

*DENIECI MORENO AZOFEIFA*

PROGRAMACIÓN AVANZADA  
EN WEB

Guía de estudio



UNED

UNIVERSIDAD ESTATAL A DISTANCIA

Institución Benemérita de la Educación y la Cultura





Producción académica  
y asesoría metodológica

*Mario Marín Romero*

Revisión filológica

*Vanessa Villalobos Rodríguez*

Diagramación

*Mario Marín Romero*

Encargado de cátedra

*Enrique Gómez Jiménez*

Esta guía de estudio se confeccionó en la UNED, en el año 2011, para utilizarse en la asignatura Programación avanzada en web, código 3101, que se imparte en el programa de Ingeniería Informática.

*Universidad Estatal a Distancia*

*Vicerrectoría Académica*

*Escuela de Ciencias Naturales y Exactas*



## PRESENTACIÓN

Los negocios por Internet son más comunes cada día y las personas se habitúan rápidamente a ellos. Consecuentemente, los requerimientos en el desempeño y el dinamismo se convierten en los puntos obligatorios por considerar cuando se desarrolla una aplicación web. El desempeño, por cuanto se requieren aplicaciones más rápidas y efectivas; el dinamismo, porque deben considerarse las lógicas contemporánea de los negocios y la funcional y habitual de los usuarios modernos: nuevas formas de negocios, de pago y de adquisición de bienes y servicios.

Le instamos al estudio de esta guía con el fin de que adquiera mejores conocimientos que le permitan iniciarse en la programación web para dispositivos móviles, capacidad que, día con día, es más demandada por billones de personas en el todo el mundo.



## CONTENIDOS

|   |          |
|---|----------|
| Presentación.....   | iii      |
| Objetivos .....   | vii      |
| Descripción .....   | viii     |
| Estructura.....   | ix       |
| <br>  |          |
| <b>Tema 1. Autenticación de usuarios en ASP .NET .....</b>                | <b>1</b> |
| Guía de lectura.....  | 2        |
| Comentarios del tema .....  | 3        |
| Referencias web .....   | 5        |
| Ejercicios de autoevaluación.....   | 6        |
| <br>  |          |
| <b>Tema 2. Establecimiento de la depuración en aplicaciones web .....</b> | <b>7</b> |
| Guía de lectura.....  | 8        |
| Comentarios del tema .....  | 9        |
| Referencias web .....   | 10       |
| Ejercicios de autoevaluación.....   | 10       |

|   |    |
|---|----|
| <b>Tema 3. Estrategias de diseño y programación de código cliente en aplicaciones web</b> ..... | 11 |
| Guía de lectura.....  | 12 |
| Comentarios del tema .....  | 13 |
| Referencias web .....   | 14 |
| Ejercicios de autoevaluación.....   | 15 |
| <br>  |    |
| <b>Tema 4. Programación web asincrónica</b> .....   | 17 |
| Guía de lectura.....  | 18 |
| Comentarios del tema .....  | 19 |
| Referencias web .....   | 22 |
| Ejercicios de autoevaluación.....   | 23 |
| <br>  |    |
| <b>Tema 5. Programación web asincrónica y gestión de datos</b> .....                            | 25 |
| Guía de lectura.....  | 26 |
| Comentarios del tema .....  | 27 |
| Referencias web .....   | 28 |
| Ejercicios de autoevaluación.....   | 28 |
| <br>  |    |
| <b>Tema 6. Estrategia de conectividad de datos mediante contexto de datos</b> .....             | 29 |
| Guía de lectura.....  | 30 |
| Comentarios del tema .....  | 31 |
| Referencias web .....   | 32 |
| Ejercicios de autoevaluación.....   | 32 |
| <br>  |    |
| <b>Tema 7. Desarrollo de aplicaciones para dispositivos móviles</b> .....                       | 33 |
| Guía de lectura.....  | 34 |
| Comentarios del tema .....  | 35 |
| Referencias web .....   | 36 |
| Ejercicios de autoevaluación.....   | 36 |

|  |           |
|--|-----------|
| <b>Tema 8. Aplicaciones para dispositivos móviles que gestionan datos.....</b> | <b>37</b> |
| Guía de lectura.....   | 38        |
| Comentarios del tema .....   | 39        |
| Referencias web .....  | 40        |
| Ejercicios de autoevaluación.....  | 40        |
| <br>   |           |
| <b>Tema 9. Nuevas herramientas para el desarrollo de sitios web .....</b>      | <b>41</b> |
| Guía de lectura.....   | 42        |
| Comentarios del tema .....   | 43        |
| Referencias web .....  | 43        |
| Ejercicios de autoevaluación.....  | 44        |
| <br>   |           |
| Respuestas a los ejercicios de autoevaluación.....                             | 45        |
| <br>   |           |
| Referencia.....  | 50        |



## OBJETIVOS

### *Objetivo general*

- ✓ Desarrollar sitios web mediante el uso de herramientas y tecnologías de avanzada que permitan una implementación efectiva y un desempeño adecuado a las necesidades de las empresas modernas.

### *Objetivos específicos*

- Analizar las diferentes estrategias de autenticación de usuarios en un sitio web.
- Analizar las diferentes estrategias para implementar la depuración en el desarrollo de un sitio web.
- Desarrollar aplicaciones web utilizando modelos de diseño y programación de código cliente.
- Desarrollar aplicaciones web que utilicen ejecución asincrónica basada en el cliente.
- Aplicar la programación asincrónica en el desarrollo de un sitio web que gestione datos.
- Construir un sitio web que gestione datos a través de contextos de datos.
- Crear aplicaciones web con soporte para dispositivos móviles.
- Crear aplicaciones para dispositivos móviles que gestionan datos.
- Analizar nuevas herramientas orientadas al desarrollo de sitios web.



## DESCRIPCIÓN

Este curso orienta al estudiante en el aprendizaje de las técnicas, métodos y estrategias existentes en el desarrollo de aplicaciones web y cómo aplicarlas en soluciones optimizadas para su uso en el ambiente comercial, científico, académico o de cualquier otra área que requiera del procesamiento de información digital. Integra, en su contenido, técnicas avanzadas de creación de scripts que ejecuta el cliente (para optimizar aplicaciones web), uso de componentes que coadyuvan a la lógica de aplicación desde el lado del cliente, aplicaciones para dispositivos móviles, entre otros temas.

Para lograr los objetivos propuestos, se cuenta con el libro del curso titulado Aplicaciones con Visual Basic .NET. ¡Programa para Escritorio, Web y Dispositivos móviles! (1<sup>o</sup> edición), del autor Enrique Gómez Jiménez. La presente guía apoyará en el estudio de ese texto.





## ESTRUCTURA

Esta guía se ha estructurado con el fin de facilitar el estudio del libro en el cumplimiento de los objetivos trazados. Para cada tema, se incluyen diversas secciones que debe considerar:

### *Guía de lectura*

Le indica las páginas respectivas del libro que usted leerá y estudiará para cada tema y subtema.

### *Comentarios generales*

Esta parte brinda aspectos importantes del tema así como su ubicación dentro de cada capítulo del libro. Los comentarios le servirán para sintetizar los conceptos presentados; de esta manera, usted podrá determinar si requiere repasar o aclarar alguna de las ideas antes de desarrollar los ejercicios.

### *Referencias web*

En este apartado, se incluyen algunas referencias web que reforzarán los temas de estudio. Puede utilizarlas para retroalimentar algunos conceptos o, simplemente, para ampliarlos.

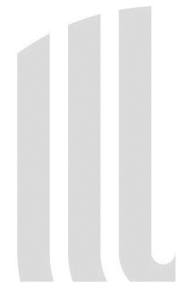
### *Preguntas de autoevaluación*

Las interrogantes se presentan con el propósito de que usted autoevalúe su nivel de comprensión y aprendizaje del tema en estudio a través de una serie de preguntas al final de cada capítulo, cuyas respuestas encontrará en las páginas finales de esta guía.

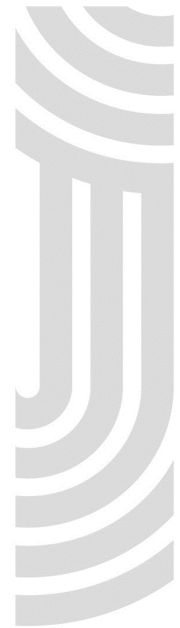


AUTENTICACIÓN DE USUARIOS  
EN ASP .NET

---



**1**





## INTRODUCCIÓN

Los temas del capítulo 1 tienen la intención de exponer las diferentes estrategias que ofrece *Microsoft* para que los usuarios de una aplicación web se autenticuen en ella. Las tres formas ofrecidas utilizan la plataforma .NET para establecer la accesibilidad a un sitio web, considerando reglas de seguridad que *Microsoft* implementa en sus productos de *software*.

Al finalizar el estudio del contenido de este capítulo, usted estará en capacidad de analizar las diferentes estrategias de autenticación de usuarios en un sitio web.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas del libro que se indican a continuación:

| Capítulo   | Temas del capítulo                           | Páginas |
|--|--|---------|
| 16.<br>Autenticación<br>de usuarios en<br>ASP .NET | 16.1. Seguridad en aplicaciones web ASP .NET | 425-426 |
|  | 16.2. Autenticación basada en Windows        | 426-430 |
|  | 16.3. Autenticación basada en formularios    | 430-432 |
|  | 16.4. Autenticación de Microsoft Passport    | 432-434 |



## COMENTARIOS DEL TEMA

### 16.1. Seguridad en aplicaciones web ASP .NET

La seguridad que se implementa en ASP .NET se conjuga perfectamente con Microsoft Information Server (IIS), empleando un mecanismo para avalar credenciales del usuario, como nombres y contraseñas mediante autenticación Windows integrada (NTLM o Kerberos), o formularios mediante Microsoft Passport. También utiliza la funcionalidad de Microsoft .NET Framework para proporcionar aplicaciones web seguras.

Se debe tener claro dos elementos fundamentales de la seguridad: la autenticación la y autorización. Por un lado, la primera comprueba que el usuario es precisamente quien dice ser, mediante el uso de credenciales; por otro lado, la segunda limita los derechos de acceso del usuario, mediante la concesión o negativa de ejecución o uso de alguna de las funcionalidades de una aplicación.

En breve, se hace una reseña de estos mecanismos de autenticación.

### 16.2. Autenticación Windows

Esta, en una aplicación web ASP .NET, utiliza el sistema de seguridad y autenticación en un servidor de red, el cual es configurado en IIS para que utilice dicho mecanismo. En primer lugar, el sistema operativo valida al usuario en la red y, posteriormente, IIS determina el perfil de este en el sistema aplicativo web. Esto permite, lógicamente, una gran ventaja, pues se utiliza la infraestructura de seguridad implementada en la red Windows, las reglas de acceso a los aplicativos de la red y de la información confidencial. Funciona a nivel de una intranet corporativa, por lo tanto su uso en un aplicativo web externo no es funcional.

Un ejemplo de autenticación Windows es el siguiente:

```
<configuration>
  <system.web>
    <authentication mode="Windows"/>
  </system.web>
</configuration>
```

### 16.3. Autenticación basada en formularios

Mediante formularios web ASP .NET, se puede autenticar las credenciales de un usuario. El ASP .NET gestiona las *cookies* de un usuario, y lo redirecciona hacia una página de registro cuando no esté autenticado. Si las credenciales son comprobadas, se crea la interfaz adecuada al perfil del usuario:

```
<configuration>
  <system.web>
    <authentication mode="Forms"/>
  </system.web>
</configuration>
```

### 16.4. Autenticación de Microsoft Passport

Esta utiliza un servicio proporcionado por Microsoft a los suscriptores. El servicio permite conectarse a un proveedor del servicio y éste le habilita el acceso a los sitios suscritos. Windows Vista no soporta este tipo de autenticación.

Entre las ventajas de utilizar este mecanismo, se encuentran:

- a) Hay un único proceso de autenticación para cualquier aplicativo.
- b) No hay bases de datos que almacenen los datos del usuario.
- c) Se puede personalizar el aspecto de la página de inscripción.

Microsoft especifica el mecanismo para implementar Passport de la siguiente manera:

1. Un cliente emite una solicitud HTTP GET para un recurso protegido, como `<http://www.contoso.com/default.aspx>`.
2. Se examinan las *cookies* del cliente para comprobar si existe un vale de autenticación de Passport. Si el sitio detecta credenciales válidas, autenticará al cliente. Si la solicitud no incluye un vale de autenticación legítimo, el servidor devolverá el código de estado 302 y redirigirá el cliente al servicio de inicio de sesión de Passport. La respuesta incluye una dirección URL en la cadena de consulta enviada al servicio de inicio de sesión de Passport para devolver el cliente al sitio original.



3. El cliente sigue la redirección, emite una solicitud HTTP GET al servidor de inicio de sesión de Passport y transmite la información de la cadena de consulta del sitio original.
4. El servidor de Passport presenta al cliente un formulario para inicio de sesión.
5. El cliente rellena el formulario y realiza una devolución automática POST al servidor de inicio de sesión, utilizando Secure Sockets Layer (SSL).
6. El servidor de inicio de sesión autentica al usuario y lo redirige de vuelta a la dirección *URL* original (<<http://www.contoso.com/default.aspx>>). La respuesta incluye una *cookie* codificada de Passport en la cadena de consulta.
7. El cliente sigue la redirección y solicita de nuevo el recurso protegido original (esta vez con la *cookie* de Passport).
8. En el servidor de origen, `PassportAuthenticationModule`, se detecta la presencia de la *cookie* de Passport y se comprueba la autenticación. Si la prueba se realiza correctamente, se autenticará a continuación la solicitud.

Vea los mecanismos autenticación de usuarios en ASP .NET, en las páginas de la 425 a la 434 del libro de texto.

#### REFERENCIAS WEB PARA EL TEMA 1

- <<http://msdn.microsoft.com/es-es/library/330a99hc%28VS.80%29.aspx>>.
- <<http://www.es-asp.net/tutoriales-asp-net/tutorial-61-118/autenticacion-y-autorizacion.aspx>>.
- <[http://www.elguille.info/colabora/puntoNET/lfoixench\\_AutenticacionUsandoFormularios.htm](http://www.elguille.info/colabora/puntoNET/lfoixench_AutenticacionUsandoFormularios.htm)>.
- <<http://technet.microsoft.com/es-es/library/cc759464%28WS.10%29.aspx>>.
- <<http://www.elguille.info/NET/ASPNET/tutorialLogin/tutorialLogin.htm>>.
- <[http://sisinfo.itc.mx/ITC-APIRGG/Fundamentos\\_ASPdotNet/capitulo14.htm](http://sisinfo.itc.mx/ITC-APIRGG/Fundamentos_ASPdotNet/capitulo14.htm)>.
- <<http://www.jasoft.org/blog/PermaLink,guid,e7174bfd-93d2-47f8-8a14-9c4db2cffd63.aspx>>.
- <<http://www.es-asp.net/tutoriales-asp-net/tutorial-61-1534/autenticacion-basada-en-formularios.aspx>>.



- <<http://www.dotnetpuebla.com/portal/Publicaciones/Articulos/245.aspx>>.
- <[http://www.programacion.com/articulo/autenticacion\\_basada\\_en\\_formularios\\_275](http://www.programacion.com/articulo/autenticacion_basada_en_formularios_275)>.

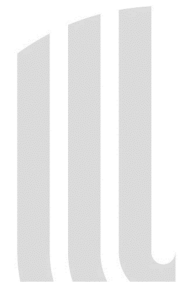
## EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 1

- 1) Cite y explique los tres mecanismos fundamentales de la autenticación de usuarios en aplicaciones web ASP.NET.
- 2) ¿Cuáles son las ventajas y las desventajas de utilizar la autenticación basada en Windows?
- 3) Cite los tres primeros pasos requeridos para habilitar la autenticación basada en Windows.
- 4) Cite tres razones por las cuales Microsoft recomienda la autenticación basada en Windows.
- 5) Cite los dos primeros pasos para habilitar la autenticación basada en formularios.
- 6) Cite las ventajas y desventajas de la autenticación basada en Microsoft Passport.

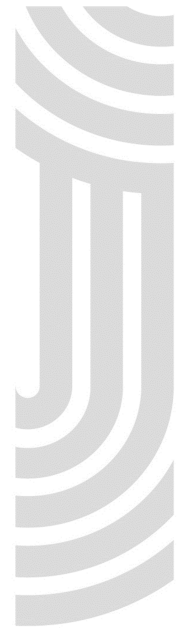


ESTABLECIMIENTO DE LA DEPURACIÓN  
EN APLICACIONES WEB

---



**2**





## INTRODUCCIÓN

El capítulo 2, resumidamente, explica cómo ASP.NET aprovecha la infraestructura de depuración en tiempo de ejecución desde un servidor web, a nivel local o remoto, entre lenguajes y equipos, así como la forma en que ASP.NET proporciona un modo de seguimiento que incluye mensajes alusivos en los formularios de un sitio web.

Los objetos depurados pueden ser administrados o no, así como aquellas aplicaciones que hayan sido desarrolladas con lenguajes compatibles con CLR del .NET Framework de Microsoft.

Al finalizar el estudio del contenido de este capítulo, usted estará en capacidad de analizar las diferentes estrategias para implementar la depuración en el desarrollo de un sitio web.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas del libro que se indican a continuación:

| Capítulo  | Temas del capítulo                               | Páginas |
|---|--|---------|
| 17.<br>Establecimiento de la depuración en aplicaciones web ASP.NET | 17.1. Depuración en sitios web ASP .NET          | 435-438 |
|   | 17.2. Uso del objeto Trace                       | 438-440 |
|   | 17.3. Visualización de resultados de seguimiento | 440-441 |



## COMENTARIOS DEL TEMA

### 17.1. Depuración en sitios web ASP.NET

La depuración de aplicaciones web ASP .NET captura los errores de una aplicación web, ya sea en tiempo de compilación o de ejecución.

En tiempo de ejecución, se debe utilizar el objeto Trace o Debug en el sistema de depuración de Visual Studio .NET, para establecer el seguimiento y los mensajes informativos del aplicativo.

Mediante el uso del objeto Trace de Microsoft ASP.NET, se muestra la información de una página web o se guarda en la memoria, mientras que con el objeto Debug se presentan datos de la depuración en la ventana de resultados del depurador.

### 17.2. Uso del objeto Trace

Mediante el objeto Trace, se escriben mensajes de seguimiento a la depuración de una página web. Para ello, se utiliza Trace.Write (escribe los resultados del seguimiento de la depuración en el editor) o Trace.Warn (los escribe en la memoria).

### 17.3. Visualización de resultados de seguimiento

Existen varias formas de ver los resultados del seguimiento, según categorías. Entre estas formas, se encuentran:

- Detalles de la petición
- Información de seguimiento
- Árbol de control
- Colección de *cookies*
- Colección de *headers*
- Colección de *form*
- Variables de servidor

|  |
|--|
| Vea los detalles de estas categorías en la página 440 del libro de texto |
|--|



## REFERENCIAS WEB PARA EL TEMA 2

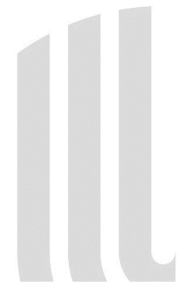
- <<http://www.slideshare.net/guest3cf6ff/5-depuracion-y-correccion-de-errores>>.
- <<http://msdn.microsoft.com/es-es/library/system.diagnostics.trace%28v=VS.100%29.aspx>>.
- <<http://www.comunidadesdeusuarios.net/cursos/aspnet35.aspx>>.

## EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 2

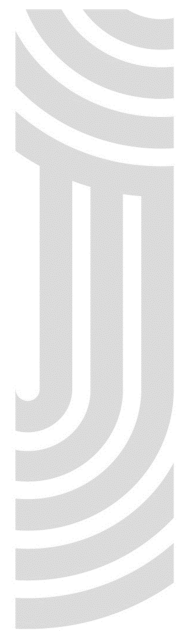
- 1) ¿Qué es la depuración en ASP.NET?
- 2) Cite las dos formas de habilitar el seguimiento de depuración de ASP.NET y su funcionamiento.
- 3) ¿Cuál es la funcionalidad del objeto Trace?
- 4) Cite y explique dos categorías de visualización de resultados de seguimiento.

ESTRATEGIAS DE DISEÑO Y  
PROGRAMACIÓN DE CÓDIGO CLIENTE  
EN APLICACIONES WEB

---



**3**





## INTRODUCCIÓN

Este capítulo explica, teórica y programáticamente, algunas de las metodologías de diseño de interfaces y desarrollo de código fuente en aplicaciones web a nivel de *script* de cliente. A nivel de interfaces, se analiza CSS y, a nivel de implementación, se usa JavaScript y JQuery, que son tecnologías de gran impacto en el desarrollo de lógica en los *script* de cliente.

Al finalizar el estudio del contenido de este capítulo, usted estará en capacidad de desarrollar aplicaciones web utilizando modelos de diseño y programación de código cliente.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas del libro que se indican a continuación:

| Capítulo                           | Temas del capítulo              | Páginas |
|------------------------------------|---------------------------------|---------|
| 18.<br>CSS, JavaScript<br>y JQuery | 18.1. CSS                       | 443-455 |
|                                    | 18.2. Introducción a JavaScript | 455-467 |
|                                    | 18.3. ¿Qué es JQuery?           | 467-475 |



## COMENTARIOS DEL TEMA

### 18.1. CSS

Las hojas de estilo en cascada, conocidas como CSS, permiten controlar la apariencia y la ubicación de objetos en una página web; representan un importante avance, en cuanto al mejoramiento del diseño de las páginas web, con lo que se deja atrás el rústico sistema de presentación de interfaces que se lograba con HTML. Con CSS, se especifica cuánto espacio se deja entre líneas, las sangrías de las líneas, colores por utilizar en texto y fondo, tamaño y tipo de letra, entre otros muchos detalles.

Mediante CSS, se define el estilo visual del documento (presentación), separando la parte semántica (HTML). En lugar de utilizar tablas para ubicar los objetos en una página web, se utiliza CSS.

Hay tres maneras de asociar las reglas de estilos a las marcas de HTML:

1. Directamente en las marcas HTML. Esta forma define el estilo mediante la propiedad *style* en el código HTML; sin embargo, poco recomendada.
2. En la página. Cuando las reglas de estilo se definen en la cabecera de la tabla; sin embargo, esta forma se establecería en cada página y redundaría en muchas de las páginas que conforman el sitio web.
3. En un archivo independiente. Es la forma más utilizada. Se crea un archivo CSS de forma independiente y, luego, se referencia a él en la página HTML.

### 18.2. Introducción a JavaScript

JavaScript permite extender las capacidades del lenguaje HTML. En realidad, no es un lenguaje de programación tal como C++, Visual Basic o Delphi, pero sí es un lenguaje orientado a documentos de un documento web, es decir, con JavaScript no se puede desarrollar *software* que se ejecute fuera de un navegador web. Un *script* de Java se embebe en una página web HTML y, luego, un navegador lo ejecuta.

JavaScript cuenta con instrucciones que permiten ejecutar una serie diversa de acciones. Tales procedimientos pueden ser desde un *document.write*, una asignación como *var x = 0* o un condicionante *if x < y*. También, se pueden crear funciones que encapsulan lógica en un mismo procedimiento, constante, variable, clases o control de flujos.



Vea como se manejan instrucciones, ciclos, funciones y otros detalles de JavaScript en las páginas del libro 457 a la 467.

### 18.3. ¿Qué es JQuery?

JQuery es un marco de trabajo cuyas funciones permiten interactuar con documentos HTML e implementan una poderosa herramienta de manejo de eventos, animaciones entre otras funcionalidades, como AJAX. Utilizando JQuery con *JavaScript*, se crea una poderosa herramienta de manejo de código a nivel de cliente.

JQuery tiene una sintaxis muy sencilla de implementar, y la poca extensión de su código hace que el desarrollador de aplicaciones web se sienta cómodo al utilizarlo.

JQuery proporciona las siguientes ventajas:

- Ahorro de líneas de código
- Soporte transparente para los principales navegadores actuales
- Mecanismos para la captura de eventos
- Funciones de animación de contenidos
- Integración funcional con AJAX

Vea cómo implementar eventos, efectos visuales y otros detalles de JQuery en las páginas del libro 473 a la 475.

### REFERENCIAS WEB PARA EL TEMA 3

- <http://html.conclase.net/w3c/html401-es/present/styles.html>.
- <http://www.htmlquick.com/es/tutorials/css.html>.
- <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.
- <http://www.sidar.org/recur/desdi/mcss/index.php>.
- <http://www.javascriptya.com.ar/temarios/descripcion.php?cod=2>.
- <https://developer.mozilla.org/es/JavaScript>.





- `<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>`.
- `<http://www.cristalab.com/tutoriales/tutorial-de-jquery-c214l/>`.
- `<http://www.javascriptya.com.ar/jquery/temarios/descripcion.php?cod=57&punto=1&inicio=0>`.

### EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 3

- 1) Defina el concepto de CSS.
- 2) Cite las tres formas de asociar las reglas de CSS en un documento HTML.
- 3) ¿A qué forma de implementación de CSS pertenece siguiente fragmento de código?

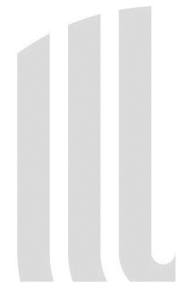
```
<html>
<head>
<title>Titulo</title>
<style type="text/css">
H1{color:White;background-color:blue}
H2{ color:yelow;background-color:black}
...
</style>
```

- 4) ¿Qué es JavaScript?
- 5) ¿Para qué se utiliza la instrucción `document.write`?
- 6) Escriba una función que devuelve el factorial de un número.
- 7) ¿Cuál es la funcionalidad de JQuery?

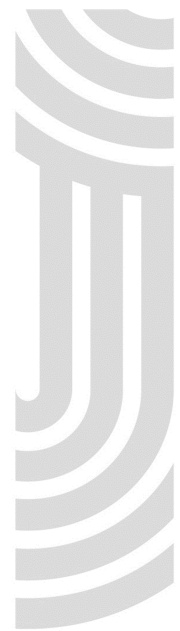


PROGRAMACIÓN WEB  
ASINCRÓNICA

---



**4**





## INTRODUCCIÓN

La programación web asincrónica introduce al funcionamiento de AJAX, una novedosa forma de gestionar la lógica funcional desde el lado del cliente en una aplicación web. El poder de su utilidad radica en la optimización de los recursos del lado del cliente, ya que envía a procesar al servidor web, solamente los datos que se requieran actualizar. En este capítulo, se incluyen los temas más importantes de AJAX, lo cual proveerá el conocimiento necesario para desarrollar aplicaciones web con un rendimiento optimizado.

Al finalizar el capítulo, usted estará en capacidad de desarrollar aplicaciones web que utilicen ejecución asincrónica basada en el cliente.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas que se indican a continuación:

| Capítulo                      | Temas del capítulo   | Páginas |
|-------------------------------|--|---------|
| 19.<br>Fundamentos<br>de AJAX | 19.1. Introducción a las técnicas de AJAX                            | 479-483 |
|                               | 19.2. Información general sobre la representación parcial de páginas | 483-498 |
|                               | 19.3. Administrar el historial del explorador                        | 498-500 |
|                               | 19.4. Servicios web en AJAX  | 500-507 |
|                               | 19.5. Comunicación cliente-servidor para los clientes                | 507-518 |
|                               | 19.6. Depuración y traza de aplicaciones habilitadas para AJAX       | 518-525 |
|                               | 19.7. <i>Script</i> compuesto  | 525-529 |
|                               | 19.8. Globalización y localización de <i>scripts</i>                 | 529-534 |



## COMENTARIOS DEL TEMA

### 19.1. Introducción a las técnicas de AJAX

AJAX no es una tecnología por sí misma, sino más bien la conjunción de varias de ellas para conformar una estrategia para el desarrollo de aplicaciones web basadas en el asincronismo; es decir, AJAX potencia la posibilidad de utilizar JavaScript y XML para cargar y renderizar una página web con el fin de que sólo actualice las áreas de esta que el servidor deba actualizar.

Vea los componentes de AJAX en la página 480 del libro.

El funcionamiento de *AJAX* se basa en realizar solicitudes, de aquella información que obligatoriamente deba ser actualizada, al servidor desde una página web. Muchas son las bondades de *AJAX* en este contexto.

Vea las prestaciones de AJAX en la página 481 del libro.

La arquitectura de AJAX se basa en dos partes: las bibliotecas de *scripts* del cliente y los componentes del servidor. Los *scripts* radican en el cliente, proporcionando una lógica depurada que permite procesar código radicado en la página del cliente. Los componentes de servidor permiten aislar cada componente para que esté en la mira de lo que es necesario actualizar en un determinado lapso de tiempo o producto de una ejecución de la lógica del *script*.

### 19.2. Información general sobre la representación parcial de páginas

La idea principal de AJAX es evitar actualizar toda la página de una aplicación web cada vez que hace una solicitud al servidor. Pretende, más bien, solo actualizar aquella zona de la página que el servidor deba actualizar; ahí entra en juego AJAX.

ASP.NET permite la representación parcial de páginas, nuevas o existentes, sin necesidad de *script* de cliente. Los escenarios que se presentan en este contexto se dan por la extensibilidad de componentes AJAX en ASP.NET.

Vea los escenarios de la extensibilidad de AJAX en ASP.NET en la página 484 del libro.



La representación parcial de páginas mediante AJAX en ASP.NET se lleva a cabo mediante controles AJAX ASP.NET y funciones cliente, también de AJAX. Los controles de servidor más conocidos son el *ScriptManager* y *UpdatePanel*. Asimismo, es posible utilizar *script* de clientes para la actualización de páginas mediante la clase *PageRequestManager*.

Vea las características principales de AJAX en ASP.NET en la página 484 del libro.

### 19.3. Administrar el historial del explorador

Con AJAX se pueden realizar varias tareas en una misma página web mediante solicitudes asincrónicas. Por ejemplo, si está navegando en Google (o muchos sitios similares) puede ir hacia adelante o hacia atrás de las páginas del sitio y observar los estados de las páginas en los momentos que navega. Dicho estado puede guardarse en un historial de las páginas visitadas. También, se registra el estado en que quedó la página una vez que se retrocedió o avanzó en el sitio.

Con AJAX se pueden crear puntos de historial. Un punto del historial representa la navegación lógica en la aplicación web que se puede representar mediante la información del estado; para poder realizar esto, se utiliza el control de servidor *ScriptManager*.

### 19.4. Servicios web en AJAX

Los servicios web se comunican con una aplicación web por medio de invocaciones o solicitudes de un cliente. Los datos que se intercambian entre un cliente y un servidor en la tecnología AJAX son asincrónicos y utilizan un formato *JSON*. Mediante JavaScript, se resuelve la funcionalidad del lado del cliente y la comunicación con el servidor gracias a servicios web que usan un intercambio XML.

Vea un ejemplo aplicativo de servicios web con AJAX en las páginas 501 a 507 del libro.



#### 19.5. Comunicación cliente-servidor para los clientes de AJAX

Si un sitio web posee una página habilitada para AJAX, cuando esta realiza una solicitud a un servidor, lo hace mediante datos asincrónicos a los servicios web y, posteriormente, el servidor los descarga a través de clases *proxy*. Estos llamados se realizan de dos formas: con la invocación del verbo *POST* de HTTP o mediante *GET* de HTTP.

Vea la arquitectura de cliente de *AJAX* en las páginas 508 a 509 y de servidor en las páginas 510 a 511 del libro.

#### 19.6. Depuración y traza de aplicaciones habilitadas para AJAX

En aplicaciones web ASP.NET que utilizan AJAX, el código de servidor y de cliente interactúan y, también, se unen las solicitudes que el explorador que aloja estas aplicaciones pueda realizar. Por ende, la depuración es un completo desafío para este tipo de aplicaciones. Los escenarios que se presentan en este entorno giran alrededor de:

- a) Depuración habilitada a través del *web.config*
- b) Depuración mediante trazas en el servidor
- c) Utilización de *Sys.Debug* para puntos de interrupción
- d) Habilitar la depuración en el explorador
- e) Asociar el *debug* de Visual Studio en el Internet Explorer
- f) Utilizar herramientas para la captura de tráfico HTTP

Vea la implementación de éstas formas de depuración y un ejemplo aplicativo en las páginas 519 a 525 del libro.

#### 19.7. *Script* compuesto

Utilizar JavaScript en una aplicación web habilitada para AJAX permite gestionar el comportamiento de los controles del cliente mediante un *script*; pero cuando hay varios controles, entonces se necesitarán varios *scripts*.



Con ello, el desempeño de la página puede ser lamentable, por lo tanto, crear un solo archivo con varios *script* entra en juego. Es lo que se conoce como *script* compuesto y se crea usando los controles *ScriptManager* y *ScriptManagerProxy*.

Vea la implementación de un *script* compuesto en las páginas 525 a 529 del libro.

#### 19.8. Globalización y localización de *scripts*

Con la globalización de las comunicaciones, las aplicaciones web también tienden a internacionalizarse y a mostrarse en muchos países donde radican personas de diferentes culturas e idiomas.

Visual Studio .NET implementa localización y globalización para las aplicaciones que deben ser utilizadas en esos escenarios. Los espacios de nombres que permiten esta creación de escenarios son *System.Globalization*, *System.Resources* y *System.Text*.

En este contexto, globalización es el proceso de diseñar y desarrollar aplicaciones que se adaptan a diversas culturas, incluso interfaces de usuarios y datos regionales.

Vea más detalles sobre globalización y localización y un ejemplo aplicativo en las páginas 530 a 534 del libro.

#### REFERENCIAS WEB PARA EL TEMA 4

- <<http://msdn.microsoft.com/es-es/library/cc488553.aspx>>.
- <<http://channel9.msdn.com/Blogs/keydet/Managing-Browser-History-Using-Client-Script-Without-ScriptManager>>.
- <<http://geeks.ms/blogs/ozonicco/archive/2006/12/10/globalizaci-243-n-y-localizaci-243-n-usando-microsoft-net-parte-i.aspx>>.
- <<http://msdn.microsoft.com/es-es/library/dd997383.aspx>>.





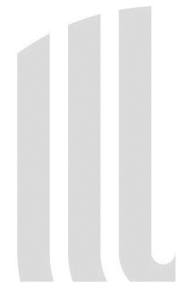
#### EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 4

- 1) ¿Cuál es el concepto principal de AJAX, en cuanto a la ejecución de páginas web en el servidor?
- 2) Cite al menos dos prestaciones funcionales AJAX que se ofrece para ASP.NET.
- 3) Cite las dos partes que conforman la arquitectura de un servidor AJAX.
- 4) Cite las dos características que proporciona la compatibilidad de los *scripts* de AJAX en ASP.NET.
- 5) Escriba la parte del *script* *AJAX* que permite activar la renderización de páginas y, así, minimizar el uso de *PostBack*.

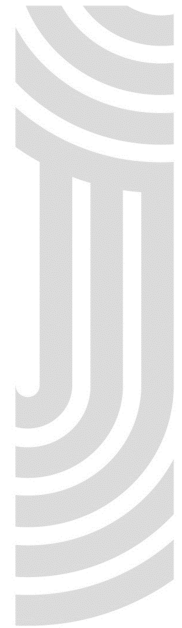


PROGRAMACIÓN WEB  
ASINCRÓNICA Y GESTIÓN DE DATOS

---



5





## INTRODUCCIÓN

Este tema se refiere al funcionamiento de AJAX y la gestión de datos en una aplicación web ASP.NET; asimismo, comprender su funcionalidad en el desarrollo de aplicaciones web ASP.NET, los objetos AJAX que pueden utilizarse en este desarrollo y la funcionalidad que significa la representación parcial de páginas.

Al finalizar el estudio del contenido de este capítulo usted estará en capacidad de aplicar la programación asincrónica en el desarrollo de un sitio web que gestione datos; para ello, utilizará AJAX.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas que se indican a continuación:

| Capítulo                | Temas del capítulo   | Páginas |
|-------------------------|--|---------|
| 20<br>AJAX y<br>ADO.NET | 20.1. <i>ScriptManager</i>   | 538-541 |
|                         | 20.2. <i>ScriptManagerProxy</i>  | 542-544 |
|                         | 20.3. Ejemplo 1: Utilización de AJAX en aplicaciones con funcionalidad ADO.NET | 545-558 |
|                         | 20.4. Ejemplo 2: Uso de AJAX en aplicaciones que utilizan ADO.NET              | 559-566 |



## COMENTARIOS DEL TEMA

### 20.1. *ScriptManager*

Mediante el objeto *ScriptManager*, es posible administrar el *script* del cliente; es decir, la funcionalidad de una aplicación web que se implementa del lado del cliente. Con este objeto se administran áreas de una página web que se encuentran en controles *UpdatePanel*; con ello, se implementa la representación parcial de páginas.

Vea la funcionalidad de *ScriptManager* de AJAX en las páginas 538 a 541 del libro.

### 20.2. *ScriptManagerProxy*

Un control *ScriptManagerProxy* permite que páginas de contenido y controles de usuario que se han incluido en una página web, agreguen a ésta referencias a *scripts* y servicios, siempre y cuando cuenten en ella con un control *ScriptManager* que los administre.

Vea la funcionalidad de *ScriptManagerProxy* de AJAX en las páginas 542 a 544 del libro.

### 20.3. Ejemplo 1: utilización de AJAX en aplicaciones con funcionalidad ADO.NET

En este ejemplo, se utiliza una base de datos SQL Server y con *LinqDataSource* se establecerá la conectividad de la aplicación web ASP.NET. Por medio de un objeto *UpdatePanel* se administrará en su página, la actualización, según las opciones que el usuario seleccione.

### 20.4. Ejemplo 2: uso de AJAX en aplicaciones que utilizan ADO.NET

Este ejemplo es similar al del punto 20.3. con la diferencia de que existen dos *UpdatePanel* para la actualización de páginas, según la necesidad del usuario (actualizar el cliente o los datos de la cuenta del cliente).



## REFERENCIAS WEB PARA EL TEMA 5

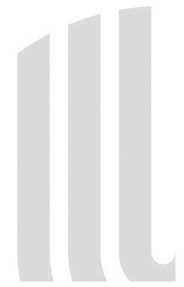
- <<http://msdn.microsoft.com/es-es/library/system.web.ui.scriptmanager.aspx>>.
- <<http://www.wrox.com/WileyCDA/Section/Using-the-ASP-NET-AJAX-ScriptManager.id-305492.html>>.
- <<http://www.aspxcode.net/free-asp-net-ajax-source-code.aspx?Topics=How%20to%20Use%20ScriptManager>>.
- <<http://msdn.microsoft.com/es-es/library/cc295370.aspx>>.
- <[http://r4r.co.in/asp.net/01/tutorial/asp.net\\_ajax/scriptmanagerproxy\\_control.shtml](http://r4r.co.in/asp.net/01/tutorial/asp.net_ajax/scriptmanagerproxy_control.shtml)>.

## EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 5

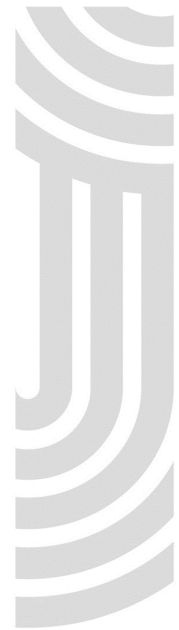
- 1) Explique la funcionalidad de un objeto *ScriptManager*.
- 2) ¿Cuál propiedad debe habilitarse en un control *ScriptManager* para que permita o no la actualización parcial de páginas?
- 3) ¿Qué significa la propiedad *AllowCustomErrorRedirect* cuando está en *true*?
- 4) ¿Para qué se utiliza la propiedad *EnableHistory* en un control *ScriptManager*?
- 5) ¿Para qué se utiliza la propiedad *Triggers* en un control *UpdatePanel*?

ESTRATEGIA DE CONECTIVIDAD DE  
DATOS MEDIANTE CONTEXTO DE DATOS

---



6





## INTRODUCCIÓN

El tema 6 explica el funcionamiento de los contextos de datos, derivados de un esquema relacional. Comprendiendo que existe un modelo relacional datos y una capa de aplicación (en el modelo cliente servidor), el contexto de datos establece una referencia basada en objetos que permite el manejo de ellos mediante un mapeo lógico.

Al finalizar el estudio del contenido de este capítulo, usted estará en capacidad de utilizar un contexto de datos derivado de una base de datos relacional, con un mapeo lógico, objetos y consultas LINQ para el acceso y mantenimiento de los datos.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas que se indican a continuación:

| Capítulo                               | Temas del capítulo   | Páginas |
|--|--|---------|
| 21.<br>LINQ y<br>ADO.NET en<br>ASP.NET | 21.1. Controles de origen de datos   | 570-571 |
|  | 21.2. Controles de enlace de datos   | 571-572 |
|  | 21.3. LINQ   | 572-574 |
|  | 21.4. Ejemplo 1: Aplicación web usando LINQ para el mantenimiento de clientes.         | 574-580 |
|  | 21.5. Ejemplo 2: Aplicación web mediante modelado de datos en una clase independiente. | 581-588 |





## COMENTARIOS DEL TEMA

### 21.1. Controles de origen de datos.

Los controles de origen de datos sirven para automatizar el proceso de acceso y mantenimiento de datos en una aplicación web. Lo importante es conocer las propiedades y métodos que estos controles ofrecen para la gestión de datos. ASP.NET brinda una cantidad considerable de estos controles, como: *AccessDataSource*, *LinqDataSource* y *ObjectDataSoure*.

Vea los controles de origen de datos que se ofrecen para ASP.NET en las páginas 570 a 572 del libro.

### 21.2. Controles de enlace a datos

Así como existen controles de conectividad a datos, también se cuenta con los que permiten presentar los datos al usuario, manteniendo y gestionando un enlace a ellos. Entre estos controles, se hallan *DataPage*, *DataList* y *DetailsView*.

Vea los controles de enlace a datos que se ofrecen para ASP.NET en las páginas 571 a 572 del libro.

### 21.3. LINQ

Este permite manejar las consultas y el mantenimiento de datos en una aplicación por medio de un contexto de datos. Este último es un mapeo de una base de datos relacional, que construye un modelo orientado a objetos; además, procesa las consultas y las operaciones de mantenimiento de datos con su propia sintaxis, la cual es muy similar a la de SQL estándar.

Vea la funcionalidad y operatividad de objetos LINQ en las páginas 572 a 574 del libro.



#### 21.4. Ejemplo 1: aplicación web usando LINQ para el mantenimiento de clientes

En este ejemplo, se demuestra el uso de LINQ para el mantenimiento de datos de una tabla de clientes. Mediante una tabla relacional diseñada en SQL Server 2005, se crea un modelo de clases LINQ sobre la cual se utilizan las instrucciones LINQ para el mantenimiento de datos de la tabla de origen relacional.

#### 21.5. Ejemplo 2: aplicación web con modelado de datos en una clase independiente

En este ejemplo, se demuestra cómo modelar una clase basado en una tabla relacional, y cómo vincularla a un proyecto web para su utilización. Esa clase funciona como el modelo de la clase por utilizar (objeto de contenido). Se edita una plantilla para el *GridView* que se utiliza como objeto de enlace.

### REFERENCIAS WEB PARA EL TEMA 6

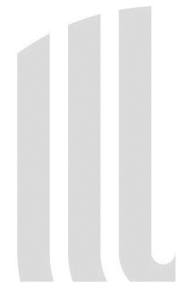
- <<http://msdn.microsoft.com/es-es/library/ms227679%28v=vs.80%29.aspx>>.
- <<http://www.slideshare.net/Tonymx/introduccion-a-linq>>.
- <<http://geeks.ms/blogs/adiazmartin/archive/2010/03/23/operadores-linq.aspx>>.

### EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 6

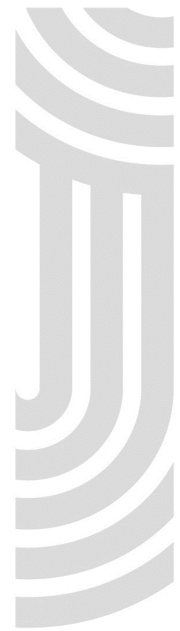
- 1) ¿Para qué se utiliza un control *XMLDataSource*?
- 2) ¿Para qué se utiliza un de acceso *Repeater*?
- 3) Cite y explique dos controles *LINQ* que se pueden utilizar en una aplicación web *ASP .NET*.
- 4) ¿Qué operaciones permite *LINQ* sobre un modelo de datos?
- 5) Cree una consulta *LINQ* para devolver los datos de clientes y empleados que hay en los países que se extraen de la tabla *países*.

DESARROLLO DE APLICACIONES PARA  
DISPOSITIVOS MÓVILES

---



7



## INTRODUCCIÓN

Este tema introduce al desarrollo de aplicaciones para dispositivos móviles. La importancia de este tipo de desarrollo radica en las nuevas tendencias del usuario, a saber: llevar los sistemas informáticos a cualquier lugar que vaya o en las empresas globalizar sus negocios y procesos sin importar la ubicación geográfica donde se encuentre. En este contexto, el desarrollo de estas aplicaciones se une a la globalización de negocios mediante el uso de las TIC, no solo vía web sino también a través de dispositivos móviles.

Al finalizar el estudio del contenido de este capítulo, usted estará en capacidad de crear aplicaciones con soporte para dispositivos móviles.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas que se indican a continuación:

| Capítulo   | Temas del capítulo  | Páginas |
|--|---|---------|
| 22.<br>Desarrollo<br>de<br>aplicaciones<br>para<br>dispositivos<br>móviles | 21.1 .NET Compact Framework                                     | 592-594 |
|  | 21.2 . Clases admitidas en .NET Framework                       | 594-596 |
|  | 21.3 . Ejemplo 1: desarrollo de una aplicación para Pocket PC   | 596-606 |
|  | 21.4 . Ejemplo 2: desarrollo de una aplicación para Smartphone. | 606-612 |



## COMENTARIOS DEL TEMA

### 22.1. .NET Compact Framework

Con este, se pueden generar y ejecutar aplicaciones que hayan sido desarrolladas para dispositivos móviles. Entre estos dispositivos, se encuentran celulares o *Pocket PC*, indispensables en el mundo de las comunicaciones de hoy día para personas comunes o empresas.

Vea los detalles sobre el .NET Compact Framework de Visual Studio .NET en las páginas 592 a 594 del libro.

### 22.2. Clases admitidas en .NET Compact Framework

*.NET Compact Framework* posee su propia arquitectura y, por ende, cuenta con una serie de clases que permiten la creación de aplicaciones orientadas a dispositivos móviles. Entre estas clases, se hallan *SqlServerCE*, que posibilita el manejo de datos *SQL Server* a pequeña escala, o *Microsoft.WindowsCE.Forms*, que permite crear interfaces similares a las creadas con *Windows Forms*.

Vea los detalles sobre las clases admitidas en el .NET Compact Framework de Visual Studio .NET en las páginas 594 a 595 del libro.

### 22.3. Ejemplo 1: desarrollo de una aplicación para *Pocket PC*

En este ejemplo, se demuestra el uso de .NET Framework y su arquitectura para el modelado y creación de una aplicación para dispositivo móvil. Con un emulador, es posible la creación de un prototipo de aplicación, el cual puede ser implementado, posteriormente (y fácilmente), en un dispositivo real.

### 22.4. Ejemplo 2: desarrollo de una aplicación para *Smartphone*

En este ejemplo, también se demuestra el uso de *.NET Framework* y su arquitectura para aplicaciones en *Smartphone*. De igual manera, con un emulador es posible la creación de un prototipo de aplicación para implementarse en otros dispositivos.



#### REFERENCIAS WEB PARA EL TEMA 7

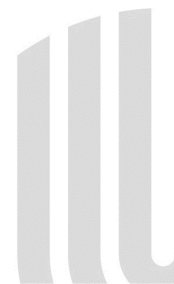
- <http://msdn.microsoft.com/es-es/vbasic/ms789096>.
- <http://javiercancela.com/2007/10/19/introduccion-al-desarrollo-de-aplicaciones-para-telefonos-moviles-windows-mobile/>.

#### EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 7

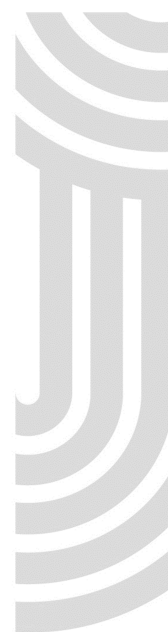
- 1) ¿Qué es el .NET CompactFramework?
- 2) ¿Cuáles son los componentes principales de .NET CompactFramework, según Microsoft?
- 3) ¿Para qué se utiliza la clase *SqlServerCE*?

APLICACIONES PARA DISPOSITIVOS  
MOVILES QUE GESTIONAN DATOS

---



8





## INTRODUCCIÓN

El tema 8 introduce al desarrollo de aplicaciones para dispositivos móviles que gestionen datos. En nuestros días, es imperativo que las aplicaciones ejecutables en cualquier dispositivo móvil estén siempre presentes, por lo cual se hace necesario que administre datos permanentemente.

Al finalizar el estudio del contenido, usted estará en capacidad de crear aplicaciones con soporte para dispositivos móviles que gestionan datos.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas que se indican a continuación:

| Capítulo   | Temas del capítulo   | Páginas |
|--|--|---------|
| 23.<br>Aplicaciones<br>ADO.NET para<br>dispositivos<br>móviles | 23.1 . Ejemplo 1: desarrollo de una aplicación para Pocket PC con gestión de datos | 616-621 |
|  | 23.2 . Ejemplo 2: creación de una aplicación para dispositivo móvil                | 622-643 |
|  | 23.3 . ASP.NET y desarrollo de aplicaciones orientadas a dispositivos móviles      | 643-645 |
|  | 23.4 . WAP   | 646-646 |
|  | 23.5 . WML   | 646-651 |





## COMENTARIOS DEL TEMA

### 23.1. Ejemplo 1: Desarrollo de una aplicación para *Pocket PC* con gestión de datos

Se muestra un ejemplo de aplicación para dispositivo móvil en el que se diseña una base de datos en SQL Server 2005. También, se muestra el procedimiento de exportación de una base de datos SQL Server 2005 mediante un utilitario llamado *GLT.SqlCopy*.

Vea los detalles sobre la creación o conversión de una base de datos SQL Server para usarse en una aplicación para *Pocket PC* en las páginas 616 a 621 del libro.

### 23.2. Ejemplo 2: Creación de una aplicación para dispositivo móvil

Este ejemplo continúa en el punto 23.1. En él, se implementa la funcionalidad requerida para la gestión de los datos. Muestra el uso de SQL Server Compact Edition y las operaciones que, generalmente, se encontrarían en una aplicación para Windows o para la Web.

Vea los detalles sobre la implementación de una aplicación para *Pocket PC* que gestione datos en las páginas 622 a 643 del libro.

### 23.3. ASP.NET y el desarrollo de aplicaciones orientadas a dispositivos móviles

Mediante *System.Web.Mobile*, se desarrollan aplicaciones web habilitadas para ser accedidas desde un dispositivo móvil; por tanto, la lógica para trabajar con *web forms* así como con *MobilePage* tendrán sus semejanzas y, por ende, no será difícil su empleo.

Existen tecnologías para dispositivos móviles que deben ser consideradas en el desarrollo de aplicaciones web habilitadas para estos dispositivos, tales como GSM, GPRS, UMTS, entre otras.

Vea los detalles sobre el desarrollo de aplicaciones orientadas a dispositivos móviles, tipos de dispositivos móviles y tecnologías a considerar, en las páginas 643 a 645 del libro.



#### 23.4. WAP

WAP o Wireless Application Protocol es un protocolo de comunicación que permite el entorno de ejecución de una aplicación web habilitada para dispositivos móviles.

#### 23.5. WML

WML o Wireless Markup Language se utiliza para el desarrollo de aplicaciones web orientadas a ejecutarse en dispositivos móviles. Existe una versión *Script* de este metalenguaje para gestionar lógica en el dispositivo.

Vea los detalles la tecnología WML y la forma de uso en las páginas 646 a 651 del libro.

#### REFERENCIAS WEB PARA EL TEMA 8

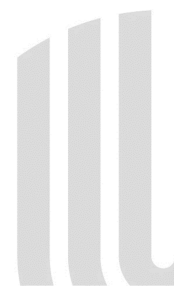
- <<http://altatorre.com/webclase/wml/wml.htm>>.
- <[http://es.wikipedia.org/wiki/Wireless\\_Application\\_Protocol](http://es.wikipedia.org/wiki/Wireless_Application_Protocol)>.
- <<http://www.melodiasmoviles.com/documentacion/intro-gprs.php>>.
- <<http://www.terra.es/tecnologia/articulo/html/tec11333.htm>>.

#### EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 8

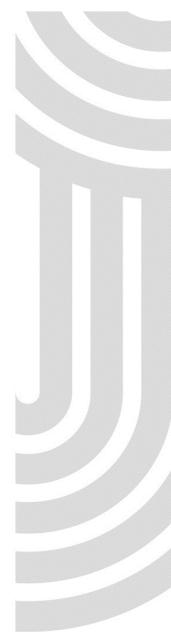
- 1) ¿Para qué se utiliza la clase *System.Web.UI.Mobile.Controls.Adpater*?
- 2) ¿Para qué se utiliza la tecnología GPRS?
- 3) ¿Qué es la tecnología WAP?
- 4) ¿Qué son barajas y cartas en WML?

NUEVAS HERRAMIENTAS PARA EL  
DESARROLLO DE SITIOS WEB

---



9



## INTRODUCCIÓN

Este tema presenta la *suite* de Visual Studio .NET, versión 2010; considera los aspectos trascendentales de aplicaciones para Windows, web y para dispositivos móviles. Este último no fue desarrollado por completo, pues la versión revisada aún no soportaba el desarrollo para dispositivos móviles. Sin embargo, se puede establecer que existe gran compatibilidad de las aplicaciones desarrolladas en Visual Studio .NET 2008 y esta nueva versión.

Al finalizar el estudio del contenido de este capítulo, usted conocerá la nueva *suite* de Visual Studio .NET 2010, su funcionamiento básico y las prestaciones que brinda al desarrollador de aplicaciones para Windows y Web.

## GUÍA DE LECTURA

Para el estudio de este tema, lea detalladamente las páginas que se indican a continuación:

| Capítulo   | Temas del capítulo  | Páginas |
|--|---|---------|
| 24.<br>Microsoft<br>Visual Studio<br>2010: la nueva<br>herramienta | 23.1. Probando la herramienta                                       | 655-659 |
|  | 23.2. Desarrollo de un sitio web ASP.NET en Visual Studio .NET 2010 | 659-665 |
|  | 23.3. Aplicaciones para dispositivos móviles                        | 665-665 |



## COMENTARIOS DEL TEMA

### 24.1. Probando la herramienta

En este apartado, se hace un recorrido por las principales características de Visual Studio .NET 2010 y las funcionalidades que aporta al desarrollador de *software*. Las aplicaciones web desarrolladas en Visual Studio .NET 2008 gozan de una total compatibilidad en esta nueva versión del producto.

Vea los detalles sobre las características de Visual Studio .NET 2010 en las páginas 653 a 659 del libro.

### 24.2. Desarrollo de un sitio web ASP.NET en Visual Studio 2010

Visual Studio .NET 2010 agrega mayor versatilidad y funcionalidad en el entorno de desarrollo de sitios web; brinda la posibilidad de crear menús basados en plantillas.

Vea la funcionalidad de Visual Studio .NET 2010 al crear sitios web mediante un ejemplo en las páginas 659 a 665 del libro.

### 24.3. Aplicaciones para dispositivos móviles

Con base en las pruebas que se realizaron con la beta 2 del producto, se comprobó que no existe soporte de Visual Studio 2010 para las aplicaciones en dispositivos móviles.

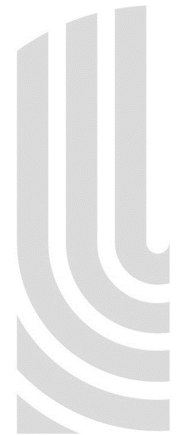
## REFERENCIAS WEB PARA EL TEMA 9

- <<http://www.microsoft.com/downloads/details.aspx?familyid=9CFB2D51-5FF4-4491-B0E5-B386F32C0992&displaylang=es>>.
- <<http://www.asp.net/dynamicdata>>.



## EJERCICIOS DE AUTOEVALUACIÓN DEL TEMA 9

- 1) ¿Cuál es la versión del Framework que incluye Visual Studio .NET 2010?
- 2) ¿Qué es un *squiggles* en Visual Studio .NET 2010?
- 3) ¿Para qué se utiliza un tipo de proyecto *Dynamic Data LINQ To SQL Web Site*?



## RESPUESTAS A LOS EJERCICIOS DE AUTOEVALUACIÓN

### Respuestas a las preguntas de autoevaluación del tema 1

1) Los tres mecanismos son: autenticación basada en Windows, en formularios y en Microsoft Passport. La primera se basa en las credenciales de seguridad que se implementan en Windows, pero que son funcionales en un ambiente de Intranet y no de Internet. La segunda se basa en *cookies* que redireccionan a una página HTML para su autenticación mediante *cookies*. En la última, se utiliza un procedimiento centralizado en Microsoft y que se prevé acceso a un servicio que inicializa un perfil básico a los perfiles aplicativos suscritos.

2) Las ventajas de utilizar autenticación basada en Windows son:

- a) Utilización de la infraestructura de seguridad implementada en la red Windows.
- b) Control de acceso a la información confidencial.
- c) Funcional en un ambiente de Intranet.

La desventaja es que no se puede utilizar en la mayoría de aplicaciones web.

3) Los tres primeros pasos para habilitar la autenticación Windows son:

- a) Establecer los usuarios de la red.
- b) Configurar IIS.
- c) Configurar la autorización en *web.config*.

El resto de pasos los puede consultar en la página 427 del libro de texto.

4) Tres razones son:

- a) Los clientes y servidores web conviven en un mismo entorno.
- b) Los administradores de red pueden estandarizar el uso de exploradores en todos los clientes.
- c) En internet, no es posible establecer un entorno de Active Directory como en Windows Server.

Vea el resto de razones en la página 427 del libro.

5) Los dos primeros pasos para habilitar la autenticación basada en formularios son:

- a) Configurar el modo de autenticación; para ello, modificar la sección de autenticación en el *web.config*.
- b) Denegar el acceso de usuarios anónimos en uno o más directorios de la aplicación web al modificar la sección *authorization* en el archivo *web.config* de los directorios privados.

Los otros dos pasos para esta habilitación se encuentran en la página 431 del libro.

6) Las ventajas de la autenticación basada en Microsoft Passport son:

- a) Existe un único proceso de autenticación para muchos sitios web.
- b) No se necesita base de datos para almacenar los datos del usuario.
- c) Personaliza el aspecto de la página de inscripción.

Las desventajas son:

- a) Se basa en *cookies*.
- b) Hay que pagar una cuota de suscripción.

## **Respuestas a las preguntas de autoevaluación del tema 2**

- 1) La depuración en ASP.NET es un mecanismo que localiza errores, les da seguimiento y les muestra una variedad de formas. Esta variedad permite al desarrollador diagnosticar problemas o analizar el rendimiento de una aplicación web.
- 2) Las dos formas de habilitar el seguimiento en una aplicación ASP.NET son: a nivel de página o a nivel de aplicación. A nivel de página, significa que se puede habilitar el seguimiento para que muestre instrucciones en la página configurada. A nivel de aplicación, se puede habilitar el seguimiento para que aplique a todas las páginas del sitio web.



- 3) Mediante el uso del objeto *Trace*, se pueden escribir mensajes de seguimiento de la depuración en una página web o en la memoria. Se utiliza *Trace.Write* o *Trace.Warn* para establecer la depuración en páginas web ASP.NET.
- 4) Dos categorías de resultados de seguimiento son:
  - a) Detalles de la petición.
  - b) Información de seguimiento.

Para detalles y más categorías, vea la página 440 del libro.

### Respuestas a las preguntas de autoevaluación del tema 3

- 1) CSS son hojas de estilo en cascada que permiten controlar la apariencia y ubicación de objetos en una página de un sitio web en ASP.NET.
- 2) Las tres formas son:
  - a) Directamente en las marcas HTML.
  - b) En el código de cabecera (head) de la página del sitio.
  - c) Agrupando las reglas en un archivo CSS independiente.
- 3) Es mediante definición de reglas de estilo a nivel de página.
- 4) JavaScript es un lenguaje orientado a documentos web HTML y que es interpretado por un navegador y que se utiliza para implementar lógica del lado del cliente de una aplicación web.
- 5) *document.write* se utiliza para que despliegue, por el navegador, un mensaje, el contenido de una variable o constante, o el resultado de un cálculo.
- 6) La función en JavaScript que devuelva un factorial sería:

```
<script language="javascript.4">
function factorial(numero)
{
var fact=1;
var i;
for(i=2;i<=numero;i++)
fact=fact*i;
document.write("Numero :"+num+" Factorial :"+fact<br/>);
}
</script>
</body>
</html>
```

- 7) JQuery simplifica la manera de interactuar con los documentos HTML, permitiendo manejar los eventos, animaciones, entre otras funcionalidades que son requeridas en una página web.

#### **Respuestas a las preguntas de autoevaluación del tema 4**

- 1) La funcionalidad de AJAX con respecto a la ejecución de funciones en el servidor es la de cargar y renderizar las páginas cada vez que sea necesario actualizarlas.
- 2) Las dos prestaciones funcionales AJAX en ASP.NET es la ejecución de procesos importantes del lado del cliente y la actualización de áreas específicas en la página web por parte del servidor. Más opciones en la página 481 del libro de texto.
- 3) La arquitectura AJAX está compuesta por los controles de servidor web ASP.NET y los componentes que administran la interfaz de usuarios y el flujo de una aplicación.
- 4) Las dos características de compatibilidad de los *scripts* de AJAX en ASP.NET son: el sistema de tipos y extensiones JavaScript y la utilización de devolución asíncronica de datos para la actualización de páginas web.
- 5) La parte que permite activar la renderización es `<...EnablePartialRendering="true">`.

#### **Respuestas a las preguntas de autoevaluación del tema 5**

- 1) Permite la administración de script del lado del cliente en una aplicación web ASP.NET.
- 2) La propiedad que debe habilitarse es *EnablePartialRendering*.
- 3) Establecer la utilización de la sección de errores personalizados (*customErrors*) del archivo *web.config* de una aplicación web ASP.NET.
- 4) Se utiliza para habilitar la admisión de puntos del historial en una página web ASP.NET.
- 5) Permite establecer un disparador para el control *UpdatePanel* (que genera la devolución asíncronica de datos).

#### **Respuestas a las preguntas de autoevaluación del tema 6**

- 1) Se utiliza para trabajar con controles de servidor ASP.NET que gestionan archivos XML.
- 2) Se utiliza para representar datos en una lista que se construye mediante una plantilla.
- 3) LINQ To Objects: gestiona colecciones de objetos. LINQ To XML: gestiona archivos XML en memoria. Más detalles en la página 573 del libro.

- 4) Recuperación, filtrado, clasificación y agrupamiento.
- 5) La consulta LINQ sería:

```
Dim datos = _
    (From a In tb.clientes _
     Select a.Pais) _
    .Union _
    (From emp In tb.Empleados _
     Select emp.Pais)
```

### **Respuestas a las preguntas de autoevaluación del tema 7**

- 1) Es un marco de trabajo que permite la generación y ejecución de aplicaciones para dispositivos móviles.
- 2) Bibliotecas de clases, CLR y Windows CE.
- 3) Para implementar la funcionalidad de SQL Server a nivel de aplicación en un dispositivo móvil.

### **Respuestas a las preguntas de autoevaluación del tema 8**

- 1) Para implementar adaptadores de datos en aplicaciones de dispositivos móviles.
- 2) Exclusivamente para la comunicación de datos. Es una evolución de GSM.
- 3) Es un estándar abierto de Internet basado en HTML, XML y TCP/IP.
- 4) Las páginas WML se denominan barajas, cada página es una carta. La navegación entre cartas se conoce como baraja, como en el juego de naipes.

### **Respuestas a las preguntas de autoevaluación del tema 9**

- 1) Framework 4.0.
- 2) Es un pequeño indicativo que muestra un error de validación de la página.
- 3) Para desarrollar un sitio basado en la funcionalidad de LINQ To SQL para la administración dinámica de datos.



## REFERENCIA

Gómez, E. (2010). *Aplicaciones con Visual Basic .NET*. México: Alfaomega.