atributos pertenecerán al espacio de nombres de WPF, que será el que se establezca como espacio de nombres por defecto:

```
xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation
```

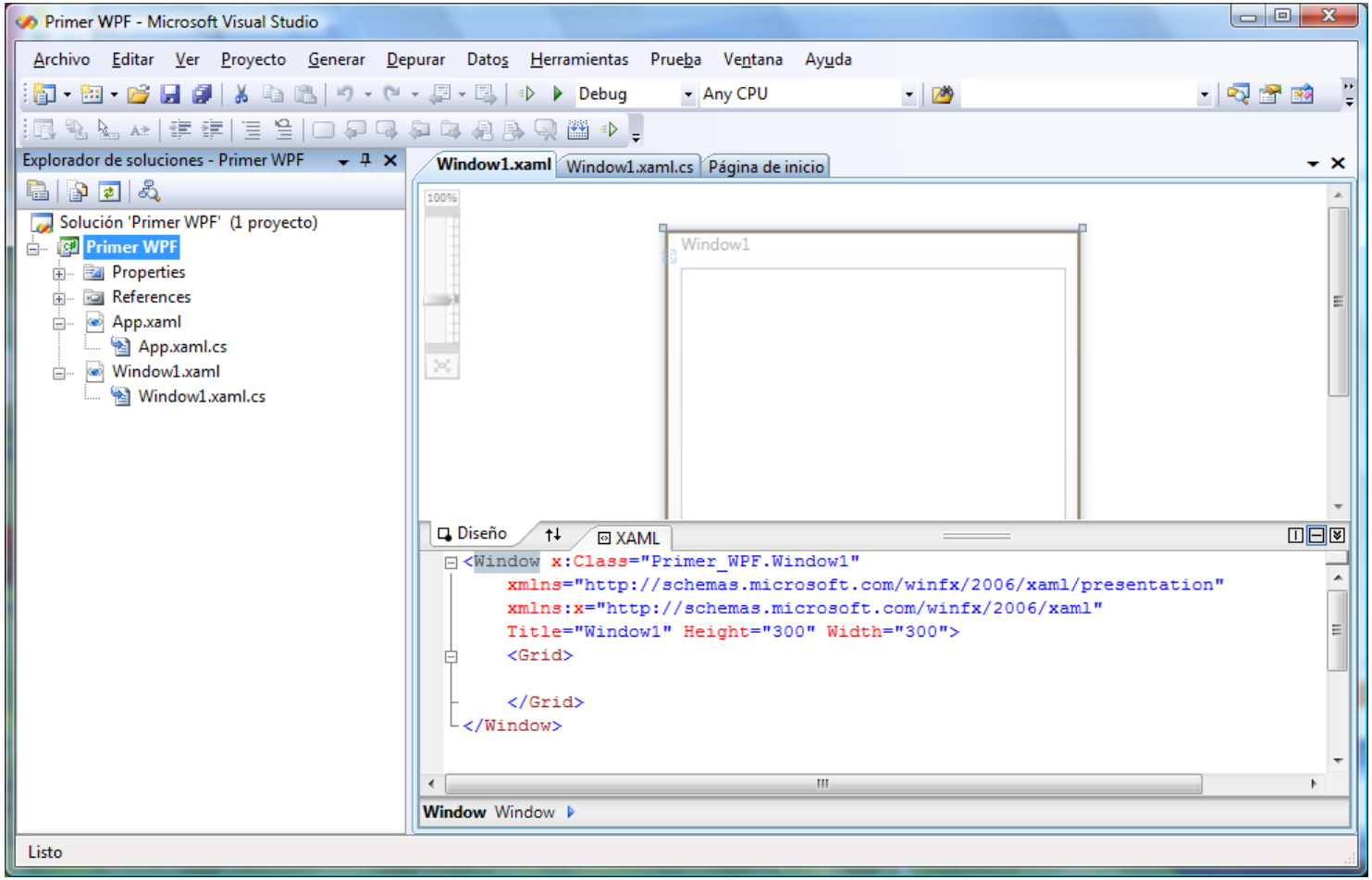
  - También se necesita el espacio de nombres que contiene las definiciones propias de XAML. Para él se define un prefijo (por ejemplo x)

```
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml
```

Creando una aplicación

- MS Visual Studio
- Crear un nuevo proyecto WPF
  - Achivo-Nuevo-Proyecto-Aplicación WPF
  - Poner nombre
- Examinar los archivos:
  - App.xaml
  - App.xaml.cs
  - Windows1.xaml o MainWindow.xaml
  - Windows1.xaml.cs o MainWindow.xaml.cs

Creando una aplicación



Creando una aplicación

- App.xaml

```
<Application x:Class="Primer_WPF.App“
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    StartupUri="Window1.xaml">
    <Application.Resources>

    </Application.Resources>
</Application>
```

Creando una aplicación

- App.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Windows;
namespace Primer_WPF
{
    /// <summary>
    /// Lógica de interacción para App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

Creando una aplicación

- Window1.xaml

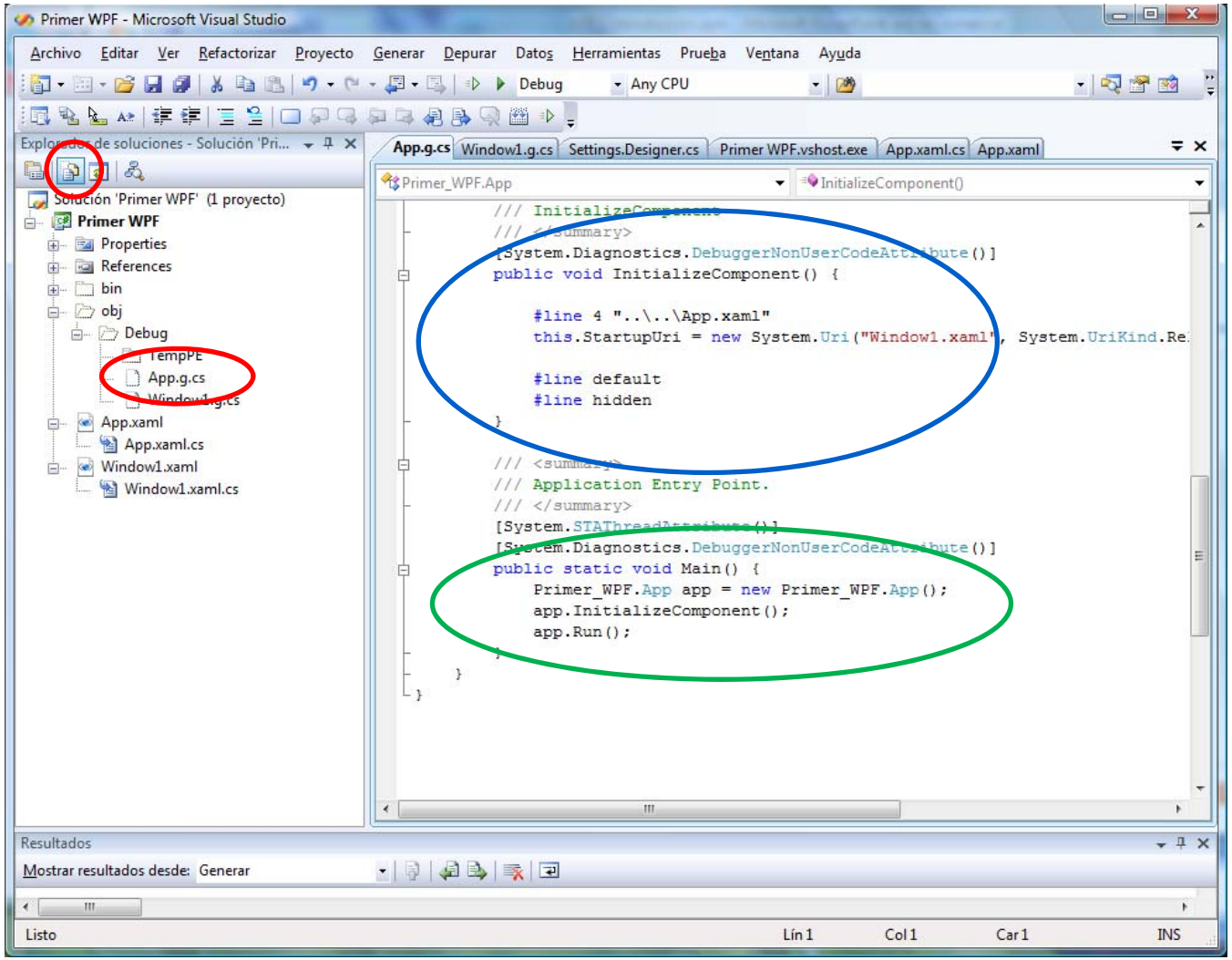
```
<Window x:Class="Primer_WPF.Window1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Window1" Height="300" Width="300">
    <Grid>

    </Grid>
</Window>
```

Creando una aplicación

- Window1.xaml.cs

```
using System;
// otros using
namespace Primer_WPF
{
    /// <summary>
    /// Lógica de interacción para Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
        }
    }
}
```

Creando una aplicación



Creando una aplicación

- Se llama a `InitializeComponent` desde el constructor de la clase en el código subyacente para combinar la interfaz de usuario definida en XAML
- `InitializeComponent` se genera automáticamente al compilar la aplicación
- La combinación de `x:Class` e `InitializeComponent` garantiza que la implementación se inicializa correctamente

Bibliografía

- **WPF 4 Unleashed.** A. Nathan. Pearson 2010
- **MSDN**. **Desarrollo .NET**

  http://msdn.microsoft.com/es-es/library/aa139615.aspx
- **Applications=Code+Markup**. Petzold. MS Press 2006
- **Programming Windows, 6th Edition**. Petzold. MS Press 2013
- **.NET Book Zero**. Petzold.

  http://www.charlespetzold.com/dotnet