

**Universidad Latinoamericana de Ciencia y
Tecnología**

Facultad de Ingeniería

Escuela de Informática

**Trabajo final para optar al grado de Licenciatura
en Ingeniería Informática con énfasis en
Desarrollo de Software**

**Título del trabajo: Herramientas de desarrollo
para aplicaciones web**

**Ing. Ever Barahona Mendoza¹
Cédula: 6-0323-0464**

Profesor tutor: Miguel Pérez

Agosto del 2007

¹ Bachiller en Informática Empresarial. Candidato a licenciado en Ingeniería Informática con énfasis en Desarrollo de Software, ULACIT.
Correo electrónico: ebarahonam@gmail.com, eibam@hotmail.com

Índice

Resumen.....	iii
Abstract.....	iii
I. Introducción	- 1 -
II. Aplicaciones web.....	- 2 -
Ventajas.....	- 3 -
Desventajas	- 3 -
Características de las aplicaciones web	- 3 -
Estándares de herramientas de desarrollo.....	- 4 -
Evolución de las aplicaciones web.....	- 5 -
1- Herramientas Microsoft.....	- 8 -
a. Información general.....	- 8 -
b. Arquitectura e infraestructura (plataforma).....	- 9 -
c. Requerimientos.....	- 12 -
d. Beneficios.....	- 12 -
e. Seguridad.....	- 13 -
f. Implementación.....	- 14 -
Lenguaje Visual Basic.NET	- 14 -
Lenguaje C#.NET.....	- 15 -
2- Lenguaje Java (Sun Microsystems)	- 15 -
a- Información general.....	- 15 -
b- Arquitectura e infraestructura (plataforma).....	- 17 -
c- Requerimientos	- 18 -
d- Beneficios	- 19 -
e- Implementación	- 19 -
IV. Análisis y discusión de resultados.....	- 20 -
V. Recomendaciones.....	- 22 -
VI. Conclusiones.....	- 23 -
VII. Referencias bibliográficas	- 24 -

Resumen

Los cambios tecnológicos y las nuevas necesidades de negocios de las empresas han sido algunos de los factores que han impulsado el desarrollo de aplicaciones web, así como la consolidación de los lenguajes de programación. Por tanto, este artículo presenta los beneficios que se obtienen al implementar aplicaciones web, así como las características, ventajas y desventajas que se deben considerar en la implementación de aplicaciones de este tipo. También se incluye la descripción y comparación de las tecnologías de diferentes herramientas de programación y los beneficios que pueden brindar en un desarrollo web con diferentes modelos que se adapten al negocio de las compañías. Este documento fomenta en el lector el análisis de las diferentes características, costos-beneficios para poder implementar y desarrollar aplicaciones web de acuerdo con la herramienta que se pueda seleccionar, así como despertar el interés en investigar sobre diversos factores que rodean este tema.

Palabras claves

Páginas web, sitios web, Internet, intranet, cliente-servidor, *browser*, programación orientada a objetos, librerías, lenguajes de programación.

Abstract

The technological changes and the new businesses' needs of the companies have been some of the factors that have led to encourage the development of Web applications, as well as the consolidation of the programming languages. Therefore this article presents the benefits that are obtained when implementing Web applications, as well as the characteristics, advantages and disadvantages that should be considered in the implementation of applications of this type. Also it is included the description and comparison of the technologies of different programming tools, and the benefits that can offer in a Web development with different models that adapt to the business of the companies. This document creates in the reader the opening of an analysis of the different characteristics, costs - benefits to be able to implement and develop Web applications according to the selected tool, as well as developing the interest to investigate about diverse factors that surround this topic.

Key Words

Web Pages, Web Sites, Internet, Intranet, Client-Server, Browser, Object Oriented Programming, Library and Programming Languages.

I. Introducción

Existen en el mercado multitud de aplicaciones que facilitan la creación y el mantenimiento de los sitios web. Las primeras páginas web se realizaron a mano, pero la complejidad de los lenguajes y sistemas implicados en el desarrollo de los portales web hicieron que se fuera abandonando la creación artesanal en favor de una más industrial. El problema con esto es que al final se corre el riesgo de que los portales se parezcan demasiado unos a otros.

En este sentido, la ingeniería de la web hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones web complejas y de gran dimensión, en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones, haciendo estas páginas más dinámicas.

La ingeniería de la web es multidisciplinar y aglutina contribuciones de diferentes áreas: arquitectura de la información, ingeniería de requisitos, diseño de interfaz de usuario, usabilidad, diseño de gráfico y de presentación, diseño y análisis de sistemas, ingeniería de software, ingeniería de datos y recuperación de información, entre otras.

La ingeniería de la web no es un clon o subconjunto de la ingeniería del software, aunque ambas incluyen desarrollo de software y programación, pues a pesar de que la ingeniería de la web utiliza principios de ingeniería de software, incluye nuevos enfoques, metodologías, herramientas, técnicas, guías y patrones para cubrir los requisitos únicos de las aplicaciones web.

En la ingeniería de la web se pueden encontrar lenguajes de programación tales como Visual Basic, .NET, C#.NET y Java, entre otros, que han venido a consolidarse y a madurarse para poder crear soluciones seguras, confiables, robustas, usables, que garanticen el rendimiento y mejoren los servicios del negocio empresarial.

Los principales aspectos del desarrollo de aplicaciones web y sus respectivas herramientas son los siguientes:

- Entornos de desarrollo de aplicaciones web integrados.
- Arquitectura de las aplicaciones web.
- Beneficios de desarrollar aplicaciones web.
- Beneficios que ofrece las herramientas de programación.

- Métodos, tecnología y técnicas que ofrecen las herramientas de programación para una buena construcción e implementación de las aplicaciones web.

II. Aplicaciones web

En la actualidad, las aplicaciones web son una herramienta que viene a revolucionar la tecnología, estándares, arquitectura, diseño e implementación de estas, entre otros. Hay una serie de aplicaciones web, según la característica para la cual sea construida; por ejemplo, hay aplicaciones meramente informativas, de búsqueda de información, transaccionales, etc.

Las aplicaciones web que más se desarrollan son de características transaccionales, y reúnen una serie de estándares, metodología y procedimientos para su desarrollo e implementación.

Las compañías se han venido preguntando cómo expandir y obtener información de una forma segura, eficaz, que pueda disminuir costos operativos, aumentar la productividad y mejorar el servicio con los clientes, proveedores y otras entidades; para esto se han visto en la necesidad de invertir en aplicaciones web que satisfagan dichos objetivos. Para la construcción de aplicaciones web hay que tomar en cuenta ciertos aspectos, tales como: diseño de interfaz, diseño de procesos, arquitectura por desarrollar, seguridad, integridad de información, eficacia en las transacciones y de procesamiento. Entre los beneficios que tienen las empresas que desarrollan aplicaciones web, se encuentran:

- Reducción de costos de comunicación con clientes, proveedores y entidades externas.
- Aumento de la productividad, haciendo más eficiente los procesos administrativos, de comercialización y de cobros, y brindando y obteniendo la información en tiempo justo.
- Incremento de las ventas.
- Mejor servicio al cliente, por tanto disminuyen los costos de servicio al cliente (físicamente) y reclamos de estos.

Ventajas

- La aplicación web se instala solamente en el servidor web, por tanto no requiere que se haga una instalación en el cliente, pues este accede por medio de un *browser*², lo cual disminuye el costo de actualización de la aplicación y su mantenimiento.
- Una de las ventajas de construir aplicaciones web es que los *browser* mediante los cuales el usuario accede a la aplicación, se han estandarizado, por lo cual no afecta la arquitectura y la herramienta en que fue desarrollada la aplicación, ya que el usuario final recibe una pagina web ya procesada.

Desventajas

- Para poder desarrollar aplicaciones web se requiere una inversión inicial considerable; sin embargo, a medida que esta produzca, la inversión será compensada con los beneficios que se vayan obteniendo.
- Por el estándar y las diversas herramientas de desarrollo que existen en el mercado, se corre el riesgo de que los diseños de las aplicaciones se parezca unos a otros, lo cual puede verse en problemas de delincuencia, como la piratería. Además, en el usuario se puede generar cierta incertidumbre por el robo de información que se pueda dar por la "duplicidad" de la aplicación web.

Características de las aplicaciones web

Las aplicaciones web deben tener ciertas características de importancia para que sean de fácil uso para el usuario final:

- **Facilidad de aprendizaje:** el diseño de la interfaz debe ser atractivo y sencillo para el usuario, con el fin de que este tenga la facilidad de navegar y realizar las diferentes tareas dentro de la aplicación sin ningún problema. Un buen diseño es un control de riesgo para el usuario, ya que le permite disminuir los errores producidos por este.
- **Velocidad de desempeño:** las aplicaciones web deben ser desarrolladas para que tengan un buen desempeño en el tiempo de respuesta y procesamiento de la información. Este aspecto se debe garantizar para que sea fácil de usar y realizar las diferentes transacciones.

² Navegador con el que se puede acceder y ver páginas web.

- **Control de errores:** la aplicación debe ser desarrollada bajo un buen diseño para evitar que el usuario final esté **propenso a evitar** una serie de errores en la aplicación. También debe ser capaz de soportar, de capturar y encapsular los errores que se puedan generar causados por agentes externos a la aplicación, y de no exponer los errores al usuario final, ya que hay aplicaciones web que le muestran al usuario todo el detalle de un error que se produjo en la base de datos, lo cual brinda cierta información que debe ser íntegra, como por ejemplo *queries*³ que se realizan a las tablas de la base de datos.
- **Control y habilidad:** el usuario debe sentir que tiene el control de la aplicación, que tiene la habilidad de realizar las diferentes transacciones, de acuerdo con la experiencia que tenga de su uso.
- **Privacidad:** la aplicación debe garantizar la privacidad de la información del usuario con respecto a otras fuentes de datos y usuarios que amenacen la integridad de esta.

Estándares de herramientas de desarrollo

Para la construcción de las aplicaciones web se deben seguir los estándares de desarrollo de este tipo de aplicación. Hay que considerar que debe existir una lógica de negocio en el servidor para que se pueda denominar aplicación web. A continuación se describen algunos estándares que se deben considerar:

- Estándares planteados por la W3C (*World Wide Web Consortium*). Algunos de los estándares que se recomiendan en el desarrollo de páginas web es la utilización de hojas de estilos (*CSS- Cascading Style Sheets*). Siguiendo las reglas de los estándares web, se debe separar la parte de codificación de la parte del contenido de la apariencia (interfaz) de las páginas.
- Se debe definir un estándar de codificación de las fuentes de la aplicación; por ejemplo: abreviatura de las variables, abreviatura de los atributos de las clases, estandarización de las interfaces, metodología de desarrollo y arquitectura de desarrollo, entre otros.

³ Instrucciones del lenguaje SQL para consultas y manipulación de datos en bases de datos.

Algunos beneficios que se obtienen de seguir los estándares son:

- El mismo contenido se puede presentar en distintos *browsers*.
- Las páginas son mucho más livianas, por tanto se descargan más rápido en el *browser*.
- Se separa el código fuente de la lógica de negocio con respecto al código de presentación.
- Se minimiza el costo de mantenimiento.

Los estándares brindan mejores prácticas en el momento del desarrollo, ya que facilitan el uso y explican los beneficios del tipo de desarrollo que se promueve.

Evolución de las aplicaciones web

En los primeros tiempos de la era de la información, cada aplicación tenía su propio programa cliente que, por tanto, tenía que ser instalado en cada uno de los computadores de los usuarios finales. Esto requería que si se realizaba una mejora en la aplicación, se tenía que reinstalar la aplicación en cada estación cliente. Esto conllevaba un costo de soporte, un lento proceso de actualización, complejidad de control de versiones instalados en las computadoras y disminuía la eficacia del personal.

Se vio entonces la necesidad de implementar nuevas tecnologías de desarrollo de aplicaciones, lo cual fue evolucionando las aplicaciones web mediante una serie de estándares. Para esto se utilizan lenguajes interpretados en el cliente como, por ejemplo, JavaScript⁴ y las reglas de negocios se ejecutan en el servidor, y se devuelven como resultado las páginas web que se visualizarán en el *browser* del cliente .

Al inicio del desarrollo de las aplicaciones web, se podían desarrollar con base en HTML (*HyperText Markup Language*)⁵ y XML (*Extensible Markup Language*- Lenguaje de marcado extensible)⁶. A medida que esta área fue evolucionando fueron surgiendo lenguajes de programación como, por ejemplo, ASP (*Active Server Pages*) y PHP (*Personal Home Page*). Debido a las nuevas demandas y necesidades de las empresas en las aplicaciones web, se han diseñado nuevos lenguajes de programación con su propia arquitectura y metodología de desarrollo. En la actualidad, uno de los lenguajes de programación que mayor utilización ha tenido en el desarrollo de aplicaciones y que ha venido a

⁴ Lenguaje de programación similar a C++ y Java, que se utiliza para crear funciones a nivel de páginas web para ejecución en el cliente.

⁵ Lenguaje que se utiliza en páginas web para diseñar y dar formato en la interfaz de usuario.

⁶ Lenguaje en donde se puede transferir datos de una forma estructurada.

innovar es Microsoft Visual Studio.NET, con en cual se pueden desarrollar aplicaciones ASP.NET (nueva versión del lenguaje ASP). El otro lenguaje que viene utilizándose con gran frecuencia es JAVA de SUN, para el desarrollo de aplicaciones web con base en JSP (*Java Server Pages*). Es importante mencionar que independientemente del lenguaje de programación que se utilice, en el desarrollo de aplicaciones web hay que aplicar un método formal de análisis y diseño.

La arquitectura estándar en el ambiente web está basada en **cliente-servidor** (véase la Figura 1.0.). A continuación se muestra una imagen que demuestra un cliente-servidor.

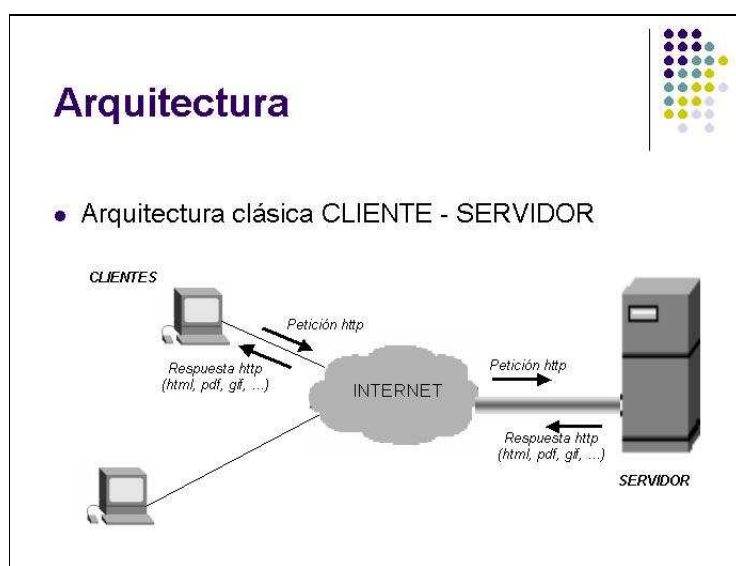


Figura 1.0. Arquitectura cliente-servidor en aplicaciones web
Fuente: Tomado de http://pericles.rediris.es/cur_web/cur_web_intro/pdf/Introducción_tecnologías_web.pdf

La arquitectura de un sitio web tiene tres componentes principales: un servidor web, una conexión de red y uno o más clientes (*browsers*). El servidor web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP.

En el desarrollo de aplicaciones web usualmente se utiliza una arquitectura de desarrollo basado en tres capas (véase la Figura 1.1.): capa de presentación, capa de reglas de negocio y capa de datos. La capa de presentación se basa en presentar y recoger los datos al usuario. La capa de regla de negocio de basa en toda aquella lógica

codificada para que la aplicación sea funcional y valide los distintos requerimientos de la aplicación. La capa de datos se basa en el conjunto de datos, procedimientos almacenados, vistas y tablas que se encuentran en una base de datos. Hay que comentar que esta tecnología pasa por metodologías de técnicas de programación y acceso a base de datos.

El siguiente es un esquema representativo de la arquitectura de tres capas:

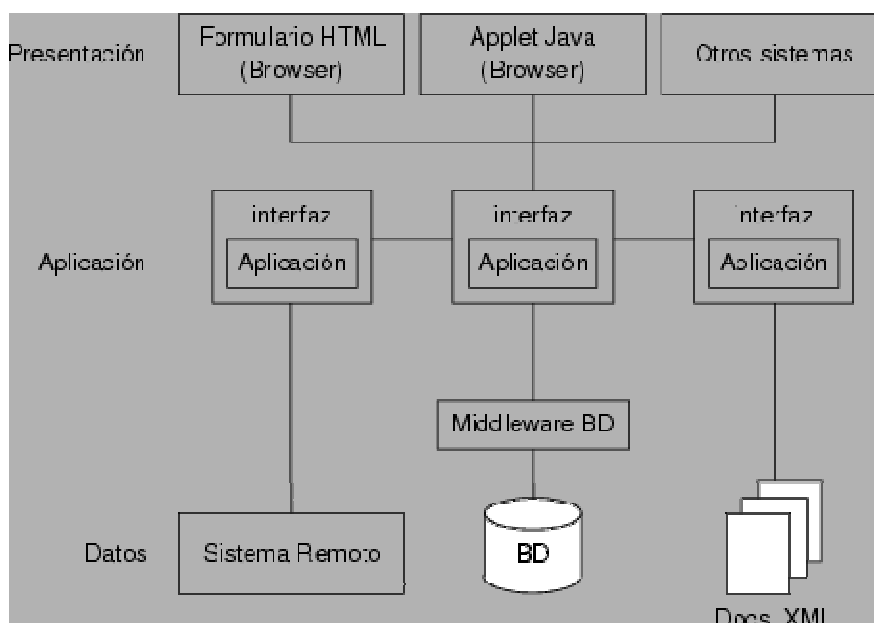


Figura 1.1. Esquema de las capas de una arquitectura diseñada para cliente-servidor.

Fuente: Tomado de <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>

Las empresas que han tenido la necesidad de implementar aplicaciones web han visto un impacto considerable en sus procesos de negocios. Por ejemplo, las grandes empresas se han dedicado a trasladar sus aplicaciones operativas en web, ya que han evaluado los costos-beneficios y han obtenido un crecimiento importante en la calidad de los servicios que brindan a sus clientes tanto internos como externos.

1- Herramientas Microsoft

a. Información general

Desde sus inicios, Microsoft ha venido a aportar una serie de herramientas, modelos e infraestructuras que permiten el desarrollo y ejecución de las aplicaciones en general. Debido a las nuevas tendencias de las tecnologías y necesidades empresariales que surgen diariamente, se requieren nuevos modelos que faciliten y agilicen el procesamiento de la información de una forma eficiente, esto ha obligado a crear nuevos modelos, herramientas y cambios de infraestructuras que puedan satisfacer las necesidades de las empresas, siguiendo los estándares de desarrollo. Con base en estos requerimientos, se han presentado una serie de innovaciones que han venido a mejorar y facilitar el procesamiento de la información y a construir aplicaciones web eficientes, con lo cual nos referimos a una nueva tecnología llamada Microsoft.NET (tecnología .NET).

La tecnología .NET es un programa que conecta información, sistemas, usuarios y dispositivos. Incluye clientes, servidores y herramientas para programadores, y está conformado por una serie de herramientas:

- o Microsoft.NET *Framework*.
- o Herramientas de programación (Visual Studio.NET) que se basan en la tecnología .NET, la cual brinda beneficios en la productividad de los programadores, con lo que se obtiene un gran beneficio de la plataforma *Framework.NET* en el desarrollo de aplicaciones web (ASP.NET⁷).
- o Una serie de servidores como, por ejemplo, Microsoft Windows Server 2003 y Microsoft SQL Server 2005, que integran, operan y administran servicios web XML y aplicaciones web.
- o Software clientes como, por ejemplo, Microsoft Windows XP, Windows CE o Microsoft XP (con la versión 2007 se le puede sacar más provecho a nivel de experiencia de usuario final en la parte de aplicaciones web y análisis de datos), que viene a causar un impacto positivo en el desarrollador y usuario final.
- o Con esta nueva tecnología se pueden construir aplicaciones web mucho más robustos, ágiles, seguras, y con una metodología orientada a objetos, lo cual mejora el rendimiento y minimiza el costo de desarrollo y de mantenimiento .

⁷ Nueva versión de ASP (*Active Server Pages*), para crear páginas dinámicas bajo la tecnología .NET.

b. Arquitectura e infraestructura (plataforma)

Windows.NET *Framework* es una plataforma que permite generar y ejecutar aplicaciones web, aplicaciones clientes inteligentes y servicios web XML. Esta plataforma tiene una serie de componentes que facilitan la integración de las aplicaciones, el compartimiento de datos, la funcionalidad en redes a través de protocolos y estándares independientemente de la plataforma en que se ejecuta, como por ejemplo XML, SOA y HTTP.

Windows.NET *Framework* está integrada por dos elementos principales:

i- *Common Language Runtime (CLR)*: este incluye un motor de ejecución, un recolector de basura (*garbage collector*), un compilador *just-in-time* (justo a tiempo), ejecución de *Thread* (hilos de procesamiento), un manejador de excepciones (control de errores) y un sistema de seguridad. Los distintos lenguajes de programación de .NET están soportados por esta herramienta. El CLR tiene un proceso importante, pues toma el código descrito en los diferentes lenguajes de programación (Visual Basic.NET, C#.NET, etc.) y lo traduce a un lenguaje intermedio (IL) para que sea ejecutado por el código nativo (*Native Code*).

Comentario [OCR1]: Buscar otra forma de expresar esto.

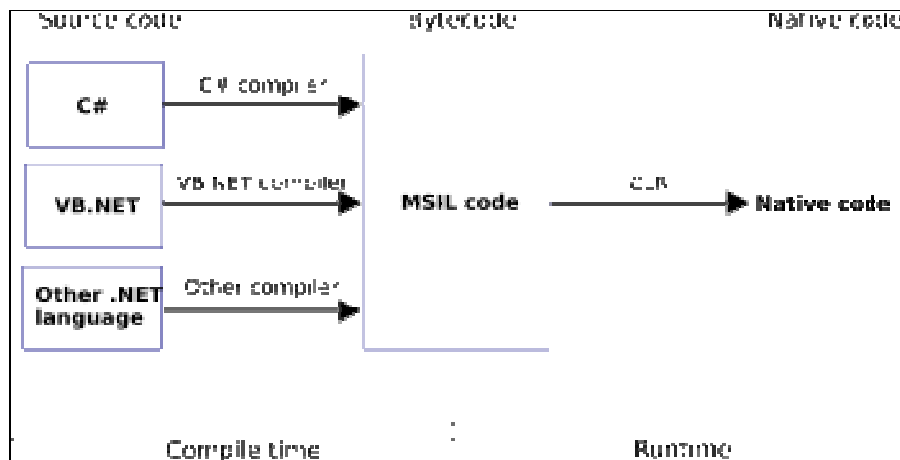


Figura 2.0. Representación del CLR y el IL.

Fuente: Tomado de http://en.wikipedia.org/wiki/Common_Language_Runtime

ii- Conjunto unificado de bibliotecas de clases, tales como ASP.NET para aplicaciones web y servicios web XML, Windows Forms para aplicaciones clientes inteligentes y ADO.NET para el acceso a datos sin rigidez.

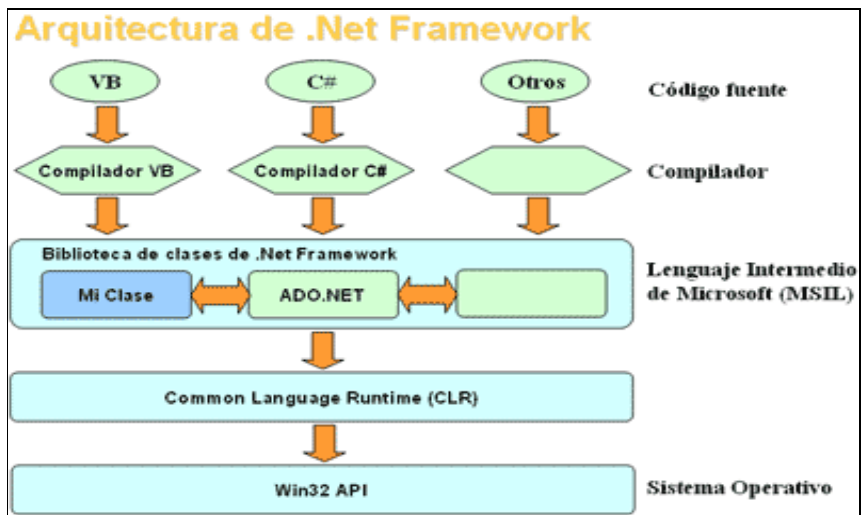


Figura 2.1. Arquitectura del .NET Framework
Fuente: Tomado de <http://www.desarrolloweb.com/articulos/1328.php>

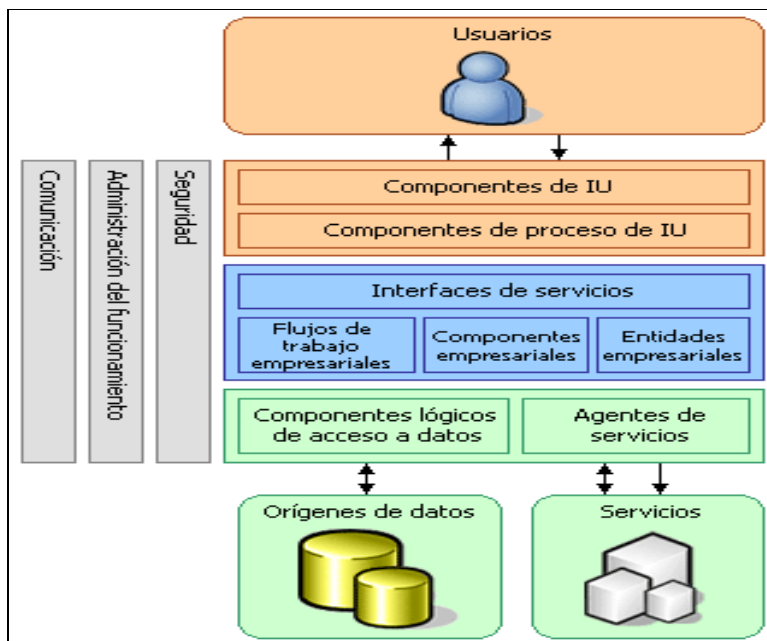


Figura 2.2. Capas de componentes de servicios y aplicaciones distribuidas creadas con .NET
Fuente: Tomado de <http://download.microsoft.com/download/e/d/1/ed1bd192-a665-49e1-b97d-e283318c4294/AppDesign.doc>

La nueva versión de herramienta de programación está cimentada en la plataforma .NET, se llama Visual Studio.NET y tiene una serie de

lenguajes de programación como, por ejemplo, Visual Basic.NET, C#.NET, J#.NET, etc.

Para desarrollar aplicaciones en Visual Studio.NET se utiliza la tecnología ASP.NET, ya que esta nueva versión se adapta al entorno .NET de Microsoft y ofrece mejoras en su estructura; por ejemplo, el ASP.NET ofrece una mejor estructuración del código fuente, ya que divide el código de presentación (HTML) y el código de lógica de negocio (VB, C#, etc), por lo que se le llama *Code Behind*. Con esta separación hay más seguridad y administración de las fuentes. Esto ayuda a que los diseñadores web no puedan ver y manipular la lógica de negocio.

Para desarrollar aplicaciones ASP.NET se pueden utilizar diversos lenguajes de programación contenidos en Visual Studio.NET, como por ejemplo el Visual Basic.NET o el C#.NET.

Unas de las facilidades en desarrollar con ASP.NET es que se trabaja bajo el modelo de "n" capas, así como la programación orientada a objetos (POO).

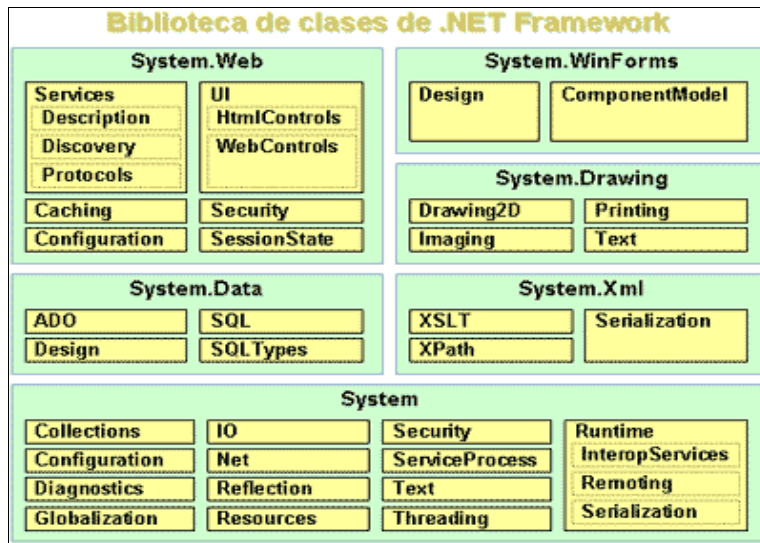


Figura 2.3. Bibliotecas y clases contenidas en el .NET Framework.

Fuente: Tomado de <http://www.desarrolloweb.com/articulos/1328.php>

c. Requerimientos

- Las herramientas de programación y las aplicaciones web que se basan en la plataforma .NET se pueden instalar y ejecutar en cualquier plataforma Microsoft, para lo cual hay una serie de software de instalación para que se puedan acoplar sin ningún problema.
- Para poder desarrollar aplicaciones web con ASP.NET se debe tener instalada una herramienta de servicio de Internet (IIS, *Internet Information Services*), cuya funcionalidad es mantener las páginas web y los sitios web en el servidor. Además, también proporciona un mecanismo para que los usuarios (visitantes) puedan acceder a la aplicación web, en otras palabras, es el encargado de mantener el sitio web disponible para el cliente.

d. Beneficios

- Uno de los beneficios de desarrollar con Visual Studio.NET es que independientemente del lenguaje en que se desarrollen las aplicaciones web, estas tienen integridad, ya que se ejecutan sobre el .NET Framework. Para esto no se requiere un cambio de infraestructura y con solo actualizar el .NET Framework, los mismos componentes de desarrollo sirven para los distintos lenguajes.
- El .NET Framework tiene una serie de librerías y componentes (véase la Figura 2.3.) que ayudan y logran que el programador realice una codificación mucho más sencilla, ya que estos componentes tienen una serie de métodos y de rutinas que facilitan la seguridad, integridad y procesamiento de los datos, así como su representación. Todo esto permite que la codificación de las aplicaciones sean más limpias y contengan menos líneas de código, por tanto hace que mejore el rendimiento y sean más rápidas al ejecutarse.
- Con ASP.NET se pueden construir aplicaciones web dinámicas, con técnicas completamente nuevas, capaces de satisfacer las necesidades del negocio ampliando las posibilidades que se ofrecen en un sitio web.
- El costo de las aplicaciones web construidas en ASP.NET es menor, ya que ofrece componentes y una serie de librerías que facilitan el mantenimiento y soporte de estas.
- Ofrece un control administrado de excepciones para la captura de errores.
- ADO.NET es un nuevo modelo de acceso a datos con mejoras significativas con respecto al ADO (*Active Data Object*). Ofrece un

nuevo objeto llamado *DataSet* (sustituye al objeto *RecordSet*) para realizar operaciones sin la necesidad de mantener una conexión permanente con el servidor de la base de datos, y también utiliza XML para proporcionar una estructura de los datos.

- El .Net Framework ofrece una serie de librerías para la seguridad de la aplicación y de los datos; estas librerías tienen subrutinas de encriptación de datos y utilizan protocolos. La seguridad de las páginas web se puede integrar con la autenticación de Windows, la cual está basada en formularios, esta es la autenticación Microsoft Passport.

e. Seguridad

Método	Ventajas	Inconvenientes
Autenticación basada en Windows	-Utiliza infraestructura Windows existente (<i>Active Directory</i>) -Controla el acceso a información confidencial. -Es un buen recurso para aplicaciones dentro de la intranet.	-No es conveniente para la mayoría de las aplicaciones de Internet.
Autenticación basada en formularios	-Apropiada para aplicaciones de Internet. -Soporta todo tipo de clientes.	-Está basada en <i>cookies</i> .
Autenticación por Microsoft Passport.	-Inicio de sesión único para muchos sitios de Internet. -No es necesario mantener una base de datos para almacenar información del usuario. -Permite a los desarrolladores personalizar el aspecto de la página de inscripción.	-Está basada en <i>cookies</i> . -Implica una cuota.

Figura 2.4. Cuadro comparativo de la seguridad en el .NET Framework
Fuente: Elaboración propia.

f. Implementación

Cuando las aplicaciones web basadas en ASP.NET se compilan, generan un archivo ensamblado que tiene cifrada toda la lógica de negocios. Hay que tener en cuenta que las aplicaciones web se deben instalar en un servidor web, al cual los clientes acceden por medio del IIS (instalado en el servidor). Una de las facilidades en estos tipos de aplicaciones es que solo se debe instalar la aplicación en el servidor, los clientes acceden por medio del *browser*.

Otra de las ventajas en el proceso de implementación es que se reduce el soporte técnico, en comparación con otros que se tienen que instalar en cada cliente. Además, se debe considerar que los componentes y las librerías que utilizan las aplicaciones se encuentran dentro del .NET Framework, lo cual reduce el sinnúmero de versiones y la dependencia que hay de estos.

- Lenguaje Visual Basic.NET

El Visual Basic ha venido a ser el lenguaje de programación de Microsoft más utilizado en los últimos años. Este permite muchas facilidades de desarrollo y de productividad. Visual Basic es un lenguaje de fácil utilización por sus expresiones, instrucciones, sintaxis y ayuda en línea en el momento de escribir el código (*Intelliscenses*).

Con la nueva tecnología Microsoft.NET, los programadores se han visto beneficiados ya que cuenta con un significativo grupo de librerías, componentes, modelo de desarrollo, manejadores de eventos, etc., lo cual ayuda a poder construir aplicaciones más eficientes, ágiles y mucho más rápidas de programar; aún más, cuenta con una característica relevante que es trabajar bajo el modelo de programación orientada a objetos (POO). En la actualidad, el Visual Basic le da muchas facilidades al programador como, por ejemplo, una gestión estructurada de errores, soporte para la ejecución de varios subprocesos (multihilos) posibilidad de crear servicios web rápidamente y de forma fácil, así como también un recolector de basura, por lo que hace que se mejore el rendimiento de desarrollo y ejecución de las aplicaciones web.

El éxito que ha tenido esta herramienta se debe a su facilidad de uso, y a la sencillez de crear interfaces de usuarios. Con la tecnología .NET, las aplicaciones desarrolladas en Visual Basic o en C# tendrán un mismo rendimiento con las mismas funcionalidades de .NET.

- Lenguaje C#.NET

C# es un lenguaje que se deriva de los lenguajes C y C++, por tanto viene a ser el primer lenguaje orientado a componentes en esta estructura. Este nuevo lenguaje de programación es simple, moderno, orientado a objetos y con un sistema de tipos seguro. Este lenguaje combina la alta productividad del Visual Basic y la eficacia de C++.

Este lenguaje innovador ha tomado gran parte de las características de C y C++, como expresiones, instrucciones y, sobre todo, la sintaxis, pero mejorando ciertos problemas que se presentan en C y C++. Este es un lenguaje de desarrollo que día a día va tomando más fuerza, por las características propias y heredadas de C y C++. Por ser un lenguaje sencillo, los programadores de Java, C y C++ en se están adaptando a él con facilidad.

Unos de los aspectos técnicos de este lenguaje que se deben mencionar son:

- Las variables deben inicializarse antes de ser usadas, lo cual elimina los problemas por variables sin inicialización.
- Las instrucciones como *if* o *while* requieren valores booleanos.
- No se aceptan casos sin cerrar instrucciones.
- El sistema de tipos emplea gestión de memoria automática (en ciertos contextos de seguridad).

C# es un lenguaje que obliga al programador a codificar de una forma más ordenada, limpia y segura, lo que disminuye la posibilidad de error de sintaxis e instrucciones y hace la lógica de desarrollo de la aplicación más rápida y fácil de darle mantenimiento.

Un aspecto que se debe considerar es que los programadores que no han tenido experiencia con C o C++ deben invertir un poco más de esfuerzo para desarrollar en este lenguaje.

2- Lenguaje Java (Sun Microsystems)

a- Información general

Java es un lenguaje de programación desarrollado por Sun Microsystems, que ha venido revolucionar el desarrollo de aplicaciones web. El Internet ha sido una de las catapultas importantes que han

hecho que este lenguaje tenga un profundo impacto en el desarrollo de aplicaciones web, lo cual extiende el universo de los objetos que se mueven en el mundo de la red de Internet.

Java es un descendiente de C y C++, por lo que ha heredado muchas de las características de estos lenguajes, aunque se han mejorado ciertas carencias de sus antecesores.

A su paso por el mercado, esta herramienta ha tenido tres ediciones:

- JSE (*Java Standard Edition*): contiene herramientas básicas para el desarrollo de aplicaciones web.
- JEE (*Java Enterprise Edition*): para el desarrollo de aplicaciones distribuidas en compartimiento de información en distintas ubicaciones físicas. También contiene librerías que satisfacen las necesidades de lógica de negocio de nuevas necesidades empresariales.
- JME (*Java Mobile Edition*): para el desarrollo de aplicaciones que se utilizan en dispositivos móviles.

Java es un lenguaje de programación de propósito general similar a otro lenguaje, con características idóneas para programación orientada a objetos (POO). Fue diseñado desde su origen con esta característica dándole un enfoque de programación sencillo, organizado, eficiente y ágil, con la idea de ir rompiendo con las técnicas de programación de forma procedimental.

Este lenguaje tiene una característica que no posee otro lenguaje, y es la posibilidad de crear aplicaciones (*applets*⁸) diseñadas para ser transmitidas y descargadas dinámicamente por medio de la red (Internet) y ejecutada por un navegador web. El Java es un lenguaje eficiente para resolver problemas de seguridad y "portabilidad" inherente a los sistemas que interactúan en red. En la tecnología Java, para la creación de aplicaciones web para ejecución en el servidor, se utiliza el JSP (Java Pages Server, que traducido sería 'Páginas de Servidor Java'), una tecnología para crear páginas web orientadas en Java. Con esta tecnología es posible crear aplicaciones web que pueden ejecutarse en múltiples plataformas, lo cual es una de las características principales de Java: ser multiplataforma.

El motor de las páginas JSP está basado en los *servlets*⁹ de Java que lleva a cabo una fase de traducción de esa página en un *servlet*, implementado en un archivo class (*Byte codes* de Java). El JSP sigue la

⁸ Módulo descargable a través de una página web y ejecutable en la JVM (Java Virtual Machine) del navegador.

⁹ Programas en Java destinados a ejecutarse en el servidor.

filosofía de Java, con respecto a que se compila una vez y se ejecuta en cualquier plataforma sin la necesidad de ser recompilado.

A continuación podemos visualizar un ejemplo de una página JSP:

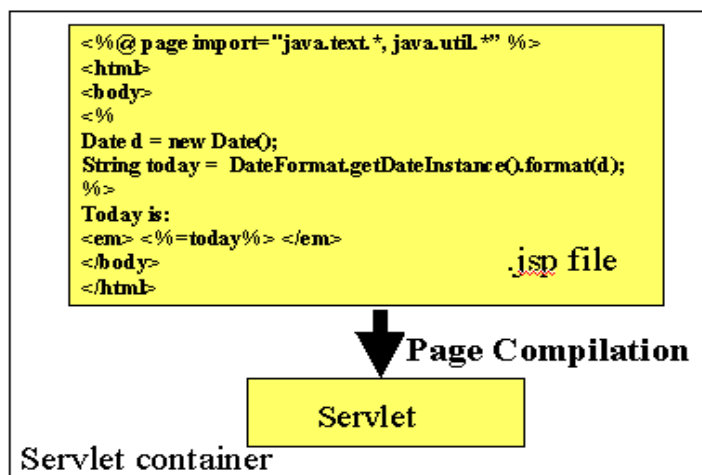


Figura 2.5. Compilación de página JSP a un Servlet.

Fuente: Tomado de <http://www.desarrolloweb.com/articulos/831.php>

b- Arquitectura e infraestructura (plataforma)

Unas de las características más sobresalientes en Java es que posee una arquitectura neutral. Esto quiere decir que se ejecuta en cualquier plataforma y hardware, lo cual les facilita a los desarrolladores construir aplicaciones web que se presenten a los usuarios (clientes) de una forma consistente y que permita su independencia de ejecución. Para que una aplicación sea independiente del medio, Java corre sobre una plataforma llamada Máquina Virtual Java (JVM, en *inglés Java Virtual Machine*). Este es un software que interpreta las instrucciones del código fuente de la aplicación y hace que corra sobre cualquier computador. La Máquina Virtual Java (véase la Figura 2.6.) representativamente es una capa lógica que está sobre el sistema operativo y por debajo de la herramienta Java, que interpreta las instrucciones y se encarga de tener la comunicación con el sistema operativo; esto hace que la aplicación no sea dependiente del sistema operativo en que se desarrolló y pueda así correr en cualquier software, eso sí, teniendo el JVM como plataforma de interpretación y comunicación con el hardware y sistema operativo (véase la Figura 2.7.).

A continuación se muestra un ejemplo que representa la posición del Java Virtual Machine:

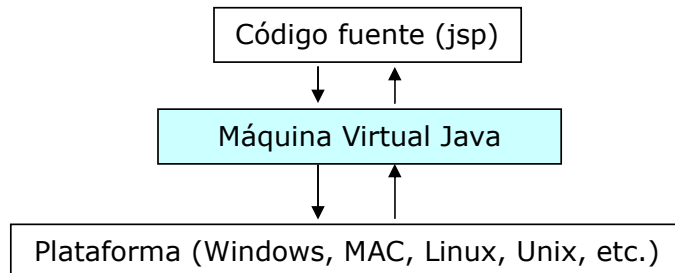


Figura 2.6. Representación lógica de la posición de la capa VJM
Fuente: Joyanes Aguilar, Luis. (2001), *Java 2, Manual de programación*.

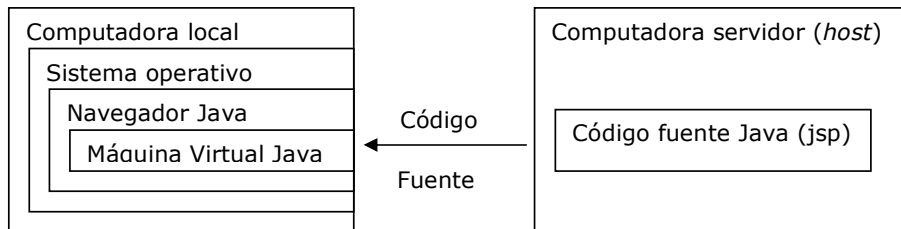


Figura 2.7. Transferencia de código fuente Java en Internet
Fuente: Joyanes Aguilar, Luis. (2001). *Java 2, Manual de programación*.

c- Requerimientos

- La herramienta Java puede ser instalada en cualquier plataforma, ya que su funcionalidad no depende del hardware o sistema operativo instalado. Lo único primordial para que funcione tanto para la parte de desarrollo de aplicaciones y utilización es instalar la Máquina Virtual Java (JVM).
- Para poder comenzar a programar aplicaciones web en Java, es suficiente con instalar el paquete JDK de Sun, que es de libre distribución.
- Para poder aprender JSP, aparte del conocimiento de HTML, es necesario comprender y tener algo de experiencia con la programación en Java, para así poder entrar a lo que es el sistema de *servlets*. También es necesario instalar la herramienta *Tomcat*, que es el contenedor de *servlets* usado en la referencia primordial en la implementación de JSP.

d- Beneficios

- El compilador Java traduce el código en instrucciones que son interpretadas (*Byte codes*) por la Máquina Virtual JAVA (JVM, Java Virtual Machine).
- Java tiene la capacidad de ejecutarse en cualquier máquina cliente, pues es multiplataforma.
- Una de las características de Java es que soporta de una forma muy eficiente el XML.
- Java es un lenguaje meramente orientado a objetos, lo cual hace que tenga como mínimo una clase.
- Java posee una colección de clases que ayudan a construir aplicaciones distribuidas. La utilización de URL (*Uniform Resource Locator*) hace que pueda accederse fácilmente desde un servidor remoto.
- Java es interpretado y necesita un intérprete, para lo cual el JVM genera un lenguaje intermedio, haciendo que sea independiente de la máquina y pueda ejecutarse en otras.
- Java es una herramienta fiable (robusta), que hace una verificación de errores, y que como consecuencia se caracteriza por ser un lenguaje tipificado. Además, posee un recolector de basura en tiempo de ejecución que asegura el rendimiento de la aplicación.
- Java es un lenguaje seguro, ya que se adapta para aplicaciones distribuidas y en red, con diversos mecanismos de seguridad que se basan en la premisa de que nada debe ser *trusted* (de confianza)¹⁰.
- Esta herramienta es de alto rendimiento, ya que le facilita al programador desarrollar diversos procesos concurrentes con la utilización de multihilos (*THREADS*).

e- Implementación

- Las aplicaciones web deben ser instaladas en un servidor web; si se hace en servidores Windows, es recomendable instalar el IIS (*Internet Information Services*).
- Para ejecutar un programa Java, es preciso que se cuente con una implementación de la máquina virtual específica donde se desea ejecutar, la cual efectúa la transformación de la aplicación compilada en un programa comprensible para la máquina.

¹⁰ Nada del entorno debe ser de confianza; Java implanta su propia seguridad.

IV. Análisis y discusión de resultados

El auge que se ha dado en el desarrollo de aplicaciones web ha traído beneficios considerables al servicio y ha permitido consolidar la visión del negocio. Además, ha ayudado a desarrollar nuevos mecanismos, metodologías, técnicas, modelos y arquitecturas en el proceso de la construcción de las aplicaciones web con una gran seguridad. En la actualidad, las empresas se han visto en la necesidad de construir e implementar sus aplicaciones en el modelo web, ya que permite más uso, flexibilidad y acceso. Hay que considerar que las compañías no siempre disponen de los suficientes recursos internos como para afrontar nuevos proyectos, por lo que contar con una empresa externa experta en esta tecnología, que supervise y certifique el trabajo de terceros proveedores durante la ejecución de nuevos proyectos, incrementa considerablemente la posibilidad de éxito y garantiza de algún modo que se obtengan los resultados deseados y convenidos, en presupuesto, tiempo y forma.

En cuanto a las herramientas de programación, la tecnología JSP de Java y ASP.NET de Microsoft sirven para hacer, en cierta medida, el mismo tipo de aplicaciones web, pero en el fondo tienen bastantes diferencias. De acuerdo con las experiencias que se han obtenido en madurar los lenguajes de programación, unas de las ventajas de JSP es que sigue la filosofía de la arquitectura JAVA de "escribe una vez, ejecuta donde quieras". Así, JSP se puede ejecutar en los sistemas operativos y servidores web más populares como, por ejemplo, Apache, Netscape o Microsoft IIS. Otro punto que se debe mencionar es que una ventaja de la programación en ASP.NET es que ésta es más fácil de aprender que JSP. Para aplicaciones web simples y dinámicas, ASP.NET es más rápido y eficaz que Java. Finalmente, la arquitectura modular de ASP.NET aporta todos los recursos de desarrollo productivo.

Debido a la serie de características que presenta cada herramienta para el desarrollo de aplicaciones web, no podemos decir que un lenguaje de programación sea mejor que el otro, ya que cada uno tiene sus propios beneficios, de acuerdo con su entorno como, por ejemplo, el modelo de negocio de la empresa, experiencia del programador, arquitectura implementada, estimación de costos-tiempo de desarrollo, implementación y mantenimiento. Por tanto, la selección de la herramienta para el desarrollo de aplicaciones web no obedece a una simple decisión, sino a una serie de factores importantes que crean el entorno.

A continuación se presentan unos gráficos que muestran una estadística de la posición y el grado de utilización de lenguajes y herramientas de programación que se usan en la construcción de aplicaciones web en nuestro país, tomado como referencia el año 2005.

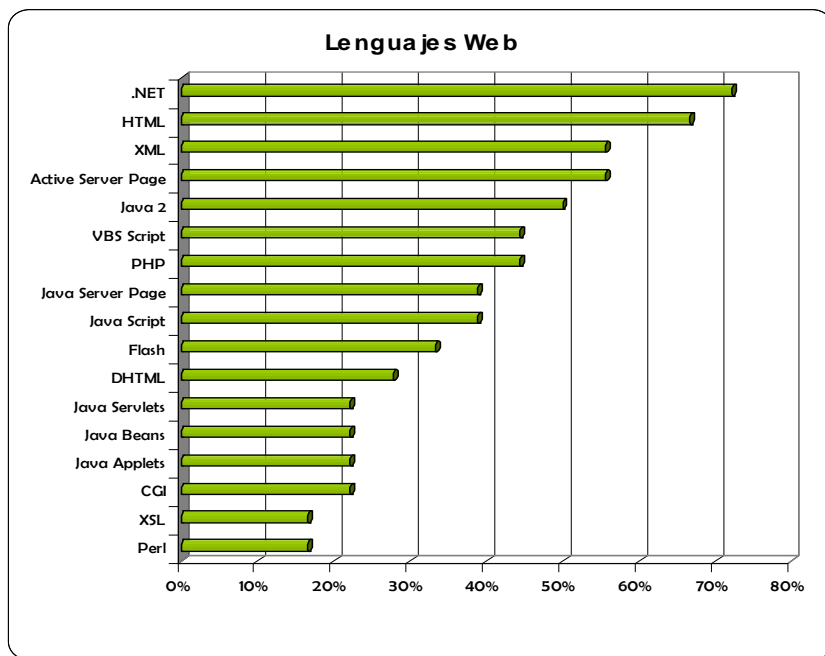


Figura 3.0. Lenguajes web utilizados

Fuente: Tomado de

http://www.camtic.org/ES/phtml/descargarDoc.phtml?arc=Estado_Nacional_de_Software_2005_Final.doc%7C65.pb.arc.333.doc

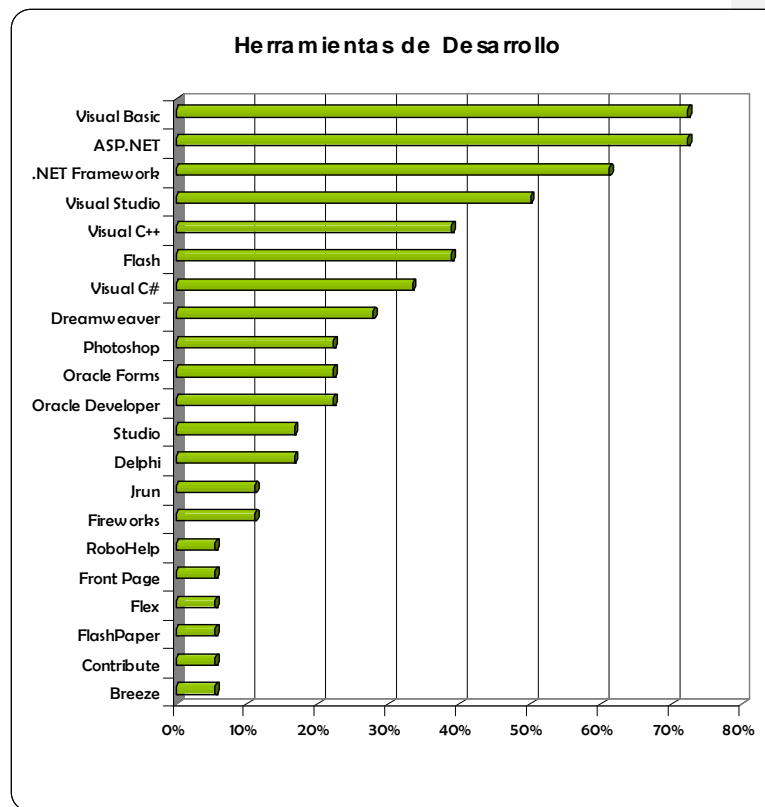


Figura 3.1. Herramientas de desarrollo utilizados.

Fuente: Tomado de

http://www.camtic.org/ES/phtml/descargarDoc.phtml?arc=Estado_Nacional_de_Software_2005_Final.doc%7C65.pb.arc.333.doc

V. Recomendaciones

El éxito de una aplicación web depende de múltiples factores, pero no hay dudas de que la facilidad de uso, tanto con respecto a su manejo como en el proceso de utilización, otorga un valor agregado considerable. Basta con chequear un listado de aplicaciones web y se verá que los servicios en línea que han triunfado son muy fáciles de usar, otorgan una experiencia agradable y proveen un servicio sencillo pero práctico.

Un buen análisis de procesos, un conocimiento profundo de las infraestructuras y sistemas, y un equipo potente de desarrollo son tres puntos fundamentales para determinar la concepción, desarrollo y puesta en producción de nuevas herramientas personalizadas, flexibles y fiables que permitan incrementar el número y calidad de los servicios ofertados tanto a los clientes externos como internos de las compañías. En proyectos de tecnología web, es conviene conocer tanto el funcionamiento y mecánica de la tecnología en sí, como de las empresas que se encargan de implementarla.

Otro punto que se debe tener en cuenta es que hay que construir un plan estratégico de capacitación a los usuarios y personal técnico para el desarrollo de las aplicaciones, ya que en estas cambia el concepto de 'usuabilidad', de diseño, arquitectura y tecnología con respecto a las aplicaciones tipo escritorios, lo cual crea un cambio de mentalidad de uso, de servicio y modelo de construcción de la aplicación.

Las empresas tienen que tener en cuenta los resultados que quieren obtener al implementar aplicaciones web, ya que una mala construcción e implementación de estos tipos de aplicaciones puede crear un impacto positivo o negativo en el negocio.

VI. Conclusiones

En el desarrollo de aplicaciones web tiene que haber una serie de cuidados, para que su construcción y su implementación sean eficientes y funcionales. Además, debe haber aceptación por parte del usuario.

En la actualidad existe una serie de herramientas y lenguajes de programación que hacen que las aplicaciones web sean más dinámicas, eficientes y seguras, para esto son diseñadas bajo un modelo cliente-servidor funcional con una serie de técnicas, metodologías y tecnologías que permiten su óptima utilización y construcción. Además, cuentan con características como la agilidad y una ordenada codificación de las aplicaciones, tanto en navegación de la interfaz, lógica de negocios y acceso a datos con mecanismos seguros que permiten mantener la integridad de la información.

Uno de los puntos más importantes que se deben considerar es conocer y comparar las herramientas de desarrollo con todo el entorno que las rodea como por ejemplo: metodologías, técnicas, arquitectura, costos (curva de aprendizaje, implementación, mantenimiento), modelos en el desarrollo de aplicaciones, tomando en cuenta la necesidad y la estructura del negocio. No hay que olvidar que para poder seleccionar el tipo de herramienta con la que se van a desarrollar aplicaciones web, hay que hacer un análisis profundo de las distintas herramientas y lenguajes de programación así como de su respectiva tecnología y la que se tiene en uso en la empresa, para aliarse con la visión de la empresa.

Por último, sería beneficioso construir una herramienta y lenguaje de desarrollo tomando como base cada ventaja y eficiencia que tienen las distintas herramientas y así poder obtener un producto único para satisfacer las grandes necesidades del mercado, pero para llegar a eso hay que empezar a unificar las arquitecturas, modelos, técnicas y metodologías de cada herramienta.

VII. Referencias bibliográficas

- Archer, Tom. (2001). *A FONDO C#*. Madrid: McGraw Hill.
- Jamsa, Kris. (2003). *Superutilidades para Visual Basic .NET*. Madrid: McGraw Hill.
- Buczek, Grez. (2003), *Superutilidades para ASP.NET*. Madrid: McGraw Hill.
- Joyanes Aguilar, Luis. (2001). *Java 2 Manual de programación*. Madrid: McGraw Hill.
- NEWPATENT y DesarrolloWeb. (2007). *Desarrollo Web, Comparando JSP con ASP*. Recuperado el 22 de julio de 2007, de www.desarrolloweb.com/articulos/832.php.
- NEWPATENT y DesarrolloWeb. (2007). *Desarrollo Web. ¿Qué es JSP?* Recuperado 22 de julio de 2007, de www.desarrolloweb.com/articulos/831.php.
- NEWPATENT y DesarrolloWeb. (2007). *Desarrollo Web, .Net Framework*. Recuperado 21 de julio de 2007, de www.desarrolloweb.com/articulos/1328.php.
- Microsoft MSDN. (2002). *Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios*. Última actualización del diciembre de 2002. Recuperado el 16, de julio 2007, de <http://download.microsoft.com/download/e/d/1/ed1bd192-a665-49e1-b97d-e283318c4294/AppDesign.doc>.
- Esposito de Wintellect, Dino. (2007). *¿Cómo aprovechar las ventajas de las características integradas de ASP.NET para rechazar los ataques a través de Internet?* (artículos técnicos sobre ASP.NET). Recuperado el 17 de julio de 2007, de www.microsoft.com/spanish/msdn/articulos/archivo/040405/voices/securitybarriers.aspx.
- NEWPATENT y DesarrolloWeb. (2007). *Desarrollo Web. 10 Beneficios de Codificar Aplicaciones Web Apegado a Estándares*. Recuperado el 16 de julio de 2007, de www.desarrolloweb.com/articulos/2503.php.

Jiménez, Luis Marco. (2003). *Cursos sobre desarrollo de aplicaciones en arquitectura Cliente/Servidor usando tecnologías web*. Recuperado el 17 de julio de 2007, de http://pericles.rediris.es/cur_web/cur_web_intro/cur_web.htm.

Jiménez, Luis Marco. (2003). *Introducción a las tecnologías web*. Recuperado el 17 de julio de 2007, de http://pericles.rediris.es/cur_web/cur_web_intro/pdf/Introducción_tecnologías_web.pdf.

Cámara Costarricense de Tecnología de Información y Comunicación (CAMTIC). (2005). *Estado Nacional del Software*. Recuperado el 3 de agosto de 2007, de www.camtic.org/ES/phtml/descargarDoc.phtml?arc=Estado_Nacional_de_Software_2005_Final.doc%7C65.pb.arc.333.doc.