

Microsoft Silverlight **en acción**

Marcos Falagán Vidales



Microsoft Silverlight en acción
Marcos Falagán Vidales

ISBN: 978-84-9383 12-9-5

EAN: 978849383 1295

Copyright © 2012 RC Libros
© RC Libros es un sello y marca comercial registrados

Microsoft Silverlight en acción.

Reservados todos los derechos. Ninguna parte de este libro incluida la cubierta puede ser reproducida, su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes intencionadamente reprodujeren o plagiaran, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución en cualquier tipo de soporte existente o de próxima invención, sin autorización previa y por escrito de los titulares de los derechos de la propiedad intelectual.

RC Libros, el Autor, y cualquier persona o empresa participante en la redacción, edición o producción de este libro, en ningún caso serán responsables de los resultados del uso de su contenido, ni de cualquier violación de patentes o derechos de terceras partes. El objetivo de la obra es proporcionar al lector conocimientos precisos y acreditados sobre el tema tratado pero su venta no supone ninguna forma de asistencia legal, administrativa ni de ningún otro tipo, si se precisase ayuda adicional o experta deberán buscarse los servicios de profesionales competentes. Productos y marcas citados en su contenido estén o no registrados, pertenecen a sus respectivos propietarios.

RC Libros
Calle Mar Mediterráneo, 2. Nave 6
28830 SAN FERNANDO DE HENARES, Madrid
Teléfono: +34 91 677 57 22
Fax: +34 91 677 57 22
Correo electrónico: info@rclibros.es
Internet: www.rclibros.es

Diseño de colección, cubierta y pre-impresión: Grupo RC
Impresión y encuadernación:
Depósito Legal:
Impreso en España

16 15 14 13 12 (1 2 3 4 5 6 7 8 9 10 11 12)

INTRODUCCIÓN E INSTALACIÓN

La herramienta Silverlight de Microsoft es una implementación multiplataforma para distintos servidores de .NET Framework con la que se generan aplicaciones web multimedia interactivas. Silverlight unifica las funciones del servidor, la web y el escritorio utilizando simultáneamente código administrado, lenguajes dinámicos, programación declarativa y tradicional, así como la eficacia de Windows Presentation Foundation (WPF).

Silverlight permite crear aplicaciones de vanguardia con las siguientes características:

- Es una tecnología multiplataforma que se ejecuta en varios exploradores. Se ejecuta en todos los exploradores web conocidos, como Microsoft Internet Explorer, Mozilla Firefox, Apple Safari y Google Chrome, así como en Microsoft Windows y Apple Mac OS X.
- Es una descarga pequeña que se instala en pocos segundos.
- Transmite los contenidos de audio y vídeo por secuencias. Ajusta el contenido de vídeo a todo tipo de calidades, desde dispositivos móviles hasta exploradores de escritorio y modos de vídeo HDTV de 720p.
- Incluye gráficos atractivos que los usuarios pueden manipular (arrastrar, girar y acercar o alejar) directamente en el explorador.

- Lee datos y actualiza la pantalla, pero no interrumpe al usuario al actualizar la página completa.
- La aplicación se puede ejecutar en el explorador web o se puede configurar de modo que los usuarios puedan ejecutarla en su equipo (ejecución fuera del explorador).

Silverlight combina varias tecnologías en una sola plataforma de desarrollo que permite seleccionar las herramientas y el lenguaje de programación apropiados según las necesidades del usuario. Silverlight ofrece las características siguientes:

- WPF y XAML. Silverlight incluye un subconjunto de la tecnología Windows Presentation Foundation (WPF), que extiende en gran medida los elementos en el explorador para crear la interfaz de usuario. Silverlight permite crear gráficos, animaciones y elementos multimedia fascinantes, así como otras características de cliente enriquecidas, extendiendo la interfaz de usuario basada en explorador más allá de lo que está disponible únicamente con HTML. El lenguaje XAML proporciona una sintaxis de marcado declarativa para crear elementos.
- Extensiones a JavaScript. Silverlight proporciona extensiones al lenguaje de scripting de explorador universal que permiten controlar la interfaz de usuario del explorador, incluida la capacidad para trabajar con elementos WPF.
- Compatibilidad con varios exploradores y plataformas. Silverlight se ejecuta de la misma manera en todos los exploradores conocidos (y en las plataformas conocidas). Es posible diseñar y desarrollar aplicaciones sin tener que preocuparse del explorador o de la plataforma de los usuarios.
- Integración con aplicaciones existentes. Silverlight se integra perfectamente con el código JavaScript y ASP.NET AJAX existente de modo que complementa la funcionalidad ya creada.
- Acceso al modelo de programación de .NET Framework. Es posible crear aplicaciones de Silverlight mediante lenguajes dinámicos, como IronPython, y lenguajes como C# y Visual Basic.
- Compatibilidad de herramientas. Se pueden utilizar herramientas de desarrollo, como Visual Studio y Expression Blend, para crear rápidamente aplicaciones de Silverlight.

- Compatibilidad de red. Silverlight incluye compatibilidad con HTTP sobre TCP. Se puede conectar a los servicios WCF, SOAP o ASP.NET AJAX y recibir datos XML, JSON o RSS. Además, es posible compilar clientes de multidifusión con Silverlight.
- LINQ. Silverlight incluye Language Integrated Query (LINQ), que permite programar el acceso a datos utilizando una sintaxis nativa intuitiva y objetos fuertemente tipados en los lenguajes de .NET Framework.

Para ejecutar una aplicación de Silverlight, los usuarios requieren un pequeño complemento en su explorador. Ese complemento es gratuito. Si los usuarios aún no tienen el complemento, aparecerá automáticamente un mensaje para indicarles que lo instalen. La descarga y la instalación solo tardan unos segundos y no requieren ninguna interacción por parte del usuario, excepto su permiso para llevar a cabo la instalación.

Silverlight garantiza que las aplicaciones se pueden ejecutar en todos los exploradores modernos, sin tener que crear código específico del explorador. Las aplicaciones de Silverlight se pueden ejecutar en el explorador o fuera del mismo.

Es posible crear aplicaciones de Silverlight mediante cualquier lenguaje compatible con .NET Framework (incluidos Visual Basic, C# y JavaScript). Visual Studio y Expression Blend permiten desarrollar aplicaciones de Silverlight. Con Silverlight, se pueden crear páginas web con elementos HTML y WPF. Al igual que HTML, XAML permite crear la interfaz de usuario de las aplicaciones basadas en web mediante una sintaxis declarativa, pero XAML proporciona elementos mucho más eficaces.

El contenido de este libro se adapta a las versiones 4 y 5 de Silverlight. Adicionalmente, en el capítulo primero se hace referencia también a los cuidados que hay que tener a la hora de trabajar con aplicaciones de Silverlight 3.

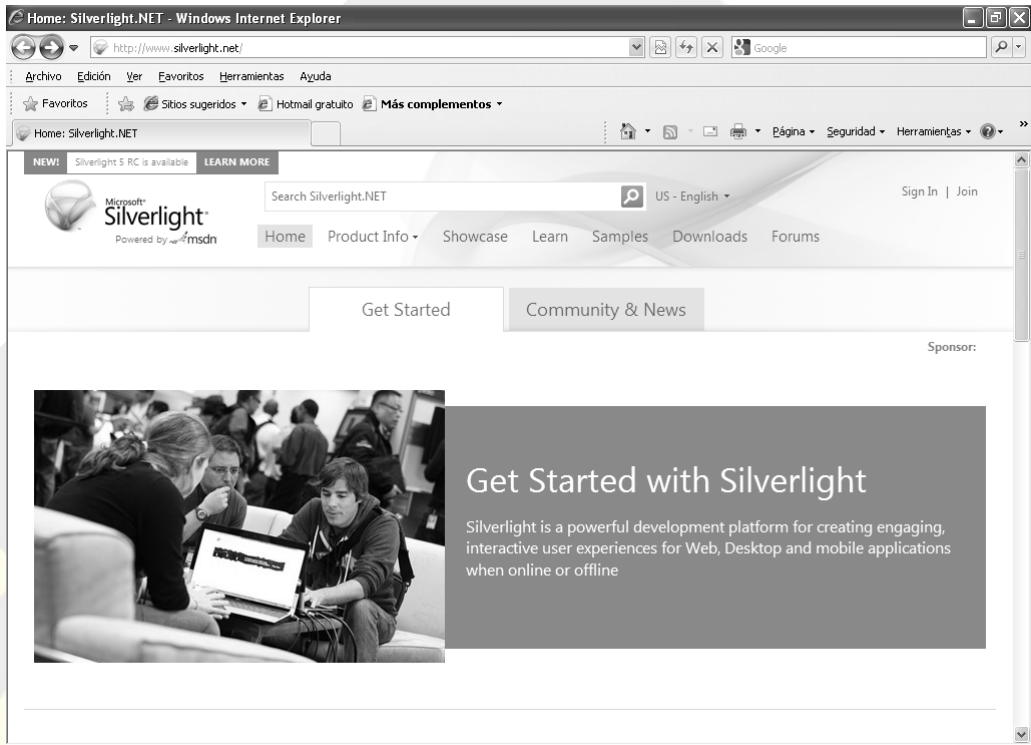
Habitualmente Silverlight se instala con el sistema operativo Windows en todas sus versiones. No obstante, Microsoft ofrece posteriormente la instalación de la aplicación y sus actualizaciones a través de Windows Update.

Asimismo, la instalación de Silverlight se puede realizar directamente desde la página web siguiente:

<http://www.silverlight.net>

El aspecto de la página se presenta en la figura representada a continuación:

MICROSOFT SILVERLIGHT EN ACCIÓN



Esta página web informa de las características del sistema necesarias para el correcto funcionamiento de Silverlight, permite comprobar si nuestro sistema cumple estas características y también permite actualizar Silverlight a la última versión disponible.

CARACTERÍSTICAS DE SILVERLIGHT

INTEGRAR SILVERLIGHT EN APLICACIONES

Una aplicación basada en Silverlight consiste en un contenido cargado por el complemento Silverlight en una página HTML. El complemento Silverlight puede rellenar la página HTML completa o simplemente parte del espacio. De forma predeterminada, un proyecto de Visual Studio permite al complemento ocupar el 100% del ancho y del alto de la página.

Si solamente desea usar Silverlight para parte de la aplicación, es posible que desee realizar llamadas en la página HTML desde el código de Silverlight, y viceversa.

XAML es un lenguaje de marcado declarativo que se puede usar para definir los elementos de la interfaz de usuario de la aplicación basada en Silverlight. Al crear un nuevo proyecto de Visual Studio, se crea automáticamente un archivo Page.xaml. En el archivo de XAML, puede crear objetos y definir sus propiedades mediante etiquetas y atributos XML

A continuación, se muestra una instrucción XAML simple que crea un rectángulo rojo.

```
<Rectangle Fill="Red" Width="150" Height="100"/>
```

Puede crear todas las interfaces de usuario en XAML, usar la superficie de diseño de Visual Studio o puede usar Microsoft Expression Blend para diseñar la aplicación. Expression Blend es una herramienta de diseñador con una superficie de diseño WYSIWYG que sirve para crear aplicaciones basadas en Silverlight. Asimismo, genera un archivo de XAML que es posible modificar directamente. También es posible enlazar eventos y escribir código subyacente con Expression Blend.

Al crear una aplicación basada en Silverlight, una de las primeras decisiones que tendrá que tomar será cómo organizar la interfaz de usuario. Silverlight proporciona 3 paneles de diseño al usuario, según indica la tabla siguiente:

Contenedor	Descripción
Canvas	Coloque los elementos secundarios en posiciones absolutas en el espacio x,y.
StackPanel	Coloque los elementos secundarios en posiciones relativas entre sí en pilas horizontales o verticales.
Grid	Coloque los elementos secundarios en filas y columnas.

El panel predeterminado es Grid, que es el panel de diseño más flexible y eficaz.

En el ejemplo siguiente, se coloca un elemento *Rectangle* en la celda 1,1 de una cuadrícula. La cuadrícula usa un índice de base cero, de modo que el rectángulo se muestra en la celda inferior derecha.

```
<Grid ShowGridLines="True">
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>

  <Rectangle Fill="Red" Width="150" Height="100"
  Grid.Column="1" Grid.Row="1"/>
</Grid>
```

Se generan unos resultados similares a la siguiente ilustración:



Controles

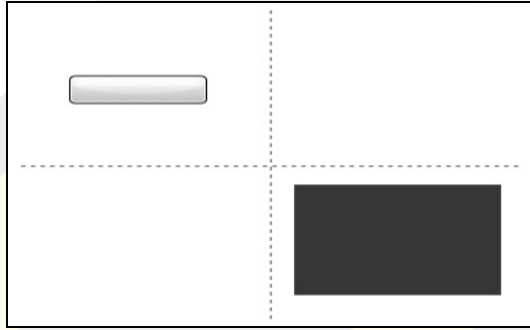
Los controles de Silverlight permiten hospedar contenido u otros controles y pueden diseñarse de modo que muestren los cambios de estado del usuario. Los controles varían en funcionalidad de elementos que permiten la interacción del usuario, como Button o TextBox, a elementos que admiten un diseño complejo de la información, como DataGrid.

Silverlight incluye plantillas predeterminadas para cada control que afectan al aspecto del control. No obstante, es posible crear plantillas personalizadas para cambiar el aspecto y el comportamiento visual de todos los controles.

En el ejemplo siguiente se toma el ejemplo anterior y se agrega un botón a la celda 0,0 de la cuadrícula.

```
<Grid ShowGridLines="True">
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>
  <Button Height="25" Width="100" Grid.Column="0"
Grid.Row="0"/>
  <Rectangle Fill="Red" Width="150" Height="100"
Grid.Column="1" Grid.Row="1"/>
</Grid>
```

Se generan unos resultados similares a la siguiente ilustración:



Agregar lógica de código

De forma predeterminada, el proyecto Visual Studio incluye un archivo de código, que también se denomina archivo de código subyacente. El archivo de código contiene uno de los lenguajes administrados admitidos por Silverlight a través de Common Language Runtime (CLR), como C# o Visual Basic.

Para el nombre del archivo de código subyacente se toma la cadena Page.xaml y se anexa el tipo de lenguaje (por ejemplo, Page.xaml.cs). El archivo de código subyacente es el lugar donde puede aplicarse lógica a los objetos de XAML. Es posible crear objetos de interfaz de usuario en el código y agregarlos al árbol de elementos visibles. Además, puede obtenerse acceso desde XAML a las clases creadas en el archivo de código subyacente (y en cualquier archivo de código incluido en el proyecto). Por ejemplo, puede definir sus propios controles y, a continuación, crear instancias de ellos en XAML.

En el ejemplo siguiente se agrega un controlador de eventos Click que cambiará el color del rectángulo a azul. En XAML, agregue el evento Click a Button y un atributo x:Name a Rectangle. x:Name permite hacer referencia al rectángulo en el archivo de código subyacente.

```
<Button Height="25" Width="100" Grid.Column="0" Grid.Row="0"
Click="Button_Click"/>
<Rectangle x:Name="rect1" Fill="Red" Width="150" Height="100"
Grid.Column="1" Grid.Row="1"/>
```

En el archivo de código subyacente, defina el controlador de eventos Click.

VB	C#	C++	F#	JScript
----	----	-----	----	---------

```

Private Sub Button_Click(ByVal sender As Object, ByVal e As RoutedEventArgs)
    rect1.Fill = New SolidColorBrush(Colors.Blue)
End Sub

```

Lenguajes dinámicos

Silverlight Tools para Visual Studio incluye Dynamic Language Runtime (DLR), que permite a los usuarios de lenguajes dinámicos, como Python y Ruby, escribir aplicaciones basadas en Silverlight. Los lenguajes dinámicos se empaquetan como código fuente, no se compilan en ensamblados, y el código puede generarse y compilarse en tiempo de ejecución. Resultan apropiados para un estilo de desarrollo flexible e interactivo. Silverlight incluye tres lenguajes dinámicos: IronPython, IronRuby y JScript administrado.

Los lenguajes dinámicos proporcionan un enfoque interactivo para el desarrollo de aplicaciones basadas en Silverlight. El DLR (Dynamic language runtime) permite que los lenguajes dinámicos se ejecuten en el Common Language Runtime y utilicen la biblioteca de clases de .NET Framework para Silverlight. El DLR y tres lenguajes dinámicos (IronPython, IronRuby y JScript administrado) están actualmente en desarrollo.

Gráficos

Silverlight proporciona muchas opciones para agregar características visuales interesantes a la aplicación. Puede usar dibujos, formas, rutas y geometrías complejas. Las áreas definidas por geometrías pueden rellenarse con efectos, como imágenes, degradados de color o clips de vídeo, mediante el uso de pinceles.

Puede transformar gráficos y otros objetos en la pantalla (por ejemplo, girarlos o inclinarlos). Incluso puede aplicar efectos 3D.

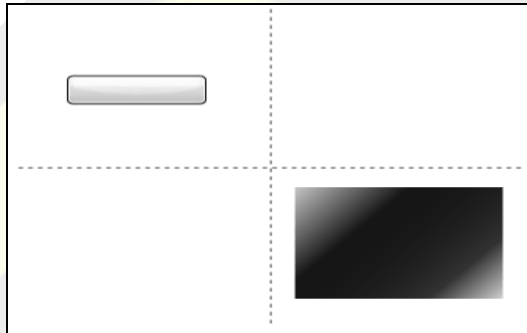
En el ejemplo siguiente el rectángulo se rellena con un pincel de degradado lineal.

```

<Rectangle x:Name="rect1" Width="150" Height="100"
Grid.Column="1" Grid.Row="1">
    <Rectangle.Fill>
        <LinearGradientBrush>
            <GradientStop Offset="0" Color="LightBlue"/>
            <GradientStop Offset="0.4" Color="Blue"/>
            <GradientStop Offset="0.8" Color="Purple"/>
            <GradientStop Offset="1.0" Color="Lavender"/>
        </LinearGradientBrush>
    </Rectangle.Fill>
</Rectangle>

```

Se generan unos resultados similares a la siguiente ilustración.



Multimedia y animación

Además de gráficos estáticos, puede agregar animaciones, audio y vídeo a la aplicación para hacerla más dinámica e interactiva.

En el ejemplo siguiente se aumenta y se reduce el tamaño del rectángulo del ejemplo anterior hasta que se hace clic en el botón **Detener**.

```
<Grid ShowGridLines="True">
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition/>
  </Grid.RowDefinitions>

  <Grid.Resources>
    <Storyboard x:Name="AnimateRectangle">
      <DoubleAnimation Storyboard.TargetName="rect1"
        Storyboard.TargetProperty="Height"
        From="0" To="100"
        AutoReverse="True" Duration="0:0:02"
        RepeatBehavior="Forever"/>
      <DoubleAnimation Storyboard.TargetName="rect1"
        Storyboard.TargetProperty="Width"
        From="0" To="100"
        AutoReverse="True" Duration="0:0:04"
        RepeatBehavior="Forever"/>
    </Storyboard>
  </Grid.Resources>
  <Image x:Name="rect1" Width="100" Height="100" Source="black.png" />
  <Button x:Name="Detener" Content="Detener" />
</Grid>
```

```

RepeatBehavior="Forever"/>
</Storyboard>
</Grid.Resources>

<Button Height="25" Width="100"
        Grid.Column="0" Grid.Row="0"
        Content="Stop"
        Click="Button_Click"/>

<Rectangle x:Name="rect1" Width="150" Height="100"
Grid.Column="1" Grid.Row="1" Loaded="OnLoaded">
    <Rectangle.Fill>
        <LinearGradientBrush>
            <GradientStop Offset="0" Color="LightBlue"/>
            <GradientStop Offset="0.4" Color="Blue"/>
            <GradientStop Offset="0.8" Color="Purple"/>
            <GradientStop Offset="1.0" Color="Lavender"/>
        </LinearGradientBrush>
    </Rectangle.Fill>
</Rectangle>

<Button Height="25" Width="100"
        Grid.Column="0" Grid.Row="1"
        Content="Start"
        Click="Button_Click_1"/>
</Grid>

```

VB C# C++ F# JScript

```

Private Sub OnLoaded(ByVal sender As Object, ByVal e As RoutedEventArgs)
    AnimateRectangle.Begin()
End Sub

Private Sub Button_Click(ByVal sender As Object, ByVal e As RoutedEventArgs)
    AnimateRectangle.Pause()
End Sub

Private Sub Button_Click_1(ByVal sender As Object, ByVal e As RoutedEventArgs)
    AnimateRectangle.[Resume]()
End Sub

```

Data, redes y ejecución de aplicaciones fuera del explorador

Muchas aplicaciones basadas en Silverlight trabajan con datos. Puede mostrar conjuntos de datos utilizando controles como DataGrid y ListBox. Para rellenar la interfaz de usuario, puede usar Enlace de datos. Si enlaza la interfaz de usuario al objeto de datos, las actualizaciones del objeto se propagarán automáticamente a la interfaz de usuario.

Los datos incluidos en la aplicación pueden proceder de varios orígenes, como una fuente RSS, pero suelen estar en formato XML. Silverlight incluye XmlReader y LINQ para analizar los datos XML. LINQ resulta más adecuado cuando se analizan fragmentos de datos más pequeños.

Silverlight proporciona varias características para comunicarse en la nube. La clase WebClient controla la descarga de contenido en el cliente. También puede usar WebClient para enviar y recibir mensajes XML sin formato. Las aplicaciones basadas en Silverlight también pueden obtener acceso a servicios web, como Windows Communication Foundation (WCF), SOAP y ASP.NET AJAX.

Además de distribuir la aplicación basada en Silverlight en línea, también puede configurarla para que los usuarios puedan instalarla desde un sitio web y, después, ejecutarla en su equipo fuera del explorador web.

ARQUITECTURA DE SILVERLIGHT

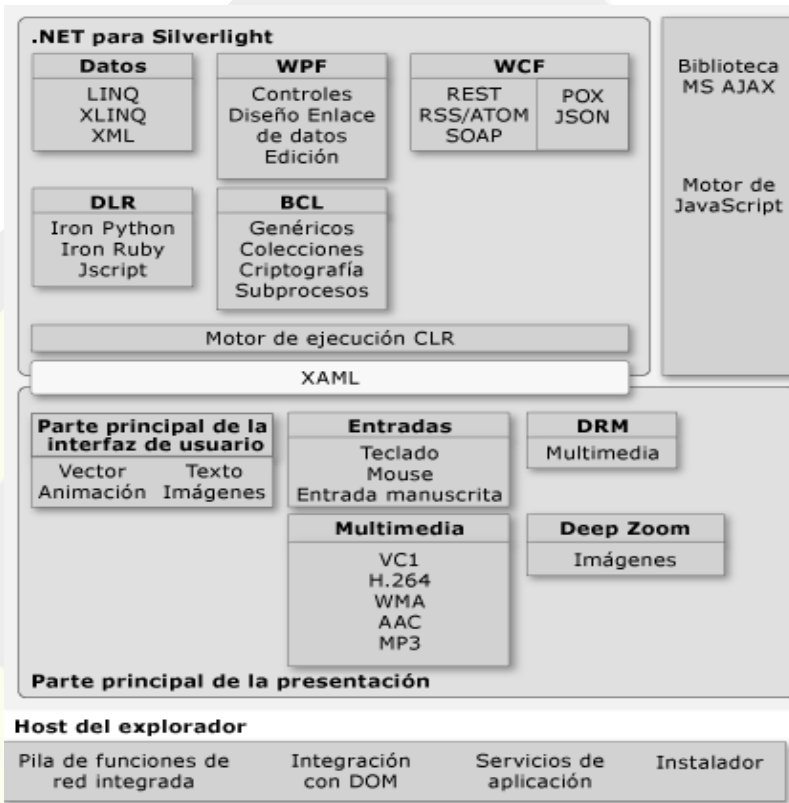
Silverlight es no solo un lienzo atractivo donde mostrar a los usuarios finales contenidos web y multimedia enriquecidos e interactivos. También es una plataforma eficaz y, a la vez, ligera, donde desarrollar aplicaciones portátiles, multiplataforma en red que integran datos y servicios de numerosos orígenes. Además, Silverlight permite generar interfaces de usuario que mejorarán de manera significativa la experiencia del usuario final típico, en comparación con las aplicaciones web tradicionales.

Aunque Silverlight parece simple y de tamaño compacto cuando se considera como un entorno en tiempo de ejecución del lado cliente, la plataforma de desarrollo de Silverlight integra diversas características y tecnologías complejas, que pone a disposición de los programadores. Los programadores necesitan conocimientos prácticos de la arquitectura de la plataforma para poder crear aplicaciones efectivas basadas en Silverlight.

La plataforma de Silverlight en su conjunto se compone de dos partes principales, además de un componente de instalador y actualización, tal y como se describe en la siguiente tabla.

Componente	Descripción
Marco de trabajo de presentación básico	Componentes y servicios orientados a la interfaz de usuario y la interacción con el usuario, incluidos los datos proporcionados por el usuario, controles de interfaz de usuario ligeros para su uso en las aplicaciones web, reproducción de elementos multimedia, administración de derechos digitales, enlaces de datos; y características de presentación, incluidos gráficos vectoriales, texto, animaciones e imágenes. También incluye el lenguaje XAML para especificar el diseño.
.NET Framework para Silverlight	<p>Subconjunto de .NET Framework que contiene componentes y bibliotecas, que incluyen integración de datos, controles de Windows extensibles, funciones de red, bibliotecas de clases base, recolección de elementos no utilizados y CLR (Common Language Runtime).</p> <p>Algunas partes de .NET Framework para Silverlight se implementan con la aplicación. Estas "bibliotecas de Silverlight" son ensamblados que no se incluyen en el motor en tiempo de ejecución de Silverlight, sino que se distribuyen en el SDK de Silverlight. Cuando se utilizan bibliotecas de Silverlight en la aplicación, se empaquetan con esta última y se descargan al explorador. Incluyen las de nuevos controles de interfaz de usuario, XLINQ, distribución (RSS/Atom), serialización XML y DLR (Dynamic Language Runtime).</p>
Componente de instalador y actualización	Control de instalación y actualización que simplifica el proceso de instalar la aplicación para los usuarios noveles y, a continuación, proporciona actualizaciones automáticas de bajo impacto.

En la ilustración siguiente se muestran estos componentes de la arquitectura de Silverlight, junto con los componentes y servicios relacionados.



El conjunto combinado de herramientas, tecnologías y servicios que se incluyen en la plataforma de Silverlight encierra un valor concreto: facilita a los programadores la tarea de crear aplicaciones enriquecidas e interactivas en red. Aunque no cabe duda de que es posible generar este tipo de aplicaciones mediante las herramientas y tecnologías web de hoy, el trabajo de los programadores se ve ralentizado por las numerosas dificultades técnicas, tales como plataformas incompatibles, protocolos y formatos de archivo dispares, y diversidad de exploradores web que presentan las páginas y administran los scripts de manera diferente. Una aplicación web enriquecida que se ejecuta perfectamente en una combinación de sistema y explorador puede funcionar de manera muy distinta, e incluso no funcionar en absoluto, en otro sistema u otro explorador. Con la gran variedad actual de herramientas, protocolos y tecnologías, constituye un esfuerzo titánico y, con frecuencia, económicamente prohibitivo, generar una aplicación que puede proporcionar al mismo tiempo las ventajas siguientes:

- Capacidad para crear la misma experiencia del usuario en la totalidad de exploradores y plataformas, de tal forma que la aplicación presente la misma apariencia y el mismo funcionamiento en todos ellos.
- Integración de datos y servicios procedentes de diversas ubicaciones de red en una misma aplicación mediante las clases y funcionalidades conocidas de .NET Framework.
- Una interfaz de usuario rica en elementos multimedia, atractiva y accesible.
- Silverlight facilita a los desarrolladores la labor de crear este tipo de aplicaciones, porque supera muchas de las incompatibilidades actuales entre las tecnologías y proporciona, dentro de una misma plataforma, las herramientas necesarias para crear aplicaciones enriquecidas, multiplataforma e integradas.

Componentes de presentación básicos

Las características de presentación básicas de la plataforma Silverlight, mostradas en la sección e ilustración anteriores, se describen en la tabla siguiente.

Característica	Descripción
Entrada	Administra datos de entrada procedentes de distintos dispositivos de hardware, como los de dibujo, el teclado y el ratón, y otros.
Representación de la interfaz de usuario	Representa gráficos vectoriales y de mapa de bits, animaciones y texto.
Multimedia	Permite la reproducción y administración de varios tipos de archivos de audio y vídeo, como los archivos .WMP y .MP3.
Deep Zoom	Permite acercar imágenes de alta resolución y realizar un movimiento panorámico alrededor de las mismas.
Controles	Admite controles extensibles que se pueden personalizar aplicando estilos y plantillas.
Diseño	Permite la colocación dinámica de los elementos de la interfaz de usuario.
Enlace de datos	Permite la vinculación de objetos de datos y elementos de la interfaz de usuario.
DRM	Permite la administración de derechos digitales de los recursos multimedia.
XAML	Proporciona un analizador para el marcado XAML.

Los programadores pueden interactuar con este marco de trabajo de presentación utilizando XAML para especificar los detalles de presentación. XAML constituye el punto primario de interacción entre .NET Framework y la capa de la presentación. Los programadores pueden manipular la capa de presentación mediante programación con código administrado.

.NET Framework para Silverlight

En la tabla siguiente se describe una lista parcial de las características de .NET Framework para Silverlight mostradas en la ilustración anterior.

Característica	Descripción
Data	Admite las características de LINQ (Language-Integrated Query) y de LINQ to XML, que facilitan el proceso de integrar datos procedentes de orígenes dispares y trabajar con ellos. También admite el uso de las clases de serialización y XML para administrar los datos.
Biblioteca de clases base	Un conjunto de bibliotecas de .NET Framework que proporcionan las funciones de programación esenciales, como la administración de cadenas, expresiones regulares, entrada y salida, reflexión, colecciones y globalización.
Windows Communication Foundation (WCF)	Proporciona características para simplificar el acceso a los servicios y datos remotos. Esto incluye un objeto de explorador, un objeto de solicitud y respuesta HTTP, compatibilidad con solicitudes HTTP entre dominios, compatibilidad con fuentes de distribución RSS/Atom así como compatibilidad con los servicios JSON, POX y SOAP.
CLR (Common Language Runtime)	Proporciona administración de memoria, recolección de elementos no utilizados, comprobación de seguridad de tipos y control de excepciones.
Controles de WPF (Windows Presentation Foundation)	Proporciona un conjunto enriquecido de controles, como son Button, Calendar, CheckBox, DataGrid, DatePicker, HyperlinkButton, ListBox, RadioButton y ScrollViewer.
DLR (Dynamic Language Runtime)	Admite la compilación y ejecución dinámicas de lenguajes de scripting como JavaScript y IronPython para programar aplicaciones basadas en Silverlight. Incluye un modelo conectable que aporta compatibilidad con otros lenguajes para su uso con Silverlight.

.NET Framework para Silverlight es un subconjunto de la plataforma .NET Framework completa. Proporciona los fundamentos del desarrollo robusto y orientado a objetos para tipos de aplicaciones (como las de Internet) que tradicionalmente no contaban con este tipo de recursos.

Los desarrolladores pueden interactuar con la capa de .NET Framework para Silverlight escribiendo código administrado en C# y Visual Basic. Los desarrolladores de .NET Framework también pueden tener acceso a la capa de presentación creando sus aplicaciones en Visual Studio o Microsoft Expression Blend.

Características de programación adicionales de Silverlight

Silverlight proporciona varias características adicionales que ayudan a los programadores a crear aplicaciones enriquecidas e interactivas, incluidas las descritas en la tabla siguiente.

Característica	Descripción
Almacenamiento aislado	Proporciona acceso seguro del cliente de Silverlight al sistema de archivos del equipo local. Permite el almacenamiento local y el almacenamiento en caché de datos aislados para un usuario determinado.
Programación asincrónica	Un subproceso de trabajo de segundo plano lleva a cabo tareas de programación, mientras la aplicación queda libre para interactuar con el usuario.
Administración de archivos	Proporciona un cuadro de diálogo Abrir archivo seguro, a fin de facilitar el proceso de crear cargas de archivo seguras.
Interacción entre HTML y código administrado	Permite a los programadores de .NET Framework manipular directamente los elementos de la interfaz de usuario en el DOM HTML de una página web. Los programadores web también pueden utilizar JavaScript para efectuar llamadas directas al código administrado y tener acceso a los objetos, propiedades, eventos y métodos que admiten el uso de scripts.
Serialización	Proporciona soporte técnico para la serialización de los tipos CLR a JSON y XML.
Empaquetar	Proporciona la clase Application y herramientas de compilación para crear paquetes .xap. El paquete .xap contiene la aplicación y el punto de entrada para que se ejecute el control del complemento Silverlight.

Bibliotecas XML	Las clases XmlWriter y XmlReader simplifican el trabajo con los datos XML de los servicios Web. La característica X.Linq permite a los desarrolladores consultar los datos XML directamente en los lenguajes de programación de .NET Framework.
-----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Herramientas y tecnologías relacionadas

Las aplicaciones de Microsoft siguientes incluyen características especiales para el desarrollo en Silverlight:

- Microsoft Expression Blend. Esta herramienta se puede utilizar para crear y modificar la capa de presentación de una aplicación; para ello, permite manipular el lienzo y los controles XAML, trabajar con gráficos y programar la capa de presentación mediante un lenguaje dinámico, como JavaScript.
- Visual Studio. Visual Studio 2010 proporciona compatibilidad de diseño visual para crear aplicaciones de Silverlight. Para construir las interfaces de usuario de las aplicaciones, arrastre los controles del **Cuadro de herramientas** y establezca las propiedades en la ventana **Propiedades**. También puede editar directamente XAML en la vista XAML.

Dado que las aplicaciones basadas en Silverlight se ejecutan en un entorno de tiempo de ejecución en el equipo cliente, no es necesario que haya ninguna aplicación concreta instalada en el servidor. Sin embargo, puede que los programadores observen que su capacidad de crear aplicaciones enriquecidas que integran en el servidor servicios y datos procedentes de diversos orígenes mejora con la integración de los tipos siguientes de servicios y aplicaciones de servidor en sus aplicaciones basadas en Silverlight:

- ASP.NET AJAX. Incluye un conjunto de controles, servicios y bibliotecas para crear aplicaciones ricas e interactivas basadas en web.
- Servicios de Microsoft Windows Communication Foundation (WCF).
- Servidores de Internet, incluidos Microsoft Internet Information Services (IIS) y el servidor web Apache.

- Aplicaciones y servicios basados en Internet, como Microsoft ASP.NET, PHP, servicios de transmisión multimedia por secuencias de Windows, servicios de Windows Live y otros servicios web abiertos.

CAMBIOS EN EL RUNTIME Y LAS HERRAMIENTAS DE SILVERLIGHT 3 A LAS NUEVAS VERSIONES 4 y 5.

De Silverlight 3 a las nuevas versiones, se han realizado cambios en el runtime de Silverlight y Silverlight Tools. Para estos cambios, se aplican los siguientes principios:

- La mayoría de las aplicaciones de Silverlight 3 funcionarán con Silverlight 4 y 5 sin ningún cambio.
- Cuando se requieren cambios importantes, Silverlight intentará mantener la compatibilidad con el comportamiento anterior y con el comportamiento nuevo mediante un *modo quirks*.

No obstante, es posible que algunos cambios realizados en los componentes de Silverlight generen un error en las aplicaciones basadas en Silverlight anteriores (tiempo de compilación, tiempo de carga de XAML o posiblemente tiempo de diseño) o hagan que dichas aplicaciones tengan un comportamiento distinto.

Cambios en el analizador de XAML

Silverlight 4 y 5 incluyen un analizador de XAML actualizado. Para mantener la compatibilidad, el runtime de las nuevas versiones también incluye el analizador de XAML original de Silverlight 3 y lo utiliza para analizar el código XAML de cualquier aplicación que tenga como destino Silverlight 3. Esto significa que si el destino sigue siendo Silverlight 3, cualquier comportamiento de XAML original que haya funcionado para Silverlight 3 seguirá funcionando, siempre y cuando el comportamiento de XAML no interfiera con los cambios importantes específicos que se describen en la sección "Cambios importantes en la actualización". Esto tiene un resultado similar a un modo quirks para la carga de XAML.

Si se actualiza un proyecto de Silverlight que tiene como destino Silverlight 3 de modo que su destino pase a ser Silverlight 4 o 5, es posible que haya que realizar cambios en el código XAML.

Cambios en modo *quirks*

El equipo de Silverlight tenía la intención de corregir varios errores de Silverlight 3 en Silverlight 4 y 5. Sin embargo, al corregir algunos de estos errores, es posible que se interrumpan algunas aplicaciones de Silverlight 3. Para evitarlo, el equipo de Silverlight ha resuelto estos cambios potencialmente problemáticos creando un "*modo quirks*" para el comportamiento en tiempo de ejecución. Un cambio en modo *quirks* representa el caso en que el *runtime* de Silverlight 4 o 5 bifurca su comportamiento si detecta que la aplicación tiene como destino Silverlight 3. De este modo, Silverlight pasa a ser "compatible con el error" como runtime. Sin embargo, es posible que haya que revisar el comportamiento en modo *quirks* si se recompila la aplicación para Silverlight 4 o 5.

Si el valor de `RuntimeVersion` de una aplicación es Silverlight 3, el runtime se ejecutará en modo *quirks*. En modo *quirks*, el runtime recurre al comportamiento de Silverlight 3 asociado a los cambios en modo *quirks* y, en estos casos, se comporta como si fuese literalmente un runtime de Silverlight 3 no actualizado. El runtime de las nuevas versiones de Silverlight detecta si la aplicación está diseñada para Silverlight 3 a través de la propiedad `RuntimeVersion`. El valor de `RuntimeVersion` se establece como un atributo en el manifiesto `AppManifest.xaml` del archivo XAP:

```
<Deployment
xmlns="http://schemas.microsoft.com/client/2007/deployment"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
EntryPointAssembly="ElevatorSilverlight"
    EntryPointType="ElevatorSilverlight.App"
    RuntimeVersion="3.0.31005.0">
    <Deployment.Parts>
        <AssemblyPart
                                x:Name="ElevatorSilverlight"
Source="ElevatorSilverlight.dll" />
    </Deployment.Parts>
</Deployment>
```

`RuntimeVersion` refleja la versión de compilación de Silverlight que había en el equipo del desarrollador cuando se compiló la aplicación.

Si se compila la aplicación con las herramientas de Silverlight 4 o 5 y se elige específicamente Silverlight 3 como destino, se podrá invocar el modo *quirks* como si se hubiera compilado la aplicación con las herramientas de Silverlight 3.

COMMON LANGUAGE RUNTIME

Common Language Runtime es la base de .NET Framework. Es responsable de administrar la ejecución del código en tiempo de ejecución y proporciona servicios básicos como compilación, administración de memoria y de subprocesos, ejecución de código, cumplimiento de seguridad de tipos y comprobación de la seguridad de código. Los compiladores tienen como destino Common Language Runtime, que define los tipos de datos básicos disponibles para los programadores de aplicaciones. Dado que proporciona un entorno administrado para la ejecución de código, Common Language Runtime mejora la productividad del programador y contribuye al desarrollo de aplicaciones sólidas.

A continuación, se proporciona información general sobre las características siguientes de Common Language Runtime:

- Administración de la memoria.
- Sistema de tipos comunes.

Administración de la memoria

El recolector de elementos no utilizados de Common Language Runtime administra la asignación y liberación de la memoria de una aplicación. Esto significa que los programadores no tienen que escribir código para realizar tareas de administración de memoria al programar aplicaciones administradas. La administración automática de la memoria puede eliminar problemas frecuentes, como olvidar liberar un objeto y causar una pérdida de memoria, o intentar tener acceso a la memoria de un objeto que ya se ha liberado.

Cuando se inicializa un nuevo proceso, el motor en tiempo de ejecución reserva una región contigua de espacio de direcciones para el proceso. Este espacio de direcciones reservado se denomina montón administrado. El montón administrado mantiene un puntero a la dirección a la que se asignará el siguiente objeto del montón. Inicialmente, este puntero se establece en la dirección base del montón administrado. Todos los tipos de referencia se asignan en el montón administrado. Cuando una aplicación crea el primer tipo de referencia, se le asigna memoria en la dirección base del montón administrado. Cuando la aplicación crea el siguiente objeto, el recolector de elementos no utilizados le asigna memoria en el espacio de direcciones que sigue inmediatamente al primer objeto. Siempre que haya espacio de direcciones disponible, el recolector de elementos no utilizados continúa asignando espacio a los objetos nuevos de este modo.

La asignación de memoria desde el montón administrado es más rápida que la asignación de memoria no administrada. Como el tiempo de ejecución asigna memoria a los objetos agregando un valor a un puntero, este método es casi tan rápido como la asignación de memoria desde la pila. Además, puesto que los nuevos objetos que se asignan consecutivamente se almacenan uno junto a otro en el montón administrado, la aplicación puede tener un acceso muy rápido a los objetos.

Además de asignar memoria, el motor de optimización del recolector de elementos no utilizados determina cuál es el mejor momento para realizar una recolección, según las asignaciones que se estén realizando. Cuando el recolector de elementos no utilizados lleva a cabo una recolección, libera la memoria de los objetos que ya no usa la aplicación.

Sistema de tipos comunes

Como es el motor en tiempo de ejecución, en lugar de un compilador del lenguaje individual, el que define los tipos base disponibles, la productividad del programador mejora. Los programadores pueden crear aplicaciones en el lenguaje que prefieran y seguir sacando todo el provecho del motor en tiempo de ejecución, la biblioteca de clases y los componentes escritos en otros lenguajes por otros colegas.

Siempre que el compilador de lenguaje tenga como destino .NET Framework y Common Language Runtime, se puede tener acceso a los componentes desarrollados con dicho compilador desde aplicaciones desarrolladas en otros lenguajes. El sistema de tipos comunes ayuda a comprender el objetivo de independencia del lenguaje; los programadores se pueden centrar en desarrollar una aplicación en su lenguaje de opción, y utilizar bibliotecas y componentes sin tener en cuenta el lenguaje en el que se escribieron.

El sistema de tipos comunes admite dos categorías generales de tipos de datos:

- *Tipos de valor.* Los tipos de valor contienen directamente sus datos y las instancias de los tipos de valor se asignan en la pila o se insertan en una estructura. Los tipos de valor pueden ser integrados (implementados por el motor en tiempo de ejecución), definidos por el usuario o enumeraciones.
- *Tipos de referencia.* Los tipos de referencia guardan una referencia a la dirección en memoria del valor y se asignan en el montón. Los tipos de referencia pueden ser tipos autodescriptivos, de puntero o de interfaz. El tipo

de un tipo de referencia se puede determinar a partir de los valores de los tipos autodescriptivos. Los tipos autodescriptivos se dividen en matrices y tipos de clase. Los tipos de clase son clases definidas por el usuario, tipos de valor a los que se ha aplicado la conversión boxing y delegados.

Las variables que son tipos de valor tienen, cada una, su propia copia de los datos y, por lo tanto, las operaciones en una variable no afectan a las demás. Las variables que son tipos de referencia pueden hacer referencia al mismo objeto y, por lo tanto, las operaciones en una variable pueden afectar al mismo objeto al que hace referencia otra variable. Salvo las interfaces, todos los tipos se derivan del tipo base System.Object.

HERRAMIENTAS DE SILVERLIGHT

En este apartado se describen varias herramientas que puede usar cuando desarrolle aplicaciones para Silverlight.

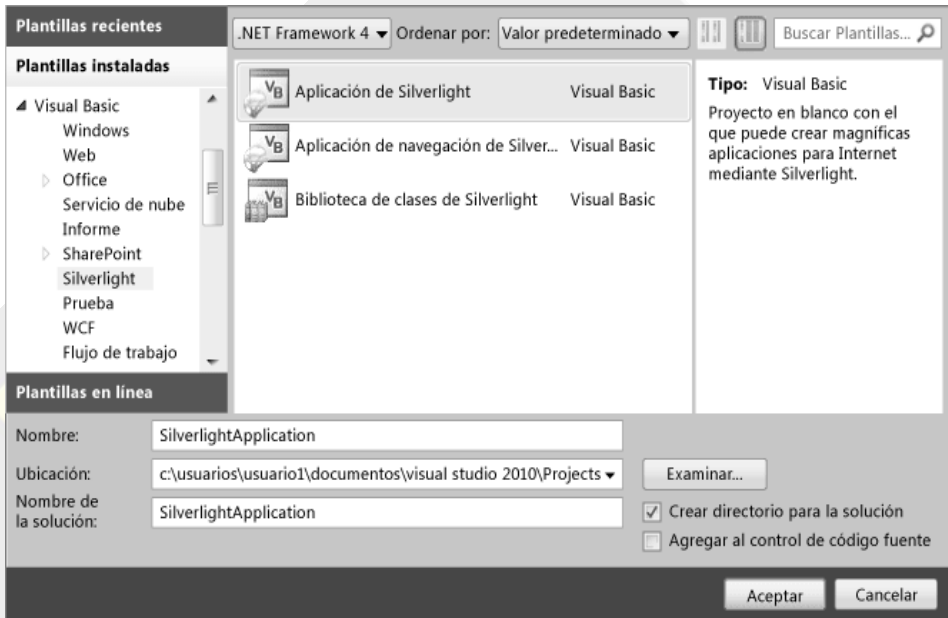
Instalar Silverlight Tools para Visual Studio 2010

Si no tiene experiencia con el desarrollo en Silverlight , la manera más rápida de ponerse al día es instalar Visual Studio 2010 o Visual Web Developer 2010 Express e instalar Silverlight Tools para Visual Studio 2010.

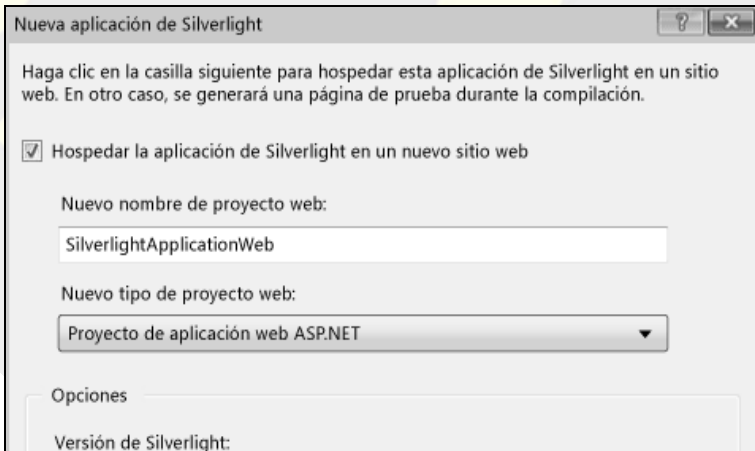
Para instalar Silverlight Tools para Visual Studio 2010, se tendrá en cuenta lo siguiente:

1. Instale Visual Studio 2010 o Visual Web Developer 2010 Express.
2. Instale Silverlight Tools para Visual Studio 2010.

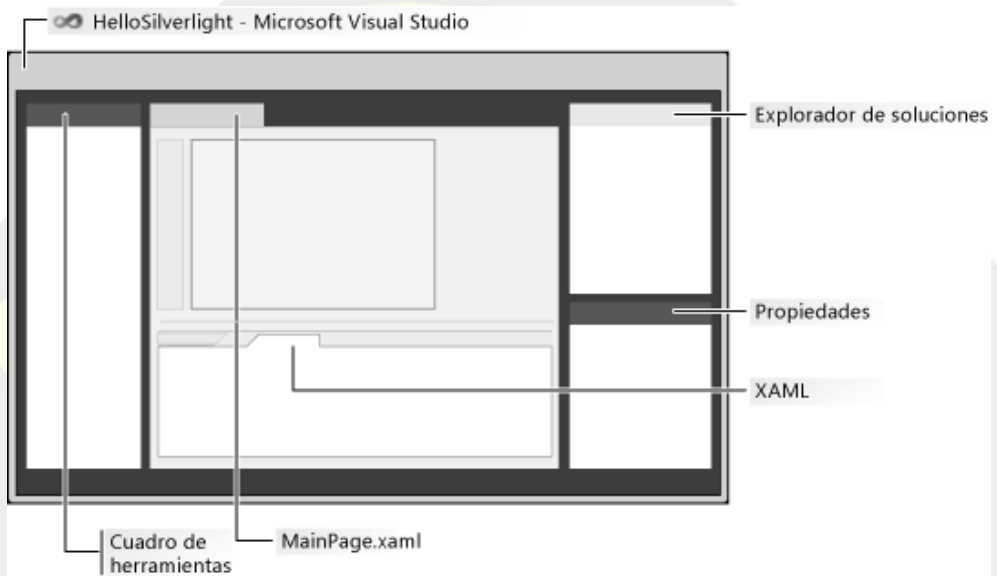
En la siguiente figura se muestra un ejemplo de las plantillas de proyecto de Silverlight.



Al instalar Silverlight Tools para Visual Studio 2010, aparecerá la opción para elegir Silverlight como destino. En la siguiente ilustración se muestra la opción Silverlight en el cuadro de diálogo *Nueva aplicación de Silverlight*.



Al crear un nuevo proyecto de Silverlight, verá un gran número de las herramientas y ventanas con las que ya se ha familiarizado en Visual Studio.



Silverlight Tools para Visual Studio 2010

Silverlight Tools for Visual Studio 2010 RC es un complemento para Visual Studio 2010 o Visual Web Developer 2010 Express que proporciona varias características para crear aplicaciones en Silverlight. Cuando se instala Silverlight Tools, se instalan los componentes siguientes.

- Silverlight developer runtime.
- SDK de Silverlight.
- Actualizaciones a Visual Studio para admitir las características de Silverlight.
- Tema sobre Servicios RIA de WCF.

A diferencia del runtime de Silverlight que instalan los usuarios finales, el runtime para desarrolladores que se instala con Silverlight Tools incluye mensajes de error que pueden ser útiles para depurar.

SDK de Silverlight

El kit de desarrollo de software (SDK) de Silverlight incluye varios componentes que ayudan a crear aplicaciones de Silverlight. El SDK de Silverlight se incluye en la instalación de Silverlight Tools, pero también lo puede instalar por separado. De forma predeterminada, el SDK de Silverlight se instala en la carpeta %ProgramFiles%\Microsoft SDKs\Silverlight. El SDK de Silverlight contiene los elementos siguientes.

- Licencia de usuario. El documento de licencia incluye las condiciones del SDK de Silverlight .
- Bibliotecas de Silverlight. El SDK de Silverlight incluye bibliotecas de cliente y servidor. Las bibliotecas de Silverlight no forman parte del complemento Silverlight. Las bibliotecas de cliente incluyen las extensiones y los controles de usuario de Silverlight. Las bibliotecas de servidor incluyen los controles de servidor de Silverlight. Para conocer las condiciones de uso, consulte la licencia de usuario del SDK de Silverlight .
- Herramientas. Incluye ensamblados de referencia, componentes de compilación y herramientas para compilar y empaquetar aplicaciones de Silverlight.

Otras herramientas de Silverlight

A medida que avanza con el desarrollo en Silverlight, es posible que le resulten útiles otras herramientas. En la tabla siguiente se indican algunas de ellas.

Herramienta	Descripción
Expression Blend	Expression Blend es una herramienta para diseñadores (y programadores) que permite crear gráficos, diseñar animaciones y generar experiencias de usuario. Expression Blend utiliza XAML y el mismo sistema de proyectos que Visual Studio. Esto permite que diseñadores y programadores compartan los mismos archivos y colaboren.

Deep Zoom Composer	Deep Zoom Composer es una herramienta que permite preparar imágenes de alta resolución para usarlas con la característica Deep Zoom. La nueva tecnología Deep Zoom de Silverlight permite al usuario acercar y alejar las imágenes de manera fluida. Deep Zoom Composer permite al usuario crear archivos de composición Deep Zoom que controlan la experiencia de aplicar el zoom y, luego, exportar los archivos necesarios para su implementación con Silverlight.
Silverlight.js	Silverlight.js es un archivo auxiliar de JavaScript que se proporciona en el SDK de Silverlight. Puede llamar a las funciones definidas en este archivo para inicializar instancias del complemento Silverlight en una página web. Silverlight.js también contiene las funciones de utilidad para determinar la versión instalada del cliente del complemento.
Silverlight.supportedUserAgent.js	Este archivo de aplicación auxiliar de JavaScript opcional incluye la función Silverlight.supportedUserAgent, que determina si el explorador del usuario admite Silverlight.