

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

Evaluación de lenguajes de Script de servidor

Autor: Héctor Ulises Gaona Campos

**Tesina presentada para obtener el título de:
Ingeniero en Sistemas Computacionales**

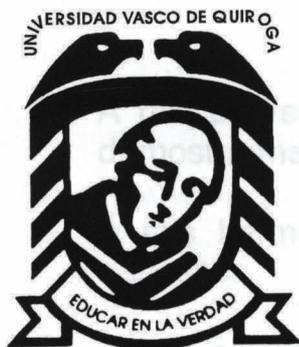
**Nombre del asesor:
Gabriel Nava Vázquez**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





UNIVERSIDAD VASCO DE QUIROGA

ESCUELA DE SISTEMAS COMPUTACIONALES

“ EVALUACIÓN DE LENGUAJES DE SCRIPT DE SERVIDOR ”

TESINA

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTA:
HÉCTOR ULISES GAONA CAMPOS

ASESOR:
M.C. GABRIEL NAVA VÁZQUEZ

CLAVE: 16PSU0049F

MORELIA, MICHOACÁN, MÉXICO

MARZO DE 2006.



Dedicatoria

Índice

1.1	A mis padres, por apoyarme siempre y en todo momento y demostrarme que las metas siempre se pueden alcanzar.	3
1.2	A mis hermanos, por siempre estar a mi lado cuando los necesité.	5
1.3	A mis profesores, por ser siempre mis amigos y darme sus conocimientos sin los cuales nunca hubiera llegado a superarme.	8
2.0	Resumen de la tesis	9
2.1	Introducción	10
2.2	Objetivos	12
2.3	Metodología	13
2.4	Marco teórico	13
2.5	Marco metodológico	14
2.6	Marco de referencia de la plataforma	16
2.7	Marco de referencia de la metodología	16
2.8	Marco de referencia de la metodología de la investigación	17
3.0	Conclusiones	19
3.1	My Cloud	19
3.2	ASP.NET	21
4.0	Experiencias de desarrollo de software	24
4.1	Experiencia de desarrollo de software en la nube	27
4.2	Experiencia de desarrollo de software en la nube	38
4.3	Experiencia de desarrollo de software en la nube	44
4.4	Experiencia de desarrollo de software en la nube	45
4.5	Experiencia de desarrollo de software en la nube	56
4.5.1	Experiencia de desarrollo de software en la nube	56
4.5.2	Experiencia de desarrollo de software en la nube	60
4.5.3	Experiencia de desarrollo de software en la nube	66
4.5.4	Experiencia de desarrollo de software en la nube	69

Índice

1.0	Introducción a los lenguajes Script de Servidor	3
1.1	Funcionamiento de PHP.....	4
1.2	Proyecto Actual de PHP	5
1.3	Funcionamiento de ASP.NET.....	6
1.4	Proyectos futuros de ASP.NET	8
2.0	Requerimientos de los Lenguajes.....	9
2.1	Sistemas Operativos	9
2.1.1	Linux	10
2.1.2	Windows	12
2.2	Servidores HTTP	13
2.2.1	Apache.....	13
2.2.2	Microsoft Internet Information Services	14
2.3	Proyectos de Plataforma Cruzada.....	16
2.3.1	Mono.....	16
2.3.2	Sun Java System Active Server Pages	17
3.0	Conexiones con Bases de Datos	19
3.1	MySQL	19
3.2	ADO.NET	21
4.0	Evaluaciones.....	24
4.1	Inversión de Plataforma y Software a largo plazo	27
4.2	Seguridad Global.....	38
4.3	Sistema de Prueba.....	44
4.4	Instalación	45
4.5	Codificación.....	56
4.5.1	Excepciones	56
4.5.2	Conexión a Base de Datos	60
4.5.3	Sesiones.....	66
4.5.4	XML	69

4.5.5 Manejo de Caché.....	73
4.6 Portabilidad	75
4.7 Velocidad de Respuesta	87
4.8 Herramientas de Desarrollo.....	96
4.9 Módulos de Valor Agregado.....	100
4.10 Disponibilidad de Código Libre.....	105
4.11 Factores Sociales.....	107
5.0 Conclusiones	110
Referencias.....	119

1.0 Introducción a los lenguajes Script de Servidor

Los Lenguajes de programación Web, son una tecnología en las cuales, los usuarios hacen una petición con un navegador Web y esta es procesada por un script dentro de un Servidor, generando dinámicamente páginas HTML.¹

El lenguaje script de servidor es usado para proveer páginas dinámicas con información que provenga de bases de datos o de alguna otra fuente de datos. La diferencia entre un script de lado de servidor, y uno de lado del cliente es que estos últimos son ejecutados por la computadora del cliente usando el navegador Web. Un ejemplo de este tipo de scripts sería el JavaScript.

La ventaja de este tipo de script es la habilidad de personalizar altamente la respuesta basada en los requerimientos de los usuarios, sus derechos de acceso o contenidos en bases de datos.

En tiempos antiguos de la Web, esto era exclusivamente utilizado por una combinación de programas en C, script hechos en Perl, y scripts de shell usando la tecnología de CGI (Common Gateway Interface). Estos scripts eran ejecutados por el Sistema Operativo.

El funcionamiento de un Servidor Web (Figura 1.1) usando un Script de Servidor es algo simple. El servidor Web recibe una petición de una Página Web de un usuario, y dependiendo del tipo de solicitud, esta se manda a la extensión o módulo que puede manejarlo. La extensión o Módulo hará conexión con la base de datos de ser necesario y al haber interpretado el Script la salida generada se envía de regreso al servidor Web, el cual, lo envía al cliente en formato apropiado para su lectura.

¹ Wikipedia Server-side scripting
http://en.wikipedia.org/wiki/Server-side_scripting
Wikipedia

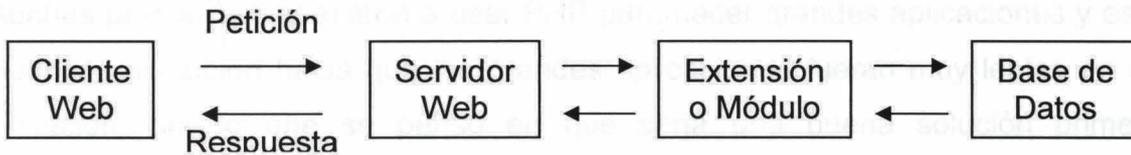


Figura 1.1 Funcionamiento de Servidor Web.

1.1 Funcionamiento de PHP

En 1994, PHP fue concebido en su versión 1.0 por Rasmus Lerdorf, la cual se basó en un conjunto de Scripts de Perl, estos códigos son basados en el lenguaje C. El lenguaje se extendió tanto y fue tan popular que para la versión 2, el código se volvió muy complejo y había cambiado mucho.

El parsing línea-por-línea de PHP 2 no se acercaba a ningún estándar. Después de la primera versión, el código había sido modificado tanto que su funcionalidad tenía muchos fallos.²

El código fue nuevamente modificado, reanalizado y reorganizado desde el principio para la versión 3 con la ayuda de Zeev Suraski y Rasmus, sin embargo el código todavía tenía hacer el parsing línea-por-línea mientras se ejecutaba, esto se muestra en la figura 1.1.1.

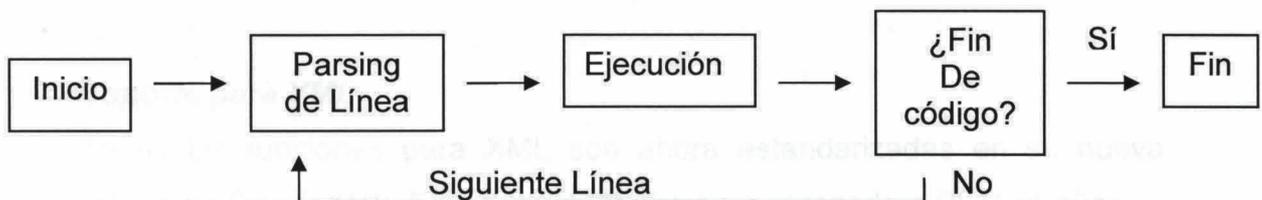


Figura 1.1.1 Funcionamiento de PHP Versión 3.

² Zend Technologies – Under The Hood of PHP4
 Zeev Suraski
<http://www.zend.com/zend/art/under-php4-hood.php>

Muchas personas empezaron a usar PHP para hacer grandes aplicaciones y este modo de ejecución hacia que las grandes aplicaciones fueran muy lentas en su ejecución, por lo que se pensó en que sería una buena solución primero interpretar el código y luego ejecutarlo.

Poco después, fue creado el Motor Zend para el nuevo PHP 4, ahora el Motor primero interpretaba todo el código y después lo ejecutaba. El Motor Zend podía ejecutar scripts de 5 a 200 veces más rápido que la versión anterior de PHP. El interpretador, el ejecutor y eran componentes distintos e independientes. El proceso se muestra en la Figura 1.1.2:

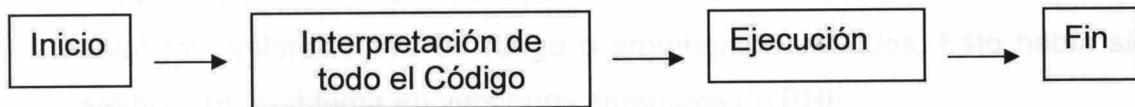


Figura 1.1.2 Funcionamiento de PHP Versión 4.

1.2 Proyecto Actual de PHP

La Corporación Zend, quien creó el Motor Zend de la cual se basa PHP para su funcionamiento, creó la versión 5 de su Motor Zend. En esta nueva versión se agregaron funciones nuevas y soporte a XML. Algunas de las nuevas mejoras, son:

Soporte para XML

Todas las funciones para XML son ahora estandarizadas en su nueva extensión SimpleXML. Esta puede ser el mayor agregado a PHP en años

SQLite

Una nueva maquinaria para SQL que viene integrada dentro de PHP. Esta nueva herramienta esta diseñada para empresas pequeñas que no tienen gran procesamiento de datos. Al hacer esto, no será necesario correr en

nuestro servidor un manejador de base de datos aparte del que viene integrado con PHP5.

Este nuevo manejador integrado, ofrece una mayor rapidez en pequeñas bases de datos de menos de 1000 filas de datos. El problema de este manejador es cuando se tienen bases de datos grandes, o demasiadas conexiones al mismo tiempo, pero hay que recordar que este manejador es para bases de datos de peso ligero.

Manejo de Errores

Ahora en su nueva versión, permite el manejo de errores mediante probar y atrapar errores. Puede manejar errores como las llamadas a funciones inválidas, valores fuera de rango o argumentos inválidos. Esto había sido siempre un problema en versiones anteriores de PHP.

1.3 Funcionamiento de ASP.NET

En los inicios de ASP, este tipo de script solo soportaba un tipo de lenguaje, el cual era el VbScript. Este tipo de lenguaje fue re-escrito bajo la tecnología .NET la cual ahora, hace que ASP tenga el poder de ejecutar diferentes lenguajes a la vez.

Algunas tecnologías usadas en .NET fueron originalmente desarrolladas por Microsoft como su versión de Java. Cuando Microsoft decidió finalizar su futuro uso de tecnologías de Java en 1998, el existente Microsoft J++ (java) fue transformado en los inicios de su proyecto .NET³

ASP.NET tuvo su aparición en 2002 y es la última tecnología de Microsoft para aplicaciones Web y Servicios, y un sucesor de las Páginas de Servidor Activas

³ Wikipedia – Microsoft .NET
http://en.wikipedia.org/wiki/Microsoft_.NET
Wikipedia

(ASP). El corazón de la tecnología .NET es el desarrollo orientado a objetos (OOD)

El Ejecutor de lenguaje común (CLR) es el entorno de ejecución para código escrito para la tecnología .NET. Trabaja con cualquier lenguaje disponible para el entorno de trabajo .NET, así que no hay necesidad de tener un ejecutor para cada lenguaje.⁴ El funcionamiento se muestra en la Figura 1.3.1

La Biblioteca de Clases de la Tecnología .NET (FCL) es un conjunto de clases orientadas a objetos que proveen una plataforma de funcionalidad básica. EL FCL contiene todas las clases que hacen a ASP.NET. Estos incluyen clases que implementan toda la funcionalidad de los objetos ASP, como clases provistas para funcionalidad adicional. El FCL también posee las clases que se usan para desarrollo de aplicaciones de consola, aplicaciones de Windows y Windows NT, así como servicios.

Una ventaja de esta tecnología, es que al haber hecho cambios, el servicio Web interpretará automáticamente el programa.

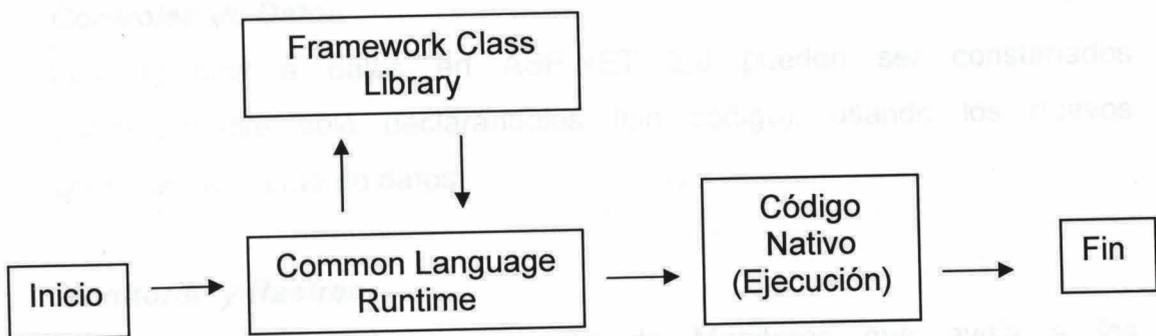


Figura 1.3.1 Funcionamiento de la Tecnología .NET.

⁴ ASP.NET in a Nutshell
G. Andrew Duthie
O'reilly 2003

1.4 Proyectos futuros de ASP.NET

Para la versión 2.0 de ASP.NET, se incorporaran varias nuevas mejoras:

Soporte de 64 Bits

Ahora se puede usar todas las ventajas del direccionamiento de memoria de los nuevos procesadores de 64 bits. Los Desarrolladores copiaran sus códigos de ASP.NET de 32 bits a los servidores de 64 bits con ASP.NET 2.0 y automáticamente se compilará y se ejecutará como código nativo de 64 bits, aparte de que no será necesario cambios en el código o recompilación manual.⁵

Mejorías en el Cache

ASP.NET ahora incluirá el manejo automático de cache de la base de datos, esto hará que el cache se regenere automáticamente. Cualquier cambio que se haga en las bases de datos, que se encuentre en el caché, si es encontrado desactualizado, se regenerara con el contenido actual de la base de datos.

Controles de Datos

Los accesos a datos en ASP.NET 2.0 pueden ser consumados completamente solo declarándolos (sin código), usando los nuevos controles de fuente de datos.

Monitoreo y Rastreo

Ahora se provee de herramienta de Monitoreo que avisa a los administradores del sistema que están ocurriendo fallas.

Se captura un cierto rango de tiempo de ejecución junto con datos solicitados para un mejor diagnostico.

⁵ ASP.NET What's New in ASP.NET 2.0
<http://beta.asp.net/QUICKSTART/aspnet/doc/whatsnew.aspx>
ASP.NET

2.0 Requerimientos de los Lenguajes

Para que un servidor Web, pueda ser eficiente, y funcione correctamente, todos sus componentes deben trabajar en conjunto. Estos componentes deben de comunicarse correctamente, ya que esto es esencial para que no sucedan errores, lo cual podría atrasar el desarrollo de nuestro sitio.

2.1 Sistemas Operativos

El sistema operativo es una decisión muy importante para elegir el lenguaje a utilizar en nuestro sitio Web. El cruce de plataformas entre código dependerá de lo tan adaptado que este el lenguaje a el sistema operativo así como que tan rápido ejecutará las aplicaciones elegidas para cada lenguaje.

En ciertos casos la combinación entre el Sistema Operativo y el lenguaje, será decisiva. Después de la elección del lenguaje, esta puede ser la segunda decisión más difícil.

La competencia entre estos 2 sistemas operativos es algo que viene desde muchos años atrás. Esto ha provocado que muchos usuarios empiecen a comparar ambos sistemas para saber cual se adapta más a sus necesidades.

Debido a que una comparación sobre los sistemas operativos más populares para Sitio Web sería muy larga, analizaremos solo algunos puntos clave para el funcionamiento de nuestro sitio. Algunos de estos puntos son:

Estabilidad

Al tener un sitio Web que este siempre activo, no nos podemos dar el lujo de que el sitio funcione en forma errática, una sola aplicación que falle puede hacer que todo el servidor tenga que ser reiniciado, perdiendo así

tiempo valioso que puede hacer que los posibles clientes que visiten el sitio y lo encuentren deshabilitado, desistan de intentar de nuevo accederlo.

Facilidad de Uso

Cuando los conocimientos sobre sistemas operativos se basen en una sola plataforma, como por ejemplo Windows, este será un factor importante para decidir el sistema operativo de nuestro sitio. Al ser novatos no podemos poner un sitio serio en un sistema operativo que podríamos no controlar cuando tengamos un problema. Muchos de los Servidores HTTP que ejecutaran nuestras aplicaciones dependen en mucho de la facilidad de uso que tenga para manejar nuestras aplicaciones.

Precio

Cuando nuestro presupuesto es reducido, tenemos que decidir cuanto se gastará en este ramo. Lo que no queremos, es que todo nuestro presupuesto se vaya simplemente en el sistema operativo, a menos que este haga una gran diferencia. En caso de que nuestro lenguaje requiera una combinación entre este y el Sistema Operativo, aquí se verá reflejado.

Adaptación al Lenguaje

Muchas de nuestros servidores HTTP vienen ya integrados en los sistemas operativos. Esto facilitará la configuración y podremos empezar a codificar casi de inmediato, aunque a veces tendremos que instalar para elegir un servidor HTTP diferente.

2.1.1 Linux

Linux es un sistema operativo muy popular entre empresas grandes para sus servidores Web. De código Libre, cualquiera puede adaptar este sistema operativo a sus necesidades.

Estabilidad

Su alta estabilidad y sobre todo su seguridad lo hacen muy resistente a fallas. Las fallas en las aplicaciones son muy raras en Linux. Cuando llegan a suceder, los procesos pueden ser terminados sin ningún problema. En cuestión de estabilidad, Linux es excelente.

Facilidad de Uso

Linux ha buscado durante todos estos años, ser más fácil de usar, esto debido a que la gente sigue pensando que Linux es un sistema que es "solo para servidores". Linux ha mejorado desde la instalación (pasando por la configuración de las particiones) hasta la instalación de aplicaciones para que al usuario se le haga más fácil de usar. Sin embargo, en lo que se refiere a configuraciones, muchas de estas siguen siendo en modo texto. A los usuarios novatos esto les será muy difícil.

Precio

Por este rubro, Linux es lo mejor, la mayoría de sus distribuciones son gratuitas, así que no afectara nuestro presupuesto, el problema con esto, es que el soporte no viene incluido. En caso de tener un problema con Linux debido por ejemplo, a la modificación del código, el soporte costará extra. Algunas otras distribuciones de Linux se venden e incluyen soporte, tal es el caso de Red Hat Enterprise.

Adaptación al lenguaje

Para el caso de PHP, nuestra opción mas obvia sería usar Apache como servidor HTTP, este ya viene incluido y preinstalado en la mayoría de las distribuciones de Linux, así que solo se tendrá que configurar detalles del servidor HTTP. En el caso de usar ASP.NET, tendríamos que utilizar un proyecto parecido a la tecnología .NET llamado MONO, en conjunto con el servidor HTTP Apache.

2.1.2 Windows

Windows surge como sistema operativo para los usuarios comunes de PC. El principal objetivo de Windows siempre fue la facilidad de uso y la versatilidad para ejecutar aplicaciones.

Para versiones de servidores, surge la tecnología NT, de la cual se desarrollaron después las versiones 2000 y 2003 que vinieron a competir con el mercado que Linux y Unix vienen ganando en los usuarios expertos y empresas grandes.

Estabilidad

En este rubro, comparado con la versión Windows NT 4.0, se ve incrementado en un 40 % la estabilidad, debido en parte a un modelo de controladores de dispositivos más robusto y a las capacidades de recuperación del sistema diseñadas para un tiempo de actividad máxima. También viene a mejorar la estabilidad, la característica de seguridad mejorada del Internet Explorer.⁶

Facilidad de Uso

Windows posee una notable facilidad de uso. Esto es muy bueno cuando tenemos un sitio Web y nos surja un problema. Aparte de que podemos tener el control del sistema con unos simples clic del ratón. Para los novatos, esto es muy bueno, porque permite a los usuarios empezar a desarrollar código casi de inmediato.

Precio

Las versiones XP que podemos usar para servidor Web tienen un costo de aproximadamente 300 dólares, y las versiones de servidor unos 500. Esto hace una difícil decisión el tener que adquirir este sistema operativo y nos

⁶ Windows Server 2003

<http://www.pc-actual.com/Actualidad/Noticias/Infraestructuras/Software/20030401010>

PC Actual

hace a pensar si realmente es necesaria esa inversión para poder tener un sitio Web usando Windows.

Adaptación al Lenguaje

El lenguaje natural para Windows es ASP, el cual puede ser instalado con unos simples clic. En el caso de ASP.NET tendríamos que instalar un agregado como lo es el agregado .NET y el Development Kit de .NET. Para el uso de un servidor HTTP se podría usar Internet Information Services (IIS) el cual es el Servidor Web que trae integrado Windows 2000 o 2003 y en el caso de Windows XP se puede usar Personal Web Server (PWS), pero si queremos tener un sitio serio, tendremos que usar IIS.

2.2 Servidores HTTP

Estos servidores, nos darán comunicación entre el cliente que hace una petición y ejecutor del código de nuestro sitio. En este ramo, hay muchos servidores, pero hay 2 que se llevan la rivalidad por estar ya integrados en los sistemas operativos. Estos servidores son el Microsoft Internet Information Services y el Apache.

2.2.1 Apache

Con alrededor del 68% del mercado⁷ y creciendo cada día más, Apache es en la actualidad, el servidor HTTP más usado en todo el mundo. Su soporte multiplataforma hace que sea posible instalar un servidor en Unix, Linux o Windows, con los mismos archivos de configuración.

El servidor Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

⁷ Netcraft
<http://www.netcraft.com>
Netcraft

Una ventaja de Apache son los sitios virtuales ya que podemos crear varios de estos, usando un solo servidor Web.

Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor parcheado).⁸

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero es criticado por la falta de una interfaz gráfica que ayude en su configuración.

Esta falta hace que la configuración de este servidor sea un poco difícil, pero no imposible. Todo se configura por medio de un archivo que sirve de base para extensiones de archivos, sitios virtuales, módulos, etc.

Para agregar funcionalidad a este servidor, existen los módulos, que hacen de intérpretes entre Apache y la función que se desea realizar.

2.2.2 Microsoft Internet Information Services

Internet Information Services (o Server), es una serie de servicios para las computadoras que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de

⁸ Wikipedia – Servidor HTTP Apache
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache
Wikipedia

2.3 Proyectos de Plataforma Cruzada

Algunas veces, las tecnología que usaremos, no estará disponible para una cierta plataforma, es por eso que necesitaremos de proyectos externos para poder utilizar, en este caso ASP.NET con la plataforma Linux.

ASP es una tecnología desarrollada por Microsoft y mejorada gracias a la tecnología .NET, pero esta fue desarrollada exclusivamente para la plataforma Windows, es por eso que varias empresas apoyan el desarrollo de proyectos para la adaptación de .NET en varias plataformas. En estos casos tenemos el proyecto MONO y a Sun Java System Active Server Pages.

2.3.1 Mono

Mono, es una plataforma de desarrollo de código abierto capaz de proveer tecnología .NET. Permite a los desarrolladores construir aplicaciones de cruce de plataforma con productividad de desarrollo mejorada.

Patrocinado por Novell, el proyecto MONO tiene una contribución de una comunidad de desarrolladores. Mono incluye herramientas de desarrollo y la infraestructura necesaria para ejecutar el cliente .NET y aplicaciones de servidor.

Internamente, Mono utiliza la misma estructura del interpretador, esto quiere decir que utiliza el CLR (Common Language Runtime).¹¹

En lo referente al servidor Web, Mono tiene su propio servidor Web llamado XSP, el cual provee un servicio de servidor Web mínimo. Al estar unido con mod_mono,

¹¹ Mono – About Mono
<http://www.mono-project.com/Mono>About>
The Mono Project

el cual es una extensión para apache puede manejar los archivos de ASP.NET junto con Apache para tener un Servidor Web ASP.NET dentro de Linux.

La Figura 2.3.1, representa la conexión que se hace desde el Servidor Web Apache hasta la arquitectura de Mono.

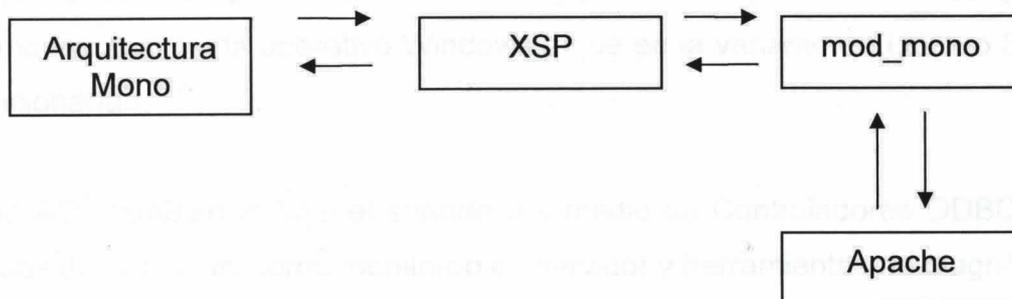


Figura 2.3.1 Conexión entre MONO y Apache.

La mayoría de las funciones de Mono son compatibles con las de la tecnología .NET, solo algunas no trabajarán de manera correcta debido a su dependencia del sistema operativo.

Al usar mono uno puede esperar errores en cuanto a interpretación de código, pero en lo referente a ASP.NET no debemos de tener muchos problemas. De hecho, podemos ver que Mono tiene una compatibilidad extra para usar MySQL lo cual es excelente para usar ASP.NET y MySQL en Linux de forma directa.

Mono es una arquitectura gratuita que esta disponible desde su sitio Web, lo cual lo hace asequible y es una alternativa a usar ASP.NET en Windows.

2.3.2 Sun Java System Active Server Pages

Este software de Sun antes llamado Chilisoft, esta especialmente enfocado en la ejecución de código ASP en plataformas diferentes a Windows

Posee una tecnología propia de soporte para la versión 5.5 de VBScript y JScript 5.5, aunque no posee soporte para ASP.NET, es por eso que no es 100% compatible y muy probablemente en la mudanza de código de la arquitectura .NET a Chilisoft, puedan surgir problemas.

Algunos de estos problemas pueden ser, ya sea de tipo de funciones que hacen llamadas al sistema operativo Windows y que en la variante de Linux o Solaris no funcionarían.

Sun ASP también incluye el soporte por medio de Controladores ODBC a varias bases de datos, así como monitoreo de servidor y herramientas de diagnóstico.

La estrategia de Sun para este producto es mudar de la tecnología de Microsoft, ya sea en sistema operativo o en el servidor Web, para usar su sistema operativo Solaris, o Linux, y según ellos, tener una base más segura en cuanto a servicios Web de ASP.

La compañía natural para este producto, es el Sun Java System Web Server, el cual tiene incluido este software. En cuanto a costo, el Sun Java System Active Server Pages cuesta \$495 Dólares por licencia de un procesador, y en la pagina principal de Sun se puede conseguir una versión de prueba gratuita.

3.0 Conexiones con Bases de Datos

Una aplicación Web, no estaría completa sin una Base de Datos, es por eso, que mencionaremos las 2 principales conexiones a Bases de Datos para nuestras aplicaciones Web. Ambas conexiones usan SQL como lenguaje primario.

Para la plataforma de PHP, la base de datos por excelencia es MySQL, la cual posee gran parte del mercado en lo que se refiere a manejadores de bases de datos relacionales en Software libre.

En lo que se refiere a ASP.NET, este usa la tecnología ADO en su versión .NET, la cual provee conectividad con bases de datos, ya sea a SQL Server de Microsoft o una bases de datos OLE.

3.1 MySQL

MySQL es uno de los manejadores de bases de datos más populares del mundo, y es desarrollado por MySQL AB. Este manejador, entrega un rápido, multi-usuario, y robusto servidor de base de datos SQL.

Bajo la Licencia Publica General (GPL), MySQL ha sido instalado en cerca de 6 millones de servidores.¹² El uso más popular de esta Base de Datos, es con el esquema LAMP (Linux-Apache-MySQL-PHP), usado para aplicaciones de Servidor Web.

No solamente es utilizado para aplicaciones Web, sino que es utilizado ampliamente en aplicaciones de datos normales con grandes resultados.

¹² MySQL Wikipedia
<http://es.wikipedia.org/wiki/MySQL>
Wikipedia

En cuanto a estabilidad, esta base de datos, es muy estable, no por nada es la más popular en el mundo. La maquinaria principal de MySQL tiene sus inicios desde la década de los 80's. Desde 1996, el código de MySQL ha sido trabajado en proyectos. Cuando se puso a disposición pública, muchas personas encontraron fallas, las cuales han sido corregidas y al mismo tiempo, se han puesto nuevas capacidades a este manejador de bases de datos.

La facilidad de uso en MySQL es algo que ésta y otras compañías se han encargado de mejorar. Esto debido a que el manejo de MySQL sigue siendo en modo de comandos a excepción de que uno agregue otros programas de la misma compañía por separado, como lo son MySQL Administrador y MySQL Query Browser (Figura 3.1). Otros agregados como Webmin facilitan la administración en MySQL y de sus bases de datos.

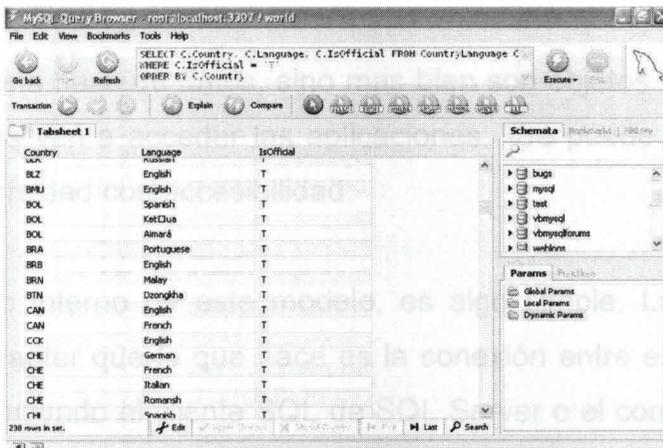


Figura 3.1 Interfaz de MySQL Query Browser.

Por el lado económico, MySQL es gratuito para uso personal, por lo que se convierte en una buena opción en el caso de que empecemos a desarrollar una aplicación, pero si esta es de uso comercial, se debe adquirir una licencia para MySQL de \$ 295 Dólares por cada servidor al año.



La adaptación con PHP no podría ser mejor. La combinación entre PHP y MySQL es natural, ya que con 3 líneas de código podemos tener acceso a la base de datos de MySQL por medio de PHP.

En lo referente a ASP.NET con MySQL, MySQL AB provee conectores ODBC para el acceso a la base de datos de MySQL, esto hace posible la vinculación entre la tecnología .NET y la base de datos de MySQL.

3.2 ADO.NET

ADO.NET es una evolución del modelo de acceso a datos de ADO (ActiveX Data Objects) que controla directamente los requisitos del usuario para programar aplicaciones escalables. Se diseñó específicamente para el Web, teniendo en cuenta la escalabilidad, la independencia y el estándar XML.

En sí no es ninguna base de datos, sino mas bien son objetos que interpretan los datos a los cuales van a acceder las aplicaciones. Se puede considerar que es una unión de diversidad con accesibilidad.

El funcionamiento interno de este modelo, es algo simple. La conexión esta a cargo del DataAdapter que lo que hace es la conexión entre el modelo y la base de datos, ya sea usando el cliente SQL de SQL Server o el conector OLE DB que puede hacer referencia a ODBC. Este funcionamiento se muestra en la Figura 3.2.

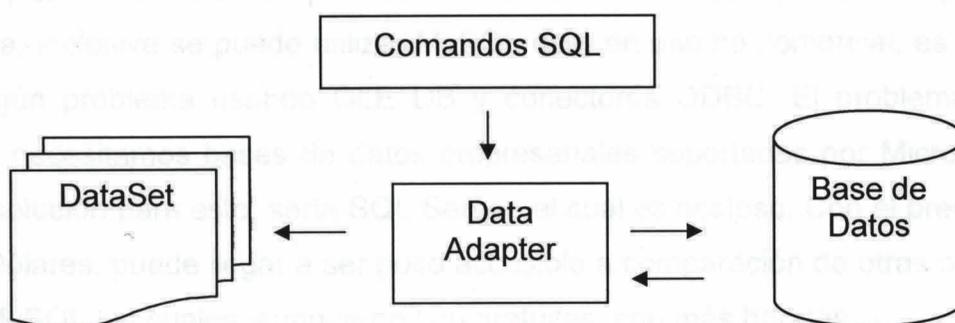


Figura 3.2 Funcionamiento de ADO.NET.

El DataAdapter tiene la función de mandar los comandos SQL a la base de datos, ya que estos sean devueltos por la base de datos, el DataAdapter lo manda al objeto DataSet, en el cual se almacenaran los datos devueltos por el comando enviado.

El objeto DataSet que usa funciones XML para manejar los datos recibidos, no conoce el origen de los datos, este mismo objeto hace seguimiento y comprobación de los datos que contiene.

La estabilidad de este modelo es muy buena, aunque tiene poco tiempo de haber sido desarrollada, las versiones anteriores de ADO.NET tienen sus raíces desde las conexiones ADO normales que surgieron con la aparición de Visual Basic 3.0.

En cuanto a programación se trata, esto es más complejo, debido a que son más líneas de código para hacer una conexión así como para hacer referencia de DataAdapter y a DataSet.

La conectividad aquí es más fuerte, debido a que hay más opciones usando el mismo conector para diferentes tipos de bases de datos usando los conectores OLE DB.

Económicamente hablando, ADO.NET es gratuito, el problema viene al elegir una base de datos para nuestra aplicación. Una base de datos hecha en Microsoft Access puede ser suficiente para una base de datos de ejemplo o una aplicación pequeña, inclusive se puede utilizar MySQL que, en uso no comercial, es gratuito sin ningún problema usando OLE DB y conectores ODBC. El problema surge cuando necesitamos bases de datos empresariales soportadas por Microsoft, ya que la solución para esto, sería SQL Server, el cual es costoso. Con el precio de \$ 1,300 Dólares, puede llegar a ser poco accesible a comparación de otras opciones como MySQL las cuales, aunque no son gratuitas, son más baratas.

ASP.NET requiere de ADO.NET para hacer conexiones externas, ya sea con el uso de ODBC u OLE DB configurados para trabajar con esas bases de datos.

La implementación de un sistema de gestión de contenidos en ASP.NET requiere de un servidor de aplicaciones que soporte ASP.NET y que también soporte ODBC u OLE DB para poder conectarse a las bases de datos.

Una vez que se ha configurado el servidor de aplicaciones y se ha instalado ASP.NET, se puede utilizar el entorno de desarrollo de ASP.NET para crear una aplicación web que utilice ADO.NET para conectarse a una base de datos.

El entorno de desarrollo de ASP.NET incluye un editor de código que permite crear y probar aplicaciones web que utilicen ADO.NET para conectarse a una base de datos.

Para poder utilizar ADO.NET en una aplicación web, se debe utilizar un proveedor de datos que soporte ADO.NET. Algunos de los proveedores de datos que soportan ADO.NET son Microsoft Access, Microsoft SQL Server y Oracle.

Los proveedores de datos que soportan ADO.NET se pueden utilizar en una aplicación web que utilice ASP.NET para conectarse a una base de datos.

Inversión de Software y Software a Largo Plazo

Debido a las altas inversiones en el desarrollo de software, es necesario que los desarrolladores de software tomen en cuenta el costo de mantenimiento y actualización del software a largo plazo. Esto implica que el software debe ser diseñado para ser fácil de mantener y actualizar.

4.0 Evaluaciones

En Internet podemos encontrar muchas comparaciones, que se inclinan casi siempre al lado de PHP. Estas generalmente son hechas por fanáticos de este lenguaje. En cambio, hay personas que defienden ASP.NET, ya sea por su facilidad de uso o por sus controles y acceso a Datos.

¿Será cierto lo que dice la mayoría sobre PHP, que es más rápido, más estable, y en resumidas cuentas, nos dejará un mejor Servidor Web? Es por eso que una evaluación más a fondo sobre ambos lenguajes puede darnos la respuesta.

En esta evaluación no trataremos de ver quien es mejor, sino cual es el más viable según las necesidades del desarrollador y de la empresa que adquirirá el sitio Web que desarrollaremos. De hecho debería ser siempre de esa manera.

Podremos hablar en esta comparación sobre los diferentes puntos a tratar, y en que tan profundo indagaremos. Un ejemplo sería indagar en el código, que en realidad sería una comparación entre C++, Visual Basic, JScript y C#. (según sea el lenguaje que se usará para codificar en ASP.NET).

Los puntos que trataremos ya en lo que se refiere a los lenguajes, serán:

Inversión de Plataforma y Software a largo plazo

Debido a las combinaciones que utilizaremos, tendremos que ver que tan costoso sería tener una plataforma instalada, junto con los programas necesarios para poner en marcha nuestro proyecto de servidor Web, así decidiremos si el presupuesto es un factor importante.

Debemos tomar en cuenta también que al nosotros comprar una plataforma, tenemos acceso a ciertos derechos, esto también lo analizaremos.

Seguridad Global

Analizaremos los posibles ataques externos hacia nuestro sistema operativo o a nuestro servidor Web (inclusive del propio lenguaje o a errores de programación de nosotros). En este punto, también trataremos virus y otros puntos débiles que tendremos que considerar al elegir la mejor combinación para nuestro lenguaje.

Instalación

La instalación iniciando desde el sistema operativo hasta el módulo o extensión que nos permitirá poder ejecutar nuestras instrucciones, serán analizadas a detalle, resaltando problemas o facilidades de uso encontradas en nuestras instalaciones.

Codificación

Para este punto, analizaremos por ejemplo, el tiempo de desarrollo, la facilidad de uso del código, las ventajas de cada uno de los lenguajes al codificar, así como sus desventajas.

Tengamos en cuenta que en PHP tenemos funciones extras de las cuales sacaremos provecho, así como en ASP.NET tenemos nuestra base de controles.

Portabilidad

La portabilidad es esencial cuando tenemos varios sitios Web que debemos controlar o ya sea que queramos mudar la información a otra plataforma. Al tener nosotros la posibilidad o ya sea, la necesidad de mudar nuestro código a otro servidor usando otra plataforma distinta, ¿Funcionará nuestro

código?, ¿Será compatible nuestro lenguaje con la nueva plataforma? Estas dudas se resolverán al realizar pruebas para ver si es en realidad compatible nuestro lenguaje con el cruce entre plataformas.

Velocidad de Respuesta

Se realizarán pruebas para ver cual de los 2 lenguajes a usar, es más rápido accediendo a bases de datos, así como las diferentes acciones que pueden hacer con esa base de datos.

En estas pruebas usaremos las 2 tecnologías que usan los lenguajes para bases de datos, MySQL y ADO.NET

Herramientas de Desarrollo

Al tener que desarrollar nosotros una página Web, podemos tener a nuestra disposición bastantes herramientas que nos pueden facilitar la vida al tener las funciones ya listas para ponerlas, incluso, tener aplicaciones que completen la escritura de código fuente al estar escribiéndolo.

Módulos de Valor Agregado

Para nosotros como desarrolladores, sería una gran ayuda, poder tener código a nuestra disposición, supongamos, un conversor de Páginas HTML a PDF, o algún álbum de fotos para que los empleados de una empresa puedan poner sus fotos en su espacio.

Esto es solo una idea de lo que podríamos hacer nosotros con agregados o script que se encuentren en Internet. Esto se vuelve muy valioso cuando queremos tener un sitio capaz de hacer varias tareas.

Por experiencia propia, hay muchas empresas que adoran este tipo de cosas, por lo que estos serán analizados en aspectos como facilidad de uso y precio.

Disponibilidad de Código Libre

Sería para nosotros, más fácil si encontráramos los scripts que necesitamos si alguien más ya lo hizo por nosotros. En caso de que nuestro tiempo sea limitado, sería bueno copiar scripts que alguien más ya se haya tomado la molestia en publicar.

Es por esto que veremos que tanto código está disponible para nosotros en la red, veremos que tan fácil es encontrar un script que necesitamos y adaptarlo a nuestro proyecto.

Factores Sociales

Como se dijo antes, PHP es el lenguaje de Script de Servidor más usado en el mundo, ¿Pero que tanto? Obtendremos unas gráficas y veremos cual ha sido su evolución, así como sus altibajos cuando sale algún otro lenguaje de desarrollo.

4.1 Inversión de Plataforma y Software a largo plazo

Generalmente cuando las empresas empiezan a incursionar en Internet, lo primero que hacen es poner una página informativa. En este punto no es muy necesario saber que lenguaje vamos a adoptar. Ya que para lo único que utilizaríamos algo de código, sería para algún libro visitas sencillo, por lo que decidir entre un lenguaje u otro es algo irrelevante.

La forma más sencilla de poner un negocio en Internet es crear páginas informativas, y un poco de código para el libro de visitas.

Ahora, supongamos que esta empresa creció, y quiere empezar a ofrecer productos por Internet. Esta es una de las decisiones más obvias, debido a que al

poder vender por Internet te reduce costos. Por ejemplo, todos los agentes de ventas se transforman en uno o dos ingenieros y en capturistas.

Es por eso que empresas que nunca imaginaron poder entrar a Internet, ahora lo hacen, es un negocio que tiene un crecimiento de 81% anual¹³, y con un mínimo de inversión.

Al crear el presupuesto, deberemos de incluir los siguientes puntos:

- **Hosting:** Esto hace que el costo de comprar Hardware, así como el costo de un sistema operativo, baje a una renta. Así mismo, no será necesario tener a un ingeniero tratando de darle mantenimiento nuestro sitio.
- **Hardware:** ya sea adquirir un servidor, o ampliación de algún servidor en componentes. En caso de tener espacio Web contratado, tendremos que ver si ese servidor es capaz de ejecutar las aplicaciones que nuestro lenguaje requiere, así como el espacio necesario, ancho de banda y soporte de bases de datos.
- **Sistema Operativo:** He aquí la gran diferencia, como vimos anteriormente, un sistema operativo puede ser un punto decisivo en la elección del lenguaje. En cuanto a inversión, los costos van desde un Sistema Operativo gratuito, como en diferentes versiones de Linux o un costo relativamente alto, como las versiones más potentes de Windows 2003.
- **Software Extra, Salarios y Cargos de Consultoría:** Aquí pondremos los costos subsecuentes de haber elegido uno u otro lenguaje, por ejemplo, el costo que pueda generar el contratar a un ingeniero para que

¹³ Yahoo Noticias -
<http://mx.news.yahoo.com/050907/40/1i5zx.html>
Yahoo!

modifique el código (dependiendo ASP.NET o PHP). Así también, hay que poner el costo que genere el contratar a alguien si es que se tiene algún problema con el servidor y este necesite soporte si es que no viene incluido. También se pondrá el costo de software extra como Bases de Datos.

En caso de ya contar con un servidor, hay que adaptarlo a nuestras necesidades. Estas serían posiblemente un Disco Duro más rápido para el acceso a bases de datos, Aumento de Memoria y una conexión dedicada con Dirección IP Fija.

Hosting

Para el caso de optar por espacio de hosting, tendremos que asegurarnos que nos sea provisto soporte para nuestras necesidades. Por ejemplo, Contar con soporte de ASP.NET, IIS y soporte, por lo menos, de bases de datos ODBC, en caso de no contar con soporte de SQL Server. Una comparación de Proveedores de Hosting, la podemos ver en la Tabla 4.1.

En esta comparación, tenemos el caso de TuSite.com con ubicación en Monterrey N.L. el cual es una empresa especializada en comercio electrónico, al proveer servicio a sitios como PCenLinea.com y PCDomino.com, los cuales venden productos de computación.

Para comparar con TuSite.com tenemos a Prodigy que es una empresa proveedora de Internet con capacidades de hosting y desarrollo de páginas Web, con presencia y promoción en todo México.

Para el Caso de PHP debe de tener soporte de Apache o IIS (según plataforma y necesidades), soporte para MySQL. En estos días es raro no encontrar un sitio de hospedaje que no te soporte el PHP y el MySQL.

Especificaciones	TuSite.com	Prodigy
Precio al Año	\$ 3,094.00	\$ 4,200.00
Ancho de Banda MB Mensual	Ilimitada	10,500
Espacio MB	1,000	350
Windows 2000	Si	Si
Linux	Si	Si
IIS	Si	Si
Apache	Si	Si
PHP	Si	Si
Asp	Si	SI
Asp.NET	Si	No
Extensiones FrontPage	Si	Si
MySQL	Si	Si
MS Access	Si	No
ODBC	Si	No
SQL Server	No	No

Tabla 4.1 Comparaciones de Hosting.

En cuestión de Hospedaje, hay muchas opciones, estas empresas, antes mencionadas, poseen ya sea soporte de Apache e IIS en plataformas Linux y Windows respectivamente. La única diferencia visible es el soporte de ASP.NET el cual, de ambos, solo en TuSite.com esta disponible.

En comparación de lenguajes usados por los proveedores de Hosting para ofrecer en sus servidores, la compatibilidad es más con PHP, esto debido a la popularidad que tiene, así como por lo estable de sus bases de datos basadas en MySQL.

Sistemas Operativos

En costos de Sistema Operativo, podemos encontrar varias opciones como lo mencionamos antes, pero en algunas, incluyen características que no incluyen los demás, (documentación, soporte, etc.).

Sistemas Operativos Basados en Windows:

Windows XP Profesional

Precio: \$ 299.99 Dls

Servidor Web: Internet Information Server 6.0

Soporte Técnico: 5 Años (Teléfono e Internet) y 10 o más en Autoayuda (Figura 4.1).¹⁴

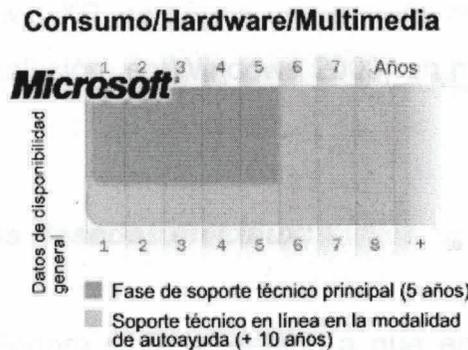


Figura 4.1 Soporte para Software de Consumo.

Windows 2003

Precio: \$599.99 Dls Versión Small Business Server

Precio: \$952.99 Dls Versión Standard Server

Precio: \$399.99 Dls Versión Web Server

Servidor Web: Internet Information Server 6.0

Soporte Técnico: 5 Años (Teléfono e Internet), y 10 o más en Autoayuda (Figura 4.2).

¹⁴Ciclo de Vida del Soporte de Microsoft
<http://support.microsoft.com/gp/lifecycle>
Microsoft

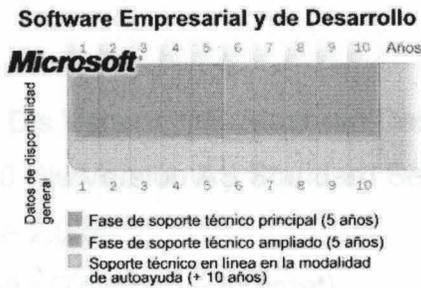


Figura 4.2 Soporte para Software Empresarial.

En el caso de Windows XP, se ofrece un pequeño manual de introducción junto con CD de Instalación, en Windows 2003, un manual de referencia y CD de Instalación.

Sistemas Operativos Basados en Linux:

Se decidió usar el Fedora Core 4, debido a que en pruebas posteriores, MONO, se puede instalar haciendo uso del YUM, el cual es un programa que posee Fedora que facilita la descarga e instalación de aplicaciones. Por otro lado, los archivos RPM que maneja Fedora y RedHat, los cuales son ampliamente distribuidos, hacen que las instalaciones sean sencillas.

RedHat Enterprise en sus versiones AS o WS poseen también el YUM, y es por esto que se decidió usar estas distribuciones para la evaluación. Las Siglas WS y AS significan Workstation y Advanced-Server respectivamente. Hay una versión aparte llamada ES, la cual significa Edge-Server.¹⁵

Fedora Core 4

Precio: Gratuito

Servidor Web: Apache 2.0.54

Soporte Técnico: Documentación en Sitio Web.

¹⁵ Red Hat Enterprise Linux
http://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux
 Wikipedia

RedHat Enterprise

Precio: \$ 299.00 DIs Versión WS Standard Desktop¹⁶

Precio: \$ 1499.00 DIs Versión AS Standard Server¹⁷

Servidor Web: Apache 2.0.54

Soporte Técnico: 1 Año (Teléfono e Internet).

En cuestión de Información impresa, el Fedora Core no contiene y los CD's tendrán que descargarse de Internet y ser grabados en CD's, pero se puede obtener la información impresa desde el sitio Web. Con el RedHat Enterprise, se obtiene toda la documentación impresa, junto con los CD's de Instalación de los Sistemas Operativos.

Costo a largo plazo

Al elegir nosotros poner nuestro propio servidor, tenemos que estar conscientes de los posibles problemas que desarrollará. El tener planes de contingencia puede ayudarnos al tener un serio problema con el servidor.

El hacer respaldos, hacer depuración y mantenimiento permitirá que trabaje con mayor eficiencia y un mínimo de errores.

Los dueños de la empresa no importaran pagar demasiado por traer a un ingeniero en una emergencia, por ejemplo, cuando nuestro servidor o el sitio Web en sí empiece a fallar, debido ya sea a una falla del sistema o el servidor HTTP.

¹⁶RedHat Client Support and Pricing
<http://www.redhat.com/software/rhel/compare/client/>

¹⁷ RedHat Server Support and Pricing
<http://www.redhat.com/software/rhel/compare/server/>

Software Extra, Salarios y Cargos de Consultas

Propuesta de Servidores Stand

Para las soluciones de Windows y Linux se tomaron en cuenta especialistas en Sistemas Operativos y Programación como ASP.NET para Windows 2003 Server, y para PHP bajo plataforma Linux. Solo se requiere un programador que sepa del lenguaje y pueda mantener el sitio Web en funcionamiento haciendo cambios menores.

Ambos salarios fueron basados según búsquedas en páginas de contratación de personal como OCC México¹⁸. Se tomó un promedio de los salarios ofrecidos según los requerimientos necesarios para un servidor Web.

La compañía que tenga servidores basados ya sea en Windows o en Linux, necesitará consultoría externa profesional, la cual también se encuentra en la tabla comparativa.

También se usarán los agregados que pueda depender el servidor, como una base de datos, las cuales dependiendo de la compatibilidad del lenguaje usar, serán los que se vayan a adquirir.

Para la plataforma Windows, se requirió de adquirir una licencia de antivirus, para seguridad del mismo servidor, para Linux no hay virus desarrollados por lo que no se requirió.

El SQL Server 2000 se tomó como referencia como una solución para ASP.NET para un uso profesional de una base de datos.

La siguiente tabla fue basada según la comparativa hecha por CyberSource¹⁹ utilizando los requerimientos solo para servidores y sus gastos de consultoría.

¹⁸ Occ México
<http://www.occ.com.mx>

¹⁹ Linux vs. Windows

Propuestas de Servidores Standard

Propuestas	Windows	Linux
Sistema Operativo	2003 Standard Server \$ 952.99 Dls	RedHat Enterprise AS Standard Server \$ 1499.00 Dls
Antivirus Symantec	\$ 51.60 Dls	-
IIS	Incluido en SO	-
Apache	-	Incluido en SO
ASP.NET	Gratis	-
PHP	-	Gratis
SQL Server 2000	\$ 4,999.00 Dls	-
MySQL	-	\$295.00 Dls/Año
Inversión Inicial	\$ 6003.59 Dls	\$ 1,794.00 Dls

Tabla 4.1.1 Propuestas por Plataformas en Inversión Inicial.

Propuestas	Windows	Linux
Salario Administrador	\$ 1,000.00 Dls/ Mes	\$ 1,300.00 Dls / Mes
Cargos de Consultas al mes	\$ 250.00 Dls / Mes	\$ 750.00 Dls / Mes
Costo al Año de Servicios	\$ 15,000.00 Dls	\$ 24,600.00 Dls

Tabla 4.1.2 Propuestas por Plataformas en Inversión a Largo Plazo.

Como podemos ver en la Figura 4.1.1, la inversión Inicial es más fuerte usando tecnología de Microsoft (esta es la propuesta de un servidor Web usando Standard Server). Mientras que la propuesta de Linux es mucho más barata. Todo esto es debido a que la base de datos en SQL Server de Microsoft es costosa.

En caso de nosotros querer hacer un sitio Web usando ASP.NET, si queremos compatibilidad, tendremos que usar ya sea una base de datos Access o una conexión con una base de datos SQL Server.

En lo referente a Linux, el uso de un sistema operativo estable y lo más posible limpio de errores, como lo es el AS Standard Server, hace que la inversión inicial por el sistema operativo sea más fuerte. Recordemos que el uso de PHP como lenguaje de programación hace que este sistema operativo sea la mejor opción, debido a su integridad con PHP.

En lo que a salario de administrador se refiere, se vé claramente en la Tabla 4.1.2 que un profesional en Windows gana menos que un profesional en Linux. Esto debido a que la administración de un servidor Windows es mucho más fácil que la de un servidor basado en Linux.

Para los cargos de consulta, las empresas cobran más por un servicio de consultaría sobre plataforma Linux o Unix que una plataforma Windows, en relación de 1 a 3.

Para un servidor con una administración profesional, en cuestión de inversión inicial, Windows es el más caro, pero en lo referente a administración y consultaría, Linux se vuelve más caro.

Propuestas	Windows	Linux
Sistema Operativo	2003 Web Server	Fedora Core 4
Precio	\$ 399.99 Dls	Gratuito
Antivirus Symantec	\$ 51.60 Dls	-
IIS	Incluido en SO	-
Apache	-	Incluido en SO
ASP.NET	Gratuito	-
PHP	-	Gratuito
ODBC	Gratuito	-
SQL Server Express 2005	Gratuito	-
MySQL	-	\$ 295.00 Dls
Inversión Inicial	\$ 451.59 Dls	\$ 295.00 Dls

Figura 4.1.3 Propuestas de Servidores Básicos.

En este concepto de servidores básicos, tenemos al Windows 2003 Web Server, el cual posee la misma arquitectura de la línea de servidores Windows 2003 Server, con la excepción de que uno no puede utilizar el Active Directory ni controladores de Dominio, ni Servidores DHCP o DNS. En la plataforma de Windows, seguiremos usando el Antivirus como se mencionó anteriormente.

En el caso de Linux, tenemos a Fedora Core 4, el proyecto antes llamado RedHat, el cual en su versión 4 incluye PHP versión 5, MySQL y Apache todos ya integrados y funcionando. En esta plataforma no se adquirirá ningún antivirus.

En lo que respecta a costos, La propuesta ASP.NET usando Windows 2003 Web Server, nos da una inversión inicial de casi 400 Dólares, lo que hace que sea imposible no gastar dinero en este sistema operativo. En lo que respecta a Fedora Core, la inversión menos costosa pero no gratuita, esto debido a que se debe adquirir una licencia de MySQL para uso comercial.

Para la opción de Fedora, aparte de la adquisición de MySQL y de los CD's en blanco, se deben de descargar las imágenes para así poder grabarlas e instalarlas. El tiempo de descarga se estima en aproximadamente 15 horas usando una conexión Prodigy Infinitum de 512 kbps.

En lo que respecta a los servidores Web, los 2 incluyen los suyos propios, Windows incluye IIS y Linux Fedora usa Apache. IIS soporta ASP.NET de manera nativa y Fedora lo mismo con PHP.

En bases de datos, IIS soporta conexiones ODBC y puede soportar el Microsoft SQL Server Express 2005, mientras que Fedora incluye MySQL para sus propias funciones, el cual ya viene preinstalado.

Ya no se tomó el costo en administración y consultoría, ya que en este rubro, uno mismo puede fungir como administrador o consultor y hacer por uno mismo los presupuestos necesarios.

En caso de que nuestro presupuesto sea limitado o muy limitado, estos serian los escenarios a los cuales acudir. La oferta de Microsoft sigue siendo costosa, mientras que la oferta de Fedora Core es por mucho la más barata.

Cuando nuestro presupuesto no es problema, la gama de opciones es muy amplia, podemos decidir exactamente que queremos sin necesidad de preocuparnos por el presupuesto, pero la realidad es muy diferente, todos nos preocupamos por cuanto nos costaran nuestras decisiones, y si en realidad valen la pena. Es por esto que el dinero invertido es un factor decisivo.

4.2 Seguridad Global

Nuestro servidor, al ser un servidor de páginas Web, tendrá que estar las 24 horas trabajando y en esas 24 horas, muchas cosas pueden pasar.

Tomemos el ejemplo de una compañía televisiva, TV Azteca. Esta compañía provee de información de noticias, así como de su canal de televisión. El 2 de Julio del 2005, su servidor fue atacado y un hacker modificó su página Web.

Vulnerabilidades de plataforma.

Una compañía llamada Security Innovation, publicó un estudio sobre la seguridad en los servidores Web²⁰. Estos se basaron en 2 tipos diferentes de servidores.

²⁰ Role Comparison Report – Web Server Role
Richard Ford
Security Innovation
Marzo 2005

Uno Ejecutando ASP.NET con Internet Information Server, y otro usando Linux AS 3.0 en conjunto con Apache para usar PHP.

Tomaron de base en los servidores, una instalación limpia, instalando solo los componentes necesarios para las aplicaciones Web a usarse, en una instalación mínima.

Las pruebas registraron su impacto en 4 diferentes rangos de peligrosidad:

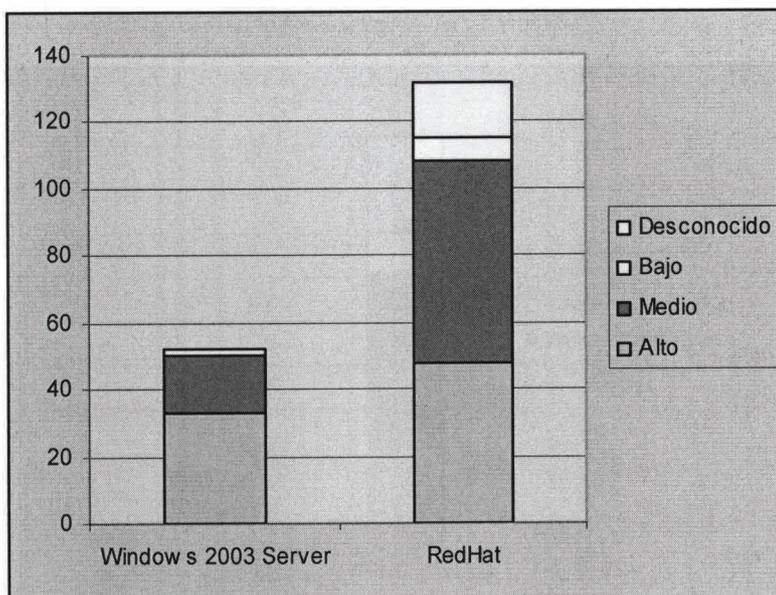
1. **Alta Peligrosidad:** Control total del sistema.
2. **Media Peligrosidad:** Acceso a información del sistema o colapso del sistema.
3. **Baja Peligrosidad:** Colapso de aplicaciones.
4. **Peligrosidad Desconocida:** Se desconoce el impacto al sistema.

En cuanto a vulnerabilidades encontradas, se sacaron los datos mostrados en la Tabla 4.2.1 y graficados en la Figura 4.2.1

Vulnerabilidades	Windows 2003 Server	RedHat Enterprise Linux
Alta Peligrosidad	33	48
Media Peligrosidad	17	60
Baja Peligrosidad	0	7
Peligrosidad Desconocida	2	17
Total	52	132

Menor número es Mejor.

Tabla 4.2.1 Vulnerabilidades.



Menor número es Mejor.

Figura 4.2.1 Vulnerabilidades.

Otros resultados hechos en el mismo estudio, tuvieron una métrica basada en los días que se toma el equipo de desarrollo desde el momento en que se descubre o se avisa de vulnerabilidad hasta el momento en que el parche se pone a disponibilidad de los clientes. Esto quiere decir, que por cada vulnerabilidad encontrada, se tomará un rango de días de seguridad comprometida.

En cuestión de días de seguridad comprometida, según el estudio, los resultados son los que se muestran en la Tabla 4.2.2 y las graficas en la Figura 4.2.2

Días de Seguridad Comprometida	Windows 2003 Server	RedHat Enterprise Linux
Alta Peligrosidad	1145	2124
Media Peligrosidad	426	4003
Baja Peligrosidad	0	921
Seguridad Desconocida	55	2142
Acumulados	1626	9190
Días de Riesgo en Promedio por Vulnerabilidad	31.9	69.6

Menor número es Mejor.

Tabla 4.2.2 Días de Seguridad Comprometida.

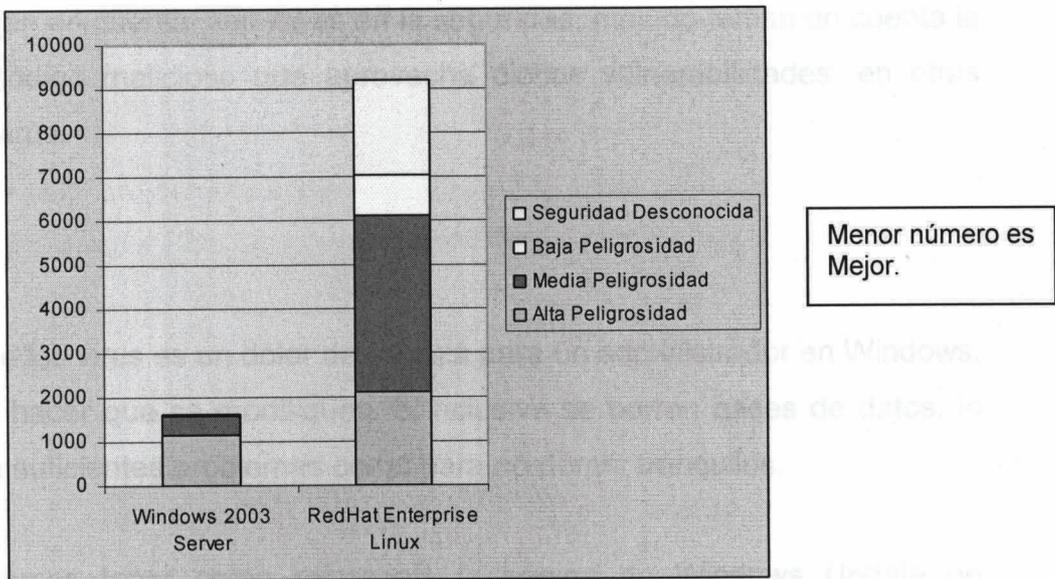


Figura 4.2.2 Días de Seguridad Comprometida.

La diferencia es muy grande en cuanto a cuestiones de seguridad, aunque históricamente las plataformas Windows son más inseguras que una plataforma Linux.

El Internet Explorer produce una importante cantidad de fallas. Este, al estar unido al sistema operativo, hace a Windows vulnerable a programas de tipo publicitario y de espionaje (Adware y Spyware) así como virus de tipo Troyano.

En Windows 2003 Server, Internet Explorer viene deshabilitado en prácticamente todas sus funciones de Scripts. Es por eso que una instalación limpia de Windows 2003 tiene menos fallas de seguridad que una de Linux.

En cuanto a días de riesgo, el problema con Linux es que cuando hay que desarrollar un parche, es más tardado el proceso de descubrir la falla y publicar el parche.

Pudiera ser que esto que anuncia Security Innovations fuera realidad, pero lo único que toman en cuenta, son fallas en la seguridad, mas no toman en cuenta la creación de código malicioso que aprovecha dichas vulnerabilidades, en otras palabras, los virus.

Virus

El problema de los virus es un dolor de cabeza para un administrador en Windows, estos pueden hacer que se modifiquen, o inclusive se borren bases de datos, lo cual nos daría suficientes problemas como para no dormir tranquilos.

También podemos tener como referencia la página de Windows Update de Microsoft. Esta página fue atacada por computadoras infectadas, haciendo que al ser detectado el problema, Microsoft no pudiera hacer público su parche en más de 50,000 sistemas afectados. Esto hizo que el servidor de Windows Update fallara y tuviera que cambiar de dirección. Este caso es del muy famoso Blaster, el cual afecto desde la versión Hogareña de Windows XP Home, hasta la versión empresarial de Windows 2003 Datacenter Server.

Este virus hizo que miles de computadoras recibieran un RPC (Remote Procedure Call) la cual hace que se ejecute código que venga de otra computadora externa ya sea a ella misma, o a la red interna de una empresa.

Lo que hacía el código era simplemente decirle a las 50,000 computadoras afectadas, ya sean computadoras de escritorio o servidores que se reiniciarán y a la vez atacaran otras computadoras. Este comportamiento en un servidor de datos crítico, es algo que puede tener a una empresa en una situación muy comprometida.

Es por eso, que en sistemas Windows, los virus siempre son un problema que se tiene que estar vigilando constantemente, por lo que aparte tendremos que conseguir un antivirus para esta función.

Linux no tiene este problema, simplemente no hay virus para Linux, es por eso que muchas empresas eligen Linux en vez de Windows

Errores de Lenguaje

En el 2004, en ASP.NET se encontró un error de programación, que podía hacer que un usuario se saltara la pantalla de inicio de sesión en un sistema protegido con contraseña, simplemente substituyendo la '\ (diagonal invertida) o el '%5C' con una '/' (diagonal) o con un espacio en la dirección o URL de la página Web.²¹

Este error opera independientemente del navegador que se use, ya sea Internet Explorer o Mozilla. Esto puede hacer que cualquiera pueda acceder a sitios administrativos.

Así también, hay errores de seguridad para PHP, por ejemplo, en la versión 4.3.7, hacia atrás y hasta la versión 5 RC3 se detecto que una función llamada memory_limit podía hacer que un atacante, activara esa función en donde una interrupción fuera insegura, pudiendo así ejecutar código malicioso.²²

Ambas fallas de seguridad han sido corregidas para esta fecha, aunque estas no son las únicas fallas, si son las más importantes de las que se tenga conocimiento.

²¹ ASP.Net Security Flaw can ByPass Password
http://news.netcraft.com/archives/2004/10/07/aspnet_security_flaw_can_bypass_password.html
Netcraft

²² PHP Zaps Security Leaks
<http://www.internetnews.com/security/article.php/3382181>
internetnews.com

Ambos lenguajes tienen referencias para tener código seguro, en Internet se pudieron encontrar métodos para codificar sin problemas un sitio seguro, haciendo prácticas comunes al usar funciones o validaciones. No importa que tan seguro sea el lenguaje, siempre se tendrán fallas que habrá que corregir.

En cuestiones de problemas de seguridad dentro del lenguaje, los 2 son muy parejos, cada uno de los lenguajes tendrá sus problemas de seguridad no descubiertos, pero en lo referente a las plataformas que usen, ese es un tema decisivo para la elección del lenguaje, que es en donde en realidad podemos ver una diferencia de seguridad.

4.4 Instalación

4.3 Sistema de Prueba

Una empresa normal, pone generalmente una computadora de reciente modelo como un servidor Web, o compra un servidor especializado para ese uso.

En el caso de estas pruebas, utilizaremos un sistema armado como servidor, las características de este sistema son las siguientes:

- Procesador Intel Pentium III a 700 Mhz
- 256 MB de Memoria SDRAM PC133
- Disco Duro Maxtor 40 GB a 5200 rpm.
- Tarjeta de Red RealTek 8139 10/100 MBps.
- Tarjeta gráfica Intel 82810
- Monitor Likom Futura 14"

Este sistema tendrá un doble arranque para poder instalar Windows 2003 Standard Server y Fedora Core 4 en el mismo equipo.

El sistema que actuará como cliente, tiene las siguientes especificaciones:

- Procesador AMD Sempron 2400

- 256 MB de Memoria DDR a 400 Mhz.
- Disco Duro Seagate 40 GB a 7200 rpm.
- Tarjeta de Red Via Rhine 10/100.
- Tarjeta Gráfica Via S3G Unichrome
- Monitor Scanport 17" LCD

Ambos están conectados a un Switch de velocidad 10/100 a 100Mbps, usando un router Efficient Networks Speedstream 5200 que les provee de direcciones dinámicas usando DHCP.

4.4 Instalación

La instalación constará de 4 partes principales para poder usar el lenguaje con todos sus componentes, las 4 partes a analizar en su instalación, serán:

- *Sistema Operativo:* Aquí instalaremos los 2 sistemas operativos en los cuales las plataformas pueden funcionar, que son Windows 2003 Standard Server y Fedora Core 4.
- *Servidor Web:* Los Servidores a Instalar serán Internet Information Server para Windows 2003 y Apache para Fedora Core y para Windows 2003.
- *Lenguaje:* Aquí se instalará el .NET Framework para la ejecución de ASP.NET en Windows y MONO para la ejecución de ASP.NET en Linux. Igualmente, se instalará PHP para su ejecución tanto en Windows como en Linux.
- *Base de Datos:* Para Estos, se requerirá la instalación de ADO.Net para las bases de datos de ASP.Net y MySQL para PHP. También se tratará de hacer conexiones entre ASP.Net y MySQL.

Se tratará de hacer la mayor cantidad de conexiones posibles usando diferentes combinaciones entre plataformas, Servidores Web, Lenguajes y Bases de Datos.

Sistema Operativo

Windows 2003 Server

En este Sistema Operativo, la instalación fue un proceso muy sencillo. Los pasos son muy fáciles de entender.

Al arrancar desde el CD nos aparecen ventanas en modo DOS que nos indican formas de recuperación de Windows cuando no se pueda arrancar. Poco después nos aparecen las particiones donde queremos instalarlo, así mismo al elegirlo, formatea la unidad.

Poco después de que la copia de archivos se hace, reinicia y empieza a preguntar sobre configuraciones de red y configuraciones regionales.

Después de hacer todas las configuraciones, nos pide crear una contraseña para el administrador y al reiniciar nos aparece un Windows ya instalado y básico al que poco después configuraremos los agregados con la pantalla de Administre su servidor (Figura 4.4.1).

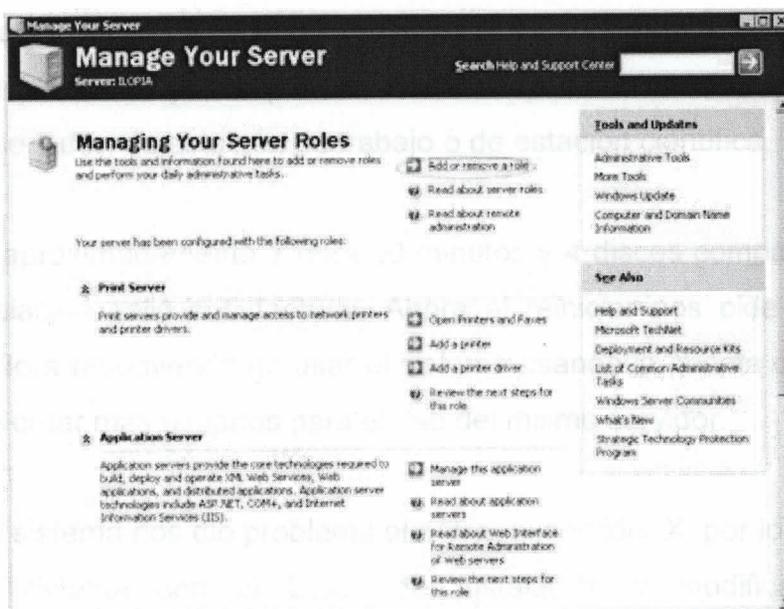


Figura 4.4.1 Interfaz de Administre su Servidor.

El resultado de la instalación fue muy satisfactorio, pues no nos dio ningún problema, aparte todos los componentes fueron detectados por la instalación e instalados correctamente.

Fedora Core 4

Al arrancar la instalación desde el CD, no hubo ningún problema, solo damos "Entrar" para entrar a la instalación grafica. Nos detecta nuestra tarjeta gráfica, nuestro monitor y todo va perfecto.

De repente nos surgió un problema, nuestro monitor aunque fue detectado, no soportó el refresco horizontal y vertical de los controladores de video.

Reiniciamos la instalación ahora en modo texto, y aquí es donde viene lo interesante, aquí podemos hacer que nuestra instalación haga las particiones automáticamente o por medio de Disk Druid, en este caso hicimos 3 particiones para Fedora, una de archivos, una de Swap y otra de arranque.

En la instalación nos permite instalar los paquetes en específico desde la misma instalación del Fedora, y nos permite darle un perfil a nuestra instalación, por ejemplo, de servidor, de estación de trabajo o de estación científica.

Después de aproximadamente 1 hora 30 minutos y 4 discos compactos después, nuestra instalación salió satisfactoria. Ahora al reiniciar nos pide que creamos usuarios. Fedora recomienda no usar el sistema usando la cuenta de "root" por lo que nos pide crear más usuarios para el uso del mismo servidor.

Al reiniciar el sistema nos dio problema el entrar al servidor X, por lo que hubo que arrancar el sistema con el Disco de instalación y modificar el archivo `/etc/X11/xorg.conf` para que el refresco del monitor fuera el aceptable.

La instalación aunque nos dio ciertos problemas con la configuración del servidor X, nos da la oportunidad de dejar todo lo que necesitamos ya instalado. Al reiniciar ya con esto configurado, nos aparece el inicio de sesión en Servidor X (Figura 4.4.2).

Aunque no es necesaria la interfaz gráfica para operar el servidor Linux, para los novatos esta es de gran ayuda, ya que permite una facilidad de uso e instalación mayor.

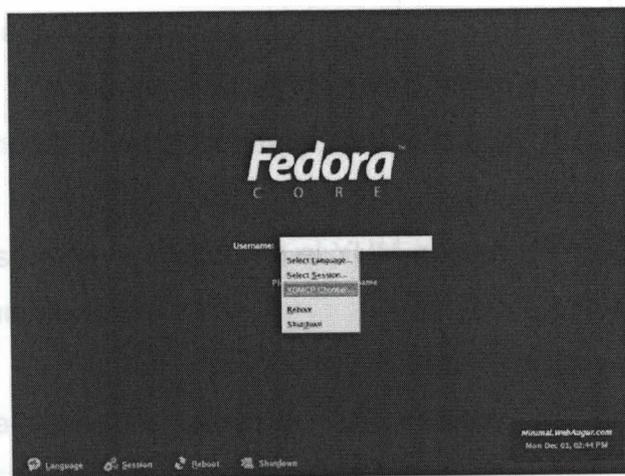


Figura 4.4.2 Inicio de Sesión de Fedora Core.

En sí, las 2 instalaciones son sencillas, el uso de interfaz gráfica nos hace mucho más fácil el trabajo. Aunque en el caso de Fedora, la instalación fue en modo texto, nos da la capacidad de instalar todo lo que necesitamos desde el mismo comienzo, lo cual es muy bueno.

Servidor Web

Internet Information Server – Windows 2003 Standard Server

La instalación de este Servidor Web solo es posible en plataformas Windows, (no existe este servidor para la plataforma Linux ni para ninguna otra).

Para instalarlo hay 2 formas, una es desde el panel de control en agregar y quitar programas y luego componentes de Windows. Ahí en Servidor de aplicaciones y luego Internet Information Services. Y con esto queda instalado.

La otra forma es desde la pantalla que aparece al inicio de administre su servidor. Ahí en agregar o quitar una función y luego servidor de archivos.

Para la configuración de IIS, usaremos el Administrador de Internet Information Server, ubicado en las Herramientas Administrativas del menú Inicio.

Apache – Fedora Core 4

La instalación de Apache en Fedora es relativamente sencilla, aquí hay varias maneras, pero la más fácil es desde la instalación.

Otra manera de instalarlo es utilizando el YUM que es un programa que instala los RPM de manera muy fácil.

La manera de instalar Apache utilizando YUM es esta:

```
# yum install httpd
```

Al hacer esto, YUM instalará todos los paquetes del demonio HTTP que usa apache para actuar como servidor Web.

La configuración de Apache se hace desde un solo archivo ubicado en /etc/httpd/conf llamado "httpd.conf".

Con este archivo podemos desde relacionar los tipos de archivos manejados por nuestro servidor Web, hasta la creación de servidores virtuales.

Apache – Windows 2003

Para poder utilizar apache junto con Windows, debemos de descargar el archivo ejecutable de Apache para Win32 de la página www.apache.org.

Al descargarlo, lo ejecutamos y aparece un asistente que nos guía por todo el proceso. Al final, queda instalado como un servicio de Windows. Tenemos que tener cuidado de no poner a funcionar Apache e IIS al mismo tiempo, ya que ambos usan el mismo puerto (80) para publicarse.

Su configuración es muy similar a la de Apache para Fedora, solo que la ubicación es diferente.

Ambos Servidores Web son relativamente sencillos de instalar, el problema surge con la configuración de Apache, ya que sigue siendo usando un archivo de texto.

Lenguaje

ASP.Net – IIS – Windows 2003

La instalación es muy sencilla, uno debe de ir al panel de control, luego Agregar y quitar programas, después agregar o quitar componentes de Windows y en Servidor de Archivos, marcamos la casilla ASP.NET para finalizar y después en aceptar.

ASP.Net se asocia inmediatamente con IIS, por lo que no habrá necesidad de hacer configuración extra.

ASP.Net (Mono) – Apache – Fedora Core

Mono es un proyecto que es financiado por Novell para hacer una versión de la Tecnología .NET en Linux. Lo que nosotros buscamos aquí es tratar de poner un servidor ASP.Net en Linux utilizando la arquitectura que nos provee Mono y hacer la conexión con Apache

Esta combinación dió problemas, la instalación de Mono fue complicada, debido a que los archivos de instalación específicos de Fedora Core, tenían dependencias de otras instalaciones, por lo que se decidió instalar un archivo ejecutable para todas las distribuciones de Linux y no la de Fedora Core en específico.

Poco después, se observó que para instalar mono, se puede usar el YUM haciendo las modificaciones necesarias en el archivo "/etc/yum.repos.d/mono.repo".

```
[mono-1.1]
name=Mono 1.1 for fedora-$releasever-$basearch
baseurl=http://go-mono.com/download/mono-1.1/fedora-3-i386/
enabled=1
gpgcheck=0

[gtk-sharp-1.0]
name=Gtk# 1.0 for fedora-$releasever-$basearch
baseurl=http://go-mono.com/download/gtk-sharp/fedora-3-i386/
enabled=1
gpgcheck=0

[gtk-sharp-2.0]
name=Gtk# 2.0 for fedora-$releasever-$basearch
baseurl=http://go-mono.com/download/gtk-sharp-2.0/fedora-3-i386/
enabled=1
gpgcheck=0
```

Después de hacer estas modificaciones, solo será necesario ejecutar:

```
# yum install mono-complete
```

Y después de eso:

```
#yum install mod_mono
```

Con estas instrucciones podremos ejecutar páginas ASP usando XSP que es el interpretador de ASP.Net, ahora tendremos que relacionar el módulo de Mono con Apache. Para esto, tendremos que modificar el archivo `"/etc/httpd/conf/httpd.conf"` y agregar estas líneas:

```
LoadModule mono_module modules/mod_mono.so
Alias /test "/usr/share/doc/xsp/test"
AddMonoApplications default "/test:/usr/share/doc/xsp/test"
<Location /test>
    SetHandler mono
</Location>
```

Estas líneas agregarán un Alias a un directorio de prueba instalado por Mono, pero en caso de que nosotros queramos tener un directorio virtual, en ese caso, debemos crearlo de esta manera:

```
<VirtualHost *:80>
    DocumentRoot /directorio/sitio
    ServerName www.ejemplo.com
    MonoApplications default "/*:/directorio/sitio"
    <Location />
        SetHandler mono
    </Location>
</VirtualHost>
```

De esta manera tenemos nuestro directorio ya listo para ejecutar ASP.Net usando Mono conectado con Apache.

En caso de que queramos que las páginas Web sigan siendo ejecutadas por el mismo Apache y no por Mono, solo le diremos que invoque a Mono cuando encuentre páginas de ASP.Net

Tendremos que reemplazar el "SetHandler mono" por:

```
AddHandler mono .aspx .ascx .asax .ashx .config .cs .asmx .axd
```

Esta es una configuración que sí dio problemas debido a que no hay una distribución exacta para Fedora Core 4, sino que se tuvo que usar una distribución creada para Fedora Core 3.

PHP – IIS o Apache – Windows 2003

Con IIS instalado o Apache, descargamos de www.php.net el Windows Binaries de PHP y lo ejecutamos. Casi al final de la instalación, pregunta el servidor Web a usar, y es cuando especificamos el servidor que usaremos (Figura 4.4.3).

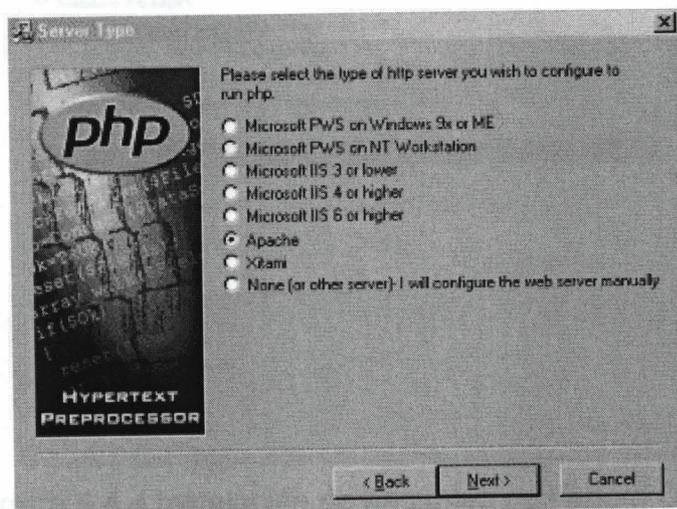


Figura 4.4.3 Instalación de PHP.

La instalación de PHP en plataformas Windows es realmente muy sencilla usando cualquier servidor.

PHP – Apache – Fedora Core 3

Al instalar Fedora Core 4 se puede especificar la instalación de PHP en las opciones de servidor Web. En caso de querer instalarlo aparte, se puede utilizar el YUM para instalarlo.

La sintaxis en una ventana de comando sería:

```
# yum install php
```

Base de Datos

MySQL – Fedora Core 4

La instalación en Fedora, se puede hacer como siempre, en la misma instalación del sistema operativo, solo es de marcar MySQL – Server en los paquetes a instalar (Figura 4.4.4).

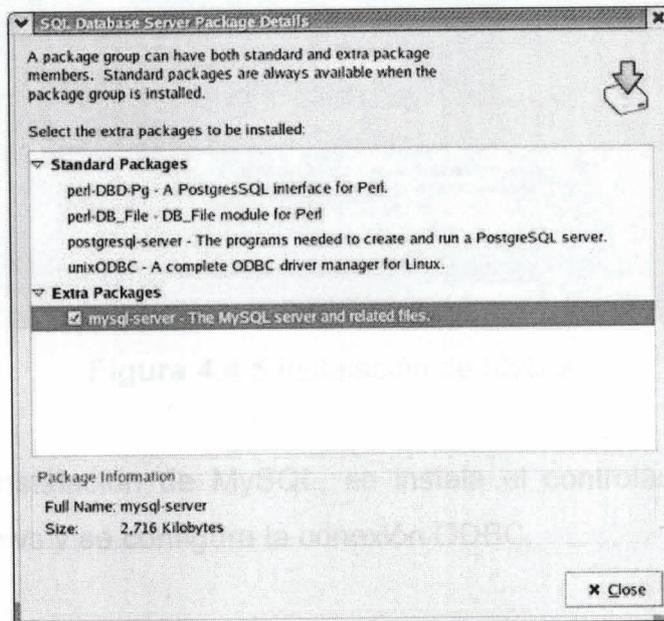


Figura 4.4.4 Instalación de paquetes de Fedora Core.

Por otro lado, se puede también instalar usando YUM y ejecutando:

```
# yum install mysql
# yum install mysql-server
```

MySQL – Windows 2003

Para esta versión, tendremos que descargar desde www.mysql.com de la sección de Windows Downloads, la versión Windows (x86). Al haber sido descargada, solo

ejecutamos el archivo que bajamos y un asistente nos aparecerá guiándonos a través de la instalación.

En la instalación de MySQL nos pregunta si deseamos que este aparezca como un servicio más de Windows. Esto se observa en la Figura 4.4.5.

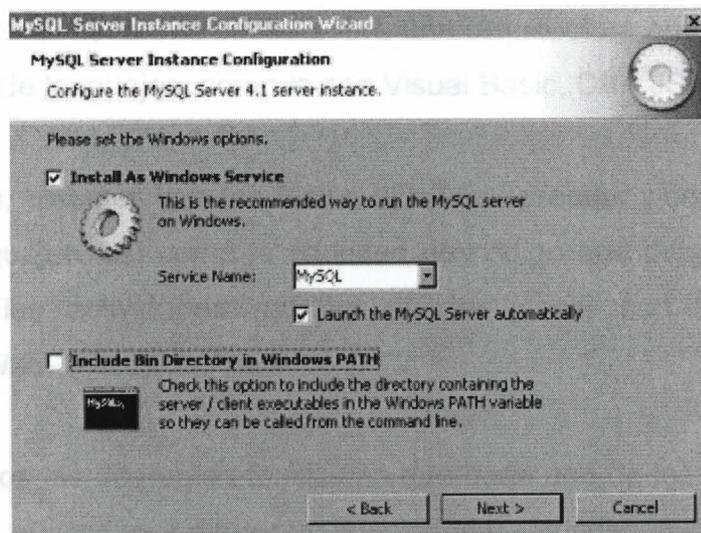


Figura 4.4.5 Instalación de MySQL.

Después de la instalación de MySQL, se instala el controlador ODBC para la plataforma Windows y se configura la conexión ODBC.

ADO.Net – Windows 2003

ADO.Net es una tecnología que viene incrustada en .NET así que no hay necesidad de instalar componentes extras para hacer conexiones de tipo OLE DB, ODBC y para SQL Server de Microsoft.

ADO.Net – Fedora Core 4

Mono tiene incrustada la misma tecnología de ADO.NET que en la versión de Windows, inclusive viene ya con conectores para MySQL y otras bases de datos populares sin tener que usar configuración extra, solamente código, lo cual lo hace muy conveniente para nuestras conexiones.

4.5 Codificación

PHP tiene sus bases en C++, y es muy evidente a la hora de programar en este lenguaje. Muchas de las funciones de C++ funcionan sobre PHP.

En cambio ASP.Net tiene sus bases sobre una arquitectura que puede manejar 3 tipos diferentes de lenguajes, como lo son Visual Basic, C# y JScript.

En esta sección, veremos que tan fácil o difícil es aprender y codificar sobre cada uno de los lenguajes, así como la cantidad de código que deberemos usar para cada una de las actividades que se pueden hacer con los lenguajes de programación Web.

También veremos las diferentes formas en que cada uno de los lenguajes maneja por ejemplo, los errores, sus diferentes funciones y sus adaptaciones a XML.

Las secciones a ver en este tema a nivel código, serán:

- Excepciones (Manejo de Errores)
- Conexión a Bases de Datos
- Sesiones
- XML
- Manejo de Caché

4.5.1 Excepciones

No se puede depurar un problema del cual no tienes conocimiento que existe. Después de que se ha puesto una aplicación a ejecutarse en un ambiente ya de producción, el desarrollador no es el único que se dará cuenta de si hay errores

Con el Manejo de Excepciones, se puede controlar los errores, haciendo que se capturen los errores, se muestre un error y se hagan las correcciones necesarias.

ASP.NET

ASP.NET en la primera versión de su tecnología .NET ya venía con manejo de excepciones.

Para ejemplificar un error en ASP.NET, haremos una división entre cero, el cual es un error muy común.

```
<%@ Page Language="VB" %>
<script language="VB" runat="server">
    Dim x As Integer = 1
    Dim y As Integer = 0
    Dim z As Integer = x / y
</script>
```

Al ejecutar ese código, ASP.NET nos mandará un error como este:

Error de servidor en la aplicación '/.

La operación Aritmética ha provocado un desbordamiento

Ahora, podemos nosotros poner un manejador de excepciones utilizando la función Try/Catch, este sería el código.

```
<%@ Page Language="VB" %>
<script language="VB" runat="server">
    Protected Sub boton1_click(sender As Object, e As EventArgs)
        Try
            Dim x As Integer = 1
            Dim y As Integer = 0
            Dim z As Integer = x / y
            Response.Write(z)
        End Try
    End Sub
</script>
```

```

Catch err as Exception
    Response.Write ("Ha Ocurrido un Error "_
        & err.ToString())

Finally
    Dim z As Integer = 0

End Try

End Sub

</script>
<form runat="server">
    <asp:button id="boton1" onclick="boton1_click"
        text="Haz Click" runat="server" />
</form>

```

Al ejecutarse el código, se observa que hay un error. Aquí ocurren 2 cosas:

- Se Mostraría "Ha Ocurrido un Error: OccurredSystem.OverflowException: La operación aritmética ha provocado un desbordamiento. at ASP.prueba_aspx.button1_click(Object sender, EventArgs e)"
- y se procedería a darle valor a "z" de cero.

La Función Try hace que el código dentro de ella, intente ser ejecutado. Después, la función Catch, que es para ejecutar instrucciones si es encontrado un error, dará informe sobre el error que ocurrió en las líneas de Try. Finally esta hecho para hacer limpieza en caso de que algún error haya ocurrido.

PHP

Hasta la versión 4, PHP todavía no conocía el manejo de las Excepciones o manejo de errores. Es por eso que desde la versión 5 se agregaron estas características, las cuales se volvieron fundamentales para aplicaciones Web.

De manera predeterminada, PHP no muestra los errores contenidos en una página Web a la hora de ejecutarla. Lo que tendremos que hacer en este caso

para ver los errores, es modificar el archivo php.ini ubicado en el caso de las distribuciones de Linux en el directorio /etc.

En la línea que dice 'display_errors = Off', hay que cambiar el "Off" por un "On", para que PHP nos pueda mostrar los errores de programación.

Para ejemplificar un error, utilizaremos una división entre cero.

```
<?php
    $a=1;
    $b=0;
    $c=1/0;
    echo $c;
?>
```

Al ser ejecutado este código, PHP mandará el siguiente mensaje:

Warning: Division by zero in prueba2.php on line 4

Si bien, el código muestra un Warning que es un aviso sobre algún funcionamiento erróneo, se puede seguir ejecutando el código.

Utilizando las funciones integradas Try/Catch de PHP 5.0, ahora se puede prever ese tipo de mensajes o errores y corregirlos. El siguiente código ejemplifica como:

```
<?php
function manejador($errno, $errstr, $errfile, $errline)
{
    throw new Exception($errstr, $errno);
}
set_error_handler('manejador');
try
{
    $a = 1;
```

```

    $b = 0;
    $c = $a / $b;
}
catch (Exception $e)
{
    echo "Excepcion Atrapada:\n";
    echo "Codigo: ".$e->getCode()."\n";
    echo "Mensaje: ".$e->getMessage()."\n";
}
?>

```

Utilizamos un manejador de Errores para crear una nueva excepción, y después de que ocurre el error dentro de el Try por la división, este es manejado y desglosado por el Catch

Este sería el resultado de la operación:

```
Excepcion Atrapada: Codigo: 2 Mensaje: Division by zero
```

De esta manera, podemos controlar los errores que se nos presenten al hacer, por ejemplo, una operación con división entre cero, y que el programa siga procesando las demás líneas del código.

4.5.2 Conexión a Base de Datos

La conexión de una base de datos es fundamental en una aplicación que maneje información.

Existen muchas bases de datos a las cuales podemos acceder, por ejemplo, MySQL, SQL Server de Microsoft, Bases de Datos de Access, etc.

ASP.NET

La tecnología .NET utiliza para conexiones a base de datos dentro de el marco de ADO.NET, un conjunto de interfaces de una variedad de fuentes de datos llamadas OLE DB (Vinculación de Objetos Contenidos de Bases de Datos) que pueden integrar ya sea Bases de Datos SQL o diferentes.²³

OLE DB viene a reemplazar el ODBC (Conectividad Abierta de Base de Datos) que era el estándar para comunicarse con base de datos hasta el surgimiento de este. Casi todas las Bases de Datos existentes son compatibles con ODBC.²⁴

Una conexión usando ODBC requiere especificar un DSN o Nombre de Fuente de Datos, el cual es único para cada controlador y tiene información y configuración específica del mismo.²⁵

Un ejemplo de una conexión a una base de datos usando OLE DB sería el siguiente:

```
<%@ Page Language="VB" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.OleDb" %>

<script runat="server">
Sub Page_Load(obj As object, e As eventargs)
Dim ConexionDB As New OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;
_Data Source=" & Server.MapPath("base_de_datos.mdb") & ";"")
ConexionDB.Open
```

²³ ADO Glossary

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/html/mddefglossary.asp>
Microsoft Developer Network

²⁴ Todo sobre ADO

<http://www.gamarod.com.ar/articulos/articulos.asp?id=35>
Gamarod

²⁵ DirList2ODBC - Terminology

<http://dirlist.sourceforge.net/docs/odbc/terms.htm>
SourceForge

```
End sub
</script>
```

En el código, se puede observar que se importan los Nombres de espacio, que son paquetes de código con funciones específicas. En este caso, se importaran el acceso a datos y el acceso a datos usando OLE DB.

Después de las importaciones, se declara la conexión como una nueva conexión OLE DB y se le dan los datos del proveedor de la conexión, en este caso, Microsoft Jet OLE DB 4.0 y el nombre de la base de datos.

En lo que se refiere a una conexión a una base de datos SQL Server, el panorama cambia, debido a que este no usa OLE DB

```
<%@ Page Language="VB" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>

<script runat="server">
Sub Page_Load(Sender As Object, E As EventArgs)
    Dim ConexionDB As SqlConnection = New SqlConnection(
        "server=(local)\Directorio;database=base_de_datos;Integrated
        Security=SSPI")
    ConexionDB.Open
End Sub
</script>
```

Cuando se hace una conexión a SQL Server, se usa otro tipo de Nombre de Espacio, en este caso, el de SqlClient. Al hacer la conexión, se da la ubicación del servidor, la base de datos y el nombre de la base de datos, así como el tipo de seguridad a usar.

En el caso de una conexión con una base de datos MySQL, usaremos un conector de tipo ODBC versión 3.51 y descargaremos de Internet el mismo de el sitio de MySQL.²⁶

Al haber hecho esto, el conector estará listo para su uso, por lo que podremos usar ahora la conexión en nuestro código.

```
<%@ Page Language="VB" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.ODBC" %>
<script runat="server">
Sub Page_Load(obj As object, e As eventargs)
Dim ConexionDB As New ODBCConnection("Driver=_
    {MySQL ODBC 3.51 Driver};uid=root;password=contraseña;_
    Server=localhost;Option=3;Database=test;")
Dim ComandoDB As ODBCCommand = New ODBCCommand_
    ("select * from orders", ConexionDB)
ConexionDB.Open()
End sub
</script>
```

Este Código es muy parecido al de una conexión OLE DB, la diferencia esta en el Nombre de Espacio utilizado que ahora será el de ODBC. Otra diferencia que encontramos es el tipo de conector a invocar, que ahora será uno del tipo ODBC MySQL.

PHP

Para PHP, las conexiones son algo muy sencillo de manejar, sobre todo si son del tipo MySQL.

²⁶ MySQL Connector/ODBC 3.51
<http://dev.mysql.com/downloads/connector/odbc/3.51.html>
MySQL AB

```

<? php
$ConexionDB=mysql_connect("localhost","nombre_usuario");
mysql_select_db("base_de_datos") or die ("No se puede abrir la base de
datos");
?>

```

Este es todo el código necesario para abrir una base de datos MySQL usando PHP. Lo único que se hace es declarar la conexión especificando el nombre del servidor, así como el usuario. Después de haber hecho la conexión, especificamos la base de datos a la que se quiere acceder.

En el caso de querer acceder una base de datos usando ODBC, solo habrá que especificar una nueva conexión con la configuración necesaria, esto es, usando el controlador específico.

Para usar PHP con Microsoft SQL Server, tendremos que descargar una extensión y modificar el archivo de configuración de PHP. Al haber hecho esto, solo tendremos que ejecutar el siguiente código para hacer la conexión.

```

<?
$server="Nombre_Servidor
$username="Nombre_Usuario";
$password="Password_Usuario";
$ConexionDBt=mssql_connect($server,$username,$password);
$BaseDatos=mssql_select_db("Base_de_Datos",$ConexionDB);
?>

```

Aquí se usan funciones parecidas a las de MySQL, solo que ahora estas son para acceder a una base de datos SQL Server. Esta opción solo trabaja, teniendo PHP funcionando en el sistema operativo Windows junto con SQL Server.

En el caso de que se requiera utilizar un servidor Microsoft SQL Server externo con PHP, utilizaremos el conjunto de bibliotecas llamadas FreeTDS.

Hablando técnicamente, TDS significa Tabular Data Stream (Flujo Tabular de Datos), el cual es el protocolo usado por Microsoft SQL Server para la comunicación con sus clientes. El funcionamiento se basa en que FreeTDS actúa como intermediario entre Microsoft SQL Server y PHP en un ambiente Linux, usando el protocolo ODBC para hacer las transacciones entre ambos.

Una vez instalado FreeTDS, PHP es capaz de acceder a un servidor SQL Server usando sus mismas funciones ODBC.

En el siguiente script, usaremos las funciones ODBC para obtener una conexión.

```
<?
$dsn="Nombre_Fuente_Datos";
$username="Nombre_Usuario";
$password="Usuario_Password";
$Conexion=odbc_connect($dsn,$username,$password);
?>
```

En los códigos anteriores, podemos observar que Linux puede acceder, por ejemplo, a MySQL de forma nativa, y a bases de datos en las cuales se tenga un controlador ODBC disponible, tal es el caso de SQL Server usando a FreeTDS como intermediario para proveer conectividad usando el controlador antes mencionado.

Por otro lado, para la creación de Conexiones, tenemos también a nuestro alcance, a PEAR (PHP Extension and Application Repository) que son Extensiones y Aplicaciones para PHP, las cuales, son también aplicables a las conexiones a bases de datos.

En el caso de una modificación mayor, solo será necesario cambiar el Data Source Name (DSN) para poder acceder a otro tipo de base de datos

El siguiente es un ejemplo de cómo trabaja PEAR:

```
<?php
require_once("DB.php");
$dbType = "mysql";
$dbUser = "Nombre_Usuario";
$dbPass = "Usuario_Password";
$dbServer = "localhost";
$dbName = "Base_de_Datos";
$db = DB::connect("$dbType://$dbUser:$dbPass@$dbServer/$dbName");
?>
```

En el código anterior, se observa que es posible cambiar el tipo de base de datos con el hecho de modificar la variable `$dbType`.

En lo que respecta al uso de PEAR con bases de datos ADO (Activex Data Objects), usamos las funciones `DB_ado`, las cuales tienen conectividad con bases de datos de Microsoft Access, SQL Server, e inclusive, archivos de Excel, todos estos en una plataforma Windows.²⁷

4.5.3 Sesiones

Uno de los desafíos al desarrollar una aplicación Web exitosa, es la de mantener la información del usuario a través de una visita o sesión, mientras que el usuario se mueve de una página a otra en la misma aplicación.

La incapacidad de recordar el estado de peticiones anteriores, presenta desafíos únicos cuando se desarrollan aplicaciones Web como un carrito de compras, que necesita rastrear los artículos que un usuario ha seleccionado cuando se mueve a través de varias páginas del catálogo.

²⁷ Pear :: Package :: DB_ado
http://pear.php.net/package/DB_ado
Pear :: The PHP Extension and Application Repository

ASP.NET

ASP.NET provee una solución para el manejo de información de sesiones vía el nombre de espacio "System.Web.SessionState". Este Nombre de Espacio describe la colección de clases usadas para almacenar datos específicos a un cliente dentro de una aplicación Web.

La información de sesiones, se guarda dentro del Navegador Web de la computadora creando así las Cookies. Estas son creadas cuando se ejecuta código de sesiones dentro de una página programada en ASP.NET.

Inmediatamente de haberse ejecutado el código, se da una sesión con un identificador llamado Session la cual esta formada por una cadena de 120 bits generada de un complejo algoritmo que garantiza que será única.

Podemos guardar datos para cada una de las sesiones, para es, utilizamos los siguientes comandos:

```
Session ("Nombre") = "Jorge"  
Session ("Apellido") = "Perez"
```

Supongamos que queremos dar la bienvenida al usuario y acceder a su nombre guardado en la sesión, entonces usaríamos:

```
Bienvenido <%= Session ( "Nombre" ) %>
```

En dado caso que no necesitemos una variable, podemos removerla con la siguiente instrucción:

```
Session.Contents.Remove ( "Nombre" )
```

Posiblemente necesitemos borrar toda la información de la sesión, así que para eso, utilizaremos el siguiente código:

```
Session.Contents.Clear ( )
```

Para deshacernos de una sesión y generar una nueva, usamos el siguiente comando:

```
Session.Abandon()
```

Nosotros podemos quitar la sesión con una instrucción como esta:

```
<%@ Page EnableSessionState=False %>
```

ASP.NET tiene un manejo de sesiones muy sencillo y sin complicaciones para los programadores, esto se convierte en una rápida programación con pocos errores.

PHP

El soporte de las sesiones le permite registrar un número arbitrario de variables que se conservarán en las siguientes peticiones. Cuando un visitante acceda a su Web, PHP comprobará automáticamente si se le ha enviado un "Session ID" específico con su petición, en cuyo caso se recrean las variables que se habían guardado anteriormente, esto quiere decir que si uno cierra el navegador, y vuelve a entrar a la misma página, la sesión queda guardada a menos que esta sea destruida.²⁸

²⁸ PHP – Funciones para el manejo de Sesiones
<http://www.php.net/session>
PHP.net

Al inicio de cada página en la cual queramos ya sea declarar o acceder a las variables guardadas en sesiones, tendremos que poner el siguiente código al inicio de cada página.

```
<? session_start(); ?>
```

Para poder tener variables dentro de la sesión, utilizamos la siguiente instrucción para declararlas:

```
$_SESSION["Nombre"] = "Jorge";  
$_SESSION["Apellido"] = "Perez";
```

Si nosotros queremos acceder a la variable, ya sea para saber su contenido o para modificarlo, debemos de usar la instrucción:

```
Echo "Bienvenido ";  
Echo $_Nombre;
```

Cuando queremos destruir la sesión, borrando así el contenido que esta pueda tener, usamos la función:

```
session_destroy();
```

El manejo de Sesiones en PHP es también muy sencillo así como en el caso de ASP.NET, el problema viene debido a que una sesión sin destruir puede ocasionar problemas de seguridad en caso de que haya información delicada.

4.5.4 XML

XML es la sigla del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C).

Su objetivo principal es conseguir una página Web más semántica. Aunque una de las principales funciones con las que nace sería suceder al HTML, separando la estructura del contenido. Tiene otras aplicaciones entre las que destaca su uso como estándar para el intercambio de datos entre diversas aplicaciones.

Al igual que el HTML, se basa en documentos de texto plano en los que se utilizan etiquetas para delimitar los elementos de un documento. Sin embargo, XML define estas etiquetas en función del tipo de datos que está describiendo y no de la apariencia final que tendrán en pantalla o en la copia impresa, además de permitir definir nuevas etiquetas y ampliar las existentes.²⁹

Un archivo normal de XML sería algo así:

```
<?xml version=" 1.0 " encoding=" UTF-8 " standalone=" yes "?>
<ficha>
<nombre> Angel </nombre>
<apellido> Barbero </apellido>
<direccion> Avenida Reforma, 36 </direccion>
</ficha>
```

Con este formato, podemos nosotros intercambiar información con otros programas.

ASP.NET

Un ejemplo para escribir lo mismo que tenemos de ejemplo de XML, pero ahora con ASP.NET sería el siguiente.

```
<%@ Import Namespace="System.Xml" %>
<%@ Import Namespace="System.Text" %>
```

²⁹ XML- Wikipedia
<http://es.wikipedia.org/wiki/XML>
Wikipedia

```

<script language="C#" runat="server">
    void Page_Load(object sender, EventArgs e)
    {
        XmlTextWriter EscritorXml = new XmlTextWriter(
            Server.MapPath("ficha.xml"), Encoding.UTF8);
        EscritorXml.WriteStartDocument();
        EscritorXml.WriteStartElement("ficha");
        EscritorXml.WriteElementString("nombre", "Angel");
        EscritorXml.WriteElementString("apellido", "Barbero");
        EscritorXml.WriteElementString("direccion", "Avenida
Reforma, 36");
        EscritorXml.WriteEndElement();
        EscritorXml.WriteEndDocument();
        EscritorXml.Close();
    }
</script>

```

ASP.NET usa el Nombre de Espacio "System.Xml" para sus funciones de escritura en formato XML.

Para la lectura de un archivo XML, este sería el código.

```

<%@ Page Language="vb" %>
<%@ import Namespace="System.Xml" %>
<%@ import Namespace="System.Text" %>
<script runat="server">
Sub ReadXML()
    Dim lectorXml As XmlTextReader = New
XmlTextReader(Server.MapPath("ficha.xml"))
    lectorXml.Read()
    Response.Write(lectorXml.AttributeCount)
    While lectorXml.Read()
        Response.Write(lectorXml.Value)
        Response.Write("<br>")
    End While
End Sub

```

```
Sub Page_Load(Sender As Object, E As EventArgs)
    ReadXML()
End Sub
</script>
```

ASP.NET con sus funciones integradas, van leyendo de nodo en nodo, e identificando el nombre y el atributo de cada una de las etiquetas XML, así como también separa los nodos del tipo de documento. La facilidad de uso es algo que queda muy claro cuando se programa una pagina XML usando ASP.NET.

PHP

Para escribir el mismo archivo XML, solo serán necesarias las siguientes líneas:

```
<?php
$file = "ficha.xml";
$xml = simplexml_load_file($file) or die ("Unable to load XML file!");
$xml->ficha->nombre = "Angel";
$xml->ficha->apellido = "Barbero";
$xml->ficha->direccion = "Avenida Reforma, 36";
file_put_contents($file, $xml->asXML());
?>
```

La lectura es un poco diferente pero igual de sencilla:

```
<?php
$file = "ficha.xml";
$xml = simplexml_load_file($file) or die ("Imposible cargar archivo
XML!");
echo "Nombre: " . $xml->ficha->nombre . "\n";
echo "Apellido: " . $xml->ficha->apellido . "\n";
echo "direccion: " . $xml->ficha->direccion . "\n";
?>
```

En PHP, el XML es algo muy sencillo de manipular, y lo hace de manera muy práctica con solo unas cuantas líneas de código para escribir.

4.5.5 Manejo de Caché

El caché es un conjunto de datos duplicados al disco duro de otros originales, en este caso, descargados de Internet, con la propiedad de que los datos de Internet son costosos de acceder debido al ancho de banda y al tiempo, respecto a la copia en el caché. Cuando se accede por primera vez a un dato en Internet, se hace una copia en el caché; los accesos siguientes se realizan a dicha copia, haciendo que el tiempo de acceso aparente al dato sea menor.³⁰

ASP.NET

En ASP.NET, la técnica más simple es la hacer caché a una página completa, esto se logra con una instrucción muy simple:

```
<%@ OutputCache Duration="60" VaryByParam="none"%>
```

Las siguientes líneas, tienen el efecto de especificar una fecha de expiración, o en este caso, una expiración de un minuto. La última línea tiene la función de refrescar la página cuando esta haya expirado.

```
Response.Cache.SetExpires(DateTime.Now.AddSeconds(60))  
Response.Cache.SetCacheability(HttpCacheability.Public)  
Response.Cache.SetSlidingExpiration(True)
```

También tenemos la posibilidad de que, en caso de que nosotros tengamos separada nuestra página en varias partes, estas partes pueden ir directamente al caché, y no toda la página completa. Esto puede ser aplicado, por ejemplo, en

³⁰ Caché - Wikipedia
<http://es.wikipedia.org/wiki/Cach%C3%A9>
Wikipedia

sitios Web donde no se refresca toda la página, sino solamente la parte inicial, pero en el caso de un menú de categorías, estas pueden ir siempre en caché.

La funcionalidad de el caché en solo fragmentos de una página, la logramos con el siguiente código:

```
<%@ OutputCache Duration="120" VaryByParam="none"
VaryByControl="Categorias" %>
<%@ Register TagPrefix="Acme" TagName="Menu" Src="Menu.ascx" %>
<Acme:Menu Categorias="MenuIzquierdo" runat=server/>
<h1>Hola, La hora es : <%=Now%> </h1>
<Acme:Menu Categorias="MenuDerecho" runat=server/>
```

Con este código, logramos poner en Caché el contenido de MenuIzquierdo y MenuDerecho dentro del archivo Menu.ascx sin tener que poner en caché toda la página. La parte que contiene la hora no irá a caché.

En caso de que nuestro deseo sea el de poner objetos en el caché, podemos hacerlo con una simple instrucción:

```
Cache("MiLlave") = MiValor
```

Para acceder al dato y comprobar que existe algo en caché, podemos hacer lo siguiente:

```
MiValor = Cache("MiLlave")
If Not (MiValor Is Nothing)
Then
    DisplayData(MiValor)
End If
```

El manejo de Caché en ASP.NET no requiere de ningún Nombre de Espacio adicional.

PHP

PHP cuenta con una función para hacer que una página completa expire en el caché. El modo de usar de esta función es el siguiente

```
// Calcular la conversión 24 Horas
$conversion= 3600 * 24;
// Calcular en GMT y agregar la conversión
$expira = "Expires: " . gmdate("D, d M Y H:i:s", time() + $conversion)
. " GMT";
//Salida del encabezado HTTP
Header($expira);
```

Con esta función puesta en el archivo PHP, se puede definir el tiempo que la página permanecerá en el caché.

4.6 Portabilidad

La portabilidad se refiere a la capacidad de un programa de ejecutarse en diferentes sistemas con mínimas modificaciones.³¹

La idea de una migración de Sistema Operativo es algo que se puede tener presente en caso de tener problemas con el mismo, o bien, en el caso de que el servidor Web que elegimos, no cumpla con las expectativas.

Cuando se hace una migración, se tiene que tener en cuenta varios aspectos, como lo son: El costo de migración, los requerimientos de Software y Hardware, así como la portabilidad de nuestro código.

³¹ Diccionario de Informática
<http://www.alegsa.com.ar/Diccionario/diccionario.php?letra=p>
Alegra Argentina

PHP

Para las pruebas de portabilidad, usaremos lo mostrado en la Tabla 4.5.1, para PHP:

Plataforma :	Windows 2003 Server	Linux Fedora Core 4
Servidor Web :	Apache 2.0	Apache 2.0
Interpretador :	PHP 5	PHP 5
Base de Datos :	MySQL usando ODBC	MySQL

Tabla 4.5.1 Pruebas de portabilidad para PHP.

El código que se usará para probar la portabilidad, analizará los siguientes aspectos: Excepciones, Conexión a Base de Datos, Sesiones, y manejo de Caché. También podremos observar si las nuevas funciones trabajan en un ambiente Windows de la misma manera que un ambiente Linux.

A continuación se presenta el código para la comprobación de Manejo de Excepciones:

```
<?
//Manejo de Excepciones
function manejador($errno, $errstr, $errfile, $errline)
{
    throw new Exception($errstr, $errno);
}
set_error_handler('manejador');
try
{
    $a = 1;
    $b = 0;
    $c = $a / $b;
}
catch (Exception $e)
{
```

```

echo "Excepcion Atrapada:\n";
echo "Codigo: ".$e->getCode()."\n";
echo "Mensaje: ".$e->getMessage()."\n";
}
?>

```

Este es el código usado para la conexión y desplgado de datos:

```

<?
//Conexión a Base de Datos
$dsn="NorthWind";
$username="root";
$password="contrasena";
$Conexion=odbc_connect($dsn,$username,$password);
$sql="SELECT * FROM orders";
$Resultado=odbc_exec($Conexion,$sql);
echo "<table><tr>";echo "<td>OrderID</td>";
echo "<td>ShipName</td></tr>";
while (odbc_fetch_row($Resultado))
{
    $compname=odbc_result($Resultado,"OrderID");
    $conname=odbc_result($Resultado,"ShipName");
    echo "<tr><td>$compname</td>";
    echo "<td>$conname</td></tr>";
}
?>

```

Código para la primera página del uso de sesiones:

```

<?session_start();?>
<html>
<body>
<?
$_SESSION ["Nombre"]='Jorge';
$_SESSION ["Apellido"]='Perez';
echo "<a href='sesion2.php'>Sesion2</a>";
?>

```

```
</body>
</html>
```

Segunda página de sesiones:

```
<?session_start();?>
<html>
<body>
<?
echo $_SESSION['Nombre'];
echo ' ';
echo $_SESSION['Apellido'];
echo "<a href='sesion2.php'>Volver a llamar con sesión destruida</a>";
session_destroy();
?>
</body>
</html>
```

Para el siguiente código, se usará la combinación entre el encabezado del manejador de Caché con el despliegado de la consulta de la base de datos anterior:

```
<?
// Manejo de caché
// Calcular la conversión 24 Horas
$conversion= 3600 * 24;
// Calcular en GMT y agregar la conversión
$expira = "Expires: " . gmdate("D, d M Y H:i:s", time() + $conversion)
. " GMT";
//Salida del encabezado HTTP
Header($expira);
?>
<html>
<head>
</head>
<body>
```

```
<?
//Conexión a Base de Datos
//...
?>
</body>
```

Todo el código anterior es ejecutado sin problemas por la combinación de Windows 2003 Server, Apache 2.0, PHP 5 y MySQL con conexión ODBC.

El código anterior es ahora puesto en la plataforma Linux y se observa la reacción del interpretador al trabajar sobre el mismo código.

Para la parte de manejo de Excepciones, la migración fue transparente, así como para las sesiones y el manejo de caché, ya que no se observó falla alguna en el cambio de plataforma.

Al hacer la migración de la base de datos, es posible hacer que PHP haga la conexión con MSQL por medio de ODBC, pero para mejores resultados, se debe de usar las funciones integradas para MySQL.

PHP ofrece en su versión 5, una mejoría en extensiones, sobre todo, en la diseñada para MySQL 4.1 llamada MySQLi (MySQL Improved). Esta instrucción no viene instalada de manera predeterminada en su versión en Windows, aunque se puede instalar en forma separada.

La compañía Zend, desarrolladora de PHP, afirma que la extensión MySQLi provee una velocidad 40 veces mayor a la extensión normal de MySQL.³²

Una vez visto esto, se modifica el código para la conexión directa entre PHP y MySQL usando la extensión MySQLi. El código resultante es el siguiente:

³² Zend Technologies – Using ext/mysqli
<http://www.zend.com/php5/articles/php5-mysqli.php>
Zend Technologies

```

<?
//Conexión a Base de Datos
$link=mysqli_connect('localhost','root')
    or die("No se puede Conectar al Servidor");
mysqli_select_db($link,'northwind')
    or die("No se puede Ingresar a la Base de Datos");
$resultado=mysqli_query($link,"SELECT * from orders");
echo "<table><tr>";echo "<td>OrderID</td>";
echo "<td>ShipName</td></tr>";
while ($res=mysqli_fetch_array($resultado))
{
    $compname=$res["OrderID"];
    $conname=$res["ShipName"];
    echo "<tr><td>$compname</td>";
    echo "<td>$conname</td></tr>";
}
?>

```

La portabilidad que tiene PHP es muy buena, solo hubo que modificar algo de código para hacer la conexión entre PHP y MySQL utilizando la nueva extensión MySQLi.

En lo que respecta a manejo del XML, este fue ejecutado usando el mismo código en una y en otra plataforma: este fue el código a ejecutar:

```

<?php
$file = "ficha.xml";
$xml = simplexml_load_file($file) or die ("Unable to load XML file!");
$xml->ficha->nombre = "Angel";
$xml->ficha->apellido = "Barbero";
$xml->ficha->direccion = "Avenida Reforma, 36";
file_put_contents($file, $xml->asXML());
?>

```

Para la página de lectura, el código fue el siguiente:

```

<?php
$file = "ficha.xml";
$xml = simplexml_load_file($file) or die ("Imposible cargar archivo
XML!");
echo "Nombre: " . $xml->ficha->nombre . "\n";
echo "Apellido: " . $xml->ficha->apellido . "\n";
echo "direccion: " . $xml->ficha->direccion . "\n";
?>

```

El código funciona tanto en plataforma Windows como en plataforma Linux en forma transparente.

ASP.NET

La Prueba que haremos para ASP.NET será portada a Linux usando dos opciones en cuanto a Infraestructura de interpretador .NET. Esto lo podemos ver en la Tabla 4.5.2.

Prueba	A	B
Plataforma :	Windows 2003 Server	Linux Fedora Core 4
Servidor Web :	Internet Information Server	Apache 2.0
Interpretador :	.NET Framework 1.1	Mono
Base de Datos :	MySQL	MySQL

Tabla 4.5.2 Pruebas de portabilidad para ASP.NET.

Para ASP.NET evaluaremos en características de portabilidad usando los siguientes temas: Excepciones, Conexión a Base de Datos, Sesiones, manejo de Caché y XML.

Prueba A: Plataforma Windows 2003 Server

Primeramente, se prueba el código en la plataforma Windows, y se verifica que funcione.

En seguida se presenta el código a utilizar:

```
<%@ Page Language="VB" %>
<script language="VB" runat="server">
Protected Sub button1_click(sender As Object, e As EventArgs)
    Try
        Dim x As Integer = 1
        Dim y As Integer = 0
        Dim z As Integer = x / y
        Response.Write(z)
    Catch err as Exception
        Response.Write ("An Error Occurred"& err.ToString())
    Finally
        Dim z As Integer = 0
    End Try
End Sub
</script>

<form runat="server">
    <asp:button id="button1" onclick="button1_click"
        text="click me" runat="server" />
</form>
```

El código anterior se usó para demostrar el manejo de excepciones de ASP.NET, así mismo, el siguiente código se usa para demostrar la conexión a una base de datos MySQL:

```
<%@ Page Language="VB" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.ODBC" %>
```

```

<script runat="server">
Sub Page_Load(obj As object, e As eventargs)
    Dim ConexionDB As New ODBCConnection("_
        Driver={MySQL ODBC 3.51 Driver};uid=root;password_
        =prueba;Server=localhost;Option=3;Database=test;")
    Dim ComandoDB As ODBCCommand = New ODBC_
        Command("select * from orders", ConexionDB)
    ConexionDB.Open()
    Dim LecturaDB As ODBCDataReader = ComandoDB.ExecuteReader()
    DatosDB.DataSource = LecturaDB
    DatosDB.DataBind()
    ConexionDB.Close()
End sub
</script>
<html>
<body>
    <asp:DataGrid      id="DatosDB"      Runat="server"      CellPadding="3"
HorizontalAlign = "Center"> </asp:DataGrid>
</body>
</html>

```

En el caso de las sesiones, utilizaremos el siguiente código en la primera página de Sesión:

```

<%@ Page Language="VB" %>
<script language="VB" runat="server">
Sub Session_start()
    Session("Nombre") = "Jorge"
    Session("Apellido") = "Perez"
End Sub
Sub Page_Load(obj As object, e As eventargs)
    Session_Start()
End Sub
</script>

<html>
<body>

```

```

Welcome <%= Session ( "Nombre" ) %> , <%= Session ( "Apellido" ) %>
<a href="sesion2.aspx">Destruir Sesion</a>
</body>
</html>

```

Código de la Segunda página de sesión:

```

<%@ Page Language="VB" %>
<script language="VB" runat="server">
Sub Page_Load(obj As object, e As eventargs)
Session.abandon()
End Sub
</script>

<html>
<body>
Welcome <%= Session ( "Nombre" ) %> , <%= Session ( "Apellido" ) %>
<a href="sesions.aspx">Confirmar Destrucción de Sesión</a>
</body>
</html>

```

El código usado para el manejo de XML en el proceso de escritura es el siguiente:

```

<%@ Import Namespace="System.Xml" %>
<%@ Import Namespace="System.Text" %>
<script language="C#" runat="server">
void Page_Load(object sender, EventArgs e)
{
    XmlTextWriter EscritorXml = new _
    XmlTextWriter(Server.MapPath("ficha.xml"), Encoding.UTF8);
    EscritorXml.WriteStartDocument();
    EscritorXml.WriteStartElement("ficha");
        EscritorXml.WriteElementString("nombre", "Angel");
        EscritorXml.WriteElementString("apellido", "Barbero");
        EscritorXml.WriteElementString("direccion", "Avenida
Reforma, 36");
    EscritorXml.WriteEndElement();
}

```

```

    EscritorXml.WriteEndDocument();
    EscritorXml.Close();
}
</script>

```

En seguida, el código del Lector de XML:

```

<%@ Page Language="vb" %>
<%@ import Namespace="System.Xml" %>
<%@ import Namespace="System.Text" %>
<script runat="server">
Sub ReadXML()
    Dim lectorXml As XmlTextReader = New
XmlTextReader(Server.MapPath("ficha.xml"))
    lectorXml.Read()
    Response.Write(lectorXml.AttributeCount)
    While lectorXml.Read()
        Response.Write(lectorXml.Value)
        Response.Write("<br>")
    End While
End Sub

Sub Page_Load(Sender As Object, E As EventArgs)
    ReadXML()
End Sub
</script>

```

En lo referente a manejo de caché, usaremos el siguiente código:

```

<%@ OutputCache Duration="60" VaryByParam="none"%>
<%@ Page Language="VB" %>
<script language="VB" runat="server">
Sub Page_Load(obj As object, e As eventargs)
    Response.Cache.SetExpires(DateTime.Now.AddSeconds(60))
    Response.Cache.SetCacheability(HttpCacheability.Public)

```

```
Response.Cache.SetSlidingExpiration(True)
```

```
End Sub
```

```
</script>
```

Terminada la ejecución del código y su verificación en la plataforma Windows en conjunto con el servidor Web de Internet Information Server, se procede a probar el mismo código en la plataforma de Fedora Core con el interpretador de Mono.

Prueba B: Plataforma Linux Fedora Core 4

En lo referente a la ejecución del código sobre MONO, esta actúa normalmente como en cualquier servidor IIS en conjunto con la tecnología .NET. El único cambio notable, fue en el manejo de base de datos para MySQL, en el cual, ahora se usará un controlador ODBC en Linux para hacer la portabilidad transparente.

Lo que se tiene que hacer para que esto suceda, será instalar el manejador de controladores ODBC, en este caso, podemos tomar el administrador iODBC para que se encargue de este proceso y agregar el controlador específico para MySQL.

Después de hacer esto, crearemos un DSN o Data Source Name (Nombre de Fuente de Datos) con el mismo nombre con el que lo usamos en Windows 2003, así, se obtiene una portabilidad transparente.

En otra manera, usando MONO, podemos obtener acceso a MySQL de forma nativa, usando las bibliotecas de ByteFX para este propósito.

La portabilidad usando ASP.NET de Windows a Linux fue relativamente sencilla, e inclusive también fue el mismo caso para la conexión a MySQL, a excepción del requisito de instalar un administrador de controladores para ODBC.

Mono es un proyecto que nos ofrece una base estable para la portabilidad de nuestras aplicaciones a un ambiente Linux. Aún cuando no es completamente compatible con la tecnología .NET debido a que ciertas funciones que hacen uso del sistema operativo, no son compatibles, la mayoría de las aplicaciones para ASP.NET no tendrán mayores problemas que la adaptación de ODBC.

4.7 Velocidad de Respuesta

La velocidad de Respuesta de un servidor Web, se hace más notable cuando se usan Bases de Datos, es por esto que se decide hacer esta sección para medir el tiempo de reacción de una base de datos usando funciones que traen integradas los mismos lenguajes de Script de Servidor.

Para esto, se harán pruebas como las de Ordenar, insertar, Borrar, modificar y consultar registros de una base de datos. En cada página, se insertará código para medir el tiempo necesario para procesar una página Web.

La base de datos que se usa es MySQL con una tabla llamada "velocidad" de 3 campos:

- Campo "numero" de tipo doble sin llave primaria.
- Campo "caracter" de 20 caracteres de longitud.
- Campo "alfa" de 20 caracteres de longitud.

Para las pruebas de velocidad, se utilizarán las configuraciones que se muestran en la tabla 4.7.1.

Prueba:	A	B
Lenguaje:	PHP	ASP.NET
Plataforma :	Linux Fedora Core 4	Windows 2003 Server
Servidor Web :	Apache 2.0	Internet Information Server 6.0
Interpretador :	PHP 5	.NET Framework 1.0
Base de Datos :	MySQL	MySQL usando ODBC

Tabla 4.7.1 Configuraciones de pruebas de Velocidad.

El método a usar en estas pruebas será el de ejecutar 3 veces el mismo código, y hacer el promedio del tiempo de las 3 pruebas. Entre cada ejecución, la tabla se regresará a su estado anterior, esto quiere decir, por ejemplo, que al haberse agregado datos, estos se borrarán y se agregarán en la siguiente prueba.

Prueba A: PHP

Al inicio de cada página Web, se toma el tiempo de ese instante con una función llamada `microtime()`, la cual mide en fracciones muy pequeñas de tiempo la hora actual. Esta función es invocada al principio y al final del código y se hace una resta sacando así el tiempo total de ejecución del código.

Para todas las páginas que usen las funciones normales de PHP para mysql se utiliza el mismo encabezado, así como el mismo pié de página.

```
<?
//Prueba Velocidad Escritura
$tiempo_inicio = microtime(true);
$link=mysql_connect('localhost','root') or die("No se puede Conectar al
Servidor");
mysql_select_db('northwind') or die("No se puede Ingresar a la Base de
Datos");
//Cuerpo..
//..
$tiempo_fin = microtime(true);
```

```

$tiempo = $time_end - $time_start;
echo "\n Tiempo total: $tiempo segundos \n";
?>

```

A continuación se presentan los tiempos y el código a ejecutar para hacer cada una de las pruebas:

Agregar 10,000 registros

```

for ($i=0;$i<=10000;$i++)
{
$numero = rand(0,9999);
$texto = "Prueba de Texto";
$numeroalfa=rand(0,500);
$alfa = "Prueba Texto".$numeroalfa;
mysql_query      ("INSERT      INTO      velocidad      values
('$numero', '$texto', '$alfa')")
      or die ("Insercion Fallida");
}

```

Ordenar y Mostrar 10,000 registros

```

$Resultado=mysql_query ("SELECT * FROM velocidad ORDER BY numero
ASC") or die ("No se puede ordenar");
echo "<table><tr>";echo "<td>Numero</td>";
echo "<td>caracter</td>";
echo "<td>alfa</td></tr>";
while ($res=mysql_fetch_array($Resultado))
{
    $numero=$res["numero"];
    $caracter=$res["caracter"];
    $alfa=$res["alfa"];
    echo "<tr><td>$numero</td>";
    echo "<td>$caracter</td>";
    echo "<td>$alfa</td></tr>";
}

```

Modificar 10,000 registros

```
mysql_query("UPDATE velocidad SET caracter='Texto Modificado' WHERE  
numero < 10000");
```

Mostrar 10,000 registros

```
$Resultado=mysql_query ("SELECT * FROM velocidad") or die ("No se  
puede ordenar");  
echo "<table><tr>";echo "<td>Numero</td>";  
echo "<td>caracter</td>";  
echo "<td>alfa</td></tr>";  
while ($res=mysql_fetch_array($Resultado))  
{  
    $numero=$res["numero"];  
    $caracter=$res["caracter"];  
    $alfa=$res["alfa"];  
    echo "<tr><td>$numero</td>";  
    echo "<td>$caracter</td>";  
    echo "<td>$alfa</td></tr>";  
}
```

Borrar 10,000 registros

```
mysql_query ("DELETE FROM velocidad WHERE numero < 10000");
```

PHP en su versión 5, integró funciones de acceso mejorado a bases de datos MySQL, llamadas MySQL Improved. Estas funciones serán también medidas en velocidad de respuesta.

Para el código de estas funciones, se hacen pequeños cambios, mostrados en la Tabla 4.7.2.:

Función para MySQL	Función para MySQL Improved
mysql_connect()	mysqli_connect()
mysql_select_db("northwind")	mysqli_select_db(\$link,'northwind')
mysql_query ("SELECT..")	mysqli_query (\$link, "SELECT..")
mysql_fetch_array()	mysqli_fetch_array()

Tabla 4.7.2 Equivalencias de Funciones MySQL y MySQL Improved

Los resultados para ambos tipos de funciones de acceso a MySQL, se muestran en la Tabla 4.7.3:

Prueba	Tiempo Total Funciones MySQL	Tiempo Total Funciones MySQL Improved
Agregar 10,000 Registros	2.625 segundos	2.416 segundos
Ordenar y Mostrar 10,000 Registros	0.889 segundos	0.661 segundos
Modificar 10,000 Registros	0.109 segundos	0.112 segundos
Mostrar 10,000 Registros	0.940 segundos	0.812 segundos
Borrar 10,000 Registros	0.101 segundos	0.086 segundos

Tabla 4.7.3 Resultados de uso de funciones MySQL y MySQL Improved en PHP

Prueba B: ASP.NET

Para la conexión de ASP.NET con MySQL, se usará el controlador ODBC para acceder a la base de datos.

Para la medición de el tiempo, ASP.NET usa una función llamada TimeSpan, la cual captura el tiempo actual, incluyendo el día y la hora. Se insertará en el código al principio y al final de la página para medir el tiempo total de las transacciones.

El Encabezado usado para las páginas de prueba de velocidad, es el siguiente:

```
<%@ Page Language="VB" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.ODBC" %>
<script runat="server">
Sub Page_Load(obj As object, e As eventargs)
    Dim spanEnd, spanStart as TimeSpan
    Dim duration as TimeSpan
    Dim dblSeconds as double
    spanStart = New TimeSpan(DateTime.Now.Ticks)
    Dim ConexionDB As New ODBCConnection_("Driver={MySQL ODBC 3.51_
        Driver};uid=root;password=contrasena;Server=localhost;Option=
3;_
        Database=northwind;")
    ConexionDB.Open()
```

El pie de página es el siguiente:

```
    ConexionDB.Close()
    spanEnd = New TimeSpan(DateTime.Now.Ticks)
    duration = spanEnd.Subtract(spanStart)
    dblSeconds = FormatNumber(duration.Total_
        Milliseconds / 1000, 2, TriState.False)
    Response.Write ("Tiempo total" & CStr(dblSeconds) & " segundos")
End sub
</script>
```

Utilizando el código apropiado para cada prueba, se obtuvieron los siguientes resultados:

Agregar 10,000 registros

```
Dim numero as double
Dim caracter as string
Dim alfa as string
```

```

Dim r As New Random()
caracter = "Prueba de Texto"
Dim miComando As New ODBC_
    Command("insert into velocidad values (?, ?, ?)", ConexionDB)
Dim I As Integer
For I = 0 To 10000
    numero= r.Next(0, 9999)
    alfa="Texto" & r.Next(0, 500)
        miComando.Parameters.Add(Nothing, numero)
        miComando.Parameters.Add(Nothing, caracter)
        miComando.Parameters.Add(Nothing, alfa)
    miComando.ExecuteNonQuery()
Next

```

Ordenar y Mostrar 10,000 registros

```

Dim ComandoDB As ODBCCommand = New ODBC_
    Command("select * from velocidad ORDER BY numero ASC", ConexionDB)
Dim LecturaDB As ODBCDataReader = ComandoDB.ExecuteReader()
DatosDB.DataSource = LecturaDB
DatosDB.DataBind()

```

```

<html>
<body>
    <asp:DataGrid id="DatosDB" Runat="server" CellPadding="3"
HorizontalAlign="Center"></asp:DataGrid>
</body>
</html>

```

Modificar 10,000 registros

```

Dim caracter as String
caracter = "Texto Modificado"
Dim miComando As New ODBCCommand("UPDATE velocidad _
    SET caracter= ? WHERE numero < 10000", ConexionDB)
miComando.Parameters.Add(Nothing, caracter)
miComando.ExecuteNonQuery()

```

Mostrar 10,000 registros

```
Dim ComandoDB As ODBCCommand = New ODBC_
    Command("select * from velocidad", ConexionDB)
Dim LecturaDB As ODBCDataReader = ComandoDB.ExecuteReader()
DatosDB.DataSource = LecturaDB
DatosDB.DataBind()

<html>
<body>
    <asp:DataGrid id="DatosDB" Runat="server" _
        CellPadding="3" HorizontalAlign="Center"></asp:DataGrid>
</body>
</html>
```

Borrar 10,000 registros

```
Dim miComando As New ODBCCommand("DELETE _
    FROM velocidad WHERE numero < 10000",ConexionDB)
miComando.ExecuteNonQuery()
```

En la Tabla 4.7.4 se muestran los resultados de las pruebas anteriores.

Prueba	Tiempo Total
Agregar 10,000 Registros	359.800 segundos
Ordenar y Mostrar 10,000 Registros	0.340 segundos
Modificar 10,000 Registros	0.090 segundos
Mostrar 10,000 Registros	0.200 segundos
Borrar 10,000 Registros	0.070 segundos

Tabla 4.7.4 Resultados de pruebas para ASP.NET

Al crear las gráficas, se puede ver una diferencia notable al agregar 10,000 registros en la Figura 4.7.5. ASP.NET requirió de aproximadamente 6 minutos para hacer esta operación, mientras que PHP usando funciones para MySQL, mostró una rapidez muy superior.

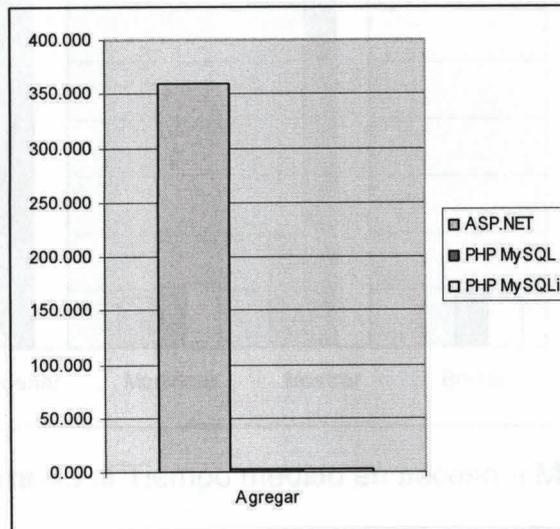


Figura 4.7.5 Tiempo medido para Agregar 10,000 registros.

El causante del exceso de tiempo para agregar datos, se debió al conector ODBC provisto por MySQL AB. Para corroborar esto, se portó PHP a Windows y usando el mismo conector ODBC, se hizo esta misma prueba, mostrando que el tiempo de agregado de datos, fue de 355.00 segundos.

4.3 Herramientas de Desarrollo

Mientras que en las demás pruebas, ASP.NET fue más rápido que PHP en sus 2 variantes de funciones de acceso a MySQL (Figura 4.7.6).

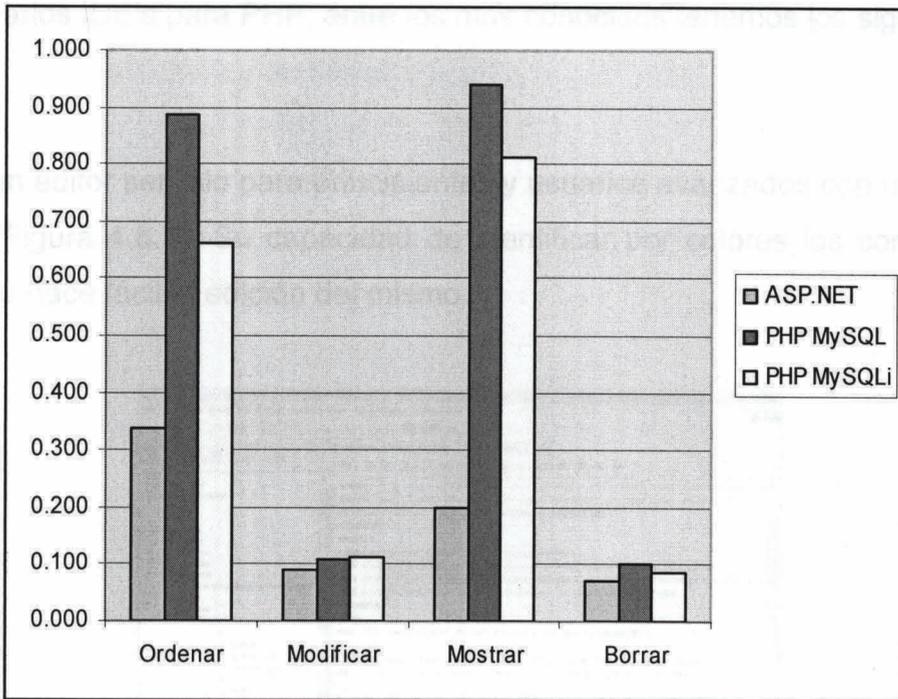


Figura 4.7.6 Tiempo medido en acceso a MySQL.

Si constantemente se están agregando datos a MySQL en gran cantidad, en la plataforma Windows el proceso será lento, debido al conector ODBC proporcionado por MySQL AB. En cambio, si lo más frecuente son consultas, modificaciones y ordenamiento de datos, ambos son buenas opciones, aunque ASP.NET es ligeramente más rápido.

4.8 Herramientas de Desarrollo

La programación ya sea en PHP o en ASP.NET puede ser más simple con el uso de programas especiales o IDEs (Integrated Development Environments) los cuales son un conjunto de herramientas como el Editor de código fuente, un compilador y un depurador³³.

³³ Integrated Development Environment – Wikipedia
http://en.wikipedia.org/wiki/Integrated_development_environment
 Wikipedia

Existen varios IDE's para PHP, entre los más conocidos tenemos los siguientes:

Quanta

Este es un editor sencillo para principiantes y usuarios avanzados con una interfaz sencilla (Figura 4.8.1). Su capacidad de identificar por colores los componentes del código hace fácil la edición del mismo.

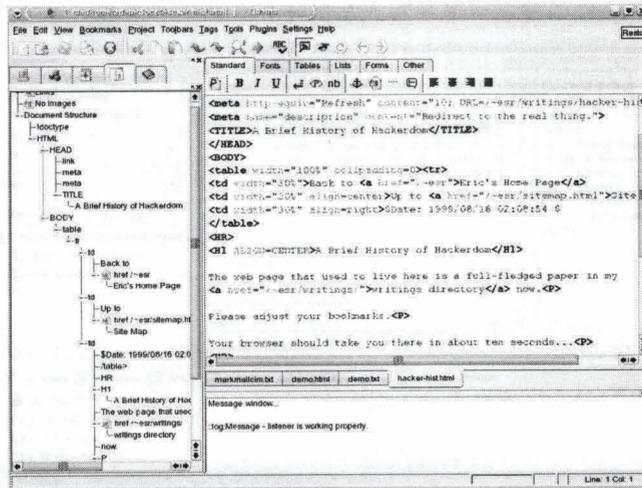


Figura 4.8.1 Interfaz de PHP Designer 2005.

Posee capacidad para identificar y ordenar Hojas de Código en Cascada (CSS)

Plataforma: Linux

Tipo de Licencia: GPL

Precio: Gratuito

KDE Web Dev

Zend Studio

La compañía Zend Technologies, desarrolla esta IDE, en la cual, se destaca la auto-completar del código cuando se escribe, por ejemplo, variables. Zend Studio hace una base de datos con todas las variables disponibles en los documentos en edición en ese momento.

Lo que también destaca, es que se pueden hacer consultas a bases de datos SQL y así tener referencias de inserción de código.

Así como PHP Designer, también posee su propio Servidor Web, bibliotecas de funciones y pantalla de depuración como se muestra en la Figura 4.8.2.

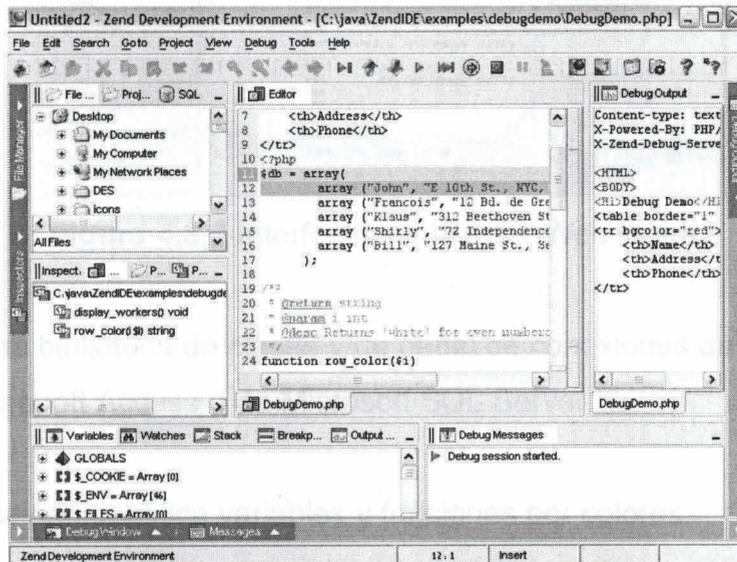


Figura 4.8.2 Depurador de Zend Studio.

Plataforma: Windows/Linux/Mac

Tipo de Licencia: Comercial

Precio: 299 Dólares

Zend Technologies

Para ASP.NET se encuentran varios, y entre los más conocidos tenemos:

Microsoft ASP.NET Web Matrix

Desarrollado por Microsoft, este editor gratuito tiene muchas funcionalidades, como lo es su panel de controles de arrastrar y soltar y su ventana de propiedades la cual posee información sobre los controles seleccionados (Figura 4.8.3).

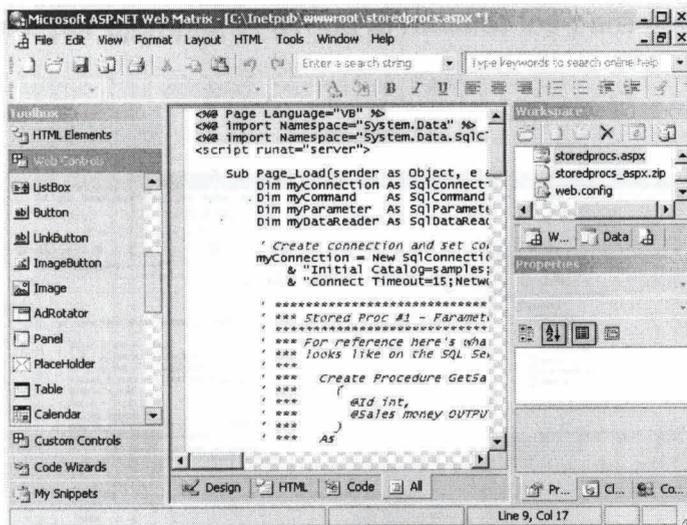


Figura 4.8.3 Interfaz de ASP.NET Web Matriz.

Lo completa una biblioteca de clases y un panel de conexiones de bases de datos, ya sean de Microsoft Access o de Microsoft SQL Server.

A nivel codificación, identifica variables y funciones por colores.

Plataforma: Windows

Tipo de Licencia: Freeware

Precio: Gratuito

Microsoft

Microsoft Visual Studio .NET 2003

Muy parecido al ASP.NET Web Matrix, con la diferencia de conectividad con bases de datos ya sea de SQL Server, conexiones ODBC y OLE DB completando así el esquema ADO.NET.

El panel de controles es más completo (Figura 4.8.4), pero Visual Studio sigue la misma desventaja de que el Auto-Completar sigue sin estar activo al diseñar Páginas ASP. Hay más manipulación sobre el código lo cual hace más sencillo el

estar escribiéndolo, por ejemplo, la ayuda al mostrar los parámetros necesarios en ciertas funciones.

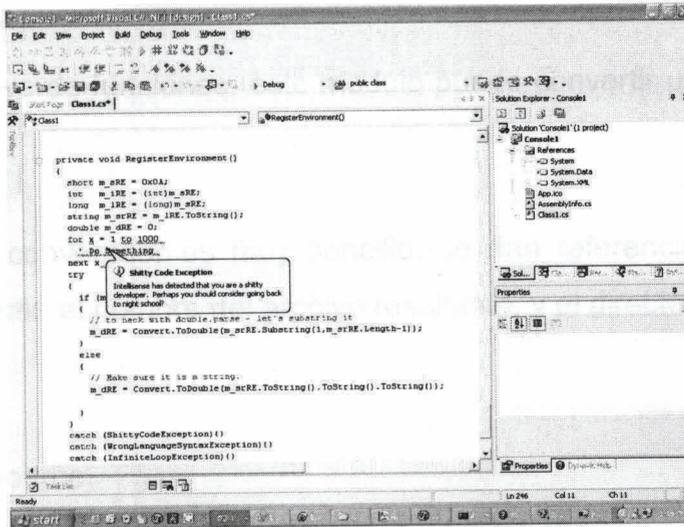


Figura 4.8.4 Interfaz de Visual Studio .NET 2003.

Visual Studio está diseñado no solo para programar en ASP.NET, sino también usando Visual Basic, Visual C#, Visual C++, aparte de incluir elementos como el SourceSafe el cual se encarga de preservar código fuente cuando se trabaja en grupo.

Plataforma: Windows

Tipo de Licencia: Comercial

Precio: 799 Dólares

Microsoft

4.9 Módulos de Valor Agregado

Hay veces que en vez de tener que programar todo lo que necesitamos para nuestro sitio, podemos descargar módulos o sitios completos, disminuyendo así el tiempo de programación o costos.

A continuación se muestran 4 categorías para agregados

Agregados para Documentos de Adobe Acrobat

HTML_ToPDF 3.3

Usando un archivo de funciones, este modulo puede convertir un archivo HTML a documento PDF.

El uso de este convertidor es muy sencillo, se dan referencias al archivo que queremos convertir, el nombre del archivo resultante, y el directorio de imágenes.

Lenguaje: PHP

Licencia: OpenSource

Precio: Gratuito

<http://www.rustyparts.com>

HTML2PDF

Un programador de ASP.NET llamado Albert Pascual, ha publicado un código capaz de utilizar un programa externo (incluido en el archivo del código) para convertir una página HTML en un archivo PDF.

No hay instalación necesaria, más que copiar los archivos a la carpeta del sitio y usar el código incluido dando como parámetro el archivo HTML a convertir

Lenguaje: ASP.NET

Licencia: OpenSource

Precio: Gratuito

<http://www.codeproject.com/aspnet/HTML2PDF.asp>

Agregados para manipulación y muestra de imágenes

phpThumb()

Este es un conjunto de funciones que puede manipular imágenes en varios aspectos, como tamaño, calidad, rotación, imagen de fondo, etc.

También se puede utilizar para recortar imágenes, agregarles marcos y otros efectos especiales. Su instalación es muy sencilla ya que se trata de un archivo PHP al que solo hay que hacer referencia de esta manera para darle, por ejemplo, un ancho de 200 píxeles:

```

```

Lenguaje: PHP

Licencia: OpenSource

Precio: Gratuito

<http://phpthumb.sourceforge.net/>

ThumbTools

Proyecto de Philipos Sakellaropoulos, el cual presenta imágenes de tamaños predeterminados. Se tienen 2 opciones para generar las miniaturas de imágenes, una forma es hacer un vínculo de imagen usando un archivo:

```
<img src='ThumbJpeg.ashx?VFilePath=/imagenes/imagen01.jpg  
&width=200&height=200&Bevel=true'>
```

Otra forma de generar estas imágenes, es usando el archivo ThumbDisplay.aspx el cual presenta una forma que uno puede rellenar con datos como el directorio donde se encuentran las imágenes, así como el tamaño de las miniaturas a generar.

Lenguaje: ASP.NET

Licencia: OpenSource

Precio: Gratuito

<http://www.codeproject.com/aspnet/thumbtools.asp>

Agregados para graficación de datos

JpGraph

Un conjunto muy grande de funciones que es capaz de generar gráficas de imágenes en forma de columnas, barras, líneas, circular, áreas y demás.

Posee la capacidad de generar estas graficas ya sea en 2 o 3 dimensiones, haciendo estas agradables a la vista.

El funcionamiento es sencillo, se dan variables como por ejemplo, el arreglo de datos que se graficará, el tipo de gráficas y colores de líneas.

Lenguaje: PHP

Licencia: OpenSource

Precio: Gratuito

<http://www.aditus.nu/jpgraph/>

.net CHARTING

Programa externo capaz de generar gráficas en tercera dimensión con transparencias, imágenes o degradados de colores. Este programa se integra con el ASP.NET Web Matrix como un componente extra.

No es necesario tener arreglos de datos listos, ya que este programa puede ejecutar secuencias SQL y con esos resultados, generar gráficas. Es capaz de generar consultas a varios tipos de bases de datos.

Lenguaje: ASP.NET

Licencia: Comercial

Precio: 395 Dolares

<http://www.dotnetcharting.com>

Agregados para comercio electrónico

osCommerce

Una tienda en línea completa para PHP la cual es completamente configurable para las necesidades de cualquier persona o empresa. El usuario tiene una interfaz donde posee una cesta de compras, configuraciones personales y menús de todo el sitio, así como productos clasificados.

Para el administrador, se tiene una interfaz en la cual se pueden configurar idioma, categorías, usuarios, altas bajas y modificaciones de productos, así como consultas de todo en relación al sitio.

<http://localhost/oscommerce/catalog/install>

Lenguaje: PHP

Licencia: OpenSource

Precio: Gratuito

<http://www.oscommerce.com/>

Para ASP.NET, se tienen también módulos externos que pueden ser agregados a nuestro Sitio Web, algunos de estos son:

ASP.NET Commerce Starter Kit

El Sitio ASP.NET provee a sus visitantes de varios Kits de Inicializacion entre los cuales podemos encontrar el Commerce Starter Kit. Este es un paquete de código listo para modificar, en el cual ya están listas conexiones a bases de datos.

El propósito de este paquete es el de que el programador, analice el código y pueda así aprender sobre sesiones, manejos de base de datos, y otras funciones básicas.

Lenguaje: ASP.NET

Licencia: OpenSource

Precio: Gratuito

<http://www.asp.net/Default.aspx?tabindex=8&tabid=47>

4.10 Disponibilidad de Código Libre

Cuando se programa un sitio, el código que se llegue a necesitar puede ser demasiado o simplemente no hay una idea clara de cómo programar lo que necesitamos. Es en estos casos cuando nos vemos en la necesidad de investigar y generalmente, copiar código disponible ya sea en Internet o en otras fuentes de información.

ASP.NET

Hay sitios en Internet con código fuente en específico, ya sea para PHP o ASP.NET, y aquí mencionaremos algunos:

PHP

PHPBuilder.com

En este sitio podemos encontrar código fuente que personas han puesto a disposición de la comunidad desarrolladora de código abierto. Podemos encontrar desde algoritmos para ordenar, hasta carros de compras y calculo de costo de envío.

Los códigos son ordenados por números de versiones, y aunque varios son con un costo, generalmente se puede encontrar código gratuito.

<http://www.phpbuilder.com>

PHP Freaks

Desde código fuente hasta artículos y foros se puede encontrar en este sitio dedicado a la programación PHP. Posee también tutoriales acerca de PHP así como una sección dedicada a referencia de MySQL.

<http://www.phpfreaks.com>

Estos son algunos ejemplos de sitios en donde se puede encontrar código gratuito y libre para modificar, e implementar en nuestro Sitio Web.

ASP.NET

Para ASP.NET podemos encontrar de ejemplos los siguientes sitios:

ASP.NET

Sitio oficial del lenguaje, el cual provee aparte de un tutorial para sus 3 variantes de código (C#, Visual Basic y JScript), Galería de controles ordenados por categorías creados por usuarios de ASP.NET y personal de Microsoft, así como los Kits de Inicialización para programadores de ASP.NET.

Provee también una sección para aplicar ASP.NET a dispositivos móviles.

www.asp.net

The Code Project

Sitio en el cual uno puede encontrar proyectos de diversas tecnologías de Microsoft. La sección de ASP.NET contiene artículos con código fuente que ha sido desarrollado por diversos usuarios de este sitio.

4.11 Factores Sociales

A nivel mundial, la presencia de PHP es muy notable. Alrededor de 22 millones de servidores lo tenían instalado y en funcionamiento ya para Abril del 2005. Para estas mismas fechas, ASP.NET tenía poco más de 4 millones y medio³⁴. Esto se muestra en la Figura 4.11.1

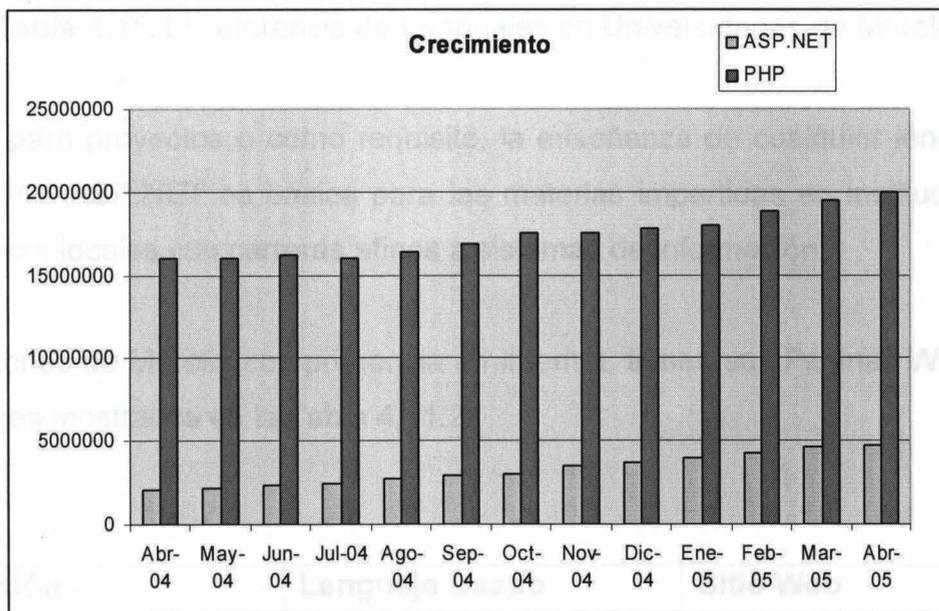


Figura 4.11.1 Presencia de PHP y ASP.NET a nivel mundial.

Netcraft, proporcionó estas estadísticas usando encuestas basadas en Dominios y Direcciones IP.

En universidades de Morelia, Michoacán, la preferencia en el uso de lenguajes, se inclina más hacia PHP. Esto se puede ver en la Tabla 4.11.1:

³⁴Netcraft: Web Server Survey Archives
http://news.netcraft.com/archives/web_server_survey.html
Netcraft

Universidad	Universidad Latina de América	Instituto Tecnológico de Morelia	Universidad Vasco de Quiroga
Lenguaje Usado	ASP.NET	PHP	PHP
Carrera	Lic. en Sistemas Computacionales	Ing. en Sistemas Computacionales	Ing. en Sistemas Computacionales
Materia en la que se requiere el Lenguaje	Programación II	Sistemas Distribuidos	Tecnologías de Información Emergente

Tabla 4.11.1 Preferencia de Lenguajes en Universidades de Morelia.

Ya sea para proyectos o como requisito, la enseñanza de cualquier lenguaje, ya sea PHP o ASP.NET es básica para las materias impartidas en instituciones de enseñanza locales con carreras afines a sistemas de información.

Instituciones de Morelia con presencia en Internet, tienen sus Páginas Web en los Lenguajes mostrados en la Tabla 4.11.2:

Institución	Lenguaje Usado	Sitio Web
Cinépolis	ASP.NET	cinepolis.com.mx
Monarcas Morelia	ASP.NET	fuerzamonarcas.com
Gobierno del Estado de Michoacán	PHP	michoacán.gob.mx
H. Ayuntamiento de Morelia	ASP.NET	morelia.gob.mx
Provincia	PHP	provincia.com.mx
Dealer de México	PHP	dealer.com.mx
SKY Código	PHP	codice1.com

Tabla 4.11.2 Páginas Web de Morelia y los Lenguajes usados.

Conclusiones

Las empresas en Michoacán están divididas entre ASP.NET y PHP. No hay clara superioridad entre preferencias de lenguaje.

El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor.

El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor. El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor.

El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor.

El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor. El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor.

El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor. El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor.

El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor. El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor.

El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor. El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor. El uso de ASP.NET y PHP en Michoacán está creciendo, pero el uso de ASP.NET es mayor.

5.0 Conclusiones

Ambos lenguajes, PHP y ASP.NET tienen sus puntos fuertes y débiles. En esta parte analizaremos los resultados de las comparaciones del capítulo anterior.

Como se mencionó anteriormente, el objetivo no es saber cuál es mejor de los 2 lenguajes, sino cuál es mejor para nuestras necesidades. He aquí los resultados de cuál es mejor para las necesidades por puntos en específico

Inversión en Plataforma y Software a largo plazo.

Al hacer una inversión, hay muchos factores que debemos tomar en cuenta. Pero cuando nuestro principal factor es el monetario, las soluciones de uno y otro lenguaje son muy distintas.

La plataforma Windows para ASP.NET es una solución que no es gratuita. De la manera que se le quiera ver, uno terminará pagando al menos \$ 451.59 Dólares por una combinación de Windows 2003 Web Server junto con IIS, ASP.NET y el antivirus.

En el caso de la plataforma Linux para PHP, Fedora Core nos ofrece gratuitamente Apache, PHP y MySQL todo integrado y listo para usar.

Si lo que queremos es un sistema operativo y todo el entorno que cubre soporte, actualizaciones y boletines de seguridad por parte de las empresas desarrolladoras, tendremos que usar la opción de RedHat Enterprise AS Standard Server con un precio de \$ 1,499.00 Dólares para nuestras necesidades en vez de Fedora Core. Y para Windows usaremos Windows 2003 Standard Server con un precio de \$ 1,004.65 Dólares con el antivirus incluido.

En cuestión de Salarios y Consultoría, La plataforma Linux es más cara, Los salarios de un Ingeniero encargado de un servidor PHP para Linux, son de \$ 1,300.00 Dólares mensuales, mientras que los de un Ingeniero programados ASP.NET con conocimientos de Windows 2003 gana \$1,000.00 Dólares al mes.

Para resumir, si el factor financiero es lo más importante dentro de la decisión de elegir un lenguaje de programación, estos serían los resultados:

En caso de que el presupuesto sea muy mínimo o simplemente no exista, la opción obvia a tomar es PHP, esto debido a que la inversión desde el Sistema Operativo hasta la base de datos, es nula.

Si el presupuesto es pequeño, pero se desea contar con soporte de la empresa del sistema operativo, la opción sería ASP.NET en combinación con Windows 2003 Web Server. Esto debido a que el sistema operativo de servidor más barato y con soporte, es el Windows 2003 Web Server. En este caso, la opción de Base de datos se inclinaría a MySQL debido a que el SQL Server de Microsoft es prohibitivamente costoso.

Cuando lo que se desea en el factor económico es acerca de costos a largo plazo, la opción de Microsoft puede ser la mejor solución. Debido a que los salarios y soporte externo son más costosos en Linux, una empresa con un año en funcionamiento puede experimentar mayores costos usando Linux.

Cuando el dinero no es problema, es aquí cuando la decisión no reside en este tema, debido a que se deben considerar principalmente otros factores como lo son seguridad, codificación, código libre, etc.

Seguridad Global.

En cuanto a las comparaciones de seguridad, el estudio realizado por Security Innovations, muestra que Windows Server 2003 es más seguro en cuestión de vulnerabilidades y Días de Seguridad Comprometida que RedHat Enterprise.

En referencia a los Virus, es obvio que Linux es el más seguro, debido a que no existen para este sistema operativo. Windows en cambio, necesita de la ayuda de un antivirus para este problema, el cual ha ocasionado serios daños, como el caso del muy famoso Blaster que puso en duda la capacidad de Windows como servidor estable.

Las vulnerabilidades de los lenguajes en sí, no son muy diferentes las unas de las otras, la realidad en este aspecto es que ambos son seguros a excepción de ciertos casos como los descritos en el capítulo anterior, aunque pueden ser descubiertas fallas más adelante.

Como conclusión en seguridad, Windows es un servidor realmente seguro en una instalación limpia sin el Internet Explorer activado, a menos que se habiliten las funciones de navegación, en dado caso, Linux es la mejor opción.

En el caso de que los virus sean algo primordial para mantener alejados, Linux es la opción correcta. Windows seguirá teniendo problemas con los virus, así que siempre se deberá tener un antivirus actualizado, pero eso no quiere decir que estaremos libres de ellos.

Instalación

Para esta parte, se tomaron en cuenta las siguientes instalaciones: Sistema Operativo, Servidor Web, Lenguaje y Base de Datos.

Refiriéndonos al Sistema Operativo, el que menos trabajo requiere para instalar es Windows 2003 Server ya que su interfaz y su guía paso a paso hace las cosas muy fáciles, inclusive para los novatos. Su compatibilidad lo hace muy sencillo y rápido de instalar.

En Fedora Core, Su interfaz de instalación es muy sencilla y fácil de entender, eso solo si se logra reconocer el refrescamiento del monitor y la tarjeta de video, ya que en caso contrario se tendrá que utilizar la instalación en modo texto.

En cuanto a instalaciones de Servidores Web, estos fueron sumamente sencillos de hacer. Por su parte Fedora Core instala Apache desde el inicio. Windows 2003 Server instala Internet Information Server por medio del Panel de Control en la sección de Agregar y Quitar Programas.

Los lenguajes también fueron fáciles de instalar por la misma vía ya sea en Fedora Core en la instalación del Sistema Operativo, e Internet Information Services para Windows 2003 por medio del Panel de Control.

La instalación de MONO en Fedora Core dió muchos problemas, ya que hay muchas guías en Internet para su instalación, pero este requiere de su servidor web XSP el cual requiere una instalación aparte y poco después hay que vincularlo con Apache.

Para MySQL como servidor de base de datos, en ambas plataformas fue muy sencilla en ambas instalaciones.

Para resumir la evaluación de la instalación, podemos decir que para ambos lenguajes, ya sea PHP o ASP.NET requieren un cierto nivel de conocimientos para las instalaciones en sus respectivos Sistemas Operativos, pero en ambos casos, fueron sencillos y rápidos de instalar.

Codificación

En esta área se puede observar que ambos lenguajes tienen capacidades de manejar excepciones, en PHP ya se incorporaron, ya que en la versión 4, estas todavía no existían.

Para las conexiones de datos, se encontró que en ambos lenguajes poseen capacidades de conexiones a bases de datos en amplias variedades, ya sean MySQL, Microsoft Access, Microsoft SQL Server, conexiones ODBC, entre otras. Cuando la base de datos es primordial, no es ningún problema crear una conexión en cualquiera de los lenguajes. Inclusive, extensiones como PEAR para PHP, hacen este proceso de manera más sencilla.

Las sesiones son una base primordial para cualquier Sitio Web, y tanto ASP.NET como PHP pudieron manejar excepciones de manera sencilla.

En XML fue el mismo caso, ambos lo manejan sin muchas complicaciones, ya sea leyendo los datos o escribiéndolos en un archivo.

Para manejo de caché, la diferencia se notó más en ASP.NET. Mientras que PHP solo puede mandar a caché la página completa, así como variables, ASP.NET puede mandarla en porciones, esto quiere decir, que si tenemos encabezados y menús que nunca cambian, pero en cambio, tenemos un cuerpo de página que cambia a diario, podemos almacenar en caché los menús y el encabezado y dejar que solo se cargue desde nuestro sitio, el cuerpo de la página.

Ambos lenguajes son muy prácticos en cuanto a su programación, aunque la diferencia es notable a la hora de programar grandes cantidades de código. Mientras que PHP necesita de poco código, ASP.NET requiere de líneas como las de inclusión de Nombres de Espacio, especificación del lenguaje a usar, llamados a controles, etc.

Por su parte, PHP tiene de desventaja el que su código es sensitivo a mayúsculas y minúsculas. Una variable mal escrita puede hacer que todo el código simplemente no sea ejecutado.

Portabilidad

La transición entre un Sistema Operativo y otro, resultó ser casi imperceptible en ambos lenguajes, ya sea de PHP al pasar de Windows a Linux, y lo mismo con ASP.NET usando el proyecto MONO en la plataforma Linux.

Los únicos inconvenientes salieron a flote en las bases de datos, se tuvo que cambiar código para cumplir con las especificaciones de las bases de datos.

En el caso de PHP, se hizo el cambio de funciones MySQL a ODBC para acceder las bases de datos en Windows.

En ASP.NET, se hizo el cambio de MySQL usando ODBC para Windows, al uso del mismo esquema en Linux, con la instalación de un administrador de controladores externos, en este caso, iODBC.

Estos fueron los únicos cambios requeridos para la portabilidad de los lenguajes.

Velocidad de Respuesta

La Metodología que se usó en estas pruebas, fue la de ejecutar 3 veces las pruebas, y hacer un promedio de estas pruebas.

Estas pruebas arrojaron resultados inesperados, ya que PHP fue más rápido por amplio margen al agregar 10,000 registros en solo 2.625 segundos, mientras que

ASP.NET lo hizo en 359.800 segundos, lo que es lo mismo, 6 minutos (Figura 5.1).

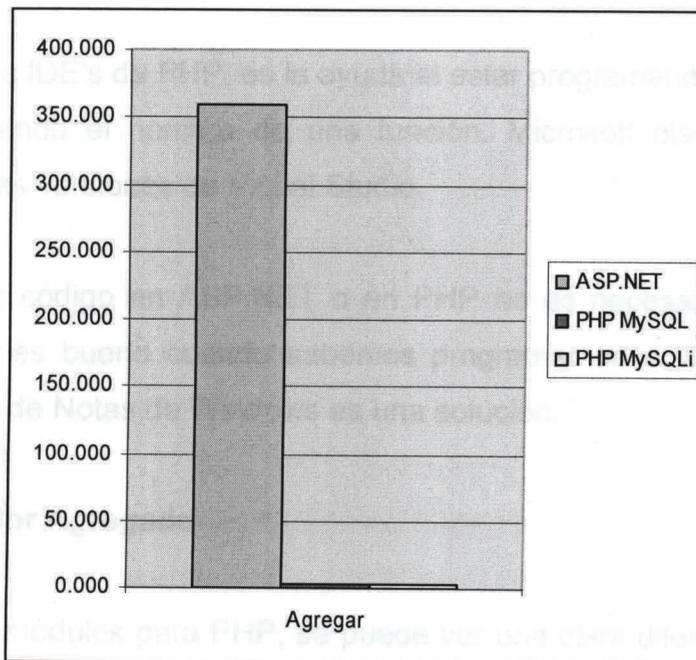


Figura 5.1 Tiempo medido para Agregar 10,000 registros.

El gran margen de tiempo entre PHP y ASP.NET al agregar registros, se debió al controlador ODBC proporcionado por MySQL AB, esto se confirmó al portar PHP de Linux a Windows y ejecutar las mismas pruebas, con los mismos resultados. Es así, que queda descartado que el lenguaje sea la causa de el retraso al hacer esta prueba en específico.

Es obvio que al agregar registros en grandes cantidades, no hay más opción que evitar el controlador ODBC de MySQL AB, El uso de PHP con Linux es mejor en esta prueba, mientras que al hacer consultas de grandes volúmenes de información, estas es mejor hacerlas en ASP.NET.

Herramientas de Desarrollo

En cuanto a las Herramientas de Desarrollo o IDE's, para PHP hay una gran variedad, mientras que para ASP.NET se encontraron unas cuantas compatibles

con ASP.NET, el cual los 2 mejores programas encontrados para desarrollar en este ultimo, son desarrollados por Microsoft.

La ventaja de los IDE's de PHP, es la ayuda al estar programando, la cual aparece al estar escribiendo el nombre de una función. Microsoft planea agregar esta función en futuras versiones de Visual Studio.

Para desarrollar código en ASP.NET o en PHP no es necesario invertir dinero, cualquier editor es bueno cuando sabemos programar en el lenguaje deseado, inclusive el Bloc de Notas de Windows es una solución.

Módulos de Valor Agregado

Al encontrar los módulos para PHP, se puede ver una clara diferencia entre estos y los de ASP.NET, La calidad.

Los Módulos hechos para PHP, son el resultado de proyectos simples que se fueron mejorando hasta la calidad que alcanzaron a lograr. osCommerce es una aplicación de muy buena calidad que aparte de ser fácil de instalar y usar, viene en varios idiomas.

Para ASP.NET hay proyectos desarrollados que aunque son sencillos fáciles de usar, no son tan completos y vistosos como los que existen para PHP, a excepción de los módulos en los que hay que pagar por una licencia.

Disponibilidad de Código Libre

Haciendo a un lado los Módulos de Valor Agregado, el código fuente disponible en Internet para PHP y ASP.NET es muy extenso, independientemente del lenguaje.

Es muy sencillo encontrar el código que buscamos en diferentes sitios en Internet, la diferencia viene cuando uno trata de comprender el código que se expone en Internet. Mucho del código encontrado tiene varias soluciones para un problema único.

Para ambos lenguajes, hay muchos sitios aparte de los mencionados en el capítulo anterior en los que podemos encontrar la solución a los problemas referentes a código.

Factores Sociales

La supremacía de PHP sobre ASP.NET es clara al observar las estadísticas, las cuales reflejan una diferencia muy grande. Esto puede ser un factor para influenciar a las demás personas a programar en este lenguaje, y no en ASP.NET

Estas estadísticas, también son las mismas para las universidades locales, las cuales reflejan una cierta preferencia por usar PHP en vez de ASP.NET

En cuanto a las empresas locales, se divide la preferencia por los lenguajes en sus Sitios Web. La mitad de los sitios investigados que requieren lenguajes de programación, usan PHP y la otra mitad ASP.NET.

La realidad, es que según las necesidades de la empresa, es el lenguaje que deberán usar. Esto quiere decir, por ejemplo, que si la empresa tiene poco presupuesto, deberá anteponer el rubro de "Inversión de Plataforma" al de "Seguridad".

Cada empresa deberá poner prioridades sobre que es lo que en realidad necesita, y dejar lo más superfluo al último en la lista. Al tener esta, se debe ir evaluando para saber cual lenguaje es mejor para las necesidades.

Referencias

Bibliografía

1. ASP.NET in a Nutshell
G. Andrew Duthie
O'reilly 2003
2. Linux vs. Windows
Total Cost of Ownership Comparison
CyberSource
2002-2004
3. Role Comparison Report – Web Server Role
Richard Ford
Security Innovation
Marzo 2005

Referencias Web

1. Wikipedia Server-side scripting
http://en.wikipedia.org/wiki/Server-side_scripting
Wikipedia
[Septiembre 2005]
2. Zend Technologies – Under The Hood of PHP4
<http://www.zend.com/zend/art/under-php4-hood.php>
Zeev Suraski
[Septiembre 2005]
3. Wikipedia – Microsoft .NET
http://en.wikipedia.org/wiki/Microsoft_.NET
Wikipedia
[Septiembre 2005]
4. ASP.NET What's New in ASP.NET 2.0
<http://beta.asp.net/QUICKSTART/aspnet/doc/whatsnew.aspx>
ASP.NET
[Octubre 2005]
5. Windows Server 2003
<http://www.pc-actual.com/Actualidad/Noticias/Infraestructuras/Software/20030401010>

13. PC Actual
[Octubre 2005]
<http://www.redhat.com/software/rhel/compare/client/>
6. Wikipedia – Servidor HTTP Apache
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache
Wikipedia
[Octubre 2005]
<http://www.netcraft.com/>
7. Wikipedia – Internet Information Services
<http://es.wikipedia.org/wiki/IIS>
Wikipedia
[Octubre 2005]
<http://news.netcraft.com/archives>
8. Netcraft
<http://www.netcraft.com>
Netcraft
[Octubre 2005]
<http://www.netcraft.com/archives>
9. Mono – About Mono
<http://www.mono-project.com/Mono:About>
The Mono Project
[Noviembre 2005]
<http://www.mono-project.com/security/article.php?id=21>
10. MySQL Wikipedia
<http://es.wikipedia.org/wiki/MySQL>
Wikipedia
[Noviembre 2005]
<http://www.mono-project.com/library/default.asp?wid=21>
11. Yahoo Noticias
<http://mx.news.yahoo.com/050907/40/1i5zx.html>
Yahoo!
[Noviembre 2005]
<http://www.mono-project.com/library/default.asp?wid=21>
12. Ciclo de Vida del Soporte de Microsoft
<http://support.microsoft.com/gp/lifecycle>
Microsoft
[Octubre 2005]
<http://www.mono-project.com/library/default.asp?wid=21>
13. Red Hat Enterprise Linux
http://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux
Wikipedia
[Octubre 2005]
<http://www.mono-project.com/library/default.asp?wid=21>
14. RedHat Client Support and Pricing
<http://www.redhat.com/software/rhel/compare/client/>
RedHat
[Octubre 2005]
<http://www.mono-project.com/library/default.asp?wid=21>

15. RedHat Server Support and Pricing
<http://www.redhat.com/software/rhel/compare/server/>
RedHat
[Octubre 2005]
16. Occ México
<http://www.occ.com.mx>
Occ Internacional
[Noviembre 2005]
17. ASP.Net Security Flaw can ByPass Password
http://news.netcraft.com/archives/2004/10/07/aspnet_security_flaw_can_bypass_password.html
Netcraft News
[Noviembre 2005]
18. PHP Zaps Security Leaks
<http://www.internetnews.com/security/article.php/3382181>
internetnews.com
[Noviembre 2005]
19. ADO Glossary
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/hm/mddefglossary.asp>
Microsoft Developer Network
[Noviembre 2005]
20. Todo sobre ADO
<http://www.gamarod.com.ar/articulos/articulos.asp?id=35>
Gamarod
[Noviembre 2005]
21. DirList2ODBC - Terminology
<http://dirlist.sourceforge.net/docs/odbc/terms.htm>
SourceForge
[Noviembre 2005]
22. MySQL Connector/ODBC 3.51
<http://dev.mysql.com/downloads/connector/odbc/3.51.html>
MySQL AB
[Octubre 2005]
23. PHP – Funciones para el manejo de Sesiones
<http://www.php.net/session>
PHP.net
[Octubre 2005]

24. XML- Wikipedia
<http://es.wikipedia.org/wiki/XML>
Wikipedia
[Octubre 2005]
25. Caché - Wikipedia
<http://es.wikipedia.org/wiki/Cach%C3%A9>
Wikipedia
[Octubre 2005]
26. Diccionario de Informática
<http://www.alegsa.com.ar/Diccionario/diccionario.php?letra=p>
Alegsa Argentina
[Octubre 2005]
27. Integrated Development Environment – Wikipedia
http://en.wikipedia.org/wiki/Integrated_development_environment
Wikipedia
[Noviembre 2005]
28. Netcraft: Web Server Survey Archives
http://news.netcraft.com/archives/web_server_survey.html
Netcraft News
[Octubre 2005]
29. Pear :: Package :: DB_ado
http://pear.php.net/package/DB_ado
Pear :: The PHP Extension and Application Repository
[Diciembre 2005]