

Argentina \$8,90 (recargo al interior \$0,20) / México: \$45

USERS

Microsoft®

Curso teórico y práctico de programación

Desarrollador .net

Con toda la potencia
de **Visual Basic .NET** y **C#**

La mejor forma de aprender
a programar desde cero



Basado en el programa
Desarrollador Cinco Estrellas
de Microsoft

18

.NET 3.0

Componentes adicionales
/ XAML / Expression / Windows
Presentation Foundation / Windows
Communication Foundation /
Windows Workflow Foundation



ISBN 978-987-1347-43-8



9 789871 347438



RedUSERS

COMUNIDAD DE TECNOLOGIA



EL SITIO Nº1 DE TECNOLOGIA

Noticias al instante // Entrevistas y coberturas exclusivas //
Análisis y opinión de los máximos referentes // Reviews de
productos // Trucos para mejorar la productividad //
Regístrate, participa, y comparte tus opiniones



SUSCRIBITE

SIN CARGO A CUALQUIERA
DE NUESTROS NEWSLETTERS
Y RECIBÍ EN TU CORREO
ELECTRÓNICO TODA LA
INFORMACIÓN DEL UNIVERSO
TECNOLÓGICO ACTUALIZADA
AL INSTANTE



INGRESÁ A
redusers.com/suscribirse-al-newsletter
¡Y REGÍSTRATE YA!

www.reduserspremium.blogspot.com.ar



Foros



Encuestas



Tutoriales



Agenda de eventos



Videos



¡Y mucho más!



redusers.com

Seguinos en:



www.facebook.com/redusers



www.twitter.com/redusers



www.youtube.com/redusersvideos



La segmentación de la red podrá lograrse mediante la instalación de un firewall con, por lo menos, tres tarjetas de red. Por un lado, tenemos una tarjeta dedicada al manejo de las entradas y salidas, o comunicación de Internet; esto es, que todas las peticiones realizadas desde Internet hacia nuestros servidores y de los equipos dentro de la red hacia Internet pasarán por ella. Una segunda tarjeta dentro del firewall estará dedicada, exclusivamente, a la comunicación entre el firewall y la zona desmilitarizada donde se encuentran los equipos que brindan los servicios externos de la empresa. Finalmente, la tercera se encargará de la comunicación con nuestra red interna.

Así, el firewall tendrá la capacidad de conmutar las peticiones entre las tarjetas de red. Como cada una pertenece a un rango de IP dife-

Es indispensable tener un firewall con, por lo menos, tres tarjetas de red para poder formar el esquema de una DMZ.

rente –por lo que cada segmento de la red es una red distinta–, éstas no son visibles entre sí de manera directa, sino que dependerán de la capacidad de intercambio de paquetes del firewall para poder “verse” entre ellas.

Una vez más, esta configuración no anula la capacidad de nuestros equipos de hacer uso de Internet, o de servidores corporativos, sino que su objetivo es aislar la red interna de posibles agresiones provenientes de Internet.

Zona desmilitarizada

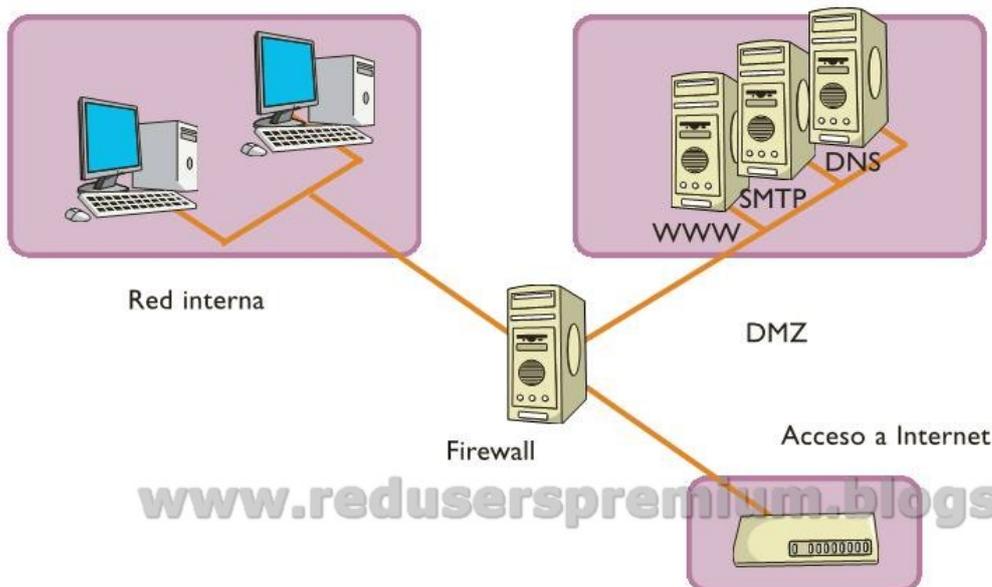


FIGURA 009 | Una zona desmilitarizada permite exponer servicios a través de un firewall de manera transparente.

LA AUTENTICACIÓN SE REFIERE A LA CAPACIDAD DE UN SISTEMA CUALQUIERA DE VALIDAR LA AUTENTICIDAD DEL USUARIO QUE INTENTA MANIPULARLO.

Autenticación y autorización

Como última alternativa está el mecanismo de “autenticación y autorización”, que, a diferencia de los dos anteriores, se utiliza para la administración y el comportamiento de usuarios de una red y los privilegios con los que contarán. Es importante distinguir estos dos términos y tratarlos como objetos separados. Si bien trabajan juntos, cada uno tiene un objetivo y pretende solucionar un problema específico.

El primero, la autenticación, se refiere a la capacidad de un sistema cualquiera (aplicación Web, de escritorio, bases de datos, acce-

so a VPN, entre otros) de validar la autenticidad del usuario que pretende manipularlo. Solemos encontrarnos con este mecanismo cuando intentamos acceder a nuestra cuenta de correo electrónico mediante una página Web. Para hacerlo, debemos ingresar nuestro nombre de usuario y contraseña, con lo cual la aplicación trata de identificarnos como poseedores de una cuenta en ese servidor. Esto no garantiza que podamos leer los mensajes aun habiendo sido “autenticados” correctamente. La autenticación, por su parte, tiene como objetivo final identificarnos como usuarios de un sistema, pero no, darnos los permisos necesarios para usarlo. Es la autorización la encargada de otorgarnos los privilegios sobre él, la que nos dirá, dentro de una red o una aplicación, qué elementos podemos manipular.

Microsoft ofrece herramientas para aplicar ambos mecanismos. A nivel de nuestra red corporativa podemos encontrar Active Directory, que permite administrar de manera centralizada a los usuarios de una red, así como los permisos y roles que poseen. A nivel de la administración de aplicaciones Web, podemos encontrar Internet Information Services, IIS, por sus siglas en inglés. No sólo sirve para hospedar nuestras aplicaciones Web, sino que además nos proporciona distintas alternativas para aplicar este modelo de seguridad.

⚠ Autenticación vs. autorización

No debemos confundir estos dos conceptos. La autenticación es el medio a través del cual nos identificamos con un sistema, en tanto que la autorización es el medio con el que obtenemos permisos o privilegios sobre él. Por ejemplo, al utilizar Hotmail, nos autenticamos con el sistema para ingresar en él, y mediante la autorización, recibimos los permisos necesarios, por ejemplo, para redactar un mensaje, obtener una lista de mensajes, eliminarlos, etc.



En la lista podemos ver estos mecanismos:

- Basic
- Digest
- NTLM
- Kerberos
- Certificados digitales

En el modelo Basic, el nombre de usuario y la contraseña son validados por IIS contra un Active Directory con el que se controlan las credenciales. Aunque es un modelo útil, resulta inseguro, ya que los datos viajan en formato de texto plano, codificados en Base 64.

El modelo Digest trabaja de manera similar al anterior, con la ventaja de que los datos

son encriptados por el navegador y viajan así hasta el servidor, que los descripta para usar. Este mecanismo requiere, además, de un navegador del tipo Internet Explorer para funcionar. Otra desventaja es que si bien las credenciales son encriptadas, no lo son el contenido y los datos de la petición. Tanto NTLM como Kerberos se usan para autenticar usuarios de Windows. Éstos tampoco pueden atravesar firewalls, por lo que no son adecuados para el uso a nivel Internet, sino ideales para intranets o redes internas con soporte Windows.

Por último, los certificados digitales pueden montarse a IIS para la implementación de SSLs en nuestras aplicaciones Web. Una red SSL puede ser reconocida en Internet por el

Sistema de autenticación

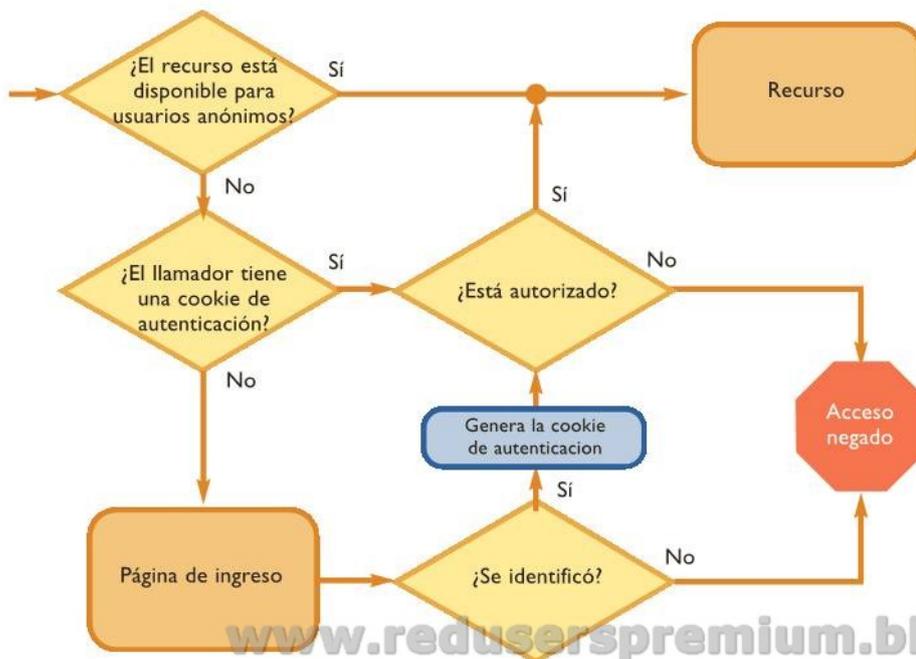


FIGURA 010 | Con la autenticación mediante formularios en ASP.NET, podemos utilizar condiciones para permitir el ingreso a un recurso.

LA AUTENTICACIÓN FORMS ES LA MÁS USADA Y COMÚN EN APLICACIONES WEB. ES AQUELLA QUE PRESENTA UN FORMULARIO CON LOS CAMPOS DE NOMBRE DE USUARIO Y CONTRASEÑA.

candado amarillo que suele aparecer en la parte inferior de los navegadores.

Autenticación en ASP.NET

En ASP.NET la autenticación puede llevarse a cabo de tres formas diferentes:

- Autenticación Windows
- Autenticación Forms
- Microsoft Passport



FIGURA 011 | Cuando tratamos de ingresar en sitios Web que requieren autenticación, nuestro navegador nos mostrará un cuadro de diálogo solicitando usuario y contraseña.

La primera usa los modelos previamente introducidos, y que están relacionados con la autenticación integrada de Windows; esto es, autenticación Basic, Digest, NTLM y Kerberos. Al igual que los modelos de IIS, este tipo de autenticación es ideal para intranets o redes Windows donde el manejo de usuarios es centralizado; por ejemplo, en un servidor con Active Directory.

La autenticación Forms es, posiblemente, la más usada y común en aplicaciones Web. Es aquella que presenta un formulario con los típicos campos de nombre de usuario y contraseña. Si bien esta solución puede ser implementada manualmente, el mecanismo presentado por ASP.NET automatiza gran parte del proceso de validación, verificación y acceso a recursos de una aplicación Web por parte de los usuarios.

La autenticación Forms automatiza la gestión de las páginas de una aplicación Web, verificando que el usuario que requiera una de ellas se haya autenticado mediante el ingreso de su nombre de usuario y contraseña en el formulario designado para tal efecto. Si no lo hubiera hecho, será redirigido para que lo haga. Si la autenticación se realiza correctamente, ASP.NET mostrará la página o recurso que requirió ver en primera instancia.

Microsoft Passport es la última alternativa presentada por ASP.NET para autenticación de usuarios en una aplicación Web. Se



trata de un servicio que permite centralizar los usuarios en un solo sistema de autenticación. Este modelo es especialmente ventajoso para dichos usuarios, ya que esta centralización les permite usar una misma credencial para autenticar en más de una aplicación Web. Por supuesto, estas credenciales no están en nuestro poder, sino que nuestras aplicaciones Web harán uso de los servicios de Microsoft Passport por medio de un ticket de autenticación.

Por último, es necesario tener en cuenta que las aplicaciones Web de ASP.NET acceden a los recursos del servidor o de la red corporativa mediante la cuenta de usuario IUSR_[NombreDeMaquina], donde [NombreDeMaquina] es el nombre del servidor donde está alojada dicha aplicación. Esta cuenta representa usuarios del tipo anónimos que ingresan en la red por medio de Internet. Como usuarios anónimos, los permisos de lectura y/o escritura o el acceso a determinados recursos pueden verse limitados. En todo caso, es posible configurar la aplicación Web para que use o tome el rol de otro usuario existente en la red corporativa en el momento en que acceda a estos recursos.

Este mecanismo, llamado en inglés *impersonation*, se implementa modificando el archivo Web.Config de la siguiente manera:

```
<identity impersonate="true"
  userName="dominio\usuario"
  password="contraseña" />
```

Es recomendable que el usuario que utilizará ASP.NET en esta “impersonación” posea privilegios restringidos sobre los recursos, por supuesto, dándole la libertad de uso para la que fue creado, pero no, por ejemplo, for-

El usuario por defecto usado por ASP.NET se considera un usuario anónimo con acceso limitado a recursos. Debemos “impersonar” para elevar estos privilegios.

mando parte del grupo de Administradores que tienen control total sobre todos los recursos de la red corporativa o el servidor. Esto se debe a que una falla de seguridad, como la que ya hemos visto en capítulos anteriores, podría verse alentada o beneficiada aún más por estos privilegios.

Seguridad en componentes

Microsoft .NET Framework implementa controles sobre nuestro código. Esto quiere decir que es posible asignar los permisos o capacidades que una pieza de código podrá tener. Por ejemplo, podría permitirse la ejecución del código en su totalidad, pero no permitir que dicho código o aplicación leyera archivos del disco duro.

Esta propiedad se denomina CAS (*Code Access Security*) o seguridad de acceso por el código. Tiene la característica de restringir aquellos recursos similares a los que podríamos restringir a un usuario dentro de nuestra red corporativa. Esto nos da mayor control sobre el código .NET en ejecución, y sirve como un puntal más para garantizar la seguridad e integridad de nuestros recursos.

Por parte de nuestro código, existen atributos decorativos para métodos y a nivel del ensamblado, que serán usados para determinar el objetivo del método o el ensamblado antes de ser ejecutado.

En el siguiente ejemplo vemos los atributos decorativos sobre un ensamblado, el que pide poder leer un archivo:

VB.NET:

```
Imports System.Security.Permissions

<Assembly: FileIOPermissionAttribute
  (SecurityAction.RequestMinimum, _
  Read := "C:\boot.ini")> _
Class Class1
  Sub Main()
  ...
```

C#:

```
using System.Security.Permissions;

[assembly:FileIOPermissionAttribute(Security
Action.RequestMinimum, Read = @"C:\boot.ini")]
namespace DeclarativeExample
```

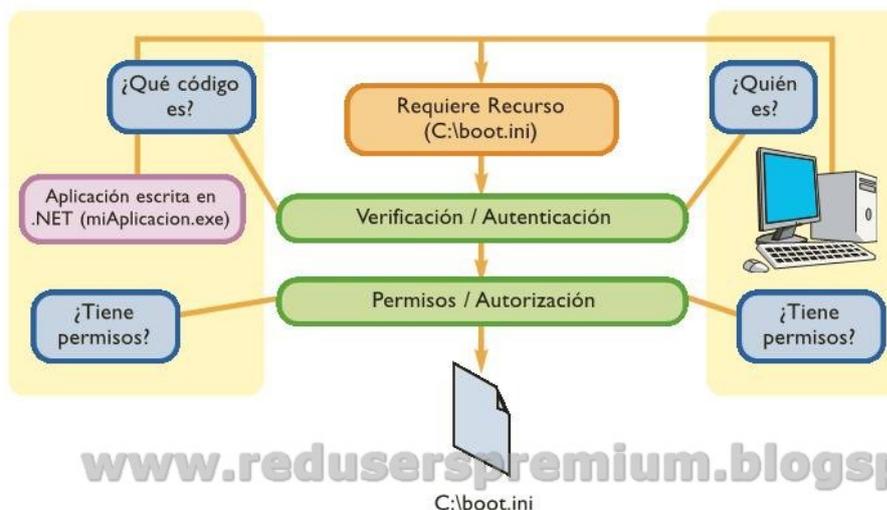
```
{
  class Program
  {
    static void Main(string[] args)
    ...
```

Así, si nuestro ensamblado no tuviera permisos de lectura sobre dicho archivo, éste simplemente no se ejecutaría.

Seguridad en bases de datos

El primer punto débil que debemos considerar es la exposición de la cadena de conexión a nuestra base de datos. Esta cadena puede ser construida en tiempo de ejecución –es decir, al momento de ejecutar la aplicación por primera vez, cuando se toman varias piezas de ella y se forma la cadena–, pero tanto las partes como

Code Access Security



www.reduserspremium.blogspot.com.ar

FIGURA 012 | Utilizando CAS, podremos controlar los permisos de una aplicación, como si se tratase de un usuario de una red corporativa.



la cadena completa suelen estar escritas en texto plano en algún archivo externo de la aplicación. En el caso de ASP.NET, el lugar comúnmente usado es el archivo **Web.Config**, como se ve a continuación:

```
...
<configuration>
  <!-- Custom configuration Settings -->
  <appSettings/>
  <!-- Custom Application Connection String -->
  <connectionStrings>
    <add name="MiCadena" connectionString=
      "CadenaDeConexión"/>
  </connectionStrings>
  ...
```

Si bien este archivo no es visible para los usuarios de Internet, si fuera obtenido por parte de un agresor, al estar almacenado como texto plano, sería posible acceder a su información fácilmente.

Por eso, es recomendable encriptar esta cadena de conexión antes de enviarla a un ambiente de producción. Esta acción puede realizarse con la ayuda del archivo **AspNet_regiis.exe**, con el atributo **-pef**. Este ejecutable viene incluido con Microsoft .NET.

Es recomendable, también, que las contraseñas almacenadas en la base de datos se mantengan encriptadas. Por ejemplo, las claves empleadas por los usuarios de nuestro sistema suelen guardarse en la base de datos como texto. Si se produjera una falla de seguridad como las nombradas anteriormente (Sql Injection o Scripting), el agresor podría obtener las contraseñas y usarlas a su favor. En cambio, si éstas estuvieran encriptadas, el resultado de esta agresión no causaría tanto daño. Uno de los algoritmos más usados para

Debemos ejecutar **aspnet_regiis -pef** “connectionStrings” “Ruta física del archivo Web.Config” para poder encriptar nuestra cadena de conexión.

esta encriptación es el denominado MD5, que tiene la particularidad de ser irreversible. Esto quiere decir que, una vez que se aplicó la encriptación, su resultado no puede ser revertido.

Seguridad en servicios Web

Pensemos que un servicio Web podría constituir todas las interfaces de comunicación necesarias para un producto completo. Podríamos crear una aplicación que sólo se encargara de brindar datos estadísticos, cotizaciones monetarias o el clima. Este producto como tal no requeriría de una interfaz, una página Web donde el usuario debiera autenticarse, y con esto, autorizarlo para usar o consumir dicho servicio.

Esto no quiere decir que estemos obligados a que el servicio brindado no pueda o deba estar restringido sólo a aquellos usuarios autorizados a usarlo. Como si esto fuera poco, un servicio Web podría ser consumido por diferentes equipos, desde servidores hasta dispositivos móviles, como celulares o Pocket PCs. Para lograr esto, existen los WSE (*Web Services Enhancement*), o mejoras en servicios Web, que comprenden los siguientes puntos:

Es importante considerar que la base de datos, y los datos que se encuentran en ella, también están sujetos a fallas de seguridad.

- Tokens de seguridad
- Tokens de seguridad bajo autenticación Kerberos
- Tokens de seguridad basados en roles
- Firmado y encriptado de tokens de seguridad
- Manejo de sesiones entre el servicio y el cliente garantizando la autenticidad

Estas mejoras nos darán las herramientas de seguridad para garantizar la estabilidad de nuestros

servicios Web. Todas estas herramientas se encuentran en las versiones de WSE 2.0 y 3.0, lo que no descarta que nuevas versiones implementen aún más características, con mayores controles de seguridad a nuestro favor.

Claves de la seguridad

En este capítulo hemos podido aprender, entender y tomar conciencia de que la seguridad en el desarrollo de software y en el ambiente que envuelve este desarrollo no es un aspecto que debemos desestimar. Una falla de seguridad podría costarnos más de lo que pensemos. Por lo tanto, las técnicas de seguridad –como encriptación de contraseñas, firewalls y buenas prácticas, entre otras– deben estar presentes durante todo el proceso de desarrollo, lo que nos ayudará, definitivamente, a reducir las posibilidades de sufrir agresiones.

Seguridad en servicios Web



FIGURA 013 | Cada vez más, los servicios Web son parte central de las aplicaciones, y la seguridad en ellos es un punto primordial.



Tercer repaso

Realicemos otro repaso de los conceptos adquiridos en las entregas que componen la estrella número 3 de nuestro curso Desarrollador .NET, con el objetivo de estar listos para rendir el examen.

En las siguientes páginas, revisaremos los conceptos más importantes de la Estrella 3, con el objetivo de estar preparados para rendir sin inconvenientes el examen de certificación de Microsoft.

Introducción a XML

- XML es el acrónimo del inglés *eXtensible Markup Language*, o lenguaje de marcas extensibles, que fue desarrollado por el W3C (Consortio de Internet). Es un lenguaje utilizado no sólo para Internet, sino también para compartir información estructurada, de manera de permitir que ésta se describa a sí misma.
- En la actualidad, es utilizado en bases de datos, procesadores de texto, planillas de cálculo, y un sinfín de utilidades más. XML descende de su antecesor GML (*General Markup Language*), creado por IBM en la década de 1970. Luego pasó a llamarse SGML, que simplemente fue una estandarización de GML. Con el nacimiento de Internet, apareció el lenguaje HTML, que es una especie de adaptación de GML y SGML. Y finalmente, apareció y se estandarizó en el mercado el lenguaje XML.

Sintaxis de XML

- Las dos secciones más importantes del lenguaje XML son el prólogo y el cuerpo:
 - El prólogo se utiliza para describir la versión de XML que se está empleando, aunque su uso no es obligatorio.
 - El cuerpo del documento es de carácter obligatorio y, a su vez, debe contener un elemento raíz. También componen un archivo XML los elementos, que pueden estar vacíos o no, o bien contener otros elementos o texto. Poseen atributos que se usan para incorporar características o propiedades a un elemento XML.

Esquemas y validaciones

- Ya sabemos que XML es un medio estructurado para guardar y compartir información. Un esquema describe la estructura en la que se alojarán los datos.

EXAMEN

- Muchas veces encontraremos que un documento XML puede estar bien construido y estructurado en cuanto a sintaxis, pero no corresponderse con el esquema que lo rige.
- Los esquemas de datos pautan restricciones en la estructura del contenido del documento. Esto lo realizan de las siguientes maneras:
 - Estableciendo el modelo de contenido de los documentos, que indica su orden y el anidamiento de los elementos.
 - Estableciendo los tipos de datos permitidos en cada caso dentro del documento.
- DTD proviene del inglés *Document Type Definition*, o definición de tipo de documento. Es una definición de la estructura y la sintaxis de un documento XML (también es aplicable a los SGML). Las DTD pueden ser internas o externas. Cabe recordar, también, que la comunidad de programadores que utilizan XML prefieren usar XML Schema antes que DTD. Para usar DTD dentro de un archivo XML, debemos definirlo junto a la declaración de la versión del archivo XML utilizada.

Estilos

- XML Schema: Los documentos que utilizan XML Schema emplean un vocabulario llamado XML-Data. XML-Schema nos permite asociar tipos de datos a los elementos, pero escribiendo este tipo de documento en formato XML.

La composición de su vocabulario es:

- Schema: Sirve como elemento raíz del documento XML-Schema.
- Datatype: Describe los tipos de datos

de elementos y atributos.

- ElementType: Describe un tipo de elemento.
 - Element: Identifica un elemento que puede aparecer dentro de otro tipo de elemento.
 - Group: Organiza elementos.
 - AttributeType: Describe un tipo de atributo.
- Se puede utilizar XML para marcar el contenido de un documento, sin importar la manera de mostrar la información que éste incluye. Para formatearlo, debemos indicar cómo hacerlo. Podemos utilizar fuentes, colores, etcétera, a través de las hojas de estilo CSS. Éstas constan de normas que determinan cómo debemos usar las reglas determinadas para formatear un documento. Se engloban dentro de selectores que indican partes de un documento XML o HTML al cual aplicar dichas reglas. El selector establece un vínculo entre el conjunto de estilos y el documento.
 - Las propiedades de estilo más comunes aceptadas por CSS son: Display, Width, Height, Border (y sus derivados), Margin, Color, TextAlign, Font, y otras. Las hojas de estilo que podemos crear pueden ser externas o internas, similar a lo que ocurre en los DTD.
 - XPath es un lenguaje de expresión que se utiliza para obtener una porción o parte de un documento XML, sin necesidad de implementar este último. Esta tecnología se reserva a situaciones en las que no se aplica el marcado XML, como en los valores de los atributos. XPath comprende que un documento XML fue analizado sintácticamente en un árbol de nodos.
 - Hay varios tipos de nodos que, seguramente, aparecerán en un árbol XPath:

www.reduserspremiumblogspot.com.ar



nodos raíz, de elementos, textos, atributos, espacio de nombres, instrucciones de procesamiento y comentarios. Cuando necesitamos buscar un nodo determinado, XPath recorre el árbol de nodos del documento XML. Dicho recorrido es llevado a cabo mediante expresiones, donde cada una, luego de ser evaluada, termina siendo un objeto de datos en uno de los siguientes tipos: node-set, boolean, float o string. XPath no es un lenguaje similar a C o Visual Basic, sino que es del tipo declarativo. Por lo tanto, las instrucciones abarcan cierta variedad de operaciones, como llamado a funciones y location paths.

ADO.NET

Los espacios de nombres

- En la categoría System.Data encontramos diversos namespaces. En ellos podemos hallar clases similares entre sí, para acceder y manipular datos almacenados en memoria.
- System.Data: Aquí hay diversas clases genéricas para almacenar datos en memoria. Entre ellas, DataSet, DataTable, DataRow, DataColumn, DataType.
- En System.Data.Common podremos crear capas de acceso a datos en forma independiente del motor. Estas clases o métodos pueden ser comunes a cualquier proveedor que incluya el framework .NET.
- Las clases principales se dividen en:
 - DataAdapter: Para definir comandos de selección y otras operaciones sobre el conjunto de datos DataSet.
 - DbConnection: Representa una conexión al motor de base de datos, basándose en la cadena de conexión.

- DbParameter: Permite el envío de parámetros de entrada y salida hacia el motor de base de datos.
- DbDataReader: Se utiliza para leer una fuente de datos. La lectura sólo será hacia adelante.
- El framework .NET posee clases específicas para cada motor de base de datos que existe en la actualidad. Esto es lo que nos permite conectarnos al motor, ejecutar consultas y obtener los resultados. La clase SQLClient permite conectarnos a SQL Server; OleDbClient, a cualquier motor de base de datos que soporte OleDb; OracleClient, a motores Oracle, etc.

DataSet

- El objeto DataSet puede contener una o más tablas del tipo DataTable, junto a las relations o relaciones que las corresponden. Cuando se crea la o las tablas en el DataSet, necesitamos indicar cuáles son las columnas que poseerá, agregando por cada una de ellas una instancia de la clase DataColumn. La clase DataSet puede generar automáticamente tablas, columnas, relaciones y demás, a partir de un archivo de esquema XML. El método que nos permitirá lograrlo es ReadXMLSchema. XSD es la extensión con la cual se creará el archivo de esquema.
- Es el elemento principal para vincular nuestro desarrollo al motor de base de datos. En dicha cadena de conexión se incluyen los datos necesarios de validación, como usuario, contraseña, timeout o límite de tiempo, opciones de seguridad, servidor, etc. Dichos parámetros se concatenan mediante punto y coma (“;”). La cadena de conexión es necesaria para instanciar objetos de tipo

Connection, cualquiera sea.

- La cadena OleDb difiere muy poco de sus pares de SQL Server, y su composición depende, exclusivamente, del proveedor de acceso que usemos.

System.Data

- En el namespace System.Data, SQLClient encontraremos las clases necesarias para acceder a motores SQL Server a partir de la versión 2000. SqlDataAdapter, SqlConnection y SqlParameter son implementaciones que cumplen la misma función que System.Data.Common.
- SqlCommand es un comando que nos permite ejecutar una instrucción del tipo Transact SQL, para acceder al motor de base de datos que fue establecido en esa conexión, y devolver el resultado de esto a través de objetos. Para ejecutar dicho comando, necesitamos pasar un parámetro del tipo string.
- SqlDataReader posee algunas cosas en particular, las cuales debemos conocer al momento de usarlo. Este objeto funciona como lector secuencial de datos, que se mueve de a un registro por vez. Su uso debe complementarse con un método While para poder realizar la lectura de la tabla de datos.
- Los Stored Procedures nos permiten escribir comandos lógicos de operaciones sobre la base de datos, y almacenarlos a través de parámetros. De esta forma, en vez de generar una consulta u operación del tipo T-SQL dentro de nuestros desarrollos, podemos realizar una llamada a un Stored Procedure y, pasándole los parámetros necesarios, hacer que nos devuelva los resultados de una manera rápida y confiable.
- La llamada a un procedimiento almace-

nado se realiza a través de un objeto SqlCommand, indicándole el nombre del procedimiento en cuestión y enviándole los parámetros necesarios para que nos devuelva los resultados.

La tecnología OleDb

- Esta tecnología que nos propone Microsoft permite acceder a prácticamente cualquier fuente de datos que tenga los drivers correspondientes. En el framework .NET encontramos un conjunto de clases específicas para usarla.
- Los objetos de acceso a datos del tipo System.Data.OleDb son iguales a los que encontraremos en System.Data.SqlClient. De todos modos, hay algunas diferencias concretas.
- Las conexiones OleDb se manejan a través de objetos OleDbConnection, que son equivalentes a SqlConnection. La cadena de conexión OleDb es particular a cada motor. Los drivers que servirán para hacer uso de esta tecnología deben buscarse en el sitio web de cada proveedor de motor de datos al cual queramos conectarnos.

Capa de acceso a datos

- La capa de acceso a datos encapsula todas las funciones para acceder a los datos de una base en nuestros sistemas o desarrollos y, así, automatizar la interacción con cualquier motor de datos.
- Existen capas físicas y capas lógicas. La primera es el acceso a un servidor Web, por ejemplo, mientras que la segunda genera un proceso separado, creando una interfaz de usuario para poder acceder a los datos. Esto se utiliza con el objetivo de reducir la cantidad de código requerido para acceder al motor y, así, poder parametrizar los puntos



necesarios que varían entre una consulta y otra. De este modo, podríamos reutilizar el código desde cualquier plataforma, a la vez que podríamos intercambiar el motor de la base de datos sin necesidad de crear nuevas compilaciones en las capas superiores.

Seguridad en bases de datos

- El concepto de seguridad es el más importante en una base de datos, ya sea que hagamos un simple sistema monousuario o desarrollemos uno de misión crítica. La seguridad es lo primero que debemos lograr en el desarrollo, dado que en cualquiera de estos casos, la información contenida en las bases de datos puede ser accedida por cualquier persona física o software con fines orientados a la piratería.
- Es necesario crear una restricción de usuarios para acceder a la información de la base de datos. Una de las formas más comunes de lograrlo, es hacer que los usuarios ingresen a través de procedimientos almacenados, que sólo entreguen la información que éstos necesitan ver. Hay que utilizar usuarios creados en la base de datos y aplicar los permisos a cada tabla en forma individual, para restringir el acceso de cualquier a cualquier tabla.
- En el desarrollo de aplicaciones Web es común utilizar el archivo Web.Config para almacenar datos, a los cuales accederemos luego desde nuestra aplicación. Estos datos son del tipo sólo lectura. En este archivo se puede incluir la cadena de conexión al motor de base de datos. Para acceder a los valores almacenados en este archivo existen diversas formas. El framework .NET, en sus versiones 1.0 y 1.1, utilizaba la clase

AppSettings. A partir de la versión 2.0, se simplificó la tarea de acceder a los parámetros de este archivo a través de la clase Configuration Manager.

WSDL

- Es el lenguaje a través del cual se describe un servicio Web, especificando qué métodos y parámetros recibe, y cuáles son los valores que retorna cada llamada que hagamos a él. Accedemos a los servicios Web en forma dinámica.
- XML maneja la información que concierne a los servicios Web. Éste describe o define la interfaz de los servicios, a la cual divide en dos partes: la concreta, que define el cómo y el dónde; y la abstracta, que establece qué hacer a través de los mensajes recibidos y retornados.

UDDI

- UDDI nos permite describir un servicio Web y, también, localizarlo en forma dinámica, para que otros clientes y servicios actúen con él creando una plataforma interoperable. UDDI se monta sobre SOAP, para que tanto al creador como al consumidor le resulte totalmente transparente.

Web Service

- Un Web Service es un servicio que permite la comunicación entre procesos y aplicaciones en forma independiente del lenguaje de programación, protocolo de comunicación y plataforma. Para crear un Web Service desde Visual Studio .NET 2005 debemos agregar un proyecto del tipo servicio Web ASP.NET. Este proyecto creará, entre otras cosas, una clase y otros archivos propios de la publicación Web, como Web.Config y Global.Asax.

Preguntas de ejemplo

Éstas son algunas de las preguntas que tendremos que superar para recibir la certificación de Microsoft.

1 | ¿Qué es XML y para qué fue ideado?

- A. Es un lenguaje de programación pensado para desarrollar aplicaciones sobre Internet.
- B. Es un lenguaje para utilizar en reemplazo de base de datos.
- C. Es un lenguaje de marcas definido para utilizar como estándar para compartir información estructurada.

Respuesta correcta C: Su uso no fue marcado sólo para trabajarlo con aplicaciones Web, sino que es de propósito general. Hoy en día, se emplea en bases de datos, editores de texto, planillas de cálculo, y un sinfín de usos más.

2 | ¿Cómo se componen las partes principales de un documento XML?

- A. Está compuesto por dos secciones: Prólogo y Cuerpo.
- B. Está compuesto por dos secciones: Encabezado y Datos.
- C. Está compuesto por tres secciones: Prólogo, Datos y EndOfFile.

Respuesta correcta A: Un documento XML consta de dos secciones principales: el Prólogo, que no es obligatorio declarar, pero que se incluye, generalmente, para describir la versión de XML que estamos empleando; y el cuerpo del documento, que es de carácter obligatorio y debe contener, sí o sí, un elemento raíz.

3 | ¿Qué función cumple un esquema en XML?

- A. Permite mostrar los registros de datos cargados en XML.
- B. Describe la estructura en la que se alojarán los datos.
- C. Indica a C# o VB.NET en qué versión de XML estamos.

Respuesta correcta B: XML es un documento que utiliza un medio estructurado para almacenar y compartir información, y el esquema es el encargado de describir la estructura en la cual se alojarán los datos. Un esquema de datos establece las restricciones para estructurar el contenido del documento, indicando el modelo para ordenar y anidar los elementos, y estableciendo qué datos están permitidos en cada caso dentro del documento.

4 | Schema, AttributeType y Group son:

- A. Una sintaxis especializada de los DTD.
- B. Elementos que componen el vocabulario de XML Schema.
- C. Partes del cuerpo de un documento XML.

Respuesta correcta B: Los esquemas XML presentan un modelo de datos abierto, que nos permite ampliar su vocabulario, estableciendo una suerte de herencia entre los distintos elementos que lo componen.

5 | Una hoja de estilo CSS se utiliza para formatear:

- A. Archivos del tipo HTML y PHP.
- B. Documentos del tipo CS o VB.
- C. Documentos XML y HTML.

Respuesta correcta B: Las hojas de estilo CSS poseen normas que son determinantes para aplicar ciertas reglas a un documento. Las reglas se engloban dentro de selectores que nos indican partes de un documento XML o HTML a las cuales se aplican. Esto nos permite, entre otras cosas, formatear la información de un archivo XML, como fuente, estilo negrita, cursiva, subrayado, etc.

6 | Applytemplates, Valueof, foreach son elementos de:

- A. XSL
- B. XML
- C. C#

Respuesta correcta A: Valueof se usa para indicar el valor que se debe colocar en ese instante dentro del resultado de la transformación, aplicando XSL a un archivo XML. Applytemplates se usa para aplicar plantillas que se definen en una hoja de estilos. Foreach se emplea para recorrer el subconjunto de datos de un archivo XML. XSL es un procesador que lleva a cabo la construcción de un árbol de resultados y la interpretación de ese árbol para darle un formato a la información.



7 | Las clases XPathDocument, XPathExpression, XPathNodeIterator pertenecen a:

- A. System.Xml.Datatype
- B. System.Xml.OleDbX
- C. System.Xml.XPath

Respuesta correcta C: XPath es un lenguaje de expresión que sirve para extraer parte de un documento XML sin necesidad de implementar este último. Esta tecnología se usa en determinadas situaciones. Xpath compone el namespace System.Xml. XPathDocument, XPathExpression, XPathNodeIterator, XPathNavigator, XPathException son clases que corresponden a System.Xml.XPath.

8 | System.Data corresponde al namespace de:

- A. OleDb
- B. DBRead
- C. Ninguna de las anteriores

Respuesta correcta C: System.Data es el namespace principal para el acceso a datos. Éste posee diversas clases genéricas para almacenar datos en memoria.

9 | Si nos referimos a los esquemas de SQL Server:

- A. Un usuario puede pertenecer a un rol y a un esquema, y un rol puede pertenecer a un esquema.
- B. Un usuario puede pertenecer a un rol y a un esquema.
- C. Un rol puede pertenecer a un esquema.

Respuesta correcta A: Desde la versión 2005 de SQL Server, cada objeto pertenece a un esquema de base de datos. Un esquema es un espacio de nombres separado de un usuario de base de datos. Un esquema se puede considerar como un contenedor de objetos. Podemos crear y modificar esquemas en una base de datos, y se le puede conceder a cada usuario un permiso de acceso a un esquema.

10 | TableAdapter, además de permitirnos abrir y cerrar la conexión con el motor automáticamente, ¿qué otra opción brinda?

- A. Usa objetos Command y DataReader internamente.
- B. Permite llenar un DataTable.
- C. Todas las respuestas son correctas.

Respuesta correcta C: En Visual Studio 2005 se amplió el acceso a datos con tipo mediante TableAdapter. Éste nos permite interactuar con objetos de datos desde el cliente, enviando actualizaciones a una base de datos. TableAdapter encapsula los objetos para establecer una comunicación con las bases de datos, para proporcionarnos métodos que permitan realizar el acceso a los datos con tipos seguros.

11 | Utilizando seguridad integrada, ¿que usuario accede a la base en una aplicación?

- A. El usuario que creó la base.
- B. El usuario que creó el objeto por utilizar.

- C. El usuario activo en el equipo cliente.
- D. El usuario Local System.

Respuesta correcta B: Cuando se inicia una aplicación realizada en .NET del tipo Escritorio -es decir, una aplicación Windows-, el CLR toma el usuario que la inició para generar el contexto en el que trabajará el programa. Entonces, cada instancia nueva de un objeto que utilice esta aplicación adoptará dicha identidad.

12 | Una correcta administración de seguridad en la base de datos implica:

- A. No utilizar "sa" como usuario de la cadena de conexión.
- B. Evitar o denegar el acceso directo a las tablas.
- C. Establecer que todas las acciones de la base de datos se realicen a través de procedimientos almacenados.
- D. Todas las anteriores.

Respuesta correcta D: Si utilizamos el usuario "sa" para la conexión a la base de datos, y si un atacante emplea SQL Injection, podemos perder todos los datos; tengamos en cuenta que "sa" es el administrador de sistemas de la base de datos. El acceso directo a tablas puede jugar en contra; por ejemplo, no conceder permisos de escritura a tablas si con el permiso de lectura es suficiente. Por último, es preferible siempre utilizar procedimientos almacenados para efectuar tareas sobre las tablas y/o consultas; esto también evita que un atacante obtenga nuestros datos empleando la técnica de SQL Injection.

Ejercicios Estrella 3

Tal como en las primeras dos estrellas, invitamos a practicar lo visto hasta aquí, realizando ejercicios y proponiendo otros para pulir más nuestro conocimiento.

EJERCICIO 1: En la página 340 hablamos sobre la tecnología XSLT, componente de transformación de la tecnología de hojas de estilos XSL. Allí realizamos un ejercicio que nos permitía mostrar, a través de ASP.NET en una página Web, clientes que estaban cargados en un documento XSL. Vamos a ampliar este documento XSL, agregándole otros clientes, y luego visualizaremos en la página el resultado de la vista de clientes.

EJERCICIO 2: A la tabla de resultados de la página Web que vimos en el ejercicio 1, vamos a aplicarle un formato de textos. Resaltaremos en negrita el nombre del cliente, y pondremos en color azul o el que prefiramos la Localidad y la Provincia, invirtiendo los colores que actualmente tienen.

EJERCICIO 3: En la página 353 de ADO.NET Avanzado nos referimos al

sitio www.ConnectionStrings.com, que nos permite obtener la cadena de conexión a cualquier tipo de base de datos. Vamos a buscar las cadenas de conexión y a armar un proyecto en Visual Basic .NET o C# (o ambos), para conectarnos con otras bases de datos, como Access. Si no disponemos de esta aplicación, podemos obtener de Internet alguno de los motores gratuitos, como MySQL o Firebird.

EJERCICIO 4: En la página 366 aprendimos sobre la seguridad en bases de datos SQL Server 2005. Busquemos nuestros primeros proyectos en los cuales trabajamos con la base de datos NorthWind, y creamos usuarios y restricciones de acceso a algunas tablas para determinado usuario. Tengamos en cuenta que esto puede afectar la base de datos NorthWind, por lo cual debemos asegurarnos de tener una copia de ella antes de proceder a modificarla.

EJERCICIO 5: En la página 361 aprendimos a realizar consultas con parámetros, y en la 363, vimos para qué sirve un procedimiento almacenado y cómo crear uno. Combinando estos conocimientos, vamos a modificar el ejemplo de la página 361, para que, en vez de escribir la consulta SQL directamente en el código, éste llame a un procedimiento almacenado y agregue los parámetros que correspondan anteponiendo la arroba (@) para cada nombre del parámetro. Estos parámetros deben llamarse tal como se nombraron dentro del procedimiento almacenado.

! Recomendación

Recomendamos también recorrer el sitio de Microsoft TechNet: www.microsoft.com/spanish/msdn/articulos/default.aspx y leer artículos técnicos acerca de Visual Studio 2005. Éstos nos ayudarán aún más a profundizar conocimientos y pulir detalles puntuales de este curso.



Framework 3.0

La nueva evolución

12

Contenidos

Repasaremos la evolución del .NET Framework de Microsoft, aprenderemos qué nuevos componentes trae la versión 3.0 y analizaremos para qué sirve cada uno. Conoceremos, también, la nueva herramienta que propone Microsoft para las identidades digitales: Windows CardSpace.

Temas tratados

- » Nuevos componentes
- » Windows CardSpace
- » Identidad digital
- » Identidades seguras

Framework 3.0

Conoceremos las nuevas herramientas que acompañan a este Framework y los roles que se integran dentro del mundo del desarrollo.

» Introducción

Nos introduciremos en este nuevo conjunto de componentes denominado Framework 3.0 y veremos cuál fue su evolución a través del tiempo.

- > Un poco de historia
- > XAML
- > Expressions

» Nuevos componentes

Aprenderemos, de manera introductoria, qué son y qué funciones cumplen tres de los cuatro nuevos componentes que conforman este Framework.

- > Windows Presentation Foundation
- > Windows Communication Foundation
- > Windows Workflow Foundation

» Windows CardSpace

Veremos de qué se trata esta nueva herramienta que propone Microsoft para la identificación y autenticación digital, sus ventajas y funcionamiento.

- > Introducción
- > Cómo funciona
- > Ventajas

» Identidad digital

Aprenderemos sobre el concepto de identidad digital, tokens, leyes, etcétera, y cómo aprovechar las características de Windows CardSpace para utilizarlas en él.

- > Leyes de identidad
- > Autenticación/Autorización
- > Tokens

» Identidades seguras

Analizaremos el esquema de seguridad que propone Microsoft con su herramienta Windows CardSpace.

- > Identidad distribuida
- > Contexto de la identidad
- > ¿Qué seguridad aporta WCS?



Introducción al Framework 3.0

Para quienes seguimos la evolución de la plataforma .NET, no debe sorprendernos cómo ha evolucionado desde sus inicios.

Desde sus orígenes hasta estos años, el núcleo de la plataforma .NET, el .NET Framework, ha pasado por varias versiones, por supuesto, una mejor que la anterior; desde la versión 1.0, 1.1 y 2.0, hasta ahora, con la 3.0, que se lanzó con el sistema operativo Windows Vista. Y, por supuesto, muy pronto llegará la 3.5, que será un componente fundamental y requisito indispensable de Visual Studio .NET 2008. Pero de esto hablaremos más adelante; ahora, nuestro punto principal es el .NET Framework 3.0.

En un principio, el .NET Framework 3.0 se llamó WinFX, y está integrado por un conjunto de cuatro APIs (*Applications Programming Interface*), que son:

- Windows Communication Foundation: WCF (antes llamado Indigo)
- Windows Presentation Foundation: WPF (antes llamado Avalon)
- Windows Workflow Foundation: WWF
- Windows CardSpace: WCS (antes llamado InfoCard)

.NET Framework 3.0 se llamó WinFX, y está integrado por un conjunto de cuatro APIs (*Applications Programming Interface*).

Microsoft Framework 3.0

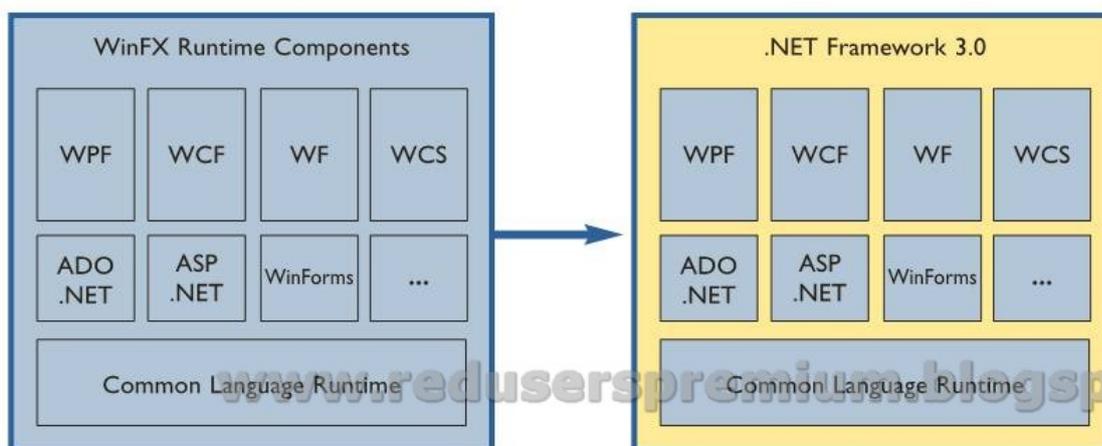


FIGURA 001 | Microsoft decidió nombrar formalmente a su nuevo conjunto de APIs como Microsoft Framework 3.0.

ANTE MUCHAS CRÍTICAS Y COMENTARIOS, MICROSOFT DECIDIÓ UNIR EL .NET FRAMEWORK 2.0 CON WINFX, LO QUE DIO COMO RESULTADO EL .NET FRAMEWORK 3.0.

Ante muchas críticas y comentarios, Microsoft decidió unir el .NET Framework 2.0 con WinFX, lo que dio como resultado el .NET Framework 3.0. Tengamos en cuenta que no hay ninguna mejora en comparación con la versión 2.0, ya que la más reciente tiene los mismos compiladores, los mismos ensamblados y, por supuesto, el mismo CLR.

Piezas adicionales

Existen dos piezas adicionales en este gran mapa de nuevas tecnologías, que no son parte directa del .NET Framework 3.0, pero que son

tecnologías muy importantes para considerar con el objetivo de desarrollar software:

XAML

Es un lenguaje de marcas, como HTML y XML, y las utiliza para representar declarativamente la interfaz de una aplicación de Windows. De este modo, se mejora la riqueza de las herramientas mediante las cuales desarrolladores y diseñadores gráficos pueden construir y modificar la interfaz de usuario. Entendamos y tengamos en cuenta que WPF utiliza XAML.

Expressions

Con la nueva interfaz 3D de Windows Vista, una aplicación actual podría verse un tanto “antiguada” o, mejor dicho, desfasada en el tiempo, ya que el usuario esperará ver aplicaciones también en 3D que utilicen al máximo el motor de Windows y, por supuesto, el sinnúmero de hardware nuevo que existe en el mercado informático.

Los productos Expression nos permiten crear gráficos de bitmaps o de vectores, animaciones en 3D, y mucho más. Un aspecto interesante es que producen código XAML, de modo que es posible incorporar los resultados en aplicaciones de escritorio o Web utilizando WPF. Así, diseñadores y desarrolladores que producen código XAML para incorporar con las tecnologías WPF podrán publicar también dichas interfaces en la Web y disponer de muchas otras funcionalidades.

* ¿Cómo desarrollar estas aplicaciones de nueva generación?

Aquí entra en escena la suite de productos Expresión. Se trata de herramientas para diseñadores gráficos que trabajen de forma recíproca con el equipo de desarrollo de software, no sólo como medios dependientes uno del otro, sino para construir interfaces animadas y en 3D, que hagan que el usuario quede impresionado y satisfecho al manejar las presentaciones de estos nuevos esquemas de interfaces.

Para obtener más información al respecto, se recomienda visitar:

www.netfx3.com
nextdesignnow.com



Nuevos componentes

Microsoft .NET Framework 3.0 es el conjunto de cuatro tecnologías. A ellas nos referiremos a continuación.

Los cuatro componentes que forman esta nueva plataforma son: Windows Communication Foundation, Windows Presentation Foundation, Windows Workflow Foundation y Windows CardSpaces. Luego de haber definido al .NET Framework 3.0 (antes WinFX) como una plataforma completa de desarrollo, examinaremos que, como tal, incluye avances en tres áreas que impactan directamente sobre la productividad y el seguimiento de la tecnología como desarrolladores.

WPF es considerado como un subsistema de presentaciones unificado para Windows, para crear aplicaciones con interfaces visuales sensacionales.

Windows Presentation Foundation

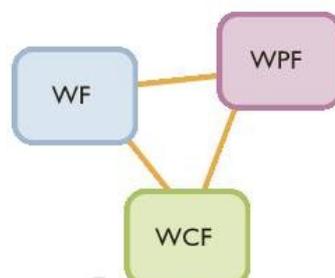
Considerado como un subsistema de presentaciones unificado de Microsoft para Windows. Consiste de un motor de visualización y un conjunto de

clases administradas que permite crear aplicaciones avanzadas y visualmente sensacionales.

Windows Presentation Foundation también introduce XAML, que permite utilizar un modelo basado en XML para manipular, mediante declaraciones, el modelo de objetos de Windows Presentation Foundation.

Las clases que constituyen esta API forman parte, en gran medida, del espacio de nombres

Componentes del Framework 3.0



www.reduserspremium.blogspot.com.ar

Windows Vista

FIGURA 002 | Componentes que integran el nuevo Framework 3.0, el futuro del desarrollo de aplicaciones para Windows Vista.

WCF es la nueva infraestructura de comunicación orientada a servicios, creada sobre la base de protocolos de servicios web.

System.Windows o sus descendientes. Los componentes principales son:

- Un modelo de aplicación con compatibilidad de exploración, ventanas y cuadros de diálogo
- Enlace de datos de Interface de Usuario
- Un avanzado conjunto de objetos de diseño y de control extensibles
- Gráficos bidimensionales y tridimensionales
- Animaciones
- Multimedia
- Documentos

Windows Communication Foundation

Es la nueva infraestructura de comunicación orientada a servicios creada sobre la base de protocolos de servicios Web. La compatibilidad del servicio Web avanzado en Windows Communication Foundation proporciona una mensajería interoperable, segura, confiable y por transacciones. El modelo de programación orientado a servicios de Windows Communication Foundation se basa en el .NET Framework y simplifica radicalmente el desarrollo de sistemas conectados.

Además, unifica una amplia gama de capacidades de sistemas distribuidos en una arquitectura ex-

tensible que se puede componer, y que admite varios transportes, patrones de mensajería, codificaciones, topologías de red y modelos de hospedaje. Es la nueva versión de varios productos existentes: los métodos Web de ASPNET (“ASMX”) y Microsoft Web Services Enhancements para Microsoft .NET (WSE), .NET Remoting, Enterprise Services y System.Messaging.

Las clases que constituyen la API de Windows Communication Foundation forman parte, en gran medida, del espacio de nombres System.ServiceModel y sus subespacios de nombres. Windows Communication Foundation admite una gran variedad de escenarios, que incluyen:

- Mensajería unidireccional y dúplex
- Llamadas a procedimientos remotos síncronos y asíncronos
- Devoluciones de llamadas
- Sesiones
- Seguridad basada en transporte y mensajes, confiabilidad y entrega ordenada
- Mensajes en cola y compatibilidad con otras transacciones

Windows Workflow Foundation

Es una nueva plataforma de desarrollo de flujo de trabajo basada en el .NET Framework. Proporciona un modelo de programación para desarrollar y ejecutar una amplia variedad de aplicaciones de flujo de trabajo persistentes, con estado y de larga duración. Brinda una funcionalidad de flujo de trabajo de fábrica para desarrollar fácilmente aplicaciones como, por ejemplo, administración de documentos, flujo de páginas comerciales, administración de IT y distintos programas de línea de negocios.



Windows CardSpace

Microsoft creó una nueva herramienta que nos permite concentrar y administrar en un solo lugar todos nuestros datos de identificación, por ejemplo, para sitios web.

Concepto de identidad digital

Para entender este concepto, tomemos como ejemplo las tarjetas que podemos tener en nuestra billetera: tarjetas de crédito expedidas por un banco, que llevan, además de nuestro nombre, un número único y alguna firma de seguridad de parte de la entidad y/o del usuario; identificación personal (o cédula, como se denomina en algunos países), extendida por el sistema gubernamental de nuestro país con información variable de cada persona; quizás alguna tarjeta de presentación personal hecha por nosotros mismos, con nuestro teléfono y dirección; y otra tarjeta de presentación de la empresa en donde trabajemos. Todas estas tarjetas indican quiénes somos, para quién trabajamos y a quién representamos. Nuestros datos en general (variable siempre) y otros privados que ni siquiera nosotros mismos determinamos (como el límite de crédito de nuestra tarjeta, que tiene el banco) están en nuestra billetera, de manera que siempre “escogemos” qué tarjeta usar.

De manera similar trabaja CardSpace, un cliente en el cual tenemos medios de identificación

Diferentes contextos requieren distintas identidades, cada una de las cuales es expresada de una manera diferente y contiene distinta información.

o tarjetas personales creadas por nosotros o extendidas por un tercer agente (banco, gobierno o empresa). Cuando decidimos entregar esa tarjeta a la entidad que nos la solicita (sitio web), ésta consulta mediante un servicio Web a la entidad que la entregó (nosotros o una empresa, por ejemplo) y, después, nosotros escogemos qué información enviarle al solicitante de la identificación.

Esto puede prestarse a confusiones, pero CardSpace es, simplemente, la implementación de un sistema de metaidentidad. Otras implementaciones de un sistema de metaidentidad son OpenID (desarrollado en conjunto con VeriSign) o Liberty Alliance. También hay implementaciones del lado de servidores, como Higgins o Heraldry; y en la actualidad ya hay plugins para browsers, como Safari y Firefox.

FIGURA 003 | Con Windows Vista, el administrador CardSpace ya viene integrado en el panel de control.



www.reduserspremium.blogspot.com.ar

Representación de identidades digitales

Las identidades digitales son comunes al ser transmitidas en la red, y cada una es representada por algún tipo de token de seguridad. Un token o ficha de seguridad es sólo un conjunto de bytes que expresan información sobre una identidad digital. La información que incluye consiste en una o más solicitudes, cada una de las cuales lleva una parte de la identidad. Por ejemplo, un token de seguridad simple puede incluir una sola solicitud que contenga el nombre de usuario, mientras que uno más complejo puede llevar solicitudes que tengan nombre y apellido, dirección, mail, etc. Algunos tokens pueden contener información delicada o sensible, como números de tarjetas de crédito.

Aunque la idea básica de cada token de seguridad es la misma, hoy en día se utiliza una variedad de formatos para representarlos. El ejemplo más simple es un nombre de usuario representado como una cadena de texto, pero también hay otros más complejos, como certificados X.509 y tickets Kerberos. Ninguno de ellos fue diseñado para permitir el transporte de un conjunto arbitrario de solicitudes, pero todos son bastante útiles para algunas identidades digitales. Los tokens generados usando *Security Assertion Markup Language* (SAML), un estándar creado por el grupo OASIS, permiten lograrlo. Basado en XML, SAML puede usarse para tokens de seguridad que contengan cualquier conjunto requerido de solicitudes.

Token de seguridad



www.reduserspremium.blogspot.com.ar

FIGURA 004 | Ejemplo de un token de seguridad que contiene una o más solicitudes.

USERS

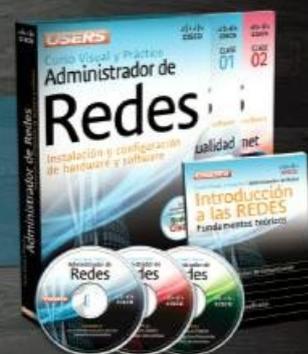


CURSOS.REUSERS.COM

CURSOS INTENSIVOS

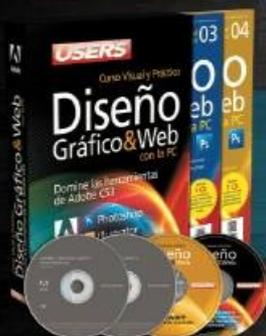


Los temas más importantes del universo de la tecnología desarrollados con la mayor profundidad y con un despliegue visual de alto impacto: Explicaciones teóricas, procedimientos paso a paso, videotutoriales, infografías y muchos recursos mas.



Brinda las habilidades necesarias para planificar, instalar y administrar redes de computadoras de forma profesional. Basada principalmente en tecnologías Cisco, es una obra actual, que busca cubrir la necesidad creciente de formar profesionales.

- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 3 CDs / 1 Libro



- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 4 CDs

Curso para dominar las principales herramientas del paquete Adobe CS3 y conocer los mejores secretos para diseñar de manera profesional. Ideal para quienes se desempeñan en diseño, publicidad, productos gráficos o sitios web.

Obra teórica y práctica que brinda las habilidades necesarias para convertirse en un profesional en composición, animación y VFX (efectos especiales).

- ▶ 25 Fascículos
- ▶ 600 Páginas
- ▶ 2 CDs / 1 DVD / 1 Libro



- ▶ 26 Fascículos
- ▶ 600 Páginas
- ▶ 2 DVDs / 2 Libros

Obra ideal para ingresar en el apasionante universo del diseño web y utilizar Internet para una profesión rentable. Elaborada por los máximos referentes en el área, con infografías y explicaciones muy didácticas.

www.reduserspremium.blogspot.com.ar

Llegamos a todo el mundo con OCA * y DHL **

✉ usershop@redusers.com ☎ +54 (011) 4110-8700

usershop.redusers.com.ar

** Válido en todo el mundo excepto Argentina. * Sólo válido para la República Argentina

Argentina \$8,90 (recargo al interior \$0,20) / México: \$45

USERS

Microsoft®

Curso teórico y práctico de programación

Desarrollador .net

Con toda la potencia
de **Visual Basic .NET** y **C#**

La mejor forma de aprender
a programar desde cero



Basado en el programa
Desarrollador Cinco Estrellas
de Microsoft

18

.NET 3.0

Componentes adicionales
/ XAML / Expression / Windows
Presentation Foundation / Windows
Communication Foundation /
Windows Workflow Foundation



ISBN 978-987-1347-43-8



9 789871 347438

