

R825e



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL ECUADOR
SEDE AMBATO**
SERÉIS MIS TESTIGOS

**DEPARTAMENTO DE INVESTIGACIÓN,
POSGRADOS Y AUTOEVALUACIÓN.**

TEMA:

**“ESTUDIO COMPARATIVO PRÁCTICO DE LAS PLATAFORMAS DE
DESARROLLO J2EE Y .NET EN APLICACIONES WEB”.**

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
MAGISTER EN GERENCIA INFORMÁTICA, MENCIÓN DESARROLLO
DE SOFTWARE Y REDES.**

Autor:

ING. JORGE BLADIMIR RUBIO PEÑAHERRERA

Asesor:

ING. MSc. JANIO JADÁN.

Ambato – Ecuador

Marzo 2010



Nº de ingreso:	005294
Precio:	/\$80.00
canje:	Donación: <input checked="" type="checkbox"/> Compra: <input type="checkbox"/>
Fecha de factura:	
Fecha de ingreso:	14042010

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

SEDE AMBATO

DEPARTAMENTO DE INVESTIGACIÓN POSGRADOS Y

AUTOEVALUACIÓN

HOJA DE APROBACIÓN

Tema :

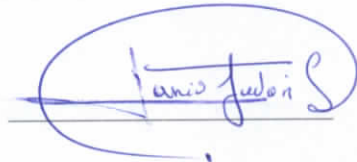
“ESTUDIO COMPARATIVO PRÁCTICO DE LAS PLATAFORMAS DE DESARROLLO
J2EE Y .NET EN APLICACIONES WEB”

Autor:

ING. JORGE BLADIMIR RUBIO PEÑAHERRERA

Janio Jadán, Ing. MSc.

f.



DIRECTOR DE TESIS

Galo López Sevilla / Ing. MSc.


f.



JURADO CALIFICADOR

Andrés López / Ing. MSc.

f.



JURADO CALIFICADOR

Telmo Viteri, Ing. MSc.

f.



DIRECTOR UNIDAD ACADÉMICA

Pablo Poveda Mora, Ab.

f.



SECRETARIO GENERAL PUCESA



DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Ing. Jorge Bladimir Rubio Peñaherrera, portador de la cédula de ciudadanía No. 050222229-2, declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del Título de Magister en Gerencia Informática, mención Desarrollo de Software y Redes, son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad.



Ing. Jorge Bladimir Rubio Peñaherrera
CI. 050222229-2

Agradecimiento

Mi agradecimiento especial a la Pontificia Universidad Católica del Ecuador Sede Ambato y a cada uno de los docentes de la misma, ya que gracias a la oportunidad que me brindaron hoy culmino una etapa más dentro de mi vida profesional.

Al Sr. Director de la Tesis, Ing. MSc. Janio Jadán, a quien debo muchas horas de intensas lecturas y sabias sugerencias para lograr la claridad y nitidez del texto de este trabajo de graduación.

A los profesores, compañeros y amigos de la Carrera de Posgrados, que de una u otra forma colaboraron en el desarrollo del presente trabajo de investigación.

A mi familia, a que con su apoyo, comprensión y solidaridad fueron siempre una fuerza incondicional para alcanzar el éxito propuesto.

Dedicatoria

Con sentimientos de gratitud, el presente trabajo lo dedico a los autores de mi vida, mis Padres, a mis Hermanas, mi Esposa e Hijo, quienes se han convertido en los pilares fundamentales para culminar mis estudios de Posgrado y obtener el Título de Magister en Gerencia Informática mención Desarrollo de Software y Redes en la Pontificia Universidad Católica del Ecuador Sede Ambato.

Jorge Blasimir Rubio Peñaherrera

RESUMEN

El presente trabajo investigativo permite informar al lector qué aspectos y conocimientos se necesitan para seleccionar la tecnología que se debe utilizar, para implementar de mejor manera el desarrollo de Aplicaciones Web, obteniendo los mejores resultados con la optimización de los recursos necesarios para desarrollar dicha tarea.

Para esto, se contempló la investigación y estudio en profundidad de las dos plataformas de desarrollo más utilizadas en el mercado, J2EE y Microsoft.NET.

Con esta información, adquirida por medio de la búsqueda y estudio a través de distintos manuales y sitios dedicados al tema, el usuario podrá familiarizarse con estas tecnologías. Así, se distinguirán las necesidades, requerimientos y aspectos importantes que se deben conocer para dedicarse por completo a una de estas plataformas de desarrollo en la realización de este tipo de aplicaciones.

Posteriormente se crearon áreas de evaluación, dentro de las cuales se definieron criterios de comparación adquiridos en el proceso de este trabajo. Esto se

plasma en tablas comparativas, las cuales permitieron dar un componente de certeza sobre esta investigación.

Esto, junto a la realización de pruebas definidas, ayudó a obtener conclusiones sobre el comportamiento y funcionalidad de Aplicaciones Web en estas tecnologías.

Luego se desarrollaron prototipos de aplicaciones Web, que van desde simples aplicaciones a sistemas con conexiones a distintos motores de bases de datos que permitieron reflejar de manera práctica aspectos que permitan resolver la elección de que tecnología ocupar.

ABSTRACT

This research work informs the reader what aspects and knowledge required to select the technology to be used to better implement the Web application development, the best results with the optimization of the resources necessary to perform that task.

For this, we looked at the research and study in depth the two most used development platform in the market, J2EE and Microsoft.NET.

With this information, gained from the search and study through various manuals and sites devoted to the topic, the user may be familiar with these technologies. Thus, distinguish the needs, requirements and aspects that are important to know in order to devote himself entirely to one of these development platforms in the conduct of such applications.

Subsequently established assessment areas within which were defined benchmarks gained in the process of this work. This is reflected in comparative tables, which permitted a component of certainty about the research.

This, along with the testing set, helped draw conclusions about the behavior and functionality of Web applications in these technologies.

Then he developed prototypes of Web applications, ranging from simple applications to systems with connections to different database engines that enabled reflect practical aspects to resolve the choice of technology occupy.

TABLA DE CONTENIDOS



Carátula	i
Hoja de Aprobación	ii
Declaración de Autenticidad y responsabilidad .	iii
Agradecimiento	iv
Dedicatoria	v
Resumen	vi
Abstract	viii
Índice General	x
Índice de Gráficos	xix
Índice de Tablas	xxi

CAPÍTULO I

1. ANTECEDENTES E INTRODUCCIÓN

1.1. Introducción	1
1.2. Definición del Problema	3
1.3. Delimitación del Problema	4
1.3.1. Delimitación de Campo	5
1.3.2. Delimitación Temporal	5
1.4. Marco Teórico	5
1.4.1. La Plataforma Java	5

1.4.2. La Plataforma .Net	7
1.5. Formulación de la Hipótesis	8
1.6. Variables e Indicadores	8
1.6.1. Variable Dependiente	9
1.6.2. Variable Independiente	9
1.7. Objetivos del Proyecto	9
1.7.1. Objetivo General	9
1.7.2. Objetivo Especifico	9
1.8. Metodología	11
1.8.1. Fuentes de Información	11
1.8.2. Instrumentos para Obtener la Información ..	11
1.9 Métodos de Investigación	12

CAPÍTULO II

2. PLATAFORMAS DE DESARROLLO

2.1. La Plataforma JAVA	13
2.1.1. La Máquina Virtual de JAVA	14
2.1.2. Herramientas de Desarrollo	15
2.1.3. Tipos de Herramientas de Aplicación ...	16
2.1.4. Arquitecturas JAVA	18
2.1.5. Componentes de Arquitectura J2EE	22
2.1.6. Principales API's de Arquitectura J2EE ..	24
2.1.7. Características de J2EE	27

2.1.8. Tipos de Aplicaciones J2EE	29
2.1.9. Acceso a Datos en J2EE	33
2.2. Introducción a la plataforma	
Microsoft .Net	34
2.2.1. Desarrollo de Aplicaciones en .NET ..	36
2.2.2. Programación WEB con Microsoft.Net ..	38
2.2.3. Introducción a .NET Framework	40
2.2.4. Descripción de la Arquitectura .NET	
Framework	41
2.2.5. CLR, Common Language Runtime	43
2.2.6. Implementación WEB con la	
Arquitectura Framework.NET	45
2.2.6. Acceso a datos en Microsoft .NET	46

CAPÍTULO III

3. INTRODUCCION A LA COMPARACION DE LAS PLATAFORMAS J2EE Y .NET

3.1. Introducción	49
3.2. Análisis de las Compañías	50
3.3. Limitación de la Comparación Web	53
3.4. Páginas Dinámicas en J2EE	55
3.4.1. Requisitos de JSP	56
3.5. Páginas Dinámicas en .NET	58

3.5.1. Web Forms	59
3.6 Programas de Desarrollo para las Plataformas J2EE y .NET	61
3.6.1. Principales IDE's de Desarrollo en J2EE..	63
3.6.2.1. Netbeans	64
3.7. Desarrollo en Microsoft.NET con Visual Studio	66
3.8. Elección de IDE's de Desarrollo para las Plataformas J2EE y .NET	68
3.8.1. Elección de IDE de Desarrollo para J2EE..	69
3.8.2. Elección de IDE de Desarrollo para .NET..	70
3.8.3. Principales Características de IDE's Seleccionados	71
3.9. Áreas de Evaluación de las Plataformas J2EE y .NET.	72
3.9.1. Definición de Áreas de Evaluación	73
3.9.1.1. Definición de Áreas de Evaluación 1 Aprendizaje	73
3.9.1.2. Definición de Áreas de Evaluación 2 Rendimiento	74
3.9.1.3. Definición de Áreas de Evaluación 3 Explotación y Uso	76
3.9.1.4. Definición de Áreas de Evaluación 4 Desarrollo	77
3.9.1.5. Definición de Áreas de Evaluación 5	

Uso de Licencias	79
3.9.1.6. Definición de Áreas de Evaluación 6	
Compatibilidad	80
3.10. Definición de Criterios	82
3.10.1. Definición de Criterios de Aprendizaje ..	82
3.10.2. Definición de Criterios de Rendimiento .	83
3.10.2.1. Rendimiento Independiente	84
3.10.3. Definición de Criterios de Explotación y Uso	86
3.10.4. Definición de Criterios de Desarrollo .	88
3.10.5. Definición de Criterios de Uso de Licencias	89
3.10.6. Definición de Criterios de Compatibilidad .	91
3.11. Postulados de Comparación de las Plataformas J2EE y .NET	92
3.11.1. Ventajas y Desventajas de las plataformas	94
3.12. Postulados de Comparación	95
3.12.1. Postulados de comparación 1	95
3.12.2. Postulados de comparación 2	97
3.12.3. Postulados de comparación 3	98
3.12.4. Postulados de comparación 4	100
3.12.5. Postulados de comparación 5	101
3.12.6. Postulados de comparación 6	104
3.12.7. Postulados de comparación 7	105

3.12.8. Postulados de comparación 8	107
3.12.9. Resumen de los postulados	108
3.13. Sistemas Web con Base de Datos en J2EE y .NET	110
3.13.1. Motores de Bases de Datos a utilizar ..	112
3.13.1.1. Mysql	112
3.13.1.2. Postgres	114
3.13.1.3. Ms Access	116
3.14. Aplicaciones Web con Base de Datos en J2EE	122
3.14. Conexión a base de datos con JDBC	123
3.17.2. Sistema de Base de Datos con Mysql y Netbeans	125
3.14.3. Sistema de Base de Datos con Microsoft Access y Netbeans	126
3.14.4. Sistemas de Base de Datos con PostgresSql y Netbeans.....	128
3.14.5. Información Adicional	129
3.15. Aplicaciones Web con base de datos en .NET ..	130
3.15.1. Sistema de Base de Datos MySQL y Visual Studio	132
3.15.2. Sistema de Base de Datos MS Access y Visual Studio	134
3.15.3. Sistema de Base de Datos PostgreSQL y Visual Studio	137

3.15.4. Información Adicional	139
-------------------------------------	-----

CAPÍTULO IV

4. DESARROLLO DE PRUEBAS DE COMPARACIÓN ENTRE LAS PLATAFORMAS.

4.1. Introducción	141
4.2. Diseño de Pruebas Comparativas	142
4.2.1. Diseño de Pruebas de Aprendizaje	143
4.2.2. Diseño de Pruebas de Rendimiento	144
4.2.3. Diseño de Pruebas de Explotación y Uso .	147
4.2.4. Diseño de Pruebas de Desarrollo	147
4.2.5. Diseño de Pruebas de Uso de Licencias ..	148
4.2.6. Diseño de Pruebas de Compatibilidad	148
4.3. Creación Tablas de Evaluación y Resultados	149
4.4. Ejecución y Resultados de Pruebas Comparativas	152
4.4.1. Ejecución y Resultados Pruebas de Aprendizaje	153
4.4.1.1. Información adicional a la tabla	156
4.4.2. Ejecución y Resultados Pruebas de Rendimiento	157
4.4.2.1. Automatización de Prueba	168

4.4.3. Ejecución y Resultados Pruebas de Explotación y Uso	170
4.4.3.1. Automatización de Prueba	173
4.4.4. Ejecución y Resultados Pruebas de Desarrollo	174
4.4.5. Ejecución y Resultados Pruebas de Uso de Licencias	177
4.4.6. Ejecución y Resultados Pruebas de Compatibilidad	180
4.5. Conclusiones Finales de Pruebas Realizadas	183
4.5.1. Conclusiones de Pruebas de Aprendizaje .	185
4.5.2. Conclusiones de Pruebas de Rendimiento .	187
4.5.3. Conclusiones de Pruebas de Explotación y Uso	190
4.5.4. Conclusiones de Pruebas de Desarrollo ..	193
4.5.5. Conclusiones de Pruebas de Uso de Licencias	198
4.5.6. Conclusiones de Pruebas de Compatibilidad	201

CAPÍTULO V**5. CONCLUSIONES Y RECOMENDACIONES**

5.1. Conclusiones	205
5.2. Recomendaciones	209
BIBLIOGRAFÍA	211
GLOSARIO DE TÉRMINOS Y SIGLAS	217
ANEXOS	226

TABLA DE GRÁFICOS

CONTENIDOS	Pág.
Figura 1. Esquema de la máquina virtual de JAVA	15
Figura 2. Esquema de Aplicaciones en Capas ...	22
Figura 3. Conjunto de API's J2EE	23
Figura 4. Estructura de un archivo JAR	30
Figura 5. Estructura de un archivo WAR	31
Figura 6. Estructura de un archivo EAR	32
Figura 7. Estructura de un archivo RAR	32
Figura 8. Arquitectura .Net	36
Figura 9. Esquema .Net Framework	41
Figura 10. Interfaz NetBeas IDE	66
Figura 11. Modelo de aplicación Web con Base de Datos	112
Figura 12. Vista de Programación en NetBeans ..	124
Figura 13. Vista de Programación en Visual Studio .Net	132
Figura 14. Gráfico de Rendimiento Normal	158
Figura 15. Gráfico de ejecución de NetBeans .	159
Figura 16. Gráfico de ejecución de Visual Studio.Net	159
Figura 17. Automatización de Rendimiento Primera instancia	169

Figura 18. Automatización de Rendimiento	
Segunda instancia	169
Figura 19. Gráfica explotación	172
Figura 20. Automatización Explotación	173

TABLAS

CONTENIDOS	Pág.
Tabla 1. Tabla comparativa, primeras Comparaciones	53
Tabla 2. Ejemplo de código fuente JSP	56
Tabla 3. Ejemplo de código fuente ASPx	61
Tabla 4. Tabla explicativa IDE's de desarrollo J2EE	64
Tabla 5. Tabla comparativa. Principales Características	71
Tabla 6. Ejemplo de conexión a Bases de datos Postgres	124
Tabla 7. Ejemplo del código cargar .jsp	126
Tabla 8. Ejemplo del código eliminar .jsp	127
Tabla 9. Ejemplo del código insertar .jsp	129
Tabla 10. Parte código inicio.aspx	133
Tabla 11. Parte de código inicio Vb	134
Tabla 12. Parte de código insertar. aspx.....	136
Tabla 13. Parte de código insertar. Vb	138
Tabla 14. Tabla de resultados	151
Tabla 15. Tabla de resultados	152
Tabla 16. Tabla de evaluación 1: Aprendizaje ..	156
Tabla 17. Tabla de evaluación 2: Rendimiento ..	170
Tabla 18. Tabla de evaluación 3: Explotación y	

uso	174
Tabla 19. Tabla de evaluación 4: Desarrollo ..	177
Tabla 20. Tabla de evaluación 5: Licencias ..	179
Tabla 21. Tabla de evaluación 6: Compatibilidad	183
Tabla 22. Tabla de resultados Aprendizaje J2EE vs .Net	187
Tabla 23. Tabla de resultados Rendimiento J2EE vs .Net	189
Tabla 24. Tabla de resultados Exploración y uso J2EE vs .Net	193
Tabla 25. Tabla de resultados Desarrollo J2EE vs .Net	197
Tabla 26. Tabla de resultados Uso de Licencias J2EE vs .Net	200
Tabla 22. Tabla de resultados compatibilidad J2EE vs .Net	204

CAPÍTULO I

1. ANTECEDENTES E INTRODUCCIÓN

1.1. INTRODUCCIÓN

El avance informático a través del tiempo es cada vez más vertiginoso y caótico. Mientras ocurren los hechos, la tecnología, las plataformas de trabajo y los lenguajes de programación avanzan a velocidades que difícilmente se lograría estar al tanto de todo. En esta especialidad, se debe estar atento a estos cambios y avances para no quedar atrás para así poder competir y mantenerse dentro de este ambiente.

En la utilización de tecnologías de información, el uso del Web es primordial. El poder tener aplicaciones que puedan mantener procesos difícilmente llevados por otro medio y, que estén actualizados y disponibles para todo el mundo, hace que sea necesario el desarrollo de éstas por las mismas empresas, proceso que difiere a otro tipo de problemas, los cuales pueden ser solventados por software's ya desarrollados.

En esta área de la programación, son dos las plataformas que destacan sobre el resto, siendo las más utilizadas y aprobadas por los usuarios. J2EE, una mejora importante del lenguaje de programación Java, y .NET, la nueva tecnología de Microsoft, que implementa de mejor manera aplicaciones desarrolladas en el pasado por esta empresa, se imponen fuertemente en el mercado de desarrollo ocupando más de la mitad de éste por sobre otras plataformas de diseño.

Tanto las empresas como los usuarios relacionados con el Desarrollo de Aplicaciones Web, ocupan estas tecnologías, las familiarizadas con tecnología open source y de sistemas operativos Linux utilizan la tecnología JAVA, que no presenta problemas con sus aplicaciones y sistemas desarrollados, teniendo el mismo rendimiento en Sistemas Microsoft.

Por su parte ésta última, con el lanzamiento de la tecnología .NET ha absorbido la mayor parte del desarrollo de aplicaciones de la mediana y pequeña empresa sin menospreciar las grandes corporaciones que han hecho suya esta norma.

Pues bien, a la hora de abordar algún problema Web, ¿Qué

tecnología se debe adoptar?, ¿Qué plataforma de trabajo es mejor? ¿Y por qué?, ¿Por qué elegir una, y no la otra?, ¿Qué plataforma requiere más recursos, tanto humanos como técnicos?, ¿Qué costos debe considerar el usuario, al acceder a una de estas plataformas? Este estudio explicará y dará respuestas, a cada una de las preguntas planteadas, dando justificaciones técnicas, teóricas y prácticas. Poniéndose en lugar del usuario y partiendo desde el inicio con cada una de las tecnologías, entregando una orientación comparativa entre ambas, analizando diferencias y parecidos en cada área de aplicación, para ser utilizado de manera específica en la toma de decisiones y el conocimiento previo que se necesita antes de dedicarse por completo a una de estas plataformas de trabajo.

1.2. DEFINICIÓN DEL PROBLEMA

En la actualidad, muchos han abordado el tema de explicar la mejor alternativa de desarrollo entre J2EE y .NET. Algunos desarrollando ideas y comentarios a base de su propia experiencia con cada una de estas plataformas, otros, tratando de explicar al usuario lo que a su modo de ver es mejor, (muchas veces cayendo en fanatismos), lo cual nubla el juicio al basarse en las tecnologías en

las cuales trabajan o prestan servicio.

Además cabe señalar que estudios hechos y sitios dedicados al tema no detallan en profundidad las ventajas y desventajas mencionadas para cada plataforma, haciendo de estas conclusiones bastante subjetivas a la hora de desarrollar aplicaciones Web.

Tomando esto en consideración, se determinó ser un medio de respuesta a la interrogante de decisión de desarrollo, la cual fuese posible demostrar mediante el estudio y posterior creación de áreas de evaluación y criterios que permitan demostrar en base a pruebas, que tecnología tiene una mejor evaluación frente a la otra en aspectos de diseño, desarrollo y ejecución de aplicaciones Web.

1.3. DELIMITACIÓN DEL PROBLEMA

1.3.1. DELIMITACIÓN DE CAMPO

Campo: Informática y sistemas computacionales.

Área: Aplicaciones WEB.

Aspecto: Aplicación de tecnologías para el desarrollo WEB.

1.3.2. DELIMITACIÓN TEMPORAL

El tiempo estipulado para la presente investigación será de 10 meses desde el 16 de marzo del 2009 al 30 de enero del 2010.

1.4. MARCO TEÓRICO

1.4.1. LA PLATAFORMA JAVA

JAVA fue desarrollado a finales de los años ochenta por Sun Microsystems, su objetivo principal nace en sus primeras versiones, como un lenguaje familiar, con similitudes a C++.

Este concepto se puede definir como un estándar de desarrollo a nivel informático que proporciona las herramientas necesarias para la creación de aplicaciones de todo tipo. Además es un lenguaje orientado a objetos, que soporta características como: la encapsulación, la herencia, el polimorfismo y el enlace dinámico.

Por otra parte JAVA es una de las plataformas que cuenta con mayor acogida para la construcción

de aplicaciones Web multinivel. Su Arquitectura J2EE (Java 2 Platform, Enterprise Edition) ofrece conceptos básicos de componentes, herramientas y ambientes de desarrollo que permiten generar programas o páginas dinámicas desplegadas en los distintos browsers de la Web.

La principal característica de JAVA es la capacidad de ejecutarse en cualquier máquina y sobre cualquier sistema operativo o arquitectura, manteniendo las facilidades básicas del lenguaje. En el ámbito Web, JAVA da la capacidad de desplazar el control de la interactividad de los servidores hacia las máquinas de los usuarios, permitiendo la utilización de distintos componentes que son interpretados por los navegadores Web y en los casos correspondientes por la máquina virtual de JAVA. Un ejemplo de esto es la carga de "applets", programas compactos y precompilados que generan animaciones y sonidos sobre páginas Web.

Otra característica propia de JAVA es el acceso a herramientas de desarrollo de licencia libre que permite a los usuarios un fácil acceso a esta tecnología.

1.4.2. LA PLATAFORMA .NET

La Plataforma .NET se refiere a la tecnología creada por Microsoft para sus productos de programación en la línea de .NET Framework y específicamente al producto de programación Visual Basic .NET, el cual se presenta como una herramienta completa para la creación de aplicaciones con esta tecnología. Para esto Microsoft, propuso una herramienta de diseño y programación que cuenta con un número mayor de funcionalidades que la ubica como una tecnología de punta en el desarrollo de aplicaciones, como lo son la programación orientada a objetos y el poderío de JAVA con su arquitectura J2EE.

Con esto Microsoft, agrego el apellido ".NET" a la mayoría de sus software comerciales creados bajo este nuevo enfoque de desarrollo, productos como Visual Basic .NET incorporan estas nuevas herramientas no incluidas en versiones anteriores, así se añade un conjunto de cambios para los programadores que deben afrontar con garantías el desarrollo de la nueva generación de aplicaciones en general, especial el ámbito Web, objetivo perseguido por todas las herramientas de desarrollo actuales.

Dentro de estas nuevas características se encuentran mejoras sobre ciertos componentes del lenguaje en sí, con la eliminación de aspectos obsoletos que han ido cambiando dentro de los modos y hábitos de programar. De esta forma se debe estar familiarizado con programación de un cierto nivel y con las herramientas que proporcionan lo necesario para la creación de aplicaciones sobre la plataforma .NET con su arquitectura NET Framework.

1.5. FORMULACIÓN DE LA HIPÓTESIS

El desconocimiento de opciones tecnológicas complementarias disminuyen los beneficios tangibles en un proyecto de desarrollo Web.

1.6. VARIABLES E INDICADORES

1.6.1. VARIABLE DEPENDIENTE

Los beneficios tangibles en un proyecto de desarrollo WEB.

1.6.2. VARIABLE INDEPENDIENTE

El desconocimiento de opciones tecnológicas

1.7. OBJETIVOS DEL PROYECTO

En el trabajo de investigación que se presenta a continuación se contemplan los siguientes objetivos.

1.7.1. OBJETIVO GENERAL.

El Objetivo principal de este proyecto es realizar un análisis comparativo - práctico de las plataformas de desarrollo J2EE y .NET en Aplicaciones Web.

1.7.2. OBJETIVOS ESPECÍFICOS.

- Investigar, documentar y clasificar información relacionada con el tema.
- Establecer un diseño de comparación en cada área, que permita identificar ventajas y desventajas de estos lenguajes.

- Realizar un análisis comparativo, desarrollando módulos de comparación y aplicaciones entre las dos plataformas.
- Definir y Realizar pruebas comparativas las cuales permitan realizar tablas explicativas sobre ventajas y desventajas sobre estas plataformas.
- Dar conclusiones puntuales a cada punto desarrollado, para que éste documento sirva como una herramienta en la toma de decisiones.

Mediante estos puntos se quiere dar respuesta al planteamiento propuesto, lo cual conlleva a poner en práctica la información recolectada para luego ir desarrollando los temas posteriores definiendo una metodología de comparación y estableciendo los criterios que se implementarán en la etapa de pruebas.

1.8. METODOLOGÍA

1.8.1. FUENTES DE INFORMACIÓN

Para realizar la presente investigación se tomará información en base a fuentes primarias y secundarias.

En cuanto a fuentes primarias se tomara información directa de las empresas de desarrollo WEB, a través de los empresarios, desarrolladores y usuarios, quienes aportarán en gran parte para tener un enfoque claro de la situación actual de la misma. Para lo que se utilizará herramientas de investigación científica como son: entrevistas, encuestas, observación directa y grupos focales, mediante la investigación descriptiva la misma que proporcionará un resumen de aspectos del medio.

En cuanto a fuentes secundarias la información se recolectará de libros, folletos y estudios especializados realizados con anterioridad.

1.8.2. INSTRUMENTOS PARA OBTENER INFORMACIÓN

- Entrevistas
- Encuestas, estructuradas y no estructuradas.

1.9. MÉTODOS DE INVESTIGACIÓN

Se realizará una investigación cuantitativa, porque todo el análisis se medirá a través de indicadores de uso y determina la factibilidad de utilizar otras tecnologías para el desarrollo de aplicaciones WEB.

Se efectuarán investigaciones de campo. También se aplicara la investigación bibliográfica documental, la misma que permitirá recolectar y analizar información o casos de éxito en la implementación de proyectos de desarrollo de Aplicaciones WEB.

CAPÍTULO II

2. PLATAFORMAS DE DESARROLLO

2.1. INTRODUCCIÓN A LA PLATAFORMA JAVA.

JAVA fue desarrollado a finales de los años ochenta por Sun Microsystems, su objetivo principal nace en sus primeras versiones, como un lenguaje familiar, con similitudes a C++. Este concepto se puede definir como un estándar de desarrollo a nivel informático que proporciona las herramientas necesarias para la creación de aplicaciones de todo tipo. Además es un lenguaje orientado a objetos, que soporta características como: la encapsulación, la herencia, el polimorfismo y el enlace dinámico.

Por otra parte Java es una de las plataformas que cuenta con mayor acogida para la construcción de aplicaciones Web multinivel. Su Arquitectura J2EE (Java 2 Platform, Enterprise Edition), ofrece conceptos básicos de componentes, herramientas y ambientes de desarrollo que permiten generar programas o páginas dinámicas

desplegables en los distintos browsers de la Web.

La principal característica de Java es la capacidad de ejecutarse en cualquier máquina y sobre cualquier sistema operativo o arquitectura, manteniendo las facilidades básicas del lenguaje. En el ámbito Web, Java da la capacidad de desplazar el control de la interactividad de los servidores hacia las máquinas de los usuarios, permitiendo la utilización de distintos componentes que son interpretados por los navegadores Web y en los casos correspondientes por la máquina virtual de Java. Un ejemplo de esto es la carga de "applets", programas compactos y recompilados que generan animaciones y sonidos sobre páginas Web. Otra característica propia de Java es el acceso a herramientas de desarrollo de licencia libre que permite a los usuarios un fácil acceso a esta tecnología. [23]

2.1.1. LA MÁQUINA VIRTUAL DE JAVA.

La máquina virtual de Java, es una capa lógica que hace creer al programa Java que se ejecuta en un computador (con registros, memoria y procesador), cuando en realidad sólo ve una reconstrucción lógica de éste. Para ejecutar un programa Java compilado, es preciso también que se

cuenta con una implementación de la máquina virtual específica donde se desea ejecutar. En la **Figura 1** se muestra el esquema de la Máquina Virtual de Java la cual efectúa la transformación del p-code en un programa comprensible para la máquina.

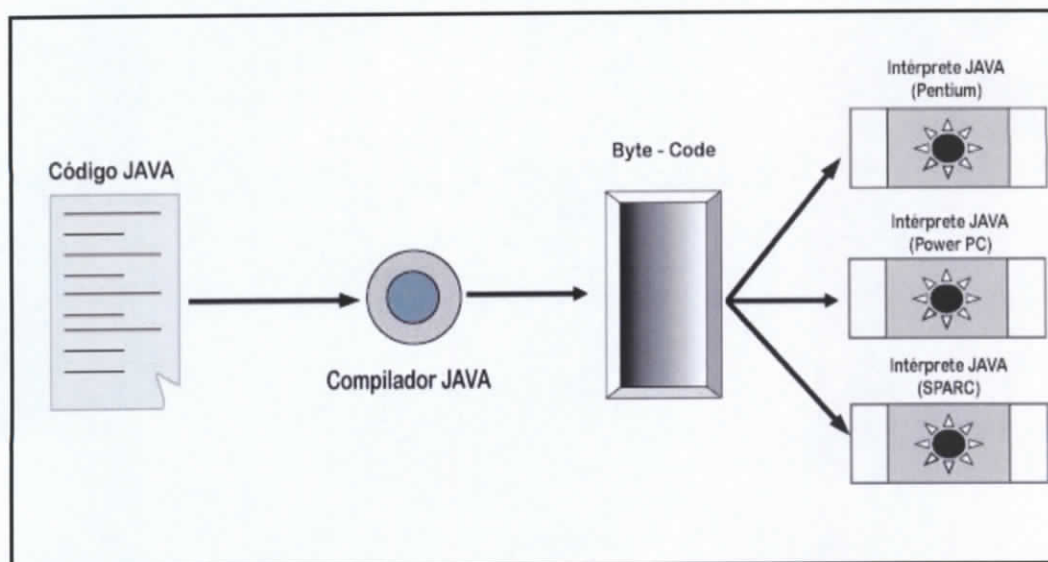


Figura 1: Esquema Máquina Virtual Java.

2.1.2. HERRAMIENTAS DE DESARROLLO.

Para desarrollar en Java, se necesita instalar el paquete JDK de Sun (Java Deployment Kit), libre de distribución y se puede descargar en varios sitios Web.

En el sitio de Sun, se encuentra información relacionada con Java como: ejemplos de programas, tutoriales, documentación, bugs junto a su solución, lo que proporciona una ayuda extra para desarrollar todo tipo de

aplicaciones en este lenguaje. La última versión disponible, Java 2 Standard Edition 5.0 y Java 2 Enterprise Edition 1.4, las cuales incluyen JDK 5.0 (Java Developer Kit), como se conocen estos "paquetes".

2.1.3. TIPOS DE HERRAMIENTAS DE APLICACIÓN.

En Java se pueden realizar todo tipo de software, estos pueden ser: [23]

- **Aplicaciones:** programas fuentes desarrollados en cualquier editor Java y guardados con extensión .java, que en su compilación pasan a ser ficheros .class, y al ser ejecutados se interpreta el byte-code, generando estas aplicaciones.
- **Applets:** códigos fuentes desarrollados en cualquier editor Java (pudiendo ser de extensión .java), compilados de forma que sean parte de una página Web que lo contenga, ejecutándolo dentro del sitio con extensión .html.
- **Servlet:** conocido como el CGI de Java (Common Gateway Interface), éste un programa independiente de la plataforma que aporta funcionalidad a la programación en

el lado del servidor. Este es código java que puede llegar a contener tags HTML.

- **Java Bean:** es una arquitectura que permite una manera de re-utilizar componentes de software que pueden ser manipulados en herramientas de desarrollo ("Builder Tools"). Estos "Beans" pueden ser tan sencillos como un botón, o complejos como el acceso a una base de datos; una característica primordial de un JavaBean son los métodos (funciones) get/set.
- **JSP:** "Java Server Pages", Este es un tipo de programa Java que contiene HTML, para ejecutarlo se requiere de un servlet engine como Tomcat o bien un Java Application Server como Websphere de IBM.

2.1.4 ARQUITECTURAS JAVA.

JAVA presenta tres tipos de arquitecturas desarrolladas por Sun Microsystems estas son:

- **J2EE. (Java 2 Enterprise Edition)** Plataforma de tecnología Java más avanzada, la cual provee un rango completo de funcionalidades empresariales para el desarrollo de aplicaciones tanto de computadores como

de servidores. Esta plataforma ha sido diseñada con el fin de proporcionar un ambiente integrado para la creación de programas Java en todos niveles requeridos por los usuarios.

- **J2SE. (Java 2 Standard Edition)** Plataforma de desarrollo JAVA que provee el pack básico de utilidades necesarias para el desarrollo de aplicaciones en este lenguaje. Este es normalmente, la versión que debemos descargar desde Internet para poder trabajar.
- **J2ME (Java 2 Micro Edition)** Plataforma de desarrollo para la implementación de aplicaciones para dispositivos móviles, tanto equipos celulares como Palm y equipos de bolsillo (Poquet PC). Ésta contiene las herramientas específicas para desarrollar aplicaciones, programas y juegos, para este tipo de dispositivos.

2.1.4. DEFINICIÓN DE JAVA 2EE.

El concepto J2EE, Java 2 Enterprise Edition es la arquitectura creada por Sun para el desarrollo de todo tipo de aplicaciones para empresas y usuarios en general

facilitando las herramientas necesarias para la programación en el lenguaje Java, las cuales facilitan el manejo de los conceptos necesarios para el desarrollo de aplicaciones más avanzadas de desarrollo, implementación y administración de software empresarial. [6]

SUN lo define como un estándar para el desarrollo de aplicaciones empresariales multicapa, simplificando las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchas de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja.

Además la plataforma J2EE incluye un amplio rango de (API's) *Application Programming Interface* y esquemas de programación como JSP y Servlets, entre otros, los cuales pueden ejecutarse dentro del servidor Web Tomcat incluido dentro de esta arquitectura. [6]

J2EE contiene cuatro grandes áreas de la ejecución de este tipo de aplicaciones, éstas son:

- **PRESENTACIÓN DEL LADO DEL CLIENTE**

Client - Side Presentation: dentro de esta área se encuentra el rango de funcionalidades que J2EE hace ejecutar en la máquina en la cual el cliente accede a las distintas herramientas. Dentro de este grupo se encuentran páginas Web estáticas (HTML), los Applets y aplicaciones de escritorio descritas en el lenguaje JAVA.

- **PRESENTACIÓN DEL LADO DEL SERVIDOR**

Server - Side Presentation: esta área incluye las aplicaciones las cuales JAVA desliga del cliente al ejecutarlas en el servidor, lo que agiliza de manera importante el despliegue de éstas. En esta área se encuentran mayormente el tipo de aplicación Web de J2EE, Java Server Pages, Java Servlet, códigos XML y los servicios Web J2EE.

- **LÓGICA DE NEGOCIOS DEL LADO DEL SERVIDOR**

Server - Side Business Logia: este grupo de componentes J2EE contiene aquellos conceptos que no necesariamente son aplicaciones, sino dispositivos encargados de generar procesos de modo no visible para el usuario, como lo son

los EJB, y los JavaBeans, los cuales permiten la interacción de componentes Web Services entre las distintas capas SOAP y WDSL. Los EJB container manejan la ejecución de los Enterprise Beans para las aplicaciones J2EE que se ejecutan del lado del servidor.

- **INFORMACIÓN DE SISTEMA DE EMPRESA**

Enterprise Information System: dentro de ésta área se incluyen los datos guardados necesarios para la correcta ejecución de los distintos componentes J2EE. Por ejemplo las bases de datos que incluyen información de software ejecutados del lado del servidor.

En la **Figura 2** se muestra un esquema de aplicación en Capas. En este modelo se encuentran los distintos componentes de una aplicación J2EE y la ubicación en la correspondiente capa del modelo de ejecución.

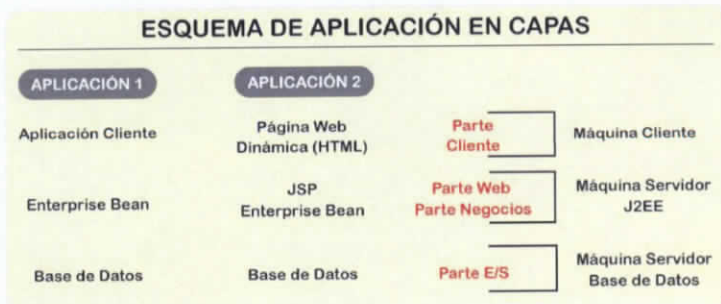


Figura 2: Esquema de Aplicación en Capas.

que corren del lado del servidor y EJB Containers que manejan la ejecución de los beans de aplicaciones J2EE que también corren del lado del servidor. Por su parte el Application Client Container maneja la ejecución de todos los componentes de servidor y a su vez el Applet Container la ejecución de dichos applets en el cliente.

2.1.6. PRINCIPALES API'S DE LA ARQUITECTURA J2EE.

Dentro de las principales APIs para el desarrollo de la tecnología Java, se encuentran algunas que permiten ciertas características, por ejemplo a los Java servlet le permiten extender la capacidad de los servidores en aplicaciones http y utilizarlos como respuestas sobre servicios Web. También permiten a los JSP agregar elementos que determinen como está compuesto el contenido dinámico de la página, están dentro de los EJB y hacen correr Applets de manera correcta en la máquina del cliente. [24]

Las API son interfaces de programación de aplicaciones (Application Programmet Interface) las cuales constituyen un conjunto de funciones o mandatos que en un entorno concreto (sistema operativo) permiten al usuario poder

actuar con éste. Algunas de las principales API's de la arquitectura son:

- **JMS** (Java Message Service API): es un mensaje estándar que permite a componentes de aplicaciones J2EE crear, enviar, recibir y leer mensajes, habilitando comunicaciones distribuidas (asincrónicas).
- **JTA** (Java Transaction API): provee una interfaz estándar para transacciones demarcadas. Maneja un auto commit por defecto para transacciones commits y rollbacks (una aplicación con manejo de datos que puede actualizarlos luego de cada operación de escritura o lectura).
- **JavaMail API**: API para envío de notificaciones email en Internet.
- **JAF** (JavaBeans Activation Framework): provee servicios para determinar tipos de datos dentro de JavaMail API.
- **JAXP** (Java API for XML Processing): procesamiento de documentos XML usando Document Object Model (DOM), Simple API for XML (SAX) y Extensible Stylesheet Language Transformations (XSLT).

- **JAX-RPC** (Java API for XML-Based RPC): Esta API usa el estándar SOAP (Simple Object Access Protocol) y HTTP para que el cliente pueda programar procedimientos en XML (Extensible Markup Language) llamados RPCs sobre Internet.
- **SAAJ** (SOAP with Attachments API for Java): API de bajo nivel que habilita la producción y consumo de mensajes SOAP.
- **JAXR** (Java API for XML Registries): registra el acceso de negocio y propósito general de aplicaciones sobre la red.
- **JDBC** (Java Data Base Connection): Quizás el API de J2EE más conocida, permite el trabajo con base de datos permitiendo comandos SQL para la programación de métodos de acceso a distintas base de datos. Posee dos partes: una de nivel de aplicación usada por el componente que accede a los datos y además un servicio que provee una interfaz que se agrega al driver JDBC dentro de la plataforma J2EE.
- **JNDI** (Java Naming and Directory Interface): provee funcionalidad de nombres y directorios a los métodos

que realicen operaciones de este tipo.

- **JAAS** (Java Authentication and Authorization Service): esta API entrega una vía para autenticar y autorizar la ejecución de aplicaciones por un usuario o grupo en particular.
- **JSTL** (Java Server Pages Standard Tag Library): este API encapsula la funcionalidad de los JSP.

2.1.7. CARACTERÍSTICAS DE J2EE.

La tecnología Java 2 Enterprise Edition (J2EE) proporciona una completa y potente plataforma orientada al desarrollo de aplicaciones corporativas distribuidas a los servicios Web. [24] Este tipo de aplicaciones suelen tener una arquitectura en capas: una capa de cliente o de presentación que proporciona el interfaz de usuario, una o más capas intermedias que proporcionan la lógica de negocio de la aplicación y una capa final con los sistemas de información que mantienen aplicaciones y bases de datos corporativas.

Entre las características de este tipo de aplicaciones se encuentran las siguientes:

- Posibilidad de altas productividades en el desarrollo de las distintas tecnologías J2EE para la integración de aplicaciones corporativa e integración de sistemas existentes.
- Mayor Escalabilidad al describir las características básicas de transacciones y desarrollando distintos tipos de componentes de aplicación J2EE con modelos flexibles de seguridad.
- Libertad de elección de plataformas de desarrollo y producción, lo que define los aspectos necesarios que puedan solucionar una determinada problemática.
- El desarrollo de aplicaciones Web utilizando conceptos de la arquitectura J2EE que permiten la construcción de este tipo de aplicaciones.
- La utilización de herramientas libres que agilizan el desarrollo de software con J2EE y que permiten el funcionamiento en los distintos módulos de ejecución.

2.1.8. TIPOS DE APLICACIONES J2EE.

La plataforma J2EE añade a Java la funcionalidad necesaria

para convertirse en un lenguaje orientado al desarrollo de servicios y páginas en Internet, realizando aplicaciones igual de poderosas que cualquier otro tipo de programa desarrollado en Java. [25]

Éstos pueden ser de uso común por diferentes organizaciones que implementan software en la red, tanto para medios publicitarios, como software de manejo de información en Internet.

Java tiene varios tipos de aplicaciones Web, que pueden implementarse a la hora de enfrentarse a problemas de desarrollo que ocupen la red. Mediante páginas dinámicas llamadas JSP (Java Servlet Page, un lenguaje de programación de páginas Web dinámicas como lo son códigos en PHP o ASP), Sun proporciona mejoras y nuevos conceptos al desarrollo Web, sumando funcionalidades a las conocidas Páginas Web, códigos HTML, las cuales hacen que páginas ya creadas puedan ser actualizadas con implementación JSP. De misma manera los Servlets, scripts CGI en el servidor que se ejecutan como script de Perl, los cuales pueden desarrollar mejoras específicas para agregar a sitios Web tecnología JAVA.

Además de los tipos de aplicaciones J2EE, servlets y JSP,

esta arquitectura presenta otro tipo de componentes o archivos, estos vienen a ser los Packaging Applications (paquetes de aplicaciones), los cuales son una encapsulación de aplicaciones con distinta orientación.

Algunas de estas aplicaciones J2EEE son:

- **JAR** (Java archive): es un formato de archivo independiente de la plataforma que permite que varios archivos puedan ser encapsulados dentro de uno, permitiendo que éste pueda ser una aplicación completa de fácil movilidad y ejecución.

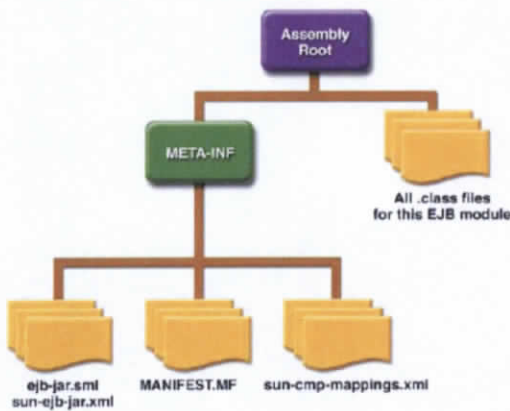


Figura 4: Estructura archivo JAR.

- **WAR** (Web Application archive): este componente es un archivo que contiene un archivo JAR que posee uno o más módulos WEB. Pudiendo ser desde un simple sistema JSP a un servicio Web.

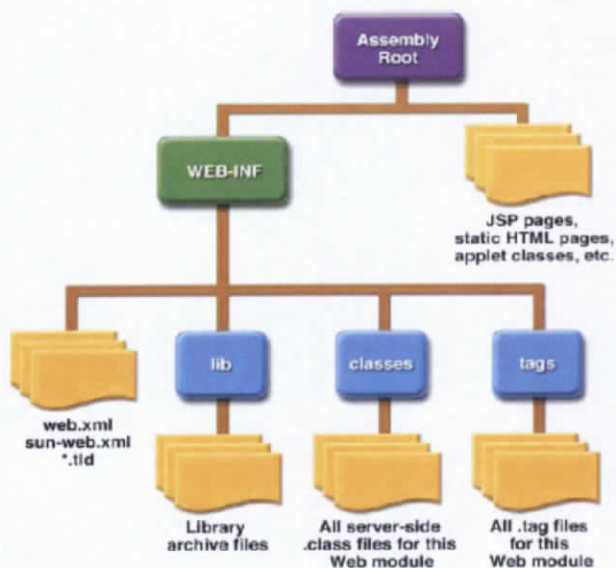


Figura 5: Estructura archivo WAR.

- **EAR** (Enterprise Archive file): un archivo EAR es un archivo JAR que contiene una aplicación J2EE. Utilizando archivos EAR es posible unir un número de diferentes aplicaciones J2EE usando los mismos componentes sin necesidad de códigos extra solo incluyéndolos dentro de este módulo.

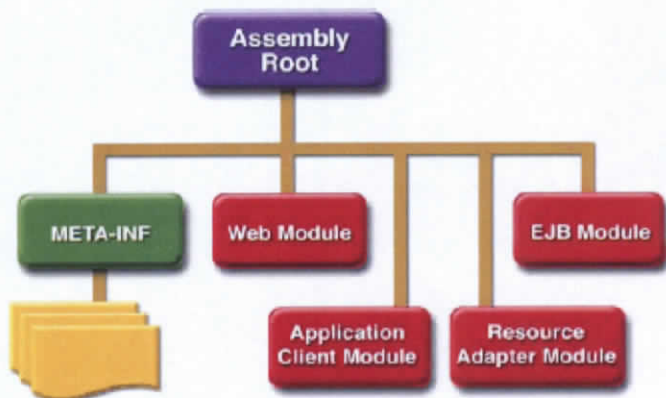


Figura 6: Estructura archivo EAR.

- **RAR** (Resource Adapter Archive): Un archivo RAR es un archivo JAR que contiene un adaptador de recursos que puede ser desplegado en un servidor J2EE. Es bastante similar a un archivo EAR e incluso un RAR puede estar contenido dentro de éste.

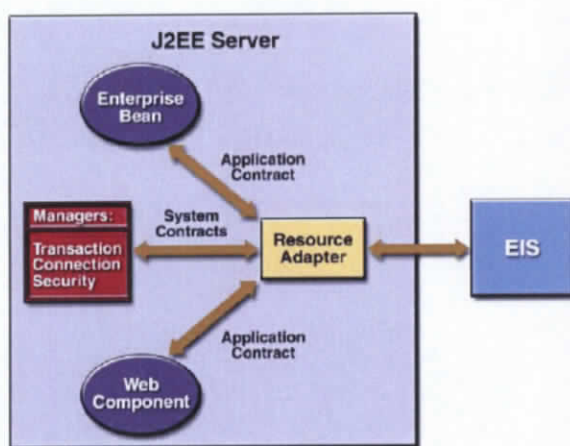


Figura 7: Estructura archivo RAR.

2.1.9. ACCESO A DATOS EN J2EE.

Para poder acceder a las bases de datos en J2EE se debe hacer referencia a la ya mencionada API JDBC, ya que este es uno de los métodos por el cual Java se puede conectar a los distintos motores de Bases de Datos ya que Java también utiliza para estas conexiones los Frameworks de persistencia.

JDBC, son las siglas de Java Data Base Connection, el

cual es un método de interconexión el cual consta de un paquete de instalación comprimido .JAR, el que contiene la librerías que hacen posible un puente entre la aplicación y el motor de base de datos, generalmente esto es conocido como Driver de base de datos.

Este es un método bastante simple y de buen desempeño, por lo que es posible conectar distintas bases de datos, con distintos motores, con solo cambiar el paquete JDBC por el correspondiente que sea necesitado.

Para este método se pueden definir características como:

- Define ocho interfaces para operaciones con bases de datos, de las que se derivan clases para su utilización.
- Contiene una clase que se encarga de cargar inicialmente todos los drivers JDBC disponibles DriverManager.
- Con la clase Connection, se realiza la conexión con la base de datos. La conexión se especifica siguiendo una sintaxis basada en la especificación más amplia de los URL (Uniform Resource Locator).

2.2. INTRODUCCIÓN A LA PLATAFORMA MICROSOFT.NET

La Plataforma .NET se refiere a la tecnología creada por Microsoft para sus productos de programación en la línea de .NET Framework y específicamente al producto de programación Visual Basic .NET, el cual se presenta como una herramienta completa para la creación de aplicaciones con esta tecnología. Para esto Microsoft, propuso una herramienta de diseño y programación que cuenta con un número mayor de funcionalidades que la ubica como una tecnología de punta en el desarrollo de aplicaciones, como lo son la programación orientada a objetos y el poderío de JAVA con su arquitectura J2EE. [3]

Con esto Microsoft, agrego el apellido “.NET” a la mayoría de sus software comerciales creados bajo este nuevo enfoque de desarrollo, productos como Visual Basic .NET incorporan estas nuevas herramientas no incluidas en versiones anteriores, así se añade un conjunto de cambios para los programadores que deben afrontar con garantías el desarrollo de la nueva generación de aplicaciones en general, especial el ámbito Web, objetivo perseguido por todas las herramientas de desarrollo actuales.

Dentro de estas nuevas características se encuentran

mejoras sobre ciertos componentes del lenguaje en sí, con la eliminación de aspectos obsoletos que han ido cambiando dentro de los modos y hábitos de programar.

De esta forma se debe estar familiarizado con programación de un cierto nivel y con las herramientas que proporcionan lo necesario para la creación de aplicaciones sobre la plataforma .NET con su arquitectura NET Framework.

2.2.1. DESARROLLO DE APLICACIONES EN .NET.

Además de programar en .NET, se pueden incluir dentro de esta tecnología, el análisis y el diseño, permitiendo entender cómo hacer uso de mejores técnicas para el desarrollo de aplicaciones. De esta forma se integran todas las etapas del ciclo de desarrollo dentro de .NET, definiendo las especificaciones y modelado del problema para llegar al resultado final reflejado en el código obtenido. [1]

La integración de Visual Basic con el resto de lenguajes del entorno de .NET, agrega mejoras al interactuar con productos de la misma plataforma, facilitando el trabajo y alcanzando óptimas soluciones en menores plazos con

códigos más mantenibles y reusables, los cuales se reflejan en los distintos niveles de desarrollo de aplicaciones .NET.

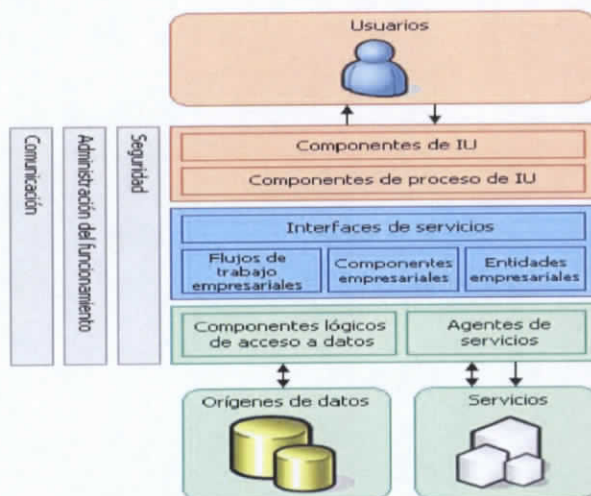


Fig. 8 Arquitectura de .NET

Haciendo referencia a la figura #8 que hace referencia a la arquitectura .NET presenta distintos niveles de aplicaciones y componentes ofreciendo distintas capacidades y servicios apoyándose en los distintos sistemas operativos Windows y sistemas compatibles.

De esta forma poder presentar a los usuarios incluidos dentro del nivel superior de la arquitectura los siguientes aspectos divididos en 3 capas:

- **La capa de presentación:** formada por los Componentes de IU Interface User y los componentes de proceso de IU.

Los componentes de IU son los cuales interactúan con el usuario (por ejemplo ventanas o páginas Web).

Los componentes de proceso de IU distinto tipo de clases del tipo controladora en UML (Unified Modeling Language), lo cuales encapsulan la lógica de navegación y control de eventos de la interface.

- **La capa de negocios:** es la cual encapsula la lógica de negocios. Las entidades empresariales representan objetos que van a ser manejados por la aplicación (modelo de objetos, xml, datasets con tipo, estructuras de datos), las cuales permiten representar objetos que han sido identificados durante el modelamiento. Los componentes empresariales contienen lógica de negocios y en algunos casos pueden ser los objetos raíz que inician las transacciones.
- **La capa de acceso a datos:** capa que contiene las clases que interactúan con las base de datos. Éstas surgen con la necesidad de mantener la cohesión o clases altamente especializadas que ayuden a reducir la dependencia entre las clases y capas. Aquí se encuentra la clase de métodos estáticos que permite uniformizar las operaciones de acceso a datos a través de un único

conjunto de métodos, esta es la clase SQLHelper.

2.2.2. PROGRAMACIÓN WEB CON MICROSOFT.NET

Después de conocer los conceptos presentados anteriormente como solución a problemas Web, se definirán las principales características y funcionalidades de la plataforma .NET en la programación de sitios, portales y páginas WEB o ASPx, las cuales, presentan un sin número de ventajas sobre conocimientos HTML y páginas con lenguajes interpretados por ejemplo PHP. Para esto se define:

- **Frontal Web principal:** éste es accesible desde un navegador de Internet y permite a los clientes consultar, acceder, informarse. Todo esto dentro de páginas dinámicas ASPx.

Para esto la tecnología ASP.NET presentan los siguientes conceptos:

- **Modelo de ASP.NET e IIS** (Internet Information Server).
- **Web Forms:** permiten diseñar páginas Web dinámicas basadas en componentes .NET.

- **Controles de usuario:** implementados por el desarrollador para ser reutilizados en distintas páginas o proyectos.
- **Sesión y caché:** usados para el mantenimiento del estado y la optimización del rendimiento.
- **Configuración e instalación** de sitios Web.
- **Seguridad:** técnicas disponibles en ASP.NET para la autenticación y autorización de usuarios.

2.2.3. INTRODUCCIÓN A .NET FRAMEWORK.

Al mencionar a .NET Framework, se dice que constituye la plataforma y elemento principal sobre el que se asienta Microsoft .NET.[9] Para los programadores, es la pieza fundamental de todo este nuevo modelo de trabajo que brinda Microsoft, ya que proporciona las herramientas y servicios que se necesitan en las labores habituales de desarrollo. .NET Framework, permite el desarrollo de aplicaciones a través del uso de un conjunto de herramientas y servicios que pueden agruparse en tres bloques principales:

El Entorno de Ejecución Común o Common Language Runtime (CLR), la jerarquía de clases básicas de la plataforma o .NET Framework Base Classes y el motor de generación de interfaz de usuario, que permite crear interfaces para la Web o para el tradicional entorno Windows. Se puede definir a .NET Framework como "Una plataforma independiente del lenguaje para el desarrollo de Aplicaciones Web", la cual tiene como objetivo el modelo de programación basado en aplicaciones y servicios, permitiendo a los desarrolladores centrarse en escribir la lógica específica a desarrollar con esta arquitectura compuesta por una serie de librerías y un entorno de ejecución común.

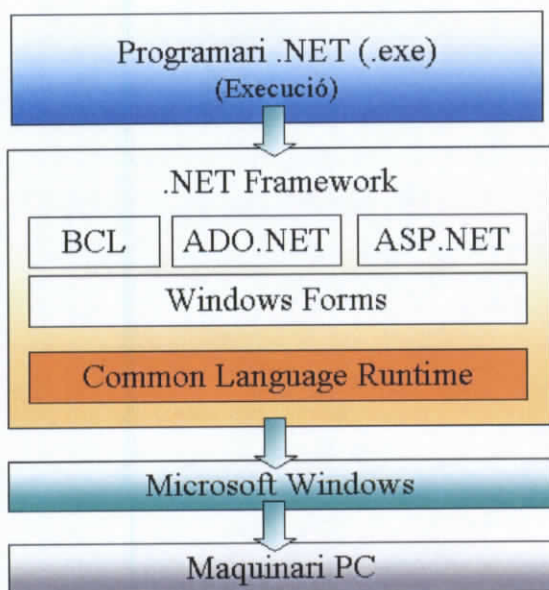


Figura 9: Esquema .NET Framework.

2.2.4. DESCRIPCIÓN DE LA ARQUITECTURA .NET FRAMEWORK.

De acuerdo al esquema que presenta la **Figura #9**, se puede diferenciar claramente los niveles presentados por el Framework de .NET.

En el nivel intermedio, se sitúa la jerarquía de clases básicas del entorno de ejecución, que constituyen un sólido API (Application Programmig Interface) de servicios a disposición del programador, para realizar múltiples tareas como, gestión del sistema de ficheros, manipulación multihebra, acceso a datos, etc.

Finalmente, en el nivel superior, se encuentran las clases que permiten el diseño del interfaz de usuario de las aplicaciones. Si se necesita desarrollar aplicaciones para Internet, se utiliza ASP.NET, la cual provee todo lo necesario para crear aplicaciones para la Red, (Web forms, Web services, etc.). Este modelo no solo se aplica para programación Web, sino también para aplicaciones de ambiente Windows.

Esta arquitectura presenta 2 nuevos ítems que son:

- Un modelo de programación consistente y sencillo completamente orientado a objetos.
- Eliminación del problema de compatibilidad entre DLLs (Dynamic Linking Librarys), conocido como Hells DLLs o "infierno de las DLLs".

2.2.5. CLR, COMMON LANGUAGE RUNTIME.

El Entorno de Ejecución Común de Lenguajes o CLR (Common Language Runtime), representa el alma de .NET Framework. Es la parte encargada de la ejecución del código de aplicaciones. Se puede considerar como un soporte que permite ejecutar los servicios y aplicaciones .NET en cualquier máquina que lo disponga.

Está basado en la idea de Java, que también tiene un módulo de ejecución independiente del sistema operativo donde se vaya a ejecutar. La gran diferencia con Java es que este lenguaje es "multilenguaje". No está limitado a un único lenguaje como .Net.

Esto permite al desarrollador utilizar una amplia variedad de lenguajes como C++, Visual Basic y C# entre otros. Algunas de las características que ofrece este

componente son:

- Proporciona un desarrollo de aplicaciones sencillo y rápido gracias a que gran parte de las funcionalidades que tradicionalmente debía crear el programador, vienen implementadas en el entorno de ejecución.
- Administra el código en tiempo de ejecución, en todo lo referente a su carga, disposición en memoria, recuperación de memoria no utilizada a través de un recolector de memoria, además de características de gestión a bajo nivel (administración de memoria, por ejemplo), que en ciertos lenguajes, eran labor del programador.
- Proporciona un sistema común de tipos para todos los lenguajes del entorno, gestionando la seguridad del código que es ejecutado. Además dispone de un diseño abierto a lenguajes y herramientas de desarrollo creadas por otros fabricantes.
- Facilita la distribución e instalación de aplicaciones, ya que en teoría, es posible instalar una aplicación simplemente copiando los ficheros que la componen en uno de los directorios del equipo en el que se vaya a

ejecutar.

Todas estas características hacen que las aplicaciones aumenten en capacidad y tiempos de respuesta al haber separado varias funciones del programador y hacerlas parte del lenguaje. Por otra parte las librerías básicas proporcionan una serie de funcionalidades que son necesarias a la hora de desarrollar tanto aplicaciones como servicios Web.

Las clases básicas gestionan operaciones como las de entrada y salida, seguridad, etc. Las Web Forms permiten desarrollar la parte gráfica de una aplicación para la Web, mientras las Windows Forms están orientadas a implementar la parte gráfica de las aplicaciones clásicas para Windows. Los compiladores producen código MSIL (Microsoft Intermediate Language).

2.2.6. IMPLEMENTACIÓN WEB CON LA ARQUITECTURA FRAMEWORK .NET. [9]

Al crearse la arquitectura Framework.NET, la empresa Microsoft pretendía imponer un cambio en la forma de desarrollar y utilizar las aplicaciones en la Red Internet. El propósito fundamental era cambiar el

concepto de aplicaciones Web a Servicios Web, y que éstas pudieran utilizarse tanto por compañías relacionadas con el negocio, como por quienes puedan requerir dichos servicios e incluso realizarlos. Todo esto se quería conseguir por medio de la creación de herramientas bajo este estándar para crear dichos servicios.

Para esto, se utilizaron una serie de tecnologías y estándares, junto a un IDE (Integrated Development Environment) unificado de desarrollo para los distintos lenguajes que forman parte de la suite de framework.NET (Visual Basic, C#, Asp.NET) con lo cual se pretendía aumentar la productividad de desarrollo. Basándose en una propuesta orientada a objetos y apuntando fuertemente en la capa middleware se podría convertir en un competidor real para soluciones de este mismo tipo, realizadas por otras tecnologías en el mercado como J2EE (Java Enterprise Edition 2).

2.2.7. ACCESO A DATOS EN MICROSOFT .NET.

Para acceder a datos en la arquitectura Microsoft .NET se debe hacer referencia a ADO .NET que es la nueva versión del modelo de objetos ADO (ActiveX Data Objects), una estrategia ofrecida por Microsoft para el acceso a datos.

ADO .NET ha sido ampliado para cubrir todas las necesidades que ADO no ofrecía, y está diseñado para trabajar con conjuntos de datos desconectados, lo que permite reducir el tráfico de red. Otra buena característica de este modelo es que utiliza XML (Lenguaje de Marcas Extensible) como formato universal de transmisión de los datos. Posee una serie de objetos que son los mismos que aparecen en la versión anterior, que permiten de mejor manera acceder a información de bases de datos, así como el objeto Connection o Command, y la introducción de nuevos objetos tales como el objeto DataReader, DataSet o DataView.

ADO .NET se puede definir como:

- Un conjunto de interfaces, clases, estructuras y enumeraciones que permiten el acceso a datos desde la plataforma .NET.
- La evolución lógica del API ADO de Microsoft a la nueva ADO.NET.
- Permite un modo de acceso desconectado a los datos, los cuales pueden provenir de múltiples fuentes de datos, de diferente arquitectura de almacenamiento.

- Soporta un completo modelo de programación y adaptación, basado en el estándar XML.

CAPÍTULO III

3. INTRODUCCIÓN A LA COMPARACIÓN DE LAS PLATAFORMAS J2EE Y .NET.

3.1. INTRODUCCIÓN.

Hasta esta parte ya se pueden diferenciar ciertos puntos sobre estas dos plataformas de desarrollo Web.

Luego de haber introducido los conceptos sobre ambas tecnologías, en ellas se hacen notar ciertas características propias, y, en ciertos casos, similitudes que hacen notar que ambas no son tan incompatibles y diferentes como se muestran en una primera etapa.

Para realizar una comparación preliminar, antes de abordar completamente el tema de la comparación dentro de las áreas de evaluación, efectuando juicios a partir de pruebas desarrolladas para demostrar que cierta plataforma puede presentar mejoras o deficiencias por sobre la otra se puede comenzar a diferenciar características con el material analizado hasta este

punto de la investigación.

En esta primera etapa se mencionarán algunos ítems de comparación que no necesitan pruebas específicas para señalar diferencias. Éstas se pueden efectuar mediante el estudio previo de las plataformas, y por medio de la información recolectada hasta esta parte.

En ella se puede notar analogías propias de diseño, desarrollo y ejecución de aplicaciones Web, arquitectura de ejecución, compañías que las apoyan e información disponible. También se puede diferenciar en una pre-etapa la complejidad en cuanto a la ejecución de códigos y la realización de procesos no vistos por el usuario.

3.2. ANÁLISIS DE LAS COMPAÑÍAS.

"Tan importante como la propia arquitectura que plantean estas dos plataformas de desarrollo, son las empresas que las apoyan, ya que contribuirán de forma decisiva en su éxito o fracaso" [48]

Independiente de lo que se opine de cada una de las empresas que patrocinan y auspician a estas dos tecnologías, es importante conocer que soporte presentan

a los usuarios y que ventajas y desventajas presentan los apoyos de determinadas empresas. Por una parte J2EE tiene el soporte de grandes empresas de software que realizan su propia implementación de éste estándar. Empresas como IBM, BEA, Oracle o la misma Sun, ofrecen productos para el desarrollo de aplicaciones en esta plataforma, que al juntarse con otros productos como bases datos, caché, y firewalls pueden entregar soluciones completas a sus clientes.

Por su parte, Microsoft, y su equipo de marketing han realizado políticas importantes para conseguir que .NET sea la plataforma de desarrollo de aplicaciones eBusiness preferida por los usuarios. Para darle apoyo, algunas empresas de renombre ya han comenzado a desarrollar soluciones usando .NET como: Accenture, Arthur Andersen, Compaq o Deutsche Bank.

Algunos estudios concuerdan que en el futuro las dos plataformas convivirán y competirán por alcanzar un mayor mercado, pero que una no reemplazará a la otra, que, realizaciones en el presente hacen entender que ambas plataformas incluso podrían fusionarse o funcionar juntas para lograr mejores objetivos.

En otro ámbito J2EE parte con la ventaja de haber llegado antes al mercado y de ser apoyada por empresas consideradas importantes las cuales ya utilizan esta tecnología.

Por otro lado Microsoft es la mayor empresa de software que con estrategias de reducción de precios y de marketing ha conseguido llegar a una importante masa de empresas y desarrolladores, los cuales están conformes con los resultados obtenidos con esta plataforma de desarrollo.

Muchos apoyan, como norma general, que J2EE se impone en las grandes compañías mientras que Microsoft lo hace en las pequeñas y medianas empresas las cuales prefieren pagar licencias en términos de productos compatibles con sus sistemas operativos Windows utilizados, teniendo plena seguridad en el funcionamiento de estos.

Característica	.NET	Java 2EE
Tipo de Tecnología	<i>Producto</i>	<i>Estándar</i>
Empresas que lo ofrecen	<i>Microsoft</i>	<i>Más de 30</i>
Librería de Desarrollo	<i>.NET Framework SDK</i>	<i>Java core API</i>
Interprete	<i>CLR</i>	<i>JRE</i>
Páginas Dinámicas	<i>ASP.NET</i>	<i>Servlets, JSP</i>
Componentes	<i>.NET Framework SDK</i>	<i>JRE</i>
Acceso a Base de Datos	<i>ADO.NET</i>	<i>JDBC, SQL/J</i>
Servicios WEB	<i>SOAP, WDSL, UDDI</i>	<i>SOAP, WDSL, UDDI</i>
Interfaces Gráficas	<i>WIN Forms y WEB Forms</i>	<i>Java Swing</i>
Herramientas de Programación	<i>Visual Studio.NET</i>	<i>Depende del Fabricante</i>
Transacciones Distribuidas	<i>MS-DTC</i>	<i>JTC</i>
Servicios de Directorios	<i>ADSI</i>	<i>JNDI</i>
Librería de Mensajes	<i>MSMQ</i>	<i>JMS 1.0</i>
Lenguajes Utilizados	<i>C#, Visual Basic, C</i>	<i>JAVA</i>
Lenguaje Intermedio	<i>IL</i>	<i>Bytecodes</i>

Tabla 1: Tabla Comparativa Primeras Comparaciones.

3.3. LIMITACIÓN DE LA COMPARACIÓN WEB.

Para poder determinar qué aspectos de amplio rango de aplicaciones Web ofrecidas por las arquitecturas J2EE y Microsoft.NET se deben considerar, y qué aspectos y funcionalidades comunes dentro de Internet se manifiestan en la utilización de páginas de tipo dinámicas. Ambas tecnologías presentan su tecnología para este desarrollo, JSP por J2EE y ASP con los nuevos ASPx de .NET, esto da condiciones que en este punto se pueden señalar como similitudes entre estos dos tipos de aplicaciones y los distintos requerimientos necesarios para su correcta ejecución.

A pesar de que ambas plataformas presentan un número considerado de posibilidades de desarrollo WEB, las que mejor representan el desarrollo de las aplicaciones de este tipo son aquellas con las cuales el usuario interactúa directamente al navegar por Internet.

Aspectos como el tipo de componentes de ejecución utilizado, las librerías incluidas o las analogías y diferencias que pueden presentar las arquitecturas J2EE con su máquina virtual y .NET con .NET Framework difieren de una comparación dentro de un ámbito Web señalando otros aspectos de estas tecnologías.

Por su parte aspectos como el dinamismo, el tipo de conexión utilizada para acceder a bases de datos, la utilización de Servidores Web para el despliegue de las aplicaciones y, aspectos comparativos relevantes como el rendimiento y escalabilidad, apuntan directamente a una línea de desarrollo Web de páginas dinámicas.

De esta forma aspectos ya definidos como las aplicaciones de índole Web de J2EE y la descripción de componentes de la arquitectura .NET quedan establecidos para poder emitir juicios posteriores dentro de la generalidad de las plataformas en comparación para, de esta forma dar paso a

una comparación directa entre páginas JSP con ASPx.

3.4. PÁGINAS DINÁMICAS EN J2EE.

La tecnología J2EE para la creación de páginas Web desarrollada en el lenguaje Java, es conocida como páginas JSP (Páginas de Servidor Java o Java Server Pages), éstas poseen la capacidad de ejecutarse en variados servidores Web y en múltiples plataformas, su sintaxis está compuesta de códigos HTML/XML mezclados con etiquetas especiales que generan estas páginas.

Éstas son desarrolladas de manera similar a como se crean las páginas ASP o PHP, las cuales se guardan como archivos de extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias JAVA a ejecutar en el servidor.

Las páginas JSP son ejecutadas por un motor basado en los servlets de Java (programas en Java ejecutados en el servidor) y antes de que sean funcionales, el motor JSP lleva a cabo una fase de traducción de la página a un servlet, implementado en un archivo class (Byte codes de Java) que es llevada a cabo habitualmente cuando se recibe la primera solicitud de la página .jsp, también existe la

opción de precompilar el código para evitar estos tiempos de espera al acceder por primera vez un cliente a la página JSP solicitada.

```

<%@ page language="java" %>
<html>
<head>
<title>Bienvenida</title>
</head>
<body>
<jsp:usebean id="clock" class="calendar.jspCalendar">
  <h1>Tiempo Inicio: <%= new java.util.Date() %></h1>
  <%
    String mensaje="Bienvenido ";
    out.println("<font color=blue size=7 >"+
      mensaje + " </font><br>");
  %>
</body>
</html>

```

Tabla 2: Ejemplo JSP

3.4.1. REQUISITOS DE JSP.

Para poder desarrollar páginas JSP se debe tener conocimientos sobre páginas Web HTML, las cuales en sí, son la base de cualquier sitio de este tipo, además será necesario comprender la programación en JAVA, lo que brindará un mejor manejo de conceptos de programación y funcionamiento de éste tipo de páginas Web dinámicas.

Además se necesitará descargar e instalar Tomcat, el contenedor de servlets usado en la referencia oficial de implementación de JSP. Esto referencia el aprendizaje de otra área dentro de la programación de páginas dinámicas de JAVA, la configuración del Servidor Web Tomcat, para

así poder acceder directamente desde el equipo, como se estuviese conectado a la Internet.

Este Servidor Web es uno de los más potentes y reconocidos servidores gratuitos de Servlet y JSP del mercado. Pertenece a la familia de Apache (otro servidor Web) y es parte del Proyecto Apache Jakarta Project. Éste puede ser instalado externamente, como un paquete o servicio dentro del Sistema Operativo, como también de forma interna al ejecutarse dentro del IDE de desarrollo de aplicaciones Web en JAVA.

La instalación y configuración varía dependiendo del Sistema Operativo que se utilice, ya que éste se desempeña correctamente en sistemas Linux a base de comandos y, para sistemas Windows la instalación viene siendo simple con respecto a los servicios que se quieren activar. En la instalación del motor de programación Java, Netbeans, se incluyen todos los servicios necesarios para poder levantar y ejecutar páginas JSP y que éstas puedan funcionar correctamente.

3.5. PÁGINAS DINÁMICAS EN .NET.

Para el desarrollo Web Microsoft .NET está diseñado para

la construcción de estos sistemas, manejados con la extensión .ASPx (Active Server Pages x), que permiten desarrollar páginas Web con contenido dinámico con la funcionalidad .NET.

Anteriormente, aplicaciones como ActiveX y las Web Classes de Visual Basic fueron la base de llevar este tipo de programación a Internet, pero su rendimiento en ejecución y complejidad en algunos casos, distaban mucho de ser la solución idónea a este problema.

Al crear aplicaciones Web con programación de páginas ASPx, junto a componentes de conexión con los distintos motores de bases de datos, hacen a las páginas ASPx junto con ADO, una solución óptima para este tipo de aplicación.

3.5.1. WEB FORMS.

Con Visual Studio.NET, la creación de Páginas ASPx, se realizan a través de Web Forms o tipo de páginas de servidor activas (ASPX, Active Server PagesX).

Estos permiten crear páginas Web programables como parte de una aplicación Web global, lo que simplifica el

desarrollo de grandes aplicaciones de este tipo, éstas proporcionan un modelo de programación basado en eventos en el servidor. Los formularios están basados en herramientas de desarrollo de Microsoft Win32 que entre sus ventajas se permite una completa separación entre el formato HTML y la lógica de la aplicación (.NET).

La lógica, o código asociado a la página, se compila y proporciona un rendimiento mucho mejor, definiendo áreas de diseño y especificando algún tipo de error si éste se encontrase. Además, este código se puede escribir en cualquier lenguaje de Microsoft.NET, por lo que permite al usuario aprovechar mejor sus habilidades.

En el área de diseño los Web Forms proporcionan una interfaz de desarrollo rápido de aplicaciones (RAD, rapid application development) para crear y administrar éstos formularios Web, los cuales son compatibles con un conjunto de controles y componentes .NET que ofrecen un modelo coherente de interconexión con cada una de estas herramientas obteniendo resultados seguros. **[38]**

Además, el marco de trabajo se presta de manera natural de mayor accesibilidad, mediante las herramientas desarrolladas por otros fabricantes.

Existen otros aspectos igualmente importantes de los Web Forms, tales como la caché, seguridad, configuración y personalización. Estos controles del servidor se usan para crear la interfaz del usuario y así generar cualquier salida adecuada para el dispositivo o explorador al que se dirigen.

Hay dos conjuntos de controles de servidor en el marco de trabajo ASPX: los controles HTML y los controles Web, que procesan el HTML que reciben los exploradores Web. Además del procesamiento, estos controles encapsulan el mecanismo para conservar el estado durante los recorridos completos y activar eventos del servidor para varios eventos cliente. Todas estas ventajas ofrecen a desarrolladores que no estén familiarizados con la creación de sitios Web poder desarrollar dichas aplicaciones.

```
<%@ Page Language="vb" AutoEventWireup="false"
Codebehind="inicio.aspx.vb" Inherits="AspxPostgres.inicio"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <meta name="GENERATOR" content="Microsoft Visual Studio.NET 7.0">
    <meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
  </HEAD>
  <body MS_POSITIONING="GridLayout" bgColor="#ccff00">
    <asp:label id="Label1"> Página Inicio </asp:label>
  </body>
</HTML>
```

Tabla 3: Ejemplo ASPx

3.6. PROGRAMAS DE DESARROLLO PARA LAS PLATAFORMAS

J2EE Y .NET.

Para comenzar a programar en ambas plataformas, se debe considerar que IDE (Integrated Development Environment) de desarrollo se utilizarán y saber el porqué de esta elección. Al ya limitar el universo de posibilidades de arquitecturas de desarrollo Web, y enfocarse solo en la propuestas por Sun Microsystems y Microsoft. Se debe conocer que IDE's, o programas de desarrollo, definen las funcionalidades y procedimientos necesarios para crear aplicaciones bajo los estándar de J2EE y .NET.

En el mercado existen varias alternativas para el desarrollo de aplicaciones de estas plataformas. En el caso de J2EE, varían desde poderosos entornos operativos para la creación de portales Web, a las más sencillas y básicas con las cuales se pueden crear sitios propios.

También el uso de licencias esta denotado bajo esta tecnología, desde soluciones libre de pago hasta IDE's comerciales de precios elevados. En el ámbito de .NET, Microsoft, aunque partió con un único lenguaje-software de desarrollo, Visual Studio.NET, de a poco el mercado ha ido lanzando nuevos productos los cuales se pueden utilizar

para el desarrollo de aplicaciones de alto nivel con esta plataforma.

Ahora se analizarán los principales IDE's de desarrollo sobre estas tecnologías, denotando las mayores características que éstos presentan para posteriormente poder decidir cuál de ellos presenta lo necesario para el desarrollo de aplicaciones Web y demuestra el poderío de las plataformas de desarrollo evaluadas.

3.6.1. PRINCIPALES IDE'S DE DESARROLLO DE J2EE.

Al querer nombrar o especificar IDE's de Desarrollo o programas para la ejecución de programas JAVA con la plataforma J2EE, se pueden mencionar una variada gama de productos, éstos, al presentar la principal diferencia, el tema de licencias, se separan en dos grandes áreas: los IDE's de licencia libre u open source y los IDEs comerciales.

En estas áreas se encuentran distintos programas de desarrollo que destacan la compatibilidad necesaria para no encontrarse con problemas de desarrollo, al realizar aplicaciones en un IDE y luego querer llevar estos códigos a ejecutarse en otro.

Producto	Sitio	Licencia	Plataforma
Eclipse (IBM)	http://www.eclipse.org	Opensource	Windows, Linux, OSX
JBuilder (Borland)	http://www.borland.com/jbuilder	La versión de Evaluación y la Personal son gratis, las avanzadas, Profesional y Enterprise son comerciales.	Windows, Linux, Solaris
Sun ONE Studio (Sun Microsystems)	http://www.sun.com/forte/ffj/	Esta basado en el IDE opensource Netbeans pero las versiones existentes son todas comerciales.	Todas con JVM.
Netbeans (SunMicrosystem)	http://www.NETbeans.org	Opensource	Todas con JVM.
Rational Application Developer (IBM)	http://www-306.ibm.com/software/awdtools/developer/application	Basado en el framework de Eclipse 3.0.	Windows, Linux.
WebSphere Developer Application (IBM)	http://www-306.ibm.com/software/awdtools/studioappdev/	Basado en el SDK versión 3.0 de Eclipse. Desarrollo de WebServices.	Windows, Linux.

Tabla 4: Tabla Explicativa IDE's de Desarrollo J2EE.

3.6.2. NETBEANS

NetBeans es el programa de desarrollo para aplicaciones Java de licencia libre más utilizado por los usuarios. Este cuenta con las funcionalidades necesarias para el desarrollo de todos los tipos soportados bajo este estándar. [6]

Este IDE de desarrollo creado por Sun, el cual mantiene una licencia Open Source sobre Netbeans, ofrece ventajas extras sobre otros IDE's y llega a ser considerado el mejor editor de programas JAVA. Este software consta por el momento de dos partes, el IDE (entorno de desarrollo)

NetBeans, y NetBeans Tools Platform (Plataforma de herramientas). Además de lo gratuito de la herramienta, esta posee un poderoso trasfondo de ejecución que permite ejecutar sin problemas aplicaciones de escritorio con la máquina virtual de Java y en aplicaciones Web, el servidor JSP Tomcat.

A pesar de que la principal característica del IDE de desarrollo es su gratuidad, Sun planea empaquetar algunos componentes con una serie de módulos propios y un contrato de mantenimiento para poder convertirlo en ediciones de lujo por la cual se pagaría una licencia. De este modo Sun pagaría los recursos utilizados para que continuase el desarrollo Open Source, para que usuarios e ingenieros, puedan obtener resultados esperados en este entorno de desarrollo verdaderamente potente.

Con el transcurso del tiempo, Netbeans ha ido mejorando e incorporando nuevas utilidades que lo han hecho un IDE más completo, el hecho de que desde la versión 3.2 pueda funcionar en Mac OS X, ha hecho que no solo los usuarios de PC posean esta arquitectura y ha ampliando notablemente el desarrollo de aplicaciones JAVA.

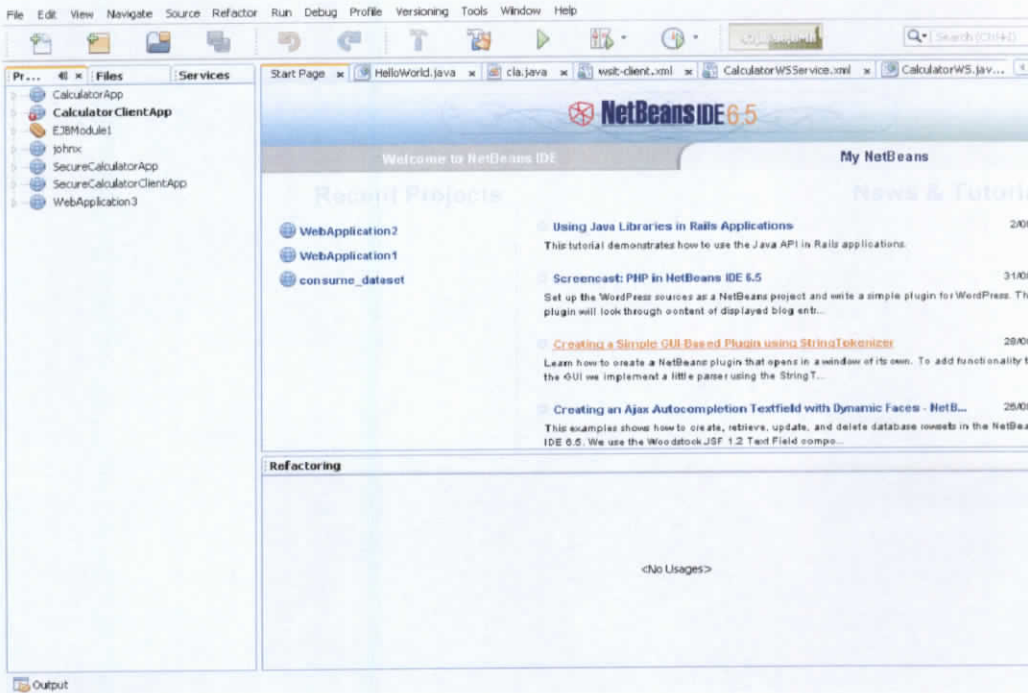


Figura # 10 Interfaz NetBeans IDE

3.7. DESARROLLO EN MICROSOFT .NET CON VISUAL STUDIO.

La empresa Microsoft al crear la plataforma de desarrollo .NET, creó toda una nueva gama de productos y nuevas versiones para su conjunto de herramientas de desarrollo de software, naturalmente orientadas a .NET Framework su nuevo entorno de programación.

Si bien es posible la escritura de programas empleando sólo el SDK de .NET Framework, este último, al estar compuesto de herramientas independientes, constituye un medio más incómodo de trabajo.

Visual Studio .NET fue creado como un entorno de desarrollo integrado (IDE) el cual contiene todas las herramientas del SDK: compiladores, editores, ayuda, etc., facilitando en gran medida la creación de programas, tanto aplicaciones de escritorio, como aplicaciones Web.

Este entorno de desarrollo contiene nuevas funcionalidades que facilitan la creación de sitios Web con contenidos programables de manera importante. Presentando herramientas gráficas, las cuales pueden desplegarse para su utilización de manera modular, lo cual representa una ventaja a la hora de compararlo con otros IDE's del mercado.

Además este editor trabaja directamente con las capas del modelo .NET Framework, la Base Class Framework que contiene las clases necesarias para generar los aspectos básicos de programación, que se interpretan desde el lenguaje CLR, el lenguaje básico de este modelo de programación.

3.8. ELECCIÓN DE IDE'S DE DESARROLLO PARA LAS PLATAFORMAS J2EE Y .NET

Después de analizar los principales IDE's de desarrollo

para las plataformas Microsoft .NET y J2EE, conocer sus funcionalidades, capacidades y herramientas, se debe empezar a definir que IDE's se utilizarán para comparar ambas plataformas y, posteriormente en una etapa de pruebas, desarrollar módulos de comparación para la evaluación de éstas.

Para esto se deben analizar las principales características de las respectivas plataformas de desarrollo, cerciorarse de que las IDE's a utilizar son fieles a conceptos y líneas de desarrollo de estas arquitecturas, darse cuenta de que en estas se puede demostrar el poderío de programas desarrollados en dichas tecnologías y, poder ejecutar sin errores, aplicaciones en un ambiente Web para cada una de éstas.

Por esto antes de emitir juicios sobre que IDE's son mejores para cada tecnología, se deben considerar las características principales de J2EE y .NET. Para el caso de J2EE en base a estudios y entrevistas de la razón principal de su utilización, se marcó el hecho de: "La disponibilidad de libre licencia" para algunos de sus IDE's de desarrollo.

Además, JAVA, por característica propia no está sujeta a

pago de licencias. Esto hace que este atributo sea de importancia en la elección del software de desarrollo para aplicaciones en J2EE. Por su parte, Microsoft .NET, por los estudios previos y consulta hacia personas que utilizan esta tecnología, se menciona como su principal característica: "La facilidad de creación de aplicaciones WEB", y en un mismo nivel, refiriéndose a Visual Studio.NET como un potente y fácil programa de desarrollo de este tipo de aplicaciones. [17]

3.8.1. ELECCIÓN DE IDE DE DESARROLLO PARA J2EE.

Al basarse en las características definidas para la elección del IDE de desarrollo para la plataforma J2EE, el aspecto de la licencia libre, limitó el conjunto de posibilidades, quedando solo dos IDE's de desarrollo con las características necesarias: Netbeans y Eclipse.

En la elección de Netbeans destacó por sobre otros programas de desarrollo por su característica de open source, principal atributo de JAVA, desechando las alternativas comerciales como se comentó con anterioridad.

Para determinar qué IDE utilizar para la plataforma J2EE, éste debería presentar las siguientes características:

IDE de Licencia Libre y Fácil utilización, aspectos considerados por Netbeans.

Además éste maneja el conjunto de API's y proceso de servlets para la ejecución de software Web.

3.8.2. ELECCIÓN DE IDE DE DESARROLLO PARA .NET.

Para poder decidir qué Plataforma de Diseño utilizar para el ambiente .NET predominó el hecho de que la mayoría de los criterios a considerar concluían en el producto, Visual Studio .NET, pues éste presenta toda la funcionalidad ofrecida por la arquitectura Framework.NET. Dejando de lado el tema relacionado con el manejo de licencias y costos, Visual Studio.NET, presenta características propias, y, junto a la Tecnología .NET, proporciona todo lo necesario para el desarrollo bajo esta arquitectura.

Visual Studio se presenta como la mejor herramienta de diseño de aplicaciones .NET por funcionalidades propias del lenguaje, facilidad del manejo de herramientas, un editor gráfico de aplicaciones, fácil instalación y ejecución de componentes que lo hace el IDE de Desarrollo para Microsoft.NET.

3.8.3. PRINCIPALES CARACTERÍSTICAS DE IDE'S SELECCIONADOS.

En esta sección se señalarán algunas características entre ambos programas de desarrollo, NetBeans y Visual Studio. NET.

Característica	NetBeans	Visual Studio .NET
<i>Manejo de Licencia</i>	<i>Licencia Libre</i>	<i>Comercial</i>
<i>Componentes de funcionamiento</i>	<i>JVM (Maquina Virtual)</i>	<i>Ninguno.</i>
<i>Orientación de Desarrollo.</i>	<i>Específico para Aplicaciones Java.</i>	<i>Tanto Web con Aplicación Win32.</i>
<i>Componentes de Funcionamiento Web</i>	<i>Tomcat</i>	<i>IIS</i>
<i>Duración de Instalación</i>	<i>Más menos 15 min.</i>	<i>Más menos 1 hr 40 min.</i>
<i>Tamaño del IDE</i>	<i>90 MB (más JDK)</i>	<i>1.751 MB</i>

Tabla 5: Tabla Comparativa Principales Características

3.9. ÁREAS DE EVALUACIÓN DE LAS PLATAFORMAS J2EE Y .NET.

Para poder determinar qué aspectos se necesitan desarrollar en una comparación de tecnologías en el ámbito Web, se debe describir qué ítems son importantes

evaluar para poder realizar pruebas que determinen superioridades e inferioridades entre ambas plataformas.

Para esto se definen ciertas áreas de evaluación, las cuales contienen algunos criterios que representan los aspectos comparativos más importantes que puedan ser la base de pruebas posteriores que se podrán evaluar a las plataformas J2EE y .NET en el ámbito WEB.

La importancia de este punto radica en condicionar qué áreas del diseño, desarrollo y posterior ejecución de pruebas y aplicaciones, permiten sacar conclusiones sobre ambas tecnologías determinando grados de efectividad, debilidades y aciertos para cada aspecto. Para esto se consultó manuales y sitios relacionados con el tema, los cuales guíasen en las áreas relevantes de comparación en particular métodos de evaluación de software.

3.9.1. DEFINICIÓN DE ÁREAS DE EVALUACIÓN.

3.9.1.1. APRENDIZAJE

Al empezar a definir las áreas de evaluación se considerará al concepto Aprendizaje, a la capacidad que tiene el usuario común de acceder a la tecnología, la

facilidad hacia el conocimiento y posterior capacitación en una de estas plataformas, qué medios se utilizan para llegar a ésta, qué información debe ser considerada y cuántos profesionales están dedicados a esto. Con esto se quiere determinar qué tecnología puede ser más "fácil" y simple de abordar en una primera línea o etapa, qué podría denominarse de conocimiento previo o bien aprendizaje.

Dentro de la investigación recopilada en etapas anteriores, muchos documentos se referían a este concepto de forma personal, midiendo variables y evaluando productos por separado, refiriéndose a la plataforma de comparación opuesta de forma negativa, lo que no llegaba a algún tipo de conclusión o aspecto factible a considerar. Por esto y más dentro de esta área del documento y particularmente en el área de evaluación Aprendizaje, plantea en simples conceptos algunas características básicas a considerar para poder emitir juicios con respecto a un tema tan subjetivo como lo es el aprendizaje.

Esto involucra los siguientes criterios a considerar:

Criterios de Aprendizaje: Información Disponible.

Información Disponible en Español.

Calidad de la Información.

Curva de Aprendizaje.

3.9.1.2. RENDIMIENTO

Al referirse al área de evaluación Rendimiento, ésta se refiere al conjunto de necesidades para el manejo de hardware por la tecnología. Qué costos involucran el relacionarse con una de estas plataformas y, si el equipo computacional utilizado por el usuario, está capacitado para albergar dicha tecnología de desarrollo.

Con esto se desea considerar, hasta qué punto el Hardware limita el accionar de las plataformas, tanto como prerrequisitos de instalación del software, que permita realizar aplicaciones Web propias, como también el costo de recursos que se utiliza al ejecutar dichas aplicaciones.

A su vez en esta etapa, se debe dejar descrito, cuáles aspectos propios del rendimiento deben considerarse y cuáles no, qué aspectos externos a las plataformas de

desarrollo deberían considerarse y no ser tomados a la ligera, o bien en qué se basan las empresas para poder definir los requerimientos mínimos de usabilidad para la instalación de productos que realicen software y aplicaciones Web bajo los estándares J2EE y .NET. Este criterio será de vital importancia en futuras etapas de pruebas y podrá ser clave en la decisión de elegir una tecnología por sobre la otra.

Para esto se consideran los siguientes criterios:

Criterios de Rendimiento:

Uso de Recursos.

- Capacidad de Respuesta.
- Requerimientos Mínimos.
- Componentes Necesarios.
- Escalabilidad.

3.9.1.3. EXPLOTACIÓN Y USO

Con Explotación y Uso, se referirá a todos los componentes externos a la plataforma, que tan utilizados son estos estándar, que empresas los utilizan y qué sistemas conocidos son creados bajo estas arquitecturas.

Mediante los criterios definidos para esta área de evaluación se quiere dejar constancia, por ejemplo, sobre qué porcentaje de Internet está constituido por páginas ASP o Aspx, frente a JSP. Esto dará un indicador sobre qué tanto se pueden encontrar "fuera", en la Internet, este tipo de aplicaciones y si mediante la construcción de un número mayor de aplicaciones Web a través de estas plataformas de desarrollo, se pueden elaborar modelos de diseño y desarrollo para futuros proyectos de este mismo tipo.

La importancia de esta área de evaluación radica en el hecho de encontrar si la plataforma que se debe dedicar el usuario es utilizada por otros, y si ésta ofrece un campo tanto laboral, como de conocimiento, las cuales permitan una continua capacitación sobre la plataforma.

También el poder determinar por ejemplo una gráfica sobre desarrollo de este tipo de tecnologías Web, tanto en un medio global como el total de Internet, o de manera local refiriéndose solo a los dominios .cl, podrían dar las bases sobre en qué nivel de aplicación Web (Asp, Jsp, php, etc.)

Para esto, se consideraron los siguientes criterios.

Criterios de Explotación y Uso:

- Curva de Desarrollo.
- Uso de la Tecnología.
- Distribución.

3.9.1.4. DESARROLLO

El área de evaluación de Desarrollo hace alusión al usuario frente al editor de programas o IDE de desarrollo de la arquitectura establecida.

En esta área se quiere dejar en claro que tan importante hace a una tecnología como .NET o J2EE un buen IDE. Si su aspecto genera mayor confianza, accesibilidad, un mejor entorno operativo que haga que el desarrollar aplicaciones de cualquier tipo no sea un proceso engorroso, y si el software permite realizar conforme se crean dichas páginas, un aprendizaje frente a errores y/o el mejor manejo de las herramientas de diseño y ejecución propuestas por el IDE de desarrollo.

Esta área describe aspectos que pueden considerarse subjetivos, pero al ser tratados desde un punto de vista crítico, basado en funcionalidades mínimas y un aspecto

atractivo que incentive el hecho de desarrollar aplicaciones en el respectivo IDE, hacen que el componente personal no sea considerado, ya que ésta área de evaluación, al igual que el Aprendizaje contiene muchos aspectos que son considerados personales y poco objetivos al querer dejar en claro alguna característica en cuanto a la ejecución de aplicaciones que pueda demostrar alguna ventaja a considerar que pueda ser totalmente imparcial.

Para esto se concibieron los siguientes criterios a considerar los criterios de desarrollo:

- Interfaz de usuario.
- Accesibilidad.

3.9.1.5. USO DE LICENCIAS.

Dentro del Área de Uso de Licencias, se consideran los aspectos que involucran el acceder a los distintos IDE's de desarrollo de ambas plataformas. El cómo se presentan al usuario, el ver con qué fines se manejan estos programas y, si se considera el hecho de desarrollo comercial y el académico. Aspecto importante y limitante al enfrentarse a una elección de productos de este tipo o

la adquisición de los mismos.

Para esto, se cuentan con las respectivas licencias de cada arquitectura y sus IDE's de desarrollo, lo que permitiría a grandes rasgos saber a qué tipo de producto el usuario se está enfrentando y qué consecuencia podría tener la posterior evaluación de desarrollo comercial a través de estas herramientas.

Otro aspecto a considerar en esta área de evaluación, es el manejo de los códigos generados por cada aplicación Web, si su accesibilidad está permitida en todo momento y si la generación de archivos sin uso es clara, o no se está al tanto de qué función realiza dentro de la aplicación general.

Otro aspecto importante dentro de esta área de evaluación, es el delicado tema de costos sobre las licencias y las adquisiciones de productos, IDE's y todo lo relacionado con ambas plataformas de desarrollo. Si dentro del ámbito comercial que puedan presentar cada una de estas tecnologías, se considera la realización de distintos paquetes y precios, dependiendo del nivel del usuario que quiere comenzar a desarrollar aplicaciones propias en estos estándares. Para esto se definieron los

siguientes criterios uso de licencias:

- Licencia Libre
- Código Abierto Costos.

3.9.1.6. COMPATIBILIDAD

Al realizar un área de evaluación relacionada con la Compatibilidad, se definen criterios como: el "dónde" poder ejecutar este tipo de aplicaciones, desde el sistema operativo soportado hasta el servidor Web que permita el correcto despliegue de la aplicación en algún tipo Browser o Navegador Web.

La compatibilidad tiene que ver con la interacción entre los distintos componentes que hacen posible la ejecución del código durante las diferentes capas de ejecución. La relación entre el Sistema Operativo y el lenguaje de máquina que interpreta el respectivo lenguaje, hacen que la compatibilidad sea uno de los puntos más discordantes entre estas dos tecnologías, ya que no todas cuentan con lo necesario para este tipo de funcionamiento.

Esto genera conflictos entre ambos estándares de desarrollo que buscan vulnerabilidades sobre la otra para

encontrar ventajas. También considerar aspectos, por ejemplo la portabilidad, el poder analizar y determinar hasta qué punto son positivos y realmente provechosos a los usuarios este tipo de características.

Para esto se definieron los siguientes criterios de comparación: Criterios de Compatibilidad:

- Portabilidad
- Navegadores Web.
- Servidores Web.

3.10. DEFINICIÓN DE CRITERIOS.

3.10.1. DEFINICIÓN DE CRITERIOS DE APRENDIZAJE.

Para el Área de Evaluación 1: Aprendizaje se definieron los siguientes criterios los cuales encierran la idea principal de cómo el usuario accede a cada tecnología y tiene la capacidad de informarse sobre éstas. Para esto se describe:

Información Disponible: se refiere a la disposición de información y conocimiento que se puede conseguir por medios asequibles, como Internet, libros y revistas

especializadas. Implícitamente en la Web, cuantos sitios manejan información del tema, cursos, manuales y bugs, que permitan una mejor y más rápida integración a la plataforma. Además se puede considerar qué tipo de demanda tiene en Internet.

Información Disponible en Español: se refiere al poder conseguir información en "nuestro" idioma. Con información relacionada sobre cada plataforma, ¿Cuánta de ésta se encuentra disponible en el idioma español?, pues al encontrarse en el idioma del usuario, el proceso de aprendizaje se acelera y se torna ágil, al no tener que dedicar tiempo a traducir, sino solo a aprender.

Curva de Aprendizaje: se refiere a la asimilación entre las variables de tiempo y aprendizaje con respecto al desarrollo en la plataforma seleccionada. El comparar estas variables gráficamente demostrará una curva más elevada y corta para la plataforma con mayor facilidad de instrucción.

3.10.2. DEFINICIÓN DE CRITERIOS DE RENDIMIENTO.

Para el Área de Evaluación 2: Rendimiento, se definieron criterios que especificarán la utilización de hardware,

la necesidad de componentes externos que limiten de alguna forma el acceso a las aplicaciones Web desarrolladas. Estos se dividieron en Rendimiento Independiente y Rendimiento Base de Datos.

3.10.2.1. RENDIMIENTO INDEPENDIENTE.

Uso de Recursos: se refiere a la utilización y consumo de componentes de hardware como lo son Tiempo Cpu o también considerado tiempo de ejecución, y el espacio de disco duro utilizado por la aplicación y la cantidad de archivos generados.

Capacidad de Respuesta: se refiere al criterio encargado de considerar la duración sobre una petición y la obtención de resultados de dicha operación, como por ejemplo, el presionar un botón y que este ejecute la acción, o la búsqueda de un dato sobre un servidor de base de datos vía Web. Esto brindará una orientación sobre tipo de "página" o aplicación es más ágil en tiempos de despliegue y de actualización.

Requerimientos Mínimos: Para este criterio se consideraron aspectos mencionados en capítulos anteriores

al referirse independientemente a cada Plataforma y a sus productos de desarrollo. En éstos se puede dar cuenta de los requerimientos operativos que necesitan para poderse ejecutar, los requerimientos de hardware para los IDE's de desarrollo de ambas plataformas, y así poder demostrar qué tecnología necesita un mayor avance tecnológico para poder ejecutar correctamente sus aplicaciones.

Escalabilidad: Este criterio hace referencia sobre un sitio Web determinado, tanto ASPx para la tecnología .NET y JSP en J2EE, tenga la capacidad de funcionar correctamente en función del número de usuarios simultáneos que lo utilizan. Dicho de otro modo, la propiedad de acceso a múltiples usuarios a una misma aplicación, manteniendo la misma funcionalidad sin presentar mayores dificultades.

Rendimiento Base de Datos:

- Uso de Recursos,
- Capacidad de Respuesta,
- Requerimientos Mínimos, Escalabilidad.

(Estos criterios hacen referencia a los mismos definidos para rendimiento independiente, ahora con la salvedad de

ser orientados al trabajo y desempeño de aplicaciones Web con Base de Datos, tanto de forma local y remota.)

Componentes Necesarios: Este criterio hace referencia a todo componente, elemento o herramienta ajena al IDE de desarrollo que permita trabajar correctamente con aplicaciones Web con operaciones de bases de datos. Este criterio permitirá ver qué IDE está mejor preparado para el desarrollo de aplicaciones de este tipo.

3.10.3. DEFINICIÓN DE CRITERIOS DE EXPLOTACIÓN Y USO.

Dentro del Área de Evaluación Explotación, se definieron criterios que explicarán de buena manera la relevancia y utilización de estas plataformas dentro del ambiente de Internet. Para estos criterios se hizo una investigación en la cual se quería reflejar la utilización de las páginas web ASPx y JSP en cuanto a cantidad y calidad, las empresas realizan sus sitios bajo estos estándares, en claro apoyo a una de estas arquitecturas, también se consideró el navegar por distintos sitios en Internet y si estos reflejaban qué tipo de aplicación Web utilizaban.

Además dentro de esta área se pudo tener acceso a un documento el cual contrastará con el estudio realizado en este documento, para así poder tener más de una referencia sobre algunos de los temas tratados en éste.

Los criterios descritos para esta área de evaluación son los siguientes:

Curva de Explotación: para este criterio se considerará el cotejo entre el desarrollo de la tecnología en función del tiempo, si a pesar del tiempo una tecnología se mantiene vigente o si la utilización de ésta decrece, considerando la evolución a una tecnología de mejor rendimiento. Por otro lado, el incremento del desarrollo de aplicaciones Web en una de estas plataformas, se determinará a través de distintos sitios y aplicaciones creadas bajo estos estándares.

Uso de la Tecnología: este criterio hace referencia a la cantidad numérica de sitios desarrollados bajo estas plataformas. En el simple hecho de navegar por Internet el usuario se encontrará con páginas con distinta orientación, información, entretenimiento, de foros, etc., las cuales poco tienen que ver directamente con conceptos ASPx o JSP, pero si determinará el hecho de

saber en qué tecnología están desarrolladas.

Distribución: Bajo este criterio se encuentran las empresas que albergan sus sitios, tanto corporativos, de información y sistemas Web bajo estas tecnologías.

Considerar más que la cantidad de sitios de este tipo, ver la calidad y el nivel de importancia de la empresa o institución que apoya a estas plataformas con la inclusión de implementaciones propias bajo estas tecnologías.

3.10.4. DEFINICIÓN DE CRITERIOS DE DESARROLLO.

Para el Área de Evaluación de Desarrollo se definieron criterios directamente relacionados sobre la programación de sitios Web. Además la visualización de estos sitios sobre un Browser, definiendo la facilidad de los distintos IDE's para la realización de páginas dinámicas.

Para esto se definen los criterios descritos a continuación.

Interfaz de usuario: Este criterio hace directa referencia a la apariencia presentada por el IDE de desarrollo, tanto desde un punto de vista de atractivo,

como de funcionabilidad.

Los aspectos que se fueron: un grato ambiente para desarrollar, interfaces y apariencias llamativas y congruentes, que no lleven a error, con utilización de objetos (menús, iconos) que no hagan a la programación una tediosa escritura de códigos.

Accesibilidad: esto se refiere a la posibilidad que ofrece tanto la plataforma como los mismos IDE's de desarrollo de ofrecer funciones, procesos y atributos de forma acertada a la hora de programar. Una página o sitio Web es asequible cuando está diseñada para que sus contenidos estén disponibles a los usuarios.

3.10.5. DEFINICIÓN DE CRITERIOS DE USO DE LICENCIAS.

Para el Área de Evaluación, Uso de Licencias, se describieron criterios que señalarán facilidades en cuanto al código y a las licencias ofrecidas por ambas plataformas. Definir qué posibilidades hay de acceder a herramientas de libre costo, y si la creación de aplicaciones bajo estas plataformas genera algún tipo de incongruencia al generar licencias personales.

Esto se describe a continuación:

Licencia Libre: Este criterio se refiere a la facilidad proporcionada por la plataforma al acceder a sus productos, la consideración de versiones independientes del uso, pudiendo separar en comerciales y de conocimiento o aprendizaje.

Este criterio muestra la orientación de cada plataforma y si su objetivo es solo comercial o si se considera el ámbito educacional. También se define el hecho que, al desarrollar productos en una plataforma con facilidad de licencias, no se depende de "nadie" para realizar aplicaciones propias.

Código Abierto: Este criterio hace referencia a la permisividad de acceder a los códigos generados en el proceso de creación de una aplicación Web. Si éstos contienen algún tipo de encapsulación o encriptación que no permitan la interacción entre el código y el usuario o con los componentes del Sistema Operativo y del Hardware en sí.

Costos: Este criterio hace referencia directa al precio de las licencias que debe pagar el usuario por la

utilización de las herramientas de desarrollo, para la creación de aplicaciones bajo los estándares J2EE y .NET. Si se considera la realización de paquetes independientes para el nivel del usuario.

3.10.6. DEFINICIÓN DE CRITERIOS DE COMPATIBILIDAD.

Para el Área de Evaluación de Compatibilidad se definieron dos criterios en los cuales se describe la posibilidad de crear una misma aplicación independiente de la máquina, programa o Browser para poder ejecutarse.

Al referirse a Compatibilidad se entra en el terreno de pugnas de empresas de software, las cuales no va en beneficio de esta investigación por lo que no son consideradas. Para esto se definió:

Portabilidad: al definir el criterio de portabilidad se refiere a la posibilidad de migrar una aplicación Web realizada bajo un Sistema Operativo y, que ésta pueda ser ejecutada en otro donde no fue concebida, de ser viable una operación de este tipo qué nivel de depuración o corrección debe hacerse para que pueda funcionar correctamente. También este criterio hace relación al IDE

seleccionado el cual contenga distintas versiones para los distintos sistemas operativos ofreciendo una alta productividad para desarrollar aplicaciones multiplataforma para Linux, Solaris o Windows.

Navegadores Web: Este criterio hace referencia a la posibilidad de encontrarse con errores al momento del despliegue de aplicaciones en algún Browser o Navegador Web, que, aunque estos visualicen códigos HTML, estas páginas dinámicas tanto JSP como ASP o ASPx con sus respectivos tags agregados a los códigos HTML, a veces generan pequeñas diferencias al acceder a una aplicación desde un visor a otro, arrojando algún tipo de faltas como: tipos de letras, alineaciones, imágenes, etc.

3.11. POSTULADOS DE COMPARACIÓN DE LAS PLATAFORMAS J2EE Y .NET.

La industria del software empieza a tomar una postura crítica hacia las plataformas de desarrollo: ¿.NET o J2EE, [49], a futuro se menciona la posibilidad de que .NET y Java serán compatibles, que ambas tecnologías se comunicarán y el desarrollo será aceptado por sus clientes los cuales podrán interconectar sus aplicaciones hechas en ambas tecnologías, pero, por el momento, no

queda otra alternativa aceptable que la de decidirse por una de éstas, teniendo en claro el porqué de dicha elección y sabiendo que es la mejor alternativa.

En esta parte, luego de definir los criterios de comparación, se describe el comportamiento de ambas plataformas de desarrollo frente a frente, se acentúa en detalles técnicos y prácticos que demuestren ventajas y vulnerabilidades de estas tecnologías, el poder acabar con mitos, tanto positivos como negativos y decidir con fundamentos cuáles de estos son reales y cuáles no.

Cada uno de estos postulados se generan luego del análisis de los distintos sitios y documentación recolectada sobre estas dos plataformas, en los cuales muchas veces convergen en aseveraciones mencionadas en más de una oportunidad, de ahí la necesidad de analizarlos y presentar estos postulados los cuales indican ventajas y desventajas presentadas por las plataformas J2EE y .NET.

Algunos de los postulados han sido elaborados de temas descritos en los sitios de apoyo a la investigación y los documentos base mencionados en la bibliografía de este trabajo.

3.11.1. VENTAJAS Y DESVENTAJAS DE LAS PLATAFORMAS.

Aquí se presentan algunos postulados que afirman características de ambas plataformas las cuales se desmienten o afirman analizando tales áreas corroborando tales enunciados.

Conforme se avanza, se va introduciendo en conceptos, características y teorías que muchos afirman sin un sustento o base que los corroboren o por otra parte, que se presenten con una descripción detallada que los haga ser válidos.

En alrededor de diez postulados, se resume la mayor cantidad de información que ha sido encontrada en los sitios mencionados, analizando conceptos, mitos y aseveraciones que abordados desde una perspectiva objetiva logren establecer ventajas de una plataformas sobre la otra.

3.12. POSTULADOS DE COMPARACIÓN.

3.12.1. POSTULADOS DE COMPARACIÓN 1.

"JAVA CORRE EN CUALQUIER AMBIENTE, MIENTRAS .NET ES EXCLUSIVAMENTE PARA ENTORNOS MICROSOFT".

Este postulado se refiere a la plataforma o sistema operativo utilizada por los usuarios, el poder saber realmente, cual de ambas plataformas permitirá a éstos una mejor explotación de sus sistemas, aprovechamiento de recursos, etc., aspectos importantes a la hora de determinar a qué tecnología se dará uso, e incluso el poder adoptar nuevos sistemas operativos dentro de la empresa, tanto en corporaciones internacionales, como en pequeñas y medianas empresas.

Algunos Informáticos piensan en Java no sólo como un lenguaje de programación, sino que afirman que ésta es una poderosa tecnología con algunos objetivos fundamentales como: el entregar un entorno estándar de desarrollo y una producción independiente de la plataforma, proveyendo de escalabilidad para que la explotación de sistemas sea más eficiente, robusta, flexible, integrada y segura.

Esto quiere decir que con Java se pueden construir aplicaciones sobre máquinas Intel con sistema operativo Linux, y después migrarlas fácilmente a ambientes Windows, llevarlas a teléfonos celulares, PDAs, mainframes IBM, servidores corporativos Sun o cualquier otra plataforma con soporte en Java, sin tener que recompilar o modificar los códigos generados.

Lo mencionado anteriormente, aunque es real, no se cumple a cabalidad, luego de la investigación desarrollada, se puede afirmar que la portabilidad no se realiza de manera tan fácil como se desearía, el pensar en la posibilidad de migrar aplicaciones JAVA desarrolladas en un sistema operativo a otro como algo simple es algo totalmente erróneo.

Hechos como el simple manejo de archivos, directorios y permisos, hacen que el proceso de depuración sea algo engorroso y complicado para cierto tipo de aplicaciones, que sin ser imposible, en muchos casos genera más problemas adaptarlos que realizarlos completamente desde cero en el nuevo entorno operativo.

Además la utilización de componentes Java desarrollados para un sistema operativo en particular, no siempre

funcionan correctamente en el sistema operativo migrado, se debe tener un conocimiento sobre el manejo de estos componentes y la ubicación correcta de éstos dentro de la aplicación o proyecto.

3.12.2. POSTULADO DE COMPARACIÓN 2:

"PARA J2EE EXISTEN MÁS EMPRESAS QUE APOYAN A ESTA TECNOLOGÍA QUE A .NET"

Este postulado es real, luego de lo investigado se puede afirmar que la plataforma J2EE cuenta con una amplia gama de productos con soporte para máquinas Intel o AMD con Linux, Windows, FreeBSD, OpenBSD y NetBSD, computadores Mac, sistemas Sun, sistemas IBM, sistemas HP, celulares Nokia y Ericsson, PDAs, todo tipo de Laptops, sistemas de bases de datos Oracle, Sybase, Informix, MySQL, PostgreSQL e incluso MS-SQL Server tiene JDBC disponible.

También se pueden mencionar las herramientas de desarrollo de Sun, Borland, IBM, Oracle, Sybase, Rational, Macromedia y Apache Group, application servers de Sun, Oracle, BEA, IBM, JBoss Group, servlet containers WebSphere, con software de clase mundial como SAP, JD Edwards, y BAAN, entre otros.

Esto hace que J2EE tenga un soporte importante a la hora de contar con apoyos tecnológicos dentro del medio. Por su parte .NET, desarrollado por Microsoft, es vista como una "arma de doble filo" por los usuarios, ya que muchos de estos son fervientes seguidores o grandes detractores. Esto hace al postulado en cuestión cierto en lo que presenta.

3.12.3. POSTULADO DE COMPARACIÓN 3:

".NET RESTRINGE A LOS USUARIOS EN CAMBIO J2EE NO"

Este postulado se refiere al restringimiento que realiza la empresa Microsoft, al limitar a sus plataformas de desarrollo que solo pueden utilizar sistemas operativos Windows, por un tema netamente comercial y de negocios, que la hace un entorno totalmente cerrado. A su vez Java, al ser un cross-plataform, facilidad de correr en cualquier ambiente, le da una ventaja por sobre .NET al no restringir a los usuarios en la utilización de un Sistema Operativo para desarrollar algún tipo de aplicaciones.

Por esto J2EE ventaja a .NET, ya que éste es un punto en contra hacia la tendencia de hoy en día, la

heterogeneidad de plataformas al interior de las organizaciones, pues, al restringir las herramientas utilizadas por los usuarios, hace que éstos pierdan el interés y no se trabaje correctamente bajo un mismo estándar.

En cuanto a la estandarización, acción realizada en más de una oportunidad por Microsoft, que realizan cambios en estándares conforme a su conveniencia y de manera arbitraria, lo que imposibilita al usuario de .NET de elegir libremente.

Esta problemática no existe por el lado de Java debido a que es una tecnología completamente estándar y abierta, características que se aprecian en el gran número de proveedores de soluciones y productos para la tecnología, hechos que hacen cierto a éste postulado en su mayoría con respecto a limitar el ambiente de trabajo.

3.12.4. POSTULADO DE COMPARACIÓN 4:

"EXISTEN MÁS PROFESIONALES J2EE QUE DE .NET"

Dentro de este postulado se puede mencionar que Java es la plataforma que cuenta con más tiempo en el mercado y

siendo probada, aceptada y adoptada por los usuarios es considerada una tecnología madura, la cual cuenta con miles de proyectos en ejecución. Microsoft .NET, la cual lleva menos años en el mercado, hace que se marque esta tendencia sobre la relevancia en la cantidad de profesionales especializado en .Net que en Java, lo que la hace superior en parte en el recurso humano capacitado.

Este hecho es considerado y quiere ser solventado por las nuevas políticas de Microsoft, que quieren hacer un realce en la capacitación a profesionales en el amplio mundo de la programación de .NET. Con dicha capacitación se quiere formar una línea de profesionales, capaces de formar y poder capacitar a otros.

También se desea por parte de esta empresa, el aumento de comunidades que faciliten el compartimiento de conocimientos que aún no supera a otras como lo es en el ámbito de cooperación Web, donde existen muchos portales de comunicación y colaboración en aplicaciones en J2EE, los que llevan tiempo funcionando.

Este postulado se puede considerar como falso, en parte, desde una perspectiva de tiempo bastante subjetiva, ya

que éste hecho no debería afectar a una superioridad de una plataforma sobre otra (por haber llegado antes al mercado). Aunque este hecho fue cierto hace un par de años atrás, el fuerte potencial que ha experimentado .NET con las nuevas funcionalidades y facilidades otorgadas dentro de sus herramientas de desarrollo, han hecho que el componente de tiempo en el mercado sea menos marcado.

3.12.5. POSTULADO DE COMPARACIÓN 5:

"J2EE ES MÁS DIFÍCIL DE APRENDER QUE .NET"

Al abordar temas como la dificultad, algo tan subjetivo, se debe considerar y describir algunos mitos que relacionan a Java o J2EE, como algo mucho más difícil de aprender que la tecnología .NET. Esto varía dependiendo del nivel en que la empresa o usuario se encuentre.

El hecho de sólo conocer HTML como estructura de desarrollo Web, hace que la tecnología ASP de Visual Studio.NET sea mucho menos compleja dado que el proceso de programación es similar. JSP de J2EE, vendría a caber dentro de esta misma característica, con la salvedad de la programación JAVA.

Tomando una posición imparcial, se puede afirmar que la supuesta complejidad de Java, tiene que ver con un mito producido por el alto grado de sofisticación de J2EE y por el gran número de características con que cuenta la plataforma. Por este lado los defensores de J2EE y acérrimos detractores de .NET afirman que J2EE no realiza "funcionalidades oscuras, patrones intrincados o APIs poco entendibles", lo que hace referencia al manejo de procesos en Windows y al desarrollo de programar en formularios o Web Forms con arrastre de objetos.

Tomando este hecho de facilidad de desarrollo, llevado a un nivel de conocimiento se podría acentuar que para realizar una aplicación Web en .NET no se necesitaría ser un experto, en cambio en J2EE tendría que tener o mantener un conocimiento de programación sobre la tecnología, o sea, solo se podrían hacer aplicaciones Web en J2EE, si se supiera programar en Java. Por el lado de .NET, los conocedores de esta tecnología, concuerdan en que Microsoft realizó, luego de mucho tiempo, una herramienta certera en el desarrollo de todo tipo de aplicaciones, y que brinda lo necesario para un rápido aprendizaje y conocimiento de todas las funcionalidades insertas en los productos .NET.

Para considerar a este postulado como cierto o como errado, se debe recomendar al menos, conocer la programación en ambas plataformas, lo que es aconsejable, es informarse antes de emitir juicios sobre dificultad o facilidad y no caer en fanatismos para no dar conclusiones erróneas.

Aunque no se puede decir que .NET es más fácil que J2EE, si se puede comentar que .NET presenta un curva de aprendizaje más favorable que J2EE, y que sus herramientas son de una rápida adaptabilidad hacia el usuario, ventajas que si son superiores al estándar creado por Sun.

3.12.6. POSTULADO DE COMPARACIÓN 6:

"J2EE REQUIERE MÁS LÍNEAS DE CÓDIGOS RESPECTO DE LAS HERRAMIENTAS DE .NET".

Al analizar este postulado se debe considerar para qué tipo de funcionalidad .NET presenta menos líneas de código que J2EE. Se debe considerar que el resultado es el mismo con menos líneas de código, si tiene la misma funcionalidad y simplicidad. Si esto se concretara para ambas plataformas y una presentara menos líneas de código,

podría considerarse como un avance.

Con estos aspectos a considerar se podría separar conceptos, pues Java como .NET tienen ciertas características más desarrolladas que otras, por lo que presentan una mayor cantidad y uso de funciones para tareas determinadas, lo que presentaría a los códigos fuente con un mayor número de líneas para la plataforma destacada.

Se debe entender que al asociar menor cantidad de líneas de código con mayor eficiencia, no tiene un real sustento para ser considerada de ese modo. Esta característica no necesariamente apunta a que algo tan sencillo será óptimo, ni tampoco, que algo con mayor complejidad será mejor, las cosas deben hacerse bien, independiente del número de líneas que están rutinas contengan.

Lo que se debe perseguir es cómo hacer mejores aplicaciones, más robustas, más poderosas, más escalables, más integrables, más mantenibles y no sólo si más fáciles de escribir. Entonces, el punto realmente importante radica en determinar cuál es la plataforma que me va a permitir explotar de mejor manera estos aspectos. Por lo que hacen a este postulado como algo sin una mayor

relevancia, además para los códigos .Visual Basic se presenta un número mayor de líneas que las aplicaciones en J2EE.

3.12.7. POSTULADO DE COMPARACIÓN 7:

“.NET SE APRENDE MÁS EN LA PRÁCTICA QUE J2EE.”

Los seguidores y partidarios de .NET aseguran que esta plataforma de desarrollo realiza una demanda de mucha menor inversión tecnológica, de hardware y software que J2EE, que esta ofrece mucho mayor reutilización de componentes y requiere menores costos de desarrollo.

Otras ventajas serían la facilidad de utilización de las herramientas, que las hacen mucho más veloces y potentes para el desarrollo de producción de software y soportar diversas capas tecnológicas de ambientes de explotación para aplicaciones Web.

.NET ofrece programas de desarrollo como Visual Studio.NET, que proponen una interfaz de usuario mucho más “amigable” que el resto de IDE’s de desarrollo para otras plataformas. Ésta presenta un ambiente de objetos y herramientas “arrastrables” dentro de Formularios Web que

pueden programarse de distintas maneras: se puede migrar componentes con arrastre, al hacer clic dentro de alguno de estos componentes, se despliega el código contenido por "detrás", y se puede acceder a códigos .Visual Basic o el código de programación que se esté utilizando, además para la creación de sitios ASPx, se puede programar en ellos como si se estuviera realizando una página HTML y realizar cambios en este mismo lenguaje de programación. Esto haría que entre más se realice un tipo de aplicación con este método, la creación de sitios sea más sencilla y fácil en la reutilización de códigos ya desarrollados, ámbito no manejado de la misma manera por J2EE. Considerando esto, este postulado es cierto.

3.12.8. POSTULADO DE COMPARACIÓN 8 (COMPUESTO) .

- **"EN JAVA EXISTE UN MAQUINA VIRTUAL QUE FACILITA EL TRABAJO EN CAMBIO EN .NET NO."**
- **"EN .NET SE PUEDEN REALIZAR SISTEMAS EN MÁS DE UN LENGUAJE, EN CAMBIO EN J2EE SOLO JAVA".**

Ambos postulados se refieren al trabajo que se realiza en las capas inferiores de la programación, el lenguaje y la forma que éste se ejecuta. Por un lado J2EE, ofrece la

máquina Virtual Java, siendo una capa lógica que hace creer al programa que se ejecuta en un ordenador real (con registros, memoria y procesador), cuando en realidad sólo ve una reconstrucción lógica de un ordenador, y aunque le quita la tarea al Computador en sí, ésta consume una cantidad de recursos importantes.

Por otro lado .NET, maneja el Common Language Runtime (CLR) del Framework .NET que es un código a nivel de máquina que a diferencia de la máquina virtual de Java, ésta se presenta en las capas superiores del software, teniendo un desempeño diferente sin la excesiva necesidad de recursos por parte el computador donde se estén desarrollando y ejecutando las aplicaciones.

Además en la utilización de lenguajes de programación, .NET lleva la ventaja, al manejar más de diez lenguajes compatibles dentro de la misma aplicación, Visual Studio .NET, para comodidad de los desarrolladores los cuales pueden adaptar la herramienta en la forma que se quiera utilizar. Algunos de estos lenguajes son: C#, VB.NET, J#, C++, DELPHI, COBOL y otros, mientras que en J2EE Java vendría a ser el único lenguaje de J2EE.

Esto podría tomarse como una versatilidad mayor de .NET

hacia con J2EE.

Estos postulados favorecerían en parte a la plataforma ofrecida por Microsoft.

3.12.9. RESUMEN DE LOS POSTULADOS.

Luego de leer y analizar cada uno de los principales postulados presentados, se puede dar un énfasis en las empresas que patrocinan ambas tecnologías, las cuales mantienen una pugna por determinar cual ambiente de desarrollo es superior al otro. Ambos "bandos", Microsoft con su Sistema Operativo Windows y las herramientas de desarrollo basadas en .NET, y Java 2 Platform Enterprise Edition con su lenguaje Java con productos como WebSphere y los productos Linux de IBM, hacen referencia a una propuesta Microsoft v/s IBM más que a .NET v/s J2EE.

Para poder derribar mitos y corroborar los aciertos que muestran estos postulados, las empresas como IBM, que apoyan firmemente la plataforma J2EE, tienen que explotar sus ventajas más allá de lo cotidiano seguir desarrollando el poderío de manejar más de un ambiente operativo y hacer que su software sea aún más fácil de usar, de modo que pueda crecer en el mercado de las

pequeñas y medianas empresas.

Por su lado Microsoft debe saltar de sus herramientas de desarrollo y profundizar en servicios de sistema operativo, demostrando que posee la seguridad, escalabilidad y rendimiento necesario para soportar aplicaciones empresariales.

.NET ofrece a los clientes un manejo amigable en la construcción de aplicaciones Web con utilización de herramientas básicas como el Internet Information Server incluido en los Sistemas Operativos Windows para el despliegue de este tipo de programas. Por otra parte J2EE, con la posibilidad de instalarlo en diversos sistemas operativos posee programas de desarrollo como Netbeans, IBM WebSphere, para cada una lo que proporciona la capacidad de poder ofrecer servicios y funcionalidades Web como JSP y Servlets, entre otras herramientas para diferentes arquitecturas.

3.13. SISTEMAS WEB CON BASE DE DATOS EN J2EE Y .NET.

Al acceder a una página Web al navegar por Internet, el principal servicio que ofrecen estos sitios, además de la

capacidad de información, es el poder acceder a datos alojados en algún computador habilitado como servidor. Estas tecnologías a través de JDBC en J2EE, y ADO.NET en Microsoft.NET manejan la de conexión de sistemas Web con base de datos. Este tipo de sistema es el principal realizado por las páginas dinámicas JSP y ASPx las cuales crean tipos de conexiones confiables y utilizables por los usuarios deseosos de acceder a estos datos donde la seguridad y el rendimiento son las variables importantes a la hora de considerar con qué lenguaje se desarrollarán dichas aplicaciones.

En este capítulo se inicia en la etapa de programación de aplicaciones Web con acceso a base de datos, las cuales presentan las principales características a considerar para desarrollar un marco de trabajo basado en aspectos que consideren la conexión, despliegue, y todas las operaciones básicas al manejar datos, describiendo un modelo de programación basado en la arquitectura cliente - servidor - motor de base de datos, apareciendo así un modelo de aspectos como: el tipo de conexión y el motor de base de datos establecido.

Para esto solo se utilizó motores de libre licencia como lo son: Mysql, PostgreSQL y la inclusión de la

herramienta de Microsoft Office, Microsoft Access, descartando motores de mayor potencial como Oracle o SQL Server, los cuales presentarían un detalle mayor lo cual desviaría del tema principal, la comparación entre J2EE y .NET y se entraría a comparar el funcionamiento de estas plataformas con los distintos motores de bases de datos.



Figura 11: Modelo de Aplicación Web con Base de Datos.

3.13.1. MOTORES DE BASES DE DATOS A UTILIZAR.

3.13.1.1. MYSQL

MySQL es uno de los sistemas gestores de bases de datos más populares desarrolladas bajo la filosofía de código abierto, éste fue desarrollado y es mantenido por la empresa de software MySQL AB, la cual mantiene disponible

el código fuente para su revisión.

Inicialmente, MySQL carecía de los elementos considerados esenciales para poder manejar bases de datos relacionales, elementos como: integridad referencial y transacciones, a pesar de ello, desarrolladores de páginas Web preferían este motor para acceder a datos, por su simplicidad y buen funcionamiento para este tipo de aplicaciones.

En la actualidad los elementos faltantes en MySQL han siendo incorporados por los desarrolladores, los cuales hacen que este motor de base de datos progrese a pasos agigantados frente a otros creados por corporaciones de Software de este tipo. Entre las características disponibles en las últimas versiones se puede mencionar:

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según, si se desea velocidad en las operaciones o un mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.

- Búsqueda e indexación de campos de texto.

Además de esto, MySQL se posiciona como el motor de base de datos más utilizado por los usuarios con cifras de más o menos seis millones de copias funcionando en la actualidad, hecho que supera a cualquier otra herramienta de bases de datos y, desde la versión 5.0, se han integrado: Procedimientos Almacenados, Triggers y Vistas. Este motor de base de datos con licencia GPL obliga a distribuir cualquier producto derivado, a que permanezca bajo esta misma licencia, al necesitar incorporar MySQL en algún producto pero no desea distribuirlo bajo licencia GPL, se debe adquirir una licencia comercial que permite poder utilizar a MySQL de esta manera.

3.13.1.2. POSTGRESQL.

PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto como MySQL, Firebird y MaxDB, así como sistemas propietarios como Oracle o DB2. Algunas de sus principales características son:

- Claves ajenas también denominadas Llaves ajenas o

Llaves Foráneas.

- Disparadores (triggers).
- Vistas.
- Integridad transaccional.
- Acceso concurrente multiversión (no se bloquean las tablas, ni filas, cuando un proceso escribe).
- Capacidad de albergar programas en el servidor en varios lenguajes.
- Herencia de tablas
- Tipos de datos y operaciones geométricas

PostgreSQL se desarrolló principalmente para resolver los problemas con el modelo de base de datos relacional y al manejo de objetos. También este motor presenta una licencia BSD orientada a desarrolladores de código abierto para continuar su desarrollo. Por el lado comercial, la empresa Pervasive comercializa el motor PervasiveSQL, basado en Postgres, que también distribuye la versión libre PostgreSQL, actualmente la versión PostgreSQL 8.0.x

3.13.1.3. MICROSOFT ACCESS.

Aunque Microsoft Access no es reconocido como un motor de base de datos, por ser una aplicación incluida en el

paquete de utilitarios Microsoft Office, éste es ampliamente utilizado por pequeñas y medianas empresas, las cuales, al no contar con recursos monetarios importantes, no invierten grandes cantidades de dinero en software de este tipo y lo consideran de manejo simple y que cumple sus necesidades con este tipo de programas.

Este software es principalmente utilizado para el almacenamiento común de datos de todo tipo, mayormente para manejo de inventarios, bodega, o para la utilización de datos que no requieran un uso continuo, sino ser, un medio de respaldo al cual se pueda acceder en tiempos variados para cotejar algún tipo de información o rescatar datos.

Microsoft Access proporciona las herramientas necesarias para el almacenamiento de los datos y las consultas SQL necesarias para el trabajo con éstos. La integración de este utilitario con aplicaciones del ámbito Microsoft, lo hacen aún más valioso en las pequeñas empresas que utilizan aplicaciones ASP con servidor Web IIS (Internet Information Server), ya que estas herramientas presentan una interconexión sumamente simple de realizar por cualquier usuario, sin que éste tenga que ser un programador en este tipo de software, lo que lo hace una

herramienta valiosa sobre otros software administradores de bases de datos.

3.13.1.4. SQL SERVER

Es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea, así como de tener unas ventajas que más abajo se describen.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados **ADP** (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la

implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

La nueva base de datos contiene mayor seguridad, integración con PowerShell, remueve La consola configuración del área expuesta (consola para configurar seguridad), tiene correctores de sintaxis del lenguaje Transact-SQL e intellisense (una característica del Visual Studio que permite a la base de datos sugerir objetos existentes mientras uno escribe la mitad de la palabra). Así mismo incluye nuevos tipos de datos y funciones.

3.13.1.5. ORACLE

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones,
- Estabilidad,
- Escalabilidad y
- Soporte multiplataforma.

Ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios.

En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

Oracle surge a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre SGBD (Sistemas Gestores de Base de Datos) de George Koch. Computer World definió este estudio como uno de los más completos jamás escritos sobre bases de datos. Este artículo incluía una comparativa de productos que erigía a Relational Software como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las bases de datos relacionales, algo que por aquella época era todavía desconocido.

En la actualidad, Oracle (Nasdaq: ORCL) todavía encabeza la lista. La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo y en las oficinas de 98 de las 100 empresas Fortune 100. Oracle es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software.

Oracle a partir de la versión 10g Release 2, cuenta con 6 ediciones:

- Oracle Database Enterprise Edition (EE).
- Oracle Database Standard Edition (SE).
- Oracle Database Standard Edition One (SE1).
- Oracle Database Express Edition (XE).
- Oracle Database Personal Edition (PE).
- Oracle Database Lite Edition (LE).

La única edición gratuita es la Express Edition, que es compatible con las demás ediciones de Oracle Database 10gR2 y Oracle Database 11g.

3.14. APLICACIONES WEB CON BASE DE DATOS EN J2EE.

En este punto de la investigación, se recolecta información detallada en capítulos anteriores, en los cuales se hizo referencia al tipo de conexión que maneja la plataforma J2EE con los distintos motores de bases de datos. En ellos se puede observar que el tipo conexión JDBC, conexión basada en un cierto tipo de archivo, por lo general de extensión .jar, los cuales proveen las funcionalidades necesarias para el correcto trabajo con

este tipo de aplicaciones.

Para los siguientes sistemas desarrollados en Netbeans para la plataforma J2EE, se necesitó descargar de la red los distintos conectores JDBC que permiten conectarse con los motores seleccionados y que seguirán siendo utilizados en una posterior etapa de pruebas.

Estos distintos tipo de conexiones (.JAR) no son incluidos en la mayoría de los IDE's de desarrollo para J2EE, lo que hace que el funcionamiento y el tipo de conexión con el servidor no sea un procedimiento directamente manejado por el IDE sino que delega todo este tipo de trabajo al conector utilizado.

Dentro del mundo de los servidores de bases de datos, se encuentra una larga lista de distintos motores que proporcionan lo necesario para el manejo de información, en especial para este tipo de estudio y dentro de una evaluación comparativa se optó por seleccionar motores de licencia libre como se comentó con anterioridad, reflejando los sistemas que se apreciarán posteriormente.

3.14.1. CONEXIÓN A BASES DE DATOS CON JDBC.

Estos conectores .jar funcionan de forma parecida, al mantener la misma estructura. Con el driver correspondiente en la rutina de conexión a la base de datos se incluye:

- La Clase del Driver a utilizar (.class).
- El nombre de la base de datos, incluyendo el servidor donde está alojada y el puerto de conexión manejado como String.
- El respectivo UserName y su clave. Manejado como String.
- El objeto conexión que incluye los datos anteriores.

```
<%  
Class.forName("org.postgresql.Driver");  
String database = "jdbc:postgresql://localhost:5432/BD2";  
String username="root";  
String password="robotech";  
Connection con = DriverManager.getConnection(database,username,password);  
%>
```

Tabla 6: Ejemplo Conexión a Base de Datos Postgres.

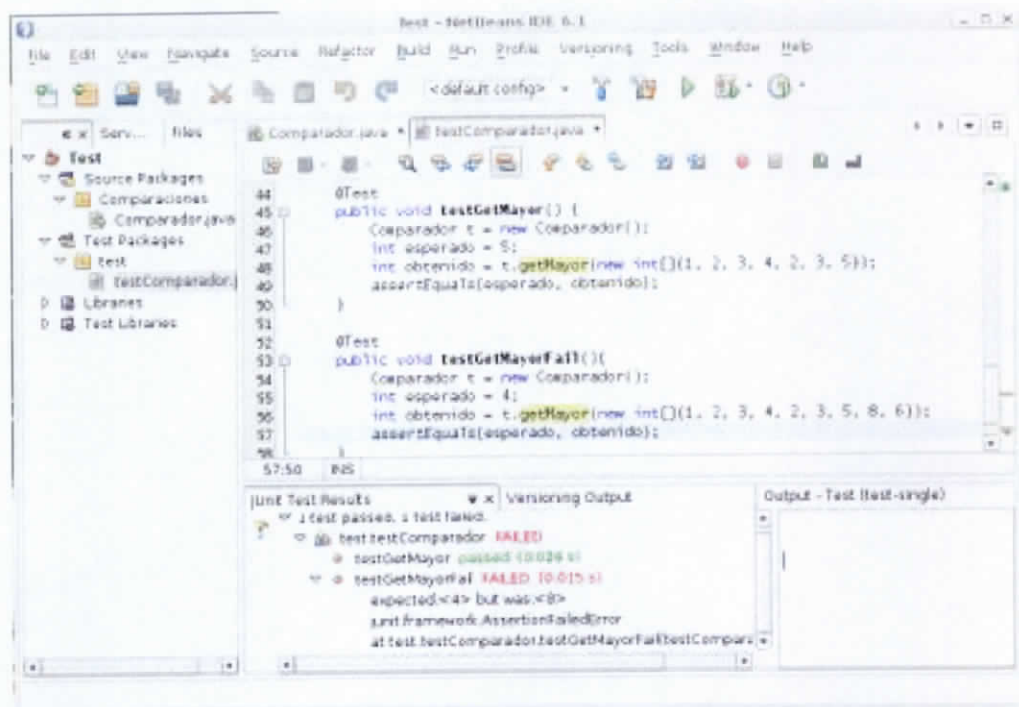


Figura 12: Vista de programación en Netbeans.

3.14.2. SISTEMA DE BASE DE DATOS CON MYSQL Y NETBEANS.

Para el desarrollo de un sistema desarrollado con el motor de base de datos MYSQL, se necesitó de un conector JDBC para poder realizar dicha conexión, para esto se descargó de Internet el paquete de instalación correspondiente, éste fue `mysql-connector-java-3.1.7`, el cual permitió conectar el IDE de desarrollo Netbeans con la base de datos.

Esta conexión se agrega en proyecto Web de Netbeans haciendo clic derecho en éste y agregando la clase para

que sea reconocida en los códigos JSP que hagan referencia a ésta.

```

<%@ page language="java" %>
<%@ page import="java.sql.*" %>
<%
    Class.forName("com.mysql.jdbc.Driver");
    String database = "jdbc:mysql://localhost:3306/peliculas";
    String username="root";
    String password="";
    Connection con =DriverManager.getConnection(database,username,password);

%>
<HTML>
<BODY background="d:\Paginas Web\Pagina Taller\fotos\fondo.gif">
    <center>
        <font color=blue size=10>Contenido de la Base de Datos </font>
        <TABLE border="10" cellspacing="10" cellpadding="10" align="center">
            <tr><td bgcolor="#78705A">Clave</td><td bgcolor="#78705A">Película</td><td
bgcolor="#78705A">Actor</td><td bgcolor="#78705A">Director</td>
            <td bgcolor="#78705A">Descripcion</td><td bgcolor="#78705A">Clasificación</td></tr>
        <%
            if (con != null)
            {
                Statement stmt = con.createStatement();
                ResultSet res = stmt.executeQuery("SELECT * FROM movies order by Nombre");
                while(res.next())
                {
                    String clave = res.getString("Clave");
                    String pelicula = res.getString("Nombre");
                    String actor = res.getString("Actor");
                    String director = res.getString("Director");
                    String descripcion = res.getString("Descripcion");
                    String clasificacion = res.getString("Clasificacion");
                    out.println("<tr><td bgcolor=blue>"+ clave + "</td>" + "<td bgcolor=blue>"+ pelicula + "</td>" + "<td
                    bgcolor=blue>"+ actor + "</td>");
                    out.println("<td bgcolor=blue>"+ director + "</td>" + "<td bgcolor=blue>"+ descripcion + "</td>" + "<td
                    bgcolor=blue>"+ clasificacion + "</td></tr>");
                }
                res.close();stmt.close(); con.close();
            }
        %>
    </table>
    <br>
    <a href="http://localhost:8084/JSPmysql/" ><h1>Volver a Pagina Principal </a>
</center>
</BODY>
</html>

```

Tabla 7: Código cargar.jsp del proyecto JSPMySQL.

3.14.3. SISTEMA DE BASE DE DATOS CON MICROSOFT ACCESS Y NETBEANS.

Para poder interconectar el software Microsoft Access con

Netbeans se necesitó de una utilidad que permitiera esto, al no existir un conector JDBC por no considerarse a Access como motor de base de datos.

En esta situación se necesitó buscar una utilidad que simulara este funcionamiento y/o ofreciera este servicio, en esta búsqueda se encontró un conector que se despliega como un servicio más el cual debe estar funcionando para poder realizar la conexión, este fue Atinav-JTA2N11R un conector externo tipo JDBC, proporcionado por la empresa Atinav aveConnect, los cuales realizan distintos tipos de conectores para J2EE y otros sistemas de desarrollo.

Con este software instalado se pudo realizar la conexión y realizar una aplicación Web con acceso a Base de Datos en Microsoft Access.

```

<%@ page language="java" %>
<%@ page import="java.sql.*" %>
<% String clavex = request.getParameter("clave");%>
<HTML>
<BODY background="d:\Paginas Web\Pagina Taller\fotos\fondo.gif">
<center>
<h1>Eliminacion de Datos con JSP<BR><BR><BR>
<TABLE border="10" cellspacing="10" cellpadding="10" align="center" td=>
<tr><td><h1>Datos Eliminados Exitosamente</td></tr>
<%
Class.forName("acs.jdbc.Driver");
String database = "jdbc:atinav:localhost:7227:d:\mibase.mdb";
String username="Admin";
String password="";
Connection con = DriverManager.getConnection(database,username,password);
String consulta ="delete from mitabla where clave="" + clavex + "" ";
PreparedStatement ps = con.prepareStatement(consulta);
ps.executeUpdate();
ps.close();
%>
<Form method=post action=cargar.jsp>
<tr><td align=center><input type=submit value="Cargar Datos"></td></tr>
</form>
</table>
<br>
<a href="http://localhost:8084/baseweb" ><h1>Volver a Pagina Principal </a>
</center>
</BODY>
</html>

```

Tabla 8: Código eliminar.jsp del proyecto baseweb.

3.14.4. SISTEMA DE BASE DE DATOS CON POSTGRESSQL Y NETBEANS.

Para el sistema desarrollado para el motor de base de datos Postgres, se realizaron las mismas acciones que para los sistemas anteriores, con la salvedad de utilizar un nuevo conector JDBC realizado para este motor.

A pesar de mantener la sintaxis y mismas funcionalidades en código, éstos presentaron alguna dificultad al presentar errores de ingreso y conexión, generalmente por falta de información y conocimiento sobre el motor de base de datos PostGresSQL.

Posteriormente al ser corregidas funcionaron a la perfección. A continuación se muestra en detalle la página JSP de ingreso de datos.

```

<%@ page language="java" %>
<%@ page import="java.sql.*" %>
<%
String clave = request.getParameter( "clave" );
String pelioulax = request.getParameter( "pelioula" );
String actorx = request.getParameter( "actor" );
String directorx = request.getParameter( "director" );
String descripcionx = request.getParameter( "descripcion" );
String clasificacionx = request.getParameter( "clasificacion" );
session.setAttribute( "laclave", clave );
session.setAttribute( "elnombre", pelioulax );
session.setAttribute( "elactor", actorx );
session.setAttribute( "eldirector", directorx );
session.setAttribute( "ladescri", descripcionx );
session.setAttribute( "laclasi", clasificacionx );
%>
<HTML>
<BODY background="d:/Paginas Web/Pagina Tallerifotos/fondo.gif">
<center> <Font color=blue size=10>Acceso a Datos con JSP </font><br><br><br>
<h1> Datos Ingresados a la Base de Datos:<BR><BR><BR>
<TABLE border="10" cellspacing="10" cellpadding="10" align="center">
<Form method=post action=cargar.jsp>
<tr><td><h1>Clave: </td> <td><%= session.getAttribute( "laclave" ) %></td></tr>
<tr><td><h1>Pelioula: </td> <td><%= session.getAttribute( "elnombre" ) %></td></tr>
<tr><td><h1>Actor: </td> <td><%= session.getAttribute( "elactor" ) %></td></tr>
<tr><td><h1>Director: </td> <td><%= session.getAttribute( "eldirector" ) %></td></tr>
<tr><td><h1>Descripcion: </td> <td><%= session.getAttribute( "ladescri" ) %></td></tr>
<tr><td><h1>Clasificacion: </td> <td><%= session.getAttribute( "laclasi" ) %></td></tr>
<%
Class.forName("org.postgresql.Driver");
String database = "jdbc:postgresql://localhost:5432/BD2";
String username="root";
String password="robotech";
Connection con = DriverManager.getConnection(database,username,password);
String query="insert into prueba values ('" + clave + "','" + Películas + "','" + actorx + "','" + directorx + "','" +
descripcionx + "','" + clasificacionx + "')";

PreparedStatement ps = con.prepareStatement(query);
ps.executeUpdate();
ps.close();
%>
</table><input type=submit value="Cargar Datos">
</form><br><br>
<a href="http://localhost:8084/JSPpost" ><h1>Volver a Pagina Principal </a>
</center>
</BODY>
</html>

```

Tabla 9: Código insertar.jsp del proyecto JSPpost.

3.14.5. INFORMACIÓN ADICIONAL.

- Todas las aplicaciones Web en Netbeans para J2EE al ejecutarse de modo local (localhost) utilizan el puerto 8084.
- En Netbeans el tipo de conexión para todos los motores de bases de datos es similar y mantiene el mismo

formato.

- La conexión y el manejo con la base de datos, sin importar el motor son tareas propias del conector JDBC (.jar) y el IDE solo se encarga de entrelazar la aplicación con dicho conector.
- JSP no maneja algún objeto el cual pueda manejar contenidos de bases de datos completos, solo maneja variables de sesión dentro de las cuales se pueden almacenar esta información.
- JSP dentro del manejo de base de datos no incluye nuevos objetos, la mayoría de los componentes utilizados son solo comandos HTML, los cuales dentro de los eventos se genera código java que permite realizar todas las operaciones necesarias.

3.15. APLICACIONES WEB CON BASE DE DATOS EN .NET.

Microsoft .NET con su producto de desarrollo Visual Studio.NET, facilita el trabajo con bases de datos con motores de la familia Microsoft, pero sin descuidar otros

motores de uso mundial y común, para los cuales realiza herramientas apartes, las cuales pueden integrarse al desarrollar en Visual Studio, el cual está diseñado para la inclusión de éstas nuevas funcionalidades para la interconexión de nuevos motores como Postgres, MySql, Oracle etc., diferentes a motores Microsoft como SQL SERVER, MS ACCESS y otros.

Como se definió anteriormente, .NET, maneja el modelo de datos ADO.NET el cual permite trabajar con conjuntos de datos desconectados, lo cual ayuda a reducir el tráfico de red e interconectar distintos administradores de bases de datos.

Para acceder a los datos ADO .NET utiliza un objeto llamado DATASET, que encapsula lo que sería la base de datos a un nivel lógico: tablas, vistas, relaciones, integridad entre todos ellos, etc., pero siempre con independencia del tipo de fabricante que la diseñó.

En el presente capítulo se pone en práctica este modelo de acceso a datos reflejando en los distintos sistemas Web de base de datos tipo ASPx la interconexión realizada para esta herramienta.

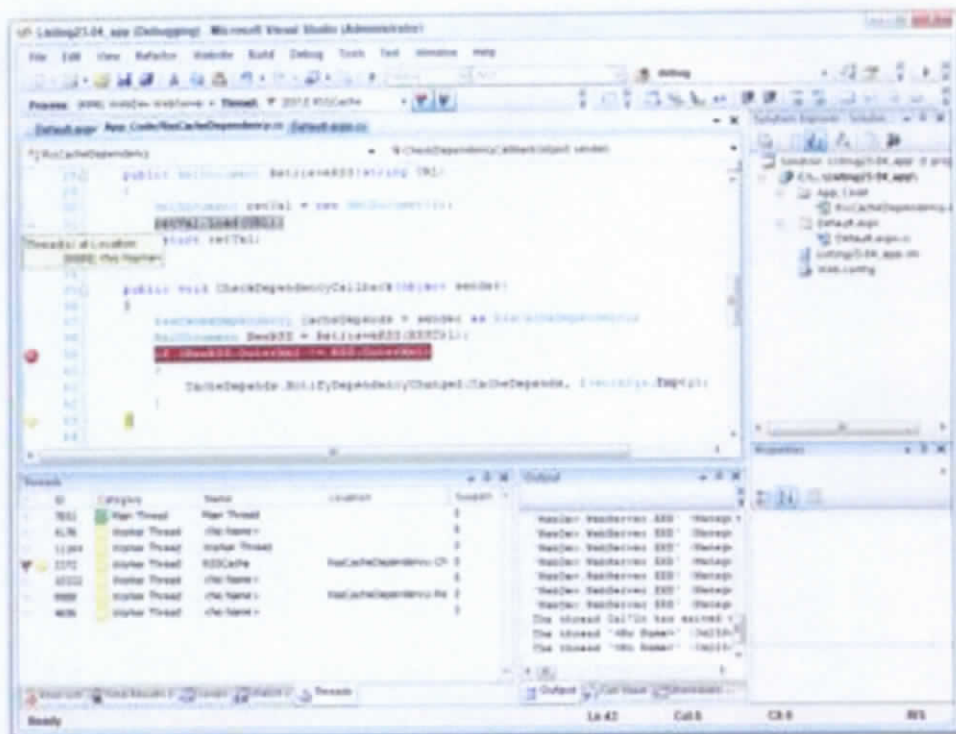


Figura 13: Vista de programación en Visual Studio.NET.

3.15.1. SISTEMA DE BASE DE DATOS CON MYSQL Y VISUAL STUDIO.

Para este sistema, al igual que para las aplicaciones Web desarrolladas en Netbeans en J2EE, se necesitó de componentes extras, los cuales permitiesen a la aplicación, el poder realizar las tareas de inserción, despliegue, eliminación, etc., para el motor de base de datos Mysql.

En la búsqueda de componentes a utilizar, se encontró y utilizó el conector mysql-connector-net-1.0.4, una

utilidad MSI (.msi) que al ejecutarse es integrada a las herramientas proporcionadas por Visual Studio.NET. Esta aplicación es desarrollada por CoreLab empresa que realiza este tipo de utilidades.

Al instalar esta aplicación, Visual Studio .NET, proporciona una utilidad de objetos y funciones agregadas en el Toolbox del editor, en la parte izquierda del IDE de desarrollo, lo cual facilita el uso de dichas herramientas y permite un uso transparente de conexión.

Algunas partes del código se ilustran a continuación:

```
<%@ Page CodeBehind="index.aspx.vb" Language="vb" AutoEventWireup="false"
Inherits="Web.index" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML> <HEAD> <title>Manejo Base de Datos Microsoft Access en .NET</title>
<meta content="Microsoft Visual Studio.NET 7.0" name="GENERATOR">
<meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
<meta content="JavaScript" name="vs_defaultClientScript">
<meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
</HEAD>
<body background="D:\backblue.gif" MS_POSITIONING="GridLayout" bgColor="#99ffff">
<center> <form id="Form1" method="post" runat="server">
<asp.button id="Button1" OnClick="ir1" runat="server" Text="Cargar"></asp.button>
<asp.button id="Button6" onclick="ir6" runat="server" Text="Modificar"></asp.button>
<asp.button id="Button5" onclick="ir5" runat="server" Text="Buscar "></asp.button>
<asp.button id="Button2" OnClick="ir2" runat="server" Text="Insertar"></asp.button>
<asp.button id="Button3" onclick="ir3" runat="server" Text="Eliminar"></asp.button>
<asp.button id="Button4" OnClick="ir4" runat="server" Text="Script"></asp.button>
<BR><asp:HyperLink id="HyperLink1" runat="server" NavigateUrl="http://localhost/Proyecto/1.aspx">
Volver a menu Principal</asp:HyperLink>
<asp:Image id="Image2" runat="server" ImageUrl="file:///net.gif"></asp:Image>
<asp:Image id="Image1" runat="server" Width="157px" Height="130px" ImageUrl="mysql2.JPG"></asp:Image>
</form></center>
</body>
</HTML>
```

Tabla 10: Parte código inicio.aspx de proyecto ASPxMysql.

```

Public Class Index
    Inherits System.Web.UI.Page
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents HyperLink1 As System.Web.UI.WebControls.HyperLink
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents Button2 As System.Web.UI.WebControls.Button
    Protected WithEvents Button3 As System.Web.UI.WebControls.Button
    Protected WithEvents Button4 As System.Web.UI.WebControls.Button
    Protected WithEvents Image1 As System.Web.UI.WebControls.Image
    Protected WithEvents Button5 As System.Web.UI.WebControls.Button
    Protected WithEvents Button6 As System.Web.UI.WebControls.Button
    Protected WithEvents Image2 As System.Web.UI.WebControls.Image
    Protected WithEvents Form1 As System.Web.UI.HtmlControls.HtmlForm
    #Region " Web Form Designer Generated Code "
    <System.Diagnostics.DebuggerStepThrough> Private Sub InitializeComponent()
    End Sub
    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Init
        InitializeComponent()
    End Sub
    #End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles Button1.Click
    End Sub
End Class

```

Tabla 11: Parte código inicio.vb de Proyecto AspXmysql.

3.15.2. SISTEMA DE BASE DE DATOS CON MICROSOFT ACCESS Y VISUAL STUDIO.

Para el sistema desarrollado para Microsoft Access, herramienta desarrollada por la misma empresa que el entorno de desarrollo, posibilita al usuario la realización de códigos de conexiones a bases de datos alojadas en este software que las opciones, funcionalidad y conexión se realizan dentro de la misma página Aspx en el proyecto. Esto hace que no se necesite otro tipo de archivo adicional, o códigos .vb extras, solo las instrucciones necesarias para dicha conexión.

A diferencia con otros sistemas .NET, con manejo de base de datos MySQL o Postgres que necesitan algún tipo de utilidad anexada a Visual Studio .NET tiene dentro de sí funcionalidades propias para aplicaciones Microsoft. En el ejemplo presentado para esta parte del capítulo, se presenta el código que carga los datos recolectados de la base de datos en el Dataset, para posteriormente ser desplegados por pantalla.

Esta no necesita de algún componente extra, facilita al usuario a realizar aplicaciones de este tipo. Dentro de Visual Studio, en el objeto de conexión directa a la base de datos, OLEDBCONNECTION, se especifica el proveedor de servicio OleDb para que pueda entender a qué tipo de aplicación u archivo se refiere.

Esto referenciado a la base de datos Microsoft Access (.mdb), almacenada de forma local o remota, queda instantáneamente conectada y ofrece los objetos necesarios para realizar cualquier tipo de operación con ésta, en otras palabras queda lista para poder ser utilizada.

```

DIM conexion AS OLEDBCONNECTION
DIM canal AS OLEDBDATAADAPTER
DIM tabla AS DATASET
SUB Page_Load(Sender As Object, E As EventArgs)
    conexion = NEW OLEDBCONNECTION ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=d:\mbase.mdb")
    canal = NEW OLEDBDATAADAPTER("select * from mitabla", conexion)
    tabla = NEW DATASET()
    canal.Fill(tabla, "mitabla")
    TABLAGRID.DATASOURCE = tabla
    TABLAGRID.DATAMEMBER = "mitabla"
    TABLAGRID.DATABIND()
    DIM cren, nvaclave AS INTEGER
    cren = tabla.Tables("mitabla").Rows.Count
    nvaclave = tabla.Tables("mitabla").Rows(cren-1)(0) + 1
    CLAVE.Text=nvaclave
END SUB
SUB INSERTAR(Sender As Object, E As EventArgs)
    DIM q AS STRING
    DIM orden as OLEDBCommand
    DIM conexion2 AS OLEDBCONNECTION
    conexion2 = NEW OLEDBCONNECTION ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=d:\mbase.mdb")
    q = "insert into mitabla(clave,nombre,edad) values(@clave, @nombre,@edad)"
    orden = NEW OLEDBCOMMAND(q, conexion2)
    orden.Parameters.Add(new OleDbParameter("@CLAVE", OleDbType Integer))
    orden.Parameters("@CLAVE").Value = CLAVE.Text
    orden.Parameters.Add(new OleDbParameter("@NOMBRE", OleDbType VarWChar, 20))
    orden.Parameters("@NOMBRE").Value = NOMBRE.Text
    orden.Parameters.Add(new OleDbParameter("@EDAD", OleDbType Integer))
    orden.Parameters("@EDAD").Value = EDAD.Text
    orden.Connection.Open()
    orden.ExecuteNonQuery()
    orden.Connection.Close()
    canal=new OleDbDataAdapter("select * from mitabla", conexion)
    tabla=new DataSet()
    canal.Fill(tabla, "mitabla")
    TABLAGRID.DataSource=tabla.Tables("mitabla").DefaultView
    TABLAGRID.DataBind()
    DIM cren, nvaclave AS INTEGER
    cren = tabla.Tables("mitabla").Rows.Count
    nvaclave =tabla.Tables("mitabla").Rows(cren-1)(0)+ 1
    CLAVE.Text=nvaclave
    NOMBRE.Text=""
    EDAD.Text=""
    conexion.Close()
END SUB

```

Tabla 12: Parte código insertar.aspx del Proyecto AspXAccess.

3.15.3. SISTEMA DE BASE DE DATOS CON POSTGRESSQL Y VISUAL STUDIO.

Para poder desarrollar este sistema, con un motor ajeno a la empresa Microsoft, se tuvieron que realizar los mismos pasos que se llevaron a cabo con el motor MySQL.

Lo que hace que para cada aplicación desarrollada por empresas u organizaciones ajenas a ésta, se deba buscar en la Internet cualquier tipo de utilidades necesarias para realizar la conexión entre el IDE de desarrollo, en este caso Visual Studio.NET, y el motor de base de datos seleccionado.

Para esto, se deben encontrar utilidades compatibles con dicho programa para que se puedan incluir en su paquete de utilidades y herramientas que trae por defecto al ejecutar la instalación. Así, para poder conectar PostGres con el programa se necesitó instalar el paquete pgsqlnet.msi, el cual es una utilidad nuevamente desarrollado por la empresa Core Lab, que en su versión PostGreSQLDirect.NET 2.11 posee lo necesario para realizar dichas operaciones.

De mismo modo que para Mysql, este instalador crea una funcionalidad extra al Toolbox ofrecido por Visual Studio.NET, en el cual contiene los objetos necesarios para realizar todo tipo de operaciones hacia las bases de datos de tipo Postgres.

A continuación se presentan extractos de códigos generados en el proyecto PostgresAspx, el cual contiene

los códigos de apariencia (.ASPx), como las funciones necesarias (.vb) de la aplicación.

```
# Declaración de Variables
#Region " Web Form Designer Generated Code "
<System.Diagnostics.DebuggerStepThrough() Private Sub InitializeComponent()
    Me.DataSet = New System.Data.DataSet()
    Me.PgSqlConnection = New CoreLab.PostgreSql.PgSqlConnection()
    Me.PgSqlDataAdapter = New CoreLab.PostgreSql.PgSqlDataAdapter()
    Me.PgSqlCommand = New CoreLab.PostgreSql.PgSqlCommand()
    Me.PgSqlCommandBuilder = New CoreLab.PostgreSql.PgSqlCommandBuilder()
    CType(Me.DataSet, System.ComponentModel.ISupportInitialize).BeginInit()
    Me.DataSet.DataSetName = "NewDataSet"
    Me.DataSet.Locale = New System.Globalization.CultureInfo("ru-RU")
    Me.PgSqlConnection.ConnectionString = "Userid=root,Database=BD2,Host=localhost,Password=robotech"
    Me.PgSqlConnection.Name = "PgSqlConnection"
    Me.PgSqlDataAdapter.SelectCommand = Me.PgSqlCommand
    Me.PgSqlCommand.CommandText = "SELECT * FROM prueba"
    Me.PgSqlCommand.Connection = Me.PgSqlConnection
    Me.PgSqlCommand.Name = "PgSqlCommand"
    Me.PgSqlCommandBuilder.DataAdapter = Me.PgSqlDataAdapter
    Me.PgSqlCommandBuilder.UpdatingFields = ""
    CType(Me.DataSet, System.ComponentModel.ISupportInitialize).EndInit()
End Sub
Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    InitializeComponent()
End Sub
#End Region
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
End Sub
Private Sub btnEjecute_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnEjecute.Click
    Dim nombre, sclave, sector, adirector, adescrp, sclavf As String
    sclave = clave.Text          nombre = pelucia.Text
    sector = actor.Text         adirector = director.Text
    adescrp = descripcion.Text  sclavf = clasif.Text
    Try
        PgSqlConnection.UserID = txtUser.Text
        PgSqlConnection.Password = txtPassword.Text
        PgSqlConnection.Host = txtHost.Text
        PgSqlConnection.Port = CInt(txtPort.Text)
        PgSqlConnection.Database = txtDatabase.Text
        PgSqlCommand.CommandText = "insert into prueba values(" & sclave & "," & nombre & "," &
            & sector & "," & adirector & "," & adescrp & "," & sclavf & ");"
        BindGrid()
        mensaje.Text = "Datos"
        mensaje.Visible = True
        Catch exception As Exception
            mensaje.Text = "Datos Ingresados Correctamente"
        End Try
End Sub
Private Sub BindGrid()
    DataSet.Clear()
    PgSqlDataAdapter.Fill(DataSet, "Table")
    dataGrid.DataSource = DataSet.Tables.Item("Table").DefaultView
    dataGrid.DataBind()
End Sub
Private Sub cargar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cargar.Click
    Try
        PgSqlConnection.UserID = txtUser.Text
        PgSqlConnection.Password = txtPassword.Text
        PgSqlConnection.Host = txtHost.Text
        PgSqlConnection.Port = CInt(txtPort.Text)
        PgSqlConnection.Database = txtDatabase.Text
        PgSqlCommand.CommandText = "select * from prueba"
        BindGrid()
        mensaje.Visible = True
        Catch exception As Exception mensaje.Text = "Datos de Base de Datos"
    End Try
End Sub
End Class
```

Tabla 13: Parte código insertar.vb del Proyecto AspxPost.

3.15.4. INFORMACIÓN ADICIONAL

- Los componentes utilizados para la interconexión entre MySQL y Postgres, se agregan al toolbox de Visual Studio

como los nombres MySQLDirect y PostgreSQLDirect respectivamente.

- El puerto de conexión utilizado por Visual Studio es el 8081, configurable al momento de la instalación, el cual no se especifica en la barra de dirección, al ser incluido automáticamente por el Internet Information Server.
- Las utilidades de conexión utilizadas en esta parte del estudio pueden ser descargados desde <http://www.crlab.com/download.html> en donde se encuentran los paquetes de instalación necesarios para conectar a Visual Studio y herramientas .NET con los distintos motores de base de datos, no solo los incluidos en este estudio.
- En Visual Studio.NET, los códigos que generan acciones pueden ser separados de los que generan las apariencias, para esto dentro del proyecto se crean: las apariencias, archivos (.ASPx), los códigos de acción (.vb) y Archivos de respuesta (.resx)
- Los archivos de respuesta (.resx), son códigos interpretados XML los cuales contienen los objetos SQL

insertos en el código generado (.ASPX), este contiene 2 objetos: data y resheader.

CAPÍTULO IV

4. DESARROLLO DE PRUEBAS DE COMPARACIÓN ENTRE LAS PLATAFORMAS J2EE Y .NET.

4.1. INTRODUCCIÓN.

Para comenzar a sacar conclusiones serias sobre ambas plataformas de desarrollo, habiendo estudiado los conocimientos necesarios para la creación de aplicaciones Web que, junto a las áreas de evaluación y criterios definidos, se pueden empezar a diseñar pruebas que puedan certificar las aseveraciones mencionadas con anterioridad por los fabricantes y empresas que apoyan a estas tecnologías y también poder realizar comentarios propios sobre estas plataformas de desarrollo los cuales se presentarán posteriormente.

En el presente capítulo se mencionarán qué criterios son los más aptos para una etapa de pruebas y si éstos, pueden corroborar dentro de algún tipo de aplicación Web o código, alguna ventaja que ofrezcan las tecnologías en cuestión y que lleven a una comparación práctica entre

las páginas dinámicas propuestas por J2EE y .NET.

Con este diseño y posterior desarrollo, se evaluará y se definirán las conclusiones que podrán ser consideradas como fuente de elección para una toma de decisiones acertada y con fundamentos que permitan a los interesados, y poder ejemplificar de mejor manera a sus propios usuarios o mismos desarrolladores sobre qué tecnología es mejor de utilizar para cada tarea. Para seleccionar qué criterios podrían llevarse acabo en esta etapa, se consideró qué aspectos de la programación podrían reflejarse en aplicaciones Web.

Conceptos como rendimiento, atractivo de la página, rapidez, etc., son nociones básicas que pueden llevarse a cabo realizando pequeñas pruebas o sistemas que determinen alguna superioridad entre las plataformas de desarrollo.

4.2. DISEÑO DE PRUEBAS COMPARATIVAS.

En esta parte del capítulo se quiere constatar en palabras como serán llevadas a cabo cada una de las pruebas comparativas a desarrollar, para luego realizarlas de forma práctica y posteriormente concluir a partir de los resultados obtenidos. En cada una de estas

pruebas se quiere dejar en claro alguna superioridad por parte de una de las plataformas, o si se encuentran en igualdad de condiciones. Para esto se definieron pruebas por separado para cada área de evaluación, estas son:

4.2.1. DISEÑO DE PRUEBAS DE APRENDIZAJE.

Para esta área de evaluación se refleja la simplicidad o complejidad de acceder a los conocimientos necesarios para acceder a una de las tecnologías en cuestión, se definió:

- **PRUEBA NÚMERO 1:**

“La prueba se define como la simple búsqueda de información disponible sobre los conceptos J2EE y .NET en el buscador Web más utilizado, Google. En una primera instancia el concepto sin importar el origen de la página y, posteriormente delimitando a páginas solo en español.”

- **PRUEBA NÚMERO 2:**

“Realizar una búsqueda sobre: Libros, Software y temas afines, en Amazon.com (página número 1 de ventas a través de Internet), sobre las plataformas de desarrollo .NET y

J2EE.”

- **PRUEBA NÚMERO 3:**

“Detallar la curva de Aprendizaje por medio de las entrevistas realizadas, información recolectada y experiencias propias con ambas tecnologías.”

4.2.2. DISEÑO DE PRUEBAS DE RENDIMIENTO.

Para medir el rendimiento en todas las áreas de las aplicaciones Web para las dos plataformas y que éstas quedaran detalladas con información real se definió lo siguiente:

Para rendimiento independiente:

- **PRUEBA NÚMERO 4:**

“Realizar una aplicación Web sencilla en la cual solo se dé un mensaje de Bienvenida para desde ahí señalar aspectos a considerar dentro de la comparación. (Rendimiento, Capacidad de respuesta, Estabilidad, etc.)”

- **PRUEBA NÚMERO 5:**

“Realizar la ejecución de un mensaje desplegado 100.000 veces y medir el tiempo utilizado para realizar esta acción”

- **PRUEBA NÚMERO 6:**

“Demostrar de forma gráfica el uso de tiempos y recursos de CPU al utilizar los distintos IDE’s de Desarrollo, tanto Netbeans para J2EE, como para Visual Studio en .NET”

- **PRUEBA NÚMERO 7:**

“Acceder remotamente por más de un usuario al mismo sitio en el mismo tiempo y considerar la escalabilidad ofrecida por la página dinámica comparando los tiempos arrojados”

Para Rendimiento Base de Datos se definió:

- **PRUEBA NÚMERO 8:**

“Realizar dos Sistemas de Base de Datos vía Web los

cuales permitan hacer todas las acciones relacionadas con el manejo de datos, para considerar tiempos de respuesta y despliegue de los datos desde el servidor hacia el cliente”

- **PRUEBA NÚMERO 9:**

“Realizar el mismo sistema de base de datos en un motor de base de datos diferente para medir el comportamiento de la plataforma con otras aplicaciones de este tipo”.

4.2.3. DISEÑO DE PRUEBAS PARA EXPLOTACIÓN Y USO.

- **PRUEBA NÚMERO 10:**

“Detallar una gráfica de explotación o tabla de porcentajes a través de la información recolectada sobre ambas plataformas”.

- **PRUEBA NÚMERO 11.**

“Detallar la cantidad de sitios realizados bajo una de estas dos tecnologías durante la navegación de distintos sitios durante un periodo de prueba y destacar los sitios

de empresas importantes realizados bajo estas tecnologías”.

4.2.4. DISEÑO DE PRUEBAS PARA DESARROLLO.

- **PRUEBA NÚMERO 12:**

“Determinar cantidad de líneas de código para una misma aplicación, determinando la facilidad de desarrollo en ambas plataformas”.

- **PRUEBA NÚMERO 13:**

“Determinar el número de funciones, tareas y procesos proporcionados por ambos IDE’s de desarrollo, detallando semejanzas en cuanto a su funcionamiento”.

4.2.5. DISEÑO DE PRUEBAS DE USO DE LICENCIAS.

- **PRUEBA NÚMERO 14:**

“Analizar las licencias entregadas con cada IDE de desarrollo, tanto para Netbeans, como para Visual Studio.NET, considerando y detallando los costos de éstos

productos”.

4.2.6. DISEÑO DE PRUEBAS DE COMPATIBILIDAD.

- **PRUEBA NÚMERO 15:**

“Llevar alguna aplicación, en el caso que corresponda, a algún sistema operativo Linux, y demostrar que esta aplicación puede ejecutarse en ambientes operativos diferentes.”

- **PRUEBA NÚMERO 16:**

“Llevar a cabo una observación detallada del funcionamiento de las aplicaciones JSP y ASPx en distintos navegadores Web o Browser”.

4.3. CREACIÓN TABLAS DE EVALUACIÓN Y RESULTADOS.

Dentro de esta parte del capítulo se definirán una serie de pasos, los cuales servirán para evaluar a las plataformas de forma comparativa, lo cual permitirá reflejar en datos estadísticos y de forma clara los resultados obtenidos por cada plataforma en las

respectivas pruebas y áreas de evaluación.

Con la creación de tablas se quiere ilustrar y dejar de forma más explícita lo asertiva y minuciosa de la comparación entre las dos plataformas de desarrollo. Las distintas clasificaciones de éstas serán un reflejo del trabajo tanto de investigación de las tecnologías como del proceso de desarrollo de cada una de las aplicaciones creadas en ellas lo que darán respuestas y resultados obtenidos de éstas. Con esto se podrá comparar cada uno de los tipos de cumplimiento observados y poder ser clasificados.

Este tipo de tabla será ocupada en la sección de ejecución y resultados de prueba. Así también para el caso en que una plataforma destaque frente a la otra se creó una tabla de certificación donde queda esto de manifiesto, las cuales serán utilizadas en las conclusiones a las pruebas.

Para esto se crearon las siguientes tablas, de resultados posibles y de certificación, las cuales reflejan el nivel de cumplimiento ofrecido por las plataformas, y la supremacía de alguna sobre la otra, estas son:

PLATAFORMA	LOGO PLATAFORMA SOBRESALIENTE
Igualdad de condiciones	 
J2EE	
.NET	

Tabla 15: Tabla de Resultados.






PRUEBA	J2EE	.NET
Cumple a Totalidad	Las plataformas cumplen todas las expectativas presentadas en el Área de Evaluación y posee ventajas adicionales.	
Cumple Satisfactoriamente	Las plataformas cumplen las expectativas del Área de Evaluación.	
Cumple	Las plataformas cumplen las expectativas del Área de Evaluación sin destacar ventajas adicionales.	
Cumple In Satisfactoriamente	Las plataformas cumplen las expectativas presentadas con ciertas dificultades en el Área de Evaluación.	
No Cumple	Las plataformas no presentan respuestas positivas a las expectativas en el área de evaluación poseyendo ciertas desventajas.	

Tabla 16: Tabla de Resultados.

4.4. EJECUCIÓN Y RESULTADOS DE PRUEBAS COMPARATIVAS.

En esta sección se dará a conocer el detalle de la puesta en marcha de las pruebas diseñadas con anterioridad y los resultados obtenidos en éstas, tanto pruebas de programación, como de simple observación y ejecución.

Estas pruebas están realizadas correctamente sin afanes personales de favorecer a alguna de las plataformas en evaluación, pues, uno de los objetivos principales de este estudio es el de no caer en favoritismos, o afirmar mitos generados por estas plataformas, sino ser un medio válido a considerar al querer evaluar ambas plataformas para su utilización. Estas pruebas se realizaron en un tiempo prolongado, desde Marzo de 2009 a Enero del 2010, tiempo suficiente para poder llevar a cabo las conclusiones tomadas a finales del proyecto.

4.4.1. EJECUCIÓN Y RESULTADOS PRUEBAS DE APRENDIZAJE.

EJECUCIÓN Y RESULTADOS PRUEBA 1.

- Para poder limitar de alguna manera el concepto J2EE, se realizó una búsqueda avanzada en el buscador Web Google, la nueva búsqueda de "plataforma J2EE" arrojó 1.400.000 sitios en 0,5 segundos, un número elevado que cuesta creer que trate de sólo páginas relacionados con esta tecnología, el buscador arroja dicha cantidad, restringiendo posibles conceptos errados a la búsqueda. Limitando el idioma J2EE arrojó 450.000 páginas en español aproximadamente en 0,43 segundos.
- Para el concepto "plataforma .NET" este buscador encontró 1.190.000 de sitios relacionados en 0,31 segundos aproximadamente. Al limitar el idioma .NET arrojó 930.000 páginas en español aproximadamente en 0,79 segundos.

(Más que mostrar una ventaja de una plataforma sobre información disponible, esta prueba quiere demostrar que el acceso a información es más que posible a través de la

red, lo que hace que éste no sea un aspecto para determinar en qué tecnología especializarse.)

EJECUCIÓN Y RESULTADOS PRUEBA 2.

- Se realizó una búsqueda sobre manuales J2EE y .NET en www.amazon.com limitando similitudes del idioma u otros aspectos que no reflejasen de manera correcta la documentación presentada. Esto arrojó:

En la búsqueda de manuales y documentos J2EE se logró en sus distintas categorías: Book (243), Software (20), Tools and Hardware (123), de la misma manera para la plataforma .NET se logró para la misma búsquedas: Book (195), Software (48), Computers (18).

EJECUCIÓN Y RESULTADOS PRUEBA 3.

- Para la prueba de curva de aprendizaje se consultó a alumnos y profesionales que desarrollaran aplicaciones o que estuviesen familiarizados con alguna de las plataformas. También accediendo a sitios de capacitación o de información sobre ambas plataformas, así se pudo modelar un gráfico como se muestra a continuación:











PRUEBA	J2EE	.NET
Sitios Relacionados		
Sitios en Español		
Calidad de la Información		
Búsqueda y Consumo		
Curva de Aprendizaje		

Tabla 16: Tabla de Evaluación 1: Pruebas de aprendizaje.

4.4.1.1. INFORMACIÓN ADICIONAL A LA TABLA

Dentro de ésta área de evaluación no se presentan grandes ventajas de una plataforma frente a la otra, o sea, ambas se encuentran en un mismo nivel de aprendizaje de nivel inicial, con la salvedad que .NET ofrece una curva de aprendizaje mayor en menos tiempo que J2EE y que el consumo de material relacionado con .NET es mayor al ofrecido para J2EE.

4.4.2. EJECUCIÓN Y RESULTADOS PRUEBAS DE RENDIMIENTO.

RESULTADOS PRUEBA 4:

- En cuanto a Capacidad de respuesta se observó:

Tiempo de respuesta de una simple aplicación JSP que desplegará un mensaje de bienvenida, esta fue:

- De manera local: instantáneo.
- De manera remota: instantáneo.

Lo mismo realizado para una aplicación ASPx reflejó:

- De manera local: instantáneo.
- De manera remota: instantáneo.

(Estos datos son promedios de pruebas reflejadas contra reloj interno de cada uno de los IDE's, lo cual indica una tendencia para cada página)

RESULTADOS PRUEBA 5:

- Al ejecutar saludo.jsp dentro de un ciclo de 100.000 repeticiones.

El tiempo de respuesta fue de: 5 m 01 sg.

- Al ejecutar saludo.jsp dentro de un ciclo de 100.000 repeticiones.

El tiempo de respuesta fue de: 5 m 37 sg.

RESULTADOS PRUEBA 6:

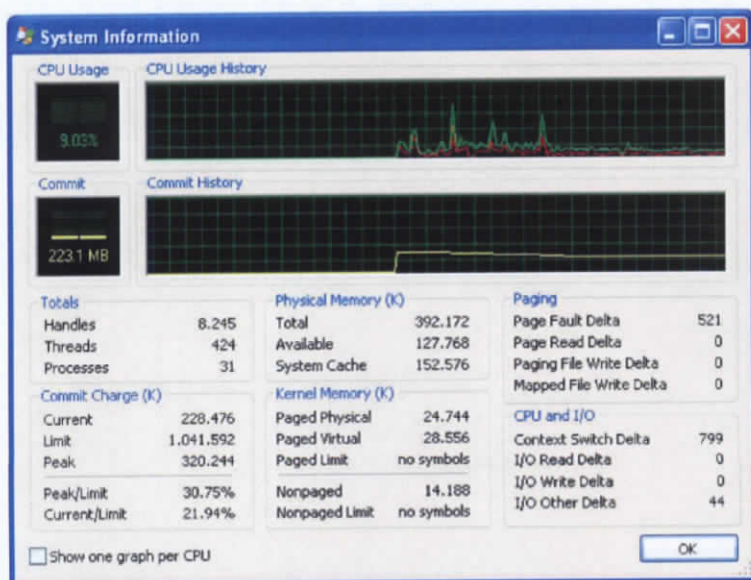


Figura 15: Gráfica rendimiento normal.

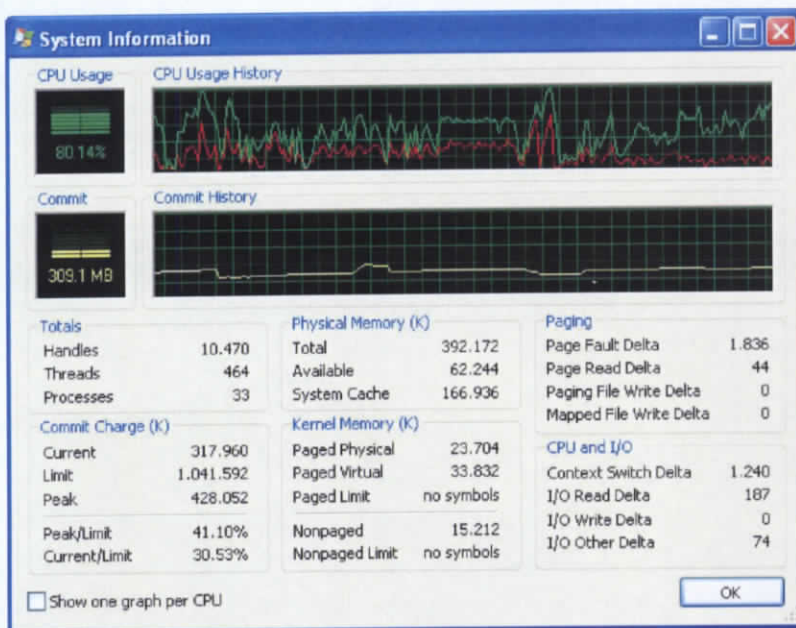


Figura 16: Gráfica de Ejecución Netbeans.

La gráfica marca su máximo al cargar la aplicación. Posteriormente presenta un promedio que varía de 47 a 52% en el desarrollo de aplicaciones).

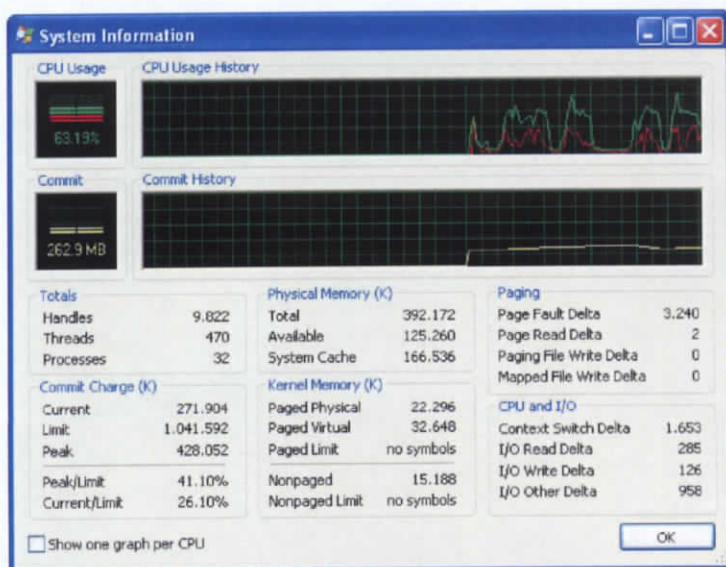


Figura 17: Gráfica de Ejecución de Visual Studio.NET.

La gráfica marca su máximo al cargar la aplicación. Posteriormente presenta un promedio que varía de 43 a 49% en el desarrollo de aplicaciones.

(Para poder medir de forma gráfica el uso de recursos por parte de cada IDE de desarrollo se recurrió a la herramienta PROCExp, la cual permite el despliegue de todos los procesos y aplicaciones de forma detallada para poder medir el consumo de CPU. Estas pruebas se realizaron en un computador laptop de características, procesador Intel Atom 1,33 Mhz, y memoria RAM de 2 GB.)

RESULTADOS PRUEBA 7:

EN CUANTO A ESTABILIDAD:

- Las respuestas para Saludo.jsp al acceder 3 usuarios a la misma página en segundos fue de:

Retardo 3 segundos aprox.

- Las respuestas para Saludo.aspx al acceder 3 usuarios simultáneamente en segundos fue de:

Retardo de 2 segundos aprox.

PARA RENDIMIENTO DE BASE DE DATOS SE OBTUVO:**RESULTADOS PRUEBA 8 Y 9:**

Estas pruebas se obtuvieron sobre una base de datos de 5 campos con 1000 registros, para ambos motores de Base de Datos, se consideró el siguiente modelo para medir los tiempos de respuesta:

Al realizar un Sistema de Base de Datos con los motores MySQL con J2EE se obtuvo:

Tiempo de respuesta al cargar datos de una base de datos:

1,6 segundos promedio, de manera local.

2,5 segundos promedio, de manera remota.

Tiempo de respuesta al ingresar datos a una base de datos:

1,4 segundos promedio, de manera local.

3,8 segundos promedio, de manera remota

Tiempo de respuesta al eliminar datos de una base de datos:

0,6 segundos promedio, de manera local.

4,2 segundos promedio, de manera remota

Tiempo de respuesta al modificar datos de una base de datos:

0,6 segundos promedio, de manera local.

3,5 segundos promedio, de manera remota.

Tiempo de respuesta al buscar datos de una base de datos:

0,6 segundos promedio, de manera local.

3,5 segundos promedio, de manera remota.

Tamaño del Proyecto: **956 KB.**

Cantidad de Archivos generados en el Proyecto:

43 archivos.

Tiempo de acceso con más de un usuario:

Más menos 4 segundos.

Al realizar un Sistema de Base de Datos con los motores MySQL con .NET se obtuvo:

Tiempo de respuesta al cargar datos de una base de datos:

2,5 segundos promedio, de manera local.

3,9 segundos promedio, de manera remota.

Tiempo de respuesta al ingresar datos a una base de datos:

1,6 segundos promedio, de manera local.

2,5 segundos promedio, de manera remota.

Tiempo de respuesta al eliminar datos de una base de datos:

1,3 segundos promedio, de manera local.

4,5 segundos promedio, de manera remota.

Tiempo de respuesta al modificar datos de una base de datos:

0,7 segundos promedio, de manera local.

4,4 segundos promedio, de manera remota.

Tiempo de respuesta al buscar datos de una base de datos:

1,3 segundos promedio, de manera local.

3,5 segundos promedio, de manera remota.

Tamaño del Proyecto: 119 KB.

Cantidad de Archivos generados en el Proyecto:

31 archivos.

Tiempo de acceso con más de un usuario:

Más menos 4 segundos.

Al realizar un Sistema de Base de Datos con los motores PostgreSQL con J2EE se obtuvo:

Tiempo de respuesta al cargar datos de una base de datos:

1,4 segundos promedio, de manera local.

2,5 segundos promedio, de manera remota.

Tiempo de respuesta al ingresar datos a una base de datos:

2,3 segundos promedio, de manera local.

2,6 segundos promedio, de manera remota.

Tiempo de respuesta al eliminar datos de una base de datos:

0,9 segundos promedio, de manera local.

4,3 segundos promedio, de manera remota.

Tiempo de respuesta al modificar datos de una base de datos:

0,7 segundos promedio, de manera local.

4,1 segundos promedio, de manera remota.

Tiempo de respuesta al buscar datos de una base de datos:

0,9 segundos promedio, de manera local.

3,2 segundos promedio, de manera remota.

Tamaño del Proyecto: 520 KB.

Cantidad de Archivos generados en el Proyecto:

34 archivos.

Tiempo de acceso con más de un usuario:

Más menos 3 segundos.

Al realizar un Sistema de Base de Datos con los motores PostgreSQL con .NET se obtuvo:

Tiempo de respuesta al cargar datos de una base de datos:

2,0 segundos promedio, de manera local.

2,8 segundos promedio, de manera remota.

Tiempo de respuesta al ingresar datos de una base de datos:

1,1 segundos promedio, de manera local.

1,8 segundos promedio, de manera remota.

Tiempo de respuesta al eliminar datos de una base de datos:

0,7 segundos promedio, de manera local.

1,4 segundos promedio, de manera remota.

Tiempo de respuesta al modificar datos de una base de datos:

1,9 segundos promedio, de manera local.

2,6 segundos promedio, de manera remota.

Tiempo de respuesta al buscar datos de una base de datos:

0,8 segundos promedio, de manera local.

1,6 segundos promedio, de manera remota.

Tamaño del Proyecto: 159 KB.

Cantidad de Archivos generados en el Proyecto:

39 archivos.

Tiempo de acceso con más de un usuario:

Más menos 4 segundos.

(Para cada una de las pruebas se realizo un promedio de ejecuciones, la cual dio los resultados obtenidos.)

4.4.2.1. AUTOMATIZACIÓN DE PRUEBA.

Además de considerar los sistemas de base de datos

anteriores junto a sus respectivos tiempos de respuesta para cada operación, se realizó un programa para poder automatizar y darle una mayor certificación a los tiempos arrojados luego de las pruebas anteriores, para esto, se definió realizarlo en una de las plataformas, para aprovechar alguna de las ventajas ofrecidas por éstas, contemplando la conexión a una base de datos MySQL alojada en algún servidor, además una variable "cantidad" para determinar el número de veces que se ejecutarían dichas operaciones incluyendo tiempos de duración de las sentencias SQL para así considerar la capacidad de respuesta mostrada por la plataforma de desarrollo.

Este sistema se aprecia en las imágenes siguientes, en primera instancia el sistema sin ejecutar, en segunda instancia luego de la ejecución y posteriormente la tabla de resultados:

The screenshot shows a software interface for automating MySQL performance tests. At the top, there are three orange-bordered boxes labeled "Tiempo Inicia Prueba:", "Tiempo Final Prueba:", and "Tiempo Total Prueba:". Below these is a purple header bar with the text "Sentencias SQL". On the left, there is a yellow-bordered form with fields for "Host", "Password", "Base de Datos", "Puerto", and "Base de Datos", with a "Conectar" button below. On the right, there is a text area titled "Sentencias SQL" containing the SQL command "SELECT * FROM usuarios". At the bottom, there is a "Cantidad" input field and an "Automatizar" button.

Figura 17. Automatización Rendimiento primera instancia

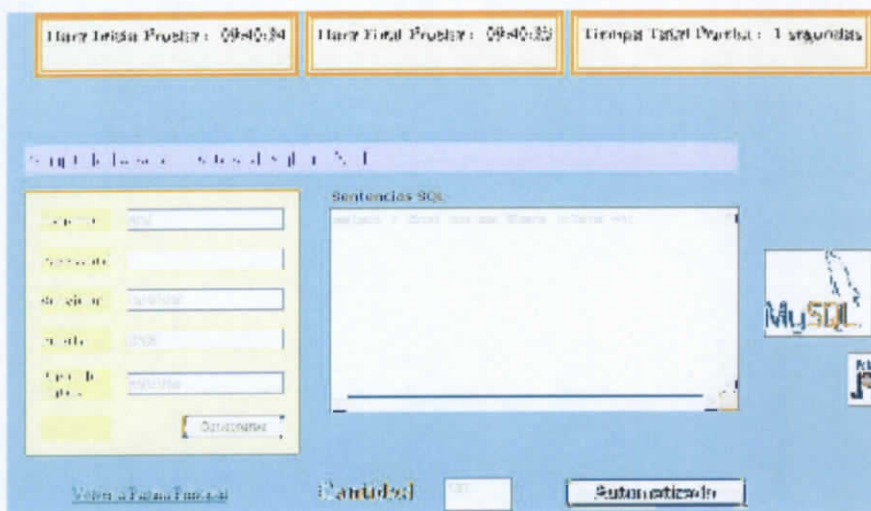


Figura 18. Automatización Rendimiento segunda instancia













PRUEBA	J2EE	.NET
Mensaje de Bienvenida		
Despliegues Repetitivos		
MySql de Manera Local		
MySql de Manera Remota		
Postgres de Manera Local		
Postgres de Manera Remota		
Cantidad de Archivos		
Tamaño del Proyecto		
Uso CPU		

Tabla 17: Tabla de Evaluación 2: Pruebas de Rendimiento.

4.4.3. EJECUCIÓN Y RESULTADOS PRUEBAS DE EXPLOTACIÓN Y USO

RESULTADOS PRUEBA 10.

GRÁFICA DE EXPLOTACIÓN JSP Y ASP (ASPX).

Para esta prueba en particular se consideró un documento realizado por Ricardo Baeza-Yates y Don Carlos Castillo del Centro de Investigación de la Web dentro del Departamento de Ciencias de la Computación de la Universidad de Chile, en el cual se realizó un Estudio de la Web chilena. Realizado en Diciembre del año 2008 ésta investigación analizó más de 50.000 sitios arrojando un crecimiento del 25% solo en ese año, para esto se utilizó crawler WIRE un programa para poder descargar sitios completos así también arrojó otro tipo de información como edad, profundidad, tamaño, largo del url, etc., para cada página. De información relacionada con el tema se puede nombrar por ejemplo:

- Del total de 50.000 sitios, de alrededor de 3.254.137, el 61,7% pertenecían a páginas estáticas y el 38,3% páginas dinámicas.

- Alrededor del 14% de los sitios de Chile son hechos con tecnología Flash y/o JavaScript, lo que impide que algún buscador pueda acceder al sitio.

En el plano de páginas Web arrojó, del 38% de páginas dinámicas:

- 77.65% de sitios realizados en PHP, primer lugar.
- 15,83% de sitios realizados en ASP, segundo lugar.
- 3,57% de sitios realizados en JSP o HTML, segundo lugar.
- 2,95% de sitios realizados en otras tecnologías.

Otro elemento a considerar es el uso de un servidor Web, esto arrojó:

- Apache 55,33%
- Microsoft IIS 30.63%
- Otro 14.05%

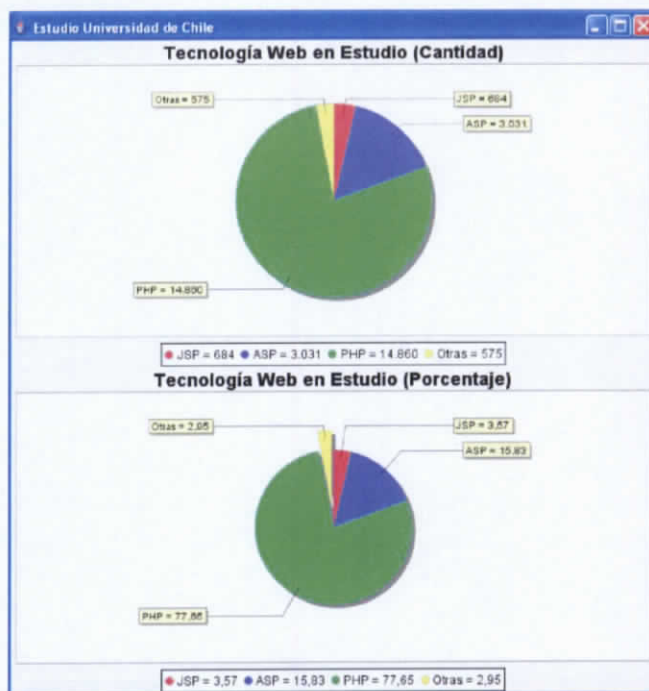


Imagen 20: Gráfica Explotación

4.4.3.1. AUTOMATIZACIÓN DE PRUEBA.

Para poder contrastar tales resultados, dentro de esta prueba de evaluación se desarrolló una aplicación en la cual se analizará el tipo de página dinámica utilizada.

Además incluiría un módulo en el cual se pudieran analizar la mayoría de los dominios .CL, para así hacer estimaciones que corroboraran los estudios realizados por la Universidad de Chile en el año 2009. Dicho programa se implementó en la Plataforma J2EE con su IDE de desarrollo Netbeans, el cual por resultado arroja la URL del sitio, su cantidad de referencias a páginas dinámicas y su tipo.

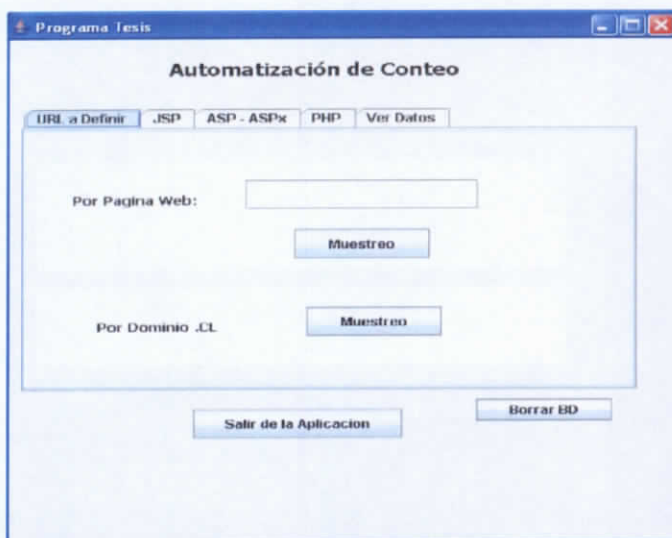


Figura 20: Automatización Explotación.

RESULTADOS PRUEBA 11.

- Cantidad de Páginas JSP observadas en un periodo de tiempo de 6 meses: 37 sitios con contenido JSP
- Cantidad de Páginas ASP o ASPx observadas en un periodo de tiempo de 6 meses: 63 sitios con contenido ASP o ASPx.

PRUEBA	J2EE	.NET
Explotación Actual		
Importancia de Sitios Internet		
Importancia de Sitios Nacional		
Proyecciones a Futuro		

Tabla 18: Tabla de Evaluación 3: Explotación y Uso.

4.4.4. EJECUCIÓN Y RESULTADOS PRUEBAS DE DESARROLLO.

RESULTADOS PRUEBA 12:

Para limitar de alguna manera la comparación del número de líneas, se dejaron dos tipos de aplicaciones Web, por una parte ver simplicidad con páginas de saludo en ambas plataformas, y complejidad al desplegar datos desde una base de datos. Estas fueron:

Para las líneas de código se tiene:

- Saludo.jsp:8 líneas. 3 Tag JSP.
- Saludo.aspx:16 líneas.7 Tag ASPx.
- Cargar.jsp:45 líneas .Mayor Cantidad TAG JSP.
- Cargar.aspx: 33 líneas. Menor Cantidad TAG ASPx.

RESULTADOS PRUEBA 13:

Para Netbeans se tiene:

- 11 menús generales (texto)
- 150 Submenus a partir del menú principal

- Ofrece impresión HTML del código
- Manejo de errores a nivel medio
- 26 objetos en ejecución
- Toolbox gráfico para aplicaciones de escritorio no web
- JSP trabaja en ambiente XML perfectamente

Para Visual Studio.NET se tiene:

- 10 menús generales (texto)
- 138 Submenus a partir del menú principal
- Toolbox gráfico para aplicaciones de escritorio y web
- 50 objetos (Ejecución)
- 6 toolbox de herramientas
- Facilidad de uso de objetos
- Manejo de errores a nivel medio

Aunque no se definió una prueba puntual para los requerimientos mínimos de cada plataforma para la instalación de sus respectivos IDE's y la ejecución de los códigos generados, a continuación se presentan dos tablas de requerimientos mínimos considerados en esta etapa de desarrollo.


















PRUEBA	J2EE	.NET
Líneas de Código simple		
Líneas de Código Complejo		
Interfaz de Usuario		
Ambiente XML		
Acceso a Código Fuente		
Manejo de Errores		
Facilidad de Diseño		
Herramientas Extras		
Funcionalidad Menús		
Requerimientos Mínimos		

Tabla 19: Tabla de Evaluación 4: Pruebas de Desarrollo.

4.4.5. EJECUCIÓN Y RESULTADOS PRUEBAS DE USO DE LICENCIAS.

RESULTADOS PRUEBA 14.

Luego de la lectura de ambas licencias sobre los IDE's de desarrollo Netbeans y Visual Studio, además de su tipo de licencias y sus costos, se agregaron los costos de otros programas e IDE's relacionados. Para esto se obtuvo:

Para costos de J2EE, se tiene:

Netbeans: IDE de desarrollo licencia libre, costo 0 uso educacional. Para poder desarrollar aplicaciones comerciales, se debe acceder a otro tipo de licencia.

JBuilder: IDE de desarrollo licencia comercial, costo Borland JBuiler

Cant	Descripción	Vr. Unitario	Valor Total
	Precios JBuilder 2008		
1	JBuilder 2008 professional-(Academic)	US\$ 109	US\$ 1090
1	JBuilder 2008 professional	US\$35	US\$35
Nota: 1090 usd.			

Para costos de .NET se tiene:

Cantidad	Descripcion	Valor Unitar	Valor Total
1	LICENCIA DE VISUAL ESTUDIO PROFESIONAL 2008 MSDN BENEFITS/SA *Visual Studio Professional 2008 *Visual Studio Tools for Office *Visual Studio 6.0 - 2008 Enterprise *Visual Studio 6.0 - 2008 Professional *Visual FoxPro 9.0 *Virtual PC OPERATING SYSTEMS (a) *SQL Server Developer Edition MSDN Library MSDN Online Concierge MSDN Managed Newsgroups *Visual Studio.Net *Visual Fox Pro 7.0 *Visual Source Safe 6.0 *Desarrollo de dispositivos inteligentes *Visual Studio SDK	594.00	594.00









PRUEBA	J2EE	.NET
Claridad de la Licencia		
Producto Licencia Libre		
Costos Productos Comerciales		
Nuevas Políticas de Costos		
Tendencias de Precios		

Tabla 20: Tabla de Evaluación Licencias.

4.4.6. EJECUCIÓN Y RESULTADOS PRUEBAS DE COMPATIBILIDAD.

RESULTADOS PRUEBA 17.

Para la migración del sistema JSPMysql a ambiente Linux se tiene:

La migración del sistema de conexión a base de datos Mysql desarrollado en un ambiente Windows para ser llevado a algún tipo de ambiente Linux presentó:

Más que llevar la aplicación Web a un sistema operativo Linux, se acordó llevar esta aplicación a un Servidor Web

con ambiente Linux, en este caso servidores con sistema operativo OpenBSD.

La migración consistió en copiar el proyecto generado en ambiente Windows en el IDE de desarrollo Netbeans y ver qué resultados se obtenían en primera instancia, estos fueron:

- No visualización de imágenes de la aplicación.
- No generar las operaciones ofrecidas por la aplicación.
- Problemas con conector JDBC.

Para esto se tuvo que realizar un proceso de investigación para poder corregir dichos errores. En el proceso de depuración, para el caso puntual de las fotos solo se tuvo que re-direccionarlas a un directorio que pudiera acceder el Servidor Web con esto corregido se pudo acceder sin problemas.

Para poder correr el conector JDBC MySql este tuvo que copiarse dentro de la carpeta lib dentro del espacio del servidor, desde ahí el Servidor Web Apache- Tomcat puede acceder al conector para ejecutar las tareas asignadas.

PARA .NET:

- Por el momento es imposible migrar aplicaciones .NET a ambientes Linux.

Cabe mencionar el proyecto Mono, una iniciativa de desarrollo abierto, que está trabajando en desarrollar una versión basada en Linux de código abierto de la plataforma de desarrollo de Microsoft .NET. Su objetivo es permitir a los desarrolladores de Linux construir e implementar aplicaciones .NET multiplataforma. Este proyecto implementará varias tecnologías desarrolladas por Microsoft que han sido enviadas al ECMA (*European Computer Manufacturers Association*) para su estandarización".

RESULTADOS PRUEBA 18.

- En la ejecución de páginas JSP, en los distintos Browser o Navegadores de Internet se observó lo siguiente:

Para Internet Explorer: no se presentan mayores notoriedades.

Para Mozilla: se observan problemas con imágenes.

- Algunas diferencias en cuanto a formatos utilizados.
- Leve cambio en tonalidades.
- TAG <jsp> sin mismo formato.
- En la ejecución de páginas ASPx, en los distintos Browser o Navegadores de Internet se observó lo siguiente:

o Para Internet Explorer: no se presentan mayores notoriedades.

o Para Mozilla: formato botones distintos.

Problemas de alineación.

No ejecución de acciones (no reconocería aspx).

PRUEBA	J2EE	.NET
Migración del Sistema		
Compatibilidad con Browser		
Formato Único (botones, etc.)		
Ejecución de Códigos		
Despliegue de Imágenes		

Tabla 21: Tabla de Evaluación Pruebas de Compatibilidad.

4.5. CONCLUSIONES DE PRUEBAS REALIZADAS.

En este apartado se presentan las conclusiones obtenidas durante la etapa de pruebas desarrolladas para esta evaluación que junto con la investigación realizada y el estudio personal de ambas tecnologías, se puede dejar ciertos aspectos más claros que en un principio del trabajo. En este punto del proyecto, se pueden notar las verdaderas ventajas y desventajas de estas plataformas de desarrollo las cuales se han ido explicando conforme el trabajo ha ido avanzando.

Al ya haber finalizado esta etapa de evaluar a J2EE y .NET en aplicaciones Web en las distintas áreas definidas, teniendo los resultados claros y explicados, se puede empezar a nombrar superioridades e inferioridades, ventajas y desventajas de ambas tecnologías, tanto considerándolas individualmente, como también desde una postura comparativa.

Dentro de estas conclusiones se hará hincapié en las claras ventajas aportadas por cada tecnología y qué aspectos se deberían considerar antes de emitir juicios sobre cada una de éstas. Dentro de esta misma parte de la evaluación de las plataformas se definen los logros

obtenidos a la problemática que se planteó desde un principio para fundamentar este trabajo. Así se podrá separar y mencionar correctamente cuales son las superioridades presentadas por cada tecnología con fundamento y sin caer en fanatismos o preferencias.

De este modo, al realizar una comparación y querer manifestar los atributos de cierta tecnología, que para opinar, al menos se debe conocerla para poder dar fundamentos válidos a considerar, sin este conocimiento, cualquier comentario realizado no pasará de ser solo una opinión y no podría ser considerado como un medio válido de referencia.

4.5.1. CONCLUSIONES DE PRUEBAS DE APRENDIZAJE.

Luego de concluir las pruebas relacionadas con el Área de Aprendizaje y especialmente en el área de inicialización a las plataformas, por medio de la búsqueda de sitios en Internet relacionados tanto en el área de J2EE y .NET, podemos concluir que estos criterios no influyen bastante en una primera etapa de conocimiento de las plataformas. Teniendo Internet como recurso, difícilmente un tema no habrá sido evaluado o considerado por alguien que manifieste algún tipo de interés y haya documentado

información que permita a otros usuarios poder interiorizarse en los temas propuestos.

Otro tema es que estas plataformas, al ser mundialmente conocidas y utilizadas, la disponibilidad de información es en grandes cantidades, que con sitios de diferente orientación, realizando una buena búsqueda, se puede encontrar lo necesario para suplir las necesidades en cuestión.

En cuanto a la curva de aprendizaje, por un lado J2EE, presenta una curva de aprendizaje menos alentadora. Por su parte .NET ofrece una curva de aprendizaje mucho más agradable, lo que hace que el usuario pueda desarrollar aplicaciones de diferentes tipos en cortos periodos de tiempo. Dentro del consumo de información referida sobre estos temas, como lo son libros, revistas, software, suscripciones en foros y todo lo relacionado con ambas plataformas de desarrollo demuestran el mayor interés sobre la tecnología planteada por Microsoft que por el estándar J2EE. Por una parte los software .NET ofrecidos, en la mayoría de las listas consultadas, las ponen en primer lugar. La documentación de manuales y guías de referencia hace ver un mayor nivel de perfeccionamiento en esta plataforma lo que en sí da una orientación al

usuario en general que aún no sabe que especialización será mejor para éste, J2EE o .NET.



EVALUACIÓN	EXPLICACIÓN	PLATAFORMA SOBRESALIENTE
Información Disponible	No se observa ninguna ventaja de una plataforma con respecto a la otra.	Microsoft .net 
Información Disponible en Español	No se observa ninguna ventaja de una plataforma con respecto a la otra.	Microsoft .net 
Curva de Aprendizaje	.NET sobresale por presentar aplicaciones funcionales en menor tiempo.	Microsoft .net
Consumo de Material Relacionado	.NET supera con creces a J2EE en cuanto al material buscado, descargado y pagado por él.	Microsoft .net
Acceso a Software Relacionado	Los Software en los primeros lugares de venta en la Web y tiendas de investigación dan a .NET como una de las más accedidas por los usuarios.	Microsoft .net

Tabla 22: Tabla de Resultados Aprendizaje J2EE v/s .NET

4.5.2. CONCLUSIONES DE PRUEBAS DE RENDIMIENTO.

Luego de desarrollar las pruebas de rendimiento, al cotejar los datos obtenidos, se puede concluir que para cada plataforma esta área de evaluación es un factor importante y valioso a la hora de compararlas. Cada una se ha preocupado de desarrollar tecnologías que permitan

llevar esto de mejor manera, considerando los avances de la competencia para no quedarse atrás.

Para el despliegue de aplicaciones ninguna plataforma mostró una ventaja sobresaliente sobre la otra. La estabilidad para páginas Aspx de .NET y JSP para J2EE no presentaron grandes retardos al incrementarse el número de usuarios que acceden a este tipo de páginas con contenidos simples, con la salvedad de leves superioridades de JSP por la simpleza de los códigos generados.

Para poder presentar tiempos semejantes cabe destacar el funcionamiento de los servidores Web correspondientes a cada tecnología, Apache-Tomcat para J2EE e IIS para .NET, los cuales presentan lo necesario para el correcto funcionamiento de este tipo de aplicaciones. Estos Servidores Web proporcionan métodos de estabilidad que permiten distribuir la carga de los sitios en función del hardware utilizado, lo que hace un mejor funcionamiento en cuanto a tiempos de respuesta.

Para el manejo de operaciones de bases de datos, tanto la tecnología JDBC como ADO.NET, presentan funciones que permiten acceder correctamente a los datos en tiempos considerados "normales". Al llevar estos funcionamientos

de manera local a remota, a pesar que los tiempos de respuesta aumentan, éstos no presentan mayores ventajas una sobre la otra. Con salvedades como que Mysql funcionó de forma más expedita con J2EE que con .NET y, de la misma forma, Postgres presentó mejores tiempos con .NET esta área no deja claro algún tipo de conclusión radical.








EVALUACIÓN	EXPLICACIÓN	PLATAFORMA SOBRESALIENTE
Retardo de Aplicaciones	Las dos plataformas presentan los mismos retardos al acceder una mayor cantidad de usuarios. En repetición de mensajes presentan el mismo tiempo de ejecución.	 
Manejo de Bases de Datos Locales	Para manejo de Bases de Datos ambas plataformas presentan buenas funcionalidades. Tiempos de respuesta mejores por J2EE y simpleza de código, aunque más complicado. .NET más simple el desarrollo de estas aplicaciones de manera local se presenta de mejor forma. MySql se comporta mejor con J2EE y Postgres con .NET	 
Manejo de Bases de Datos Remotas	Para los sistemas anteriores, ejecutados de forma remota estos presentan semejantes funcionamientos que de manera local.	 
Uso de Recursos	.NET ofrece resultados óptimos con menos recursos.	

Tabla 23: Tabla de Resultados Rendimiento J2EE v/s .NET

4.5.3. CONCLUSIONES PRUEBAS DE EXPLOTACIÓN Y USO.

Luego de concluir las pruebas sobre esta área de evaluación se puede decir que en cuanto a la explotación y uso de estas dos tecnologías, si se observan algunas diferencias marcadas.

Por un lado, luego de estar presente por muchos años en el mercado, el tipo de aplicación Web de J2EE, JSP, se ha ido degradando en su uso, obteniendo menos resultados con respecto a su similar ASP o ASPx.

Algunas de las razones de esto, es el nivel que debe presentar tanto el desarrollador Web, como el hardware necesario para la creación de estos sitios, razones que para muchos influyen en cambiar de aplicación o implementar soluciones más baratas. Por otra parte el estándar J2EE ha ido evolucionando vertiginosamente en el mercado de aplicaciones Web, con la aparición de los Web Services, ofreciendo a un nivel más elevado de usuarios y organizaciones sus macro-productos de desarrollo, como la línea WebSphere en la cual se pueden crear portales de aplicación y múltiples servicios en los cuales las páginas JSP vendrían a ser solo una pequeña parte del

grupo de funciones y aplicaciones incluidas en este tipo de software de desarrollo, eso sí a costos tanto económicos como de equipamiento bastante elevados, lo que hace que el usuario común este lejos de poder adquirirlos.

Por otro lado .NET y sus páginas dinámicas ASPx, se pueden apreciar continuamente al navegar por Internet y notar el tipo de pantalla que está desplegando el navegador, en éstas se pueden apreciar que el uso de estas extensiones es común y más presente que para las páginas dinámicas J2EE.

Dentro de los estudios analizados se pueden sacar ciertas características sobre estos criterios, pues, por ejemplo, el hecho de que los Servidores Web manejen dominios y no las dirección IP de los sitios hace que no se pueda acceder directamente al sitio por la dirección IP de ésta, sino solamente con su dominio o URL, lo que hace que se necesite conocer el dominio para poder saber en qué tecnología está desarrollado.

Otro hecho a considerar es que muchos estudios realizados sobre la explotabilidad de páginas dinámicas no suele ser objetivo, por ejemplo el hecho de que un sitio Web este

desarrollado bajo tecnología J2EE en páginas JSP con link a 30 páginas del mismo tipo, y otro sitio corresponda a ser realizado bajo ASPx de Microsoft y estos hagan referencia a 50 links de páginas desarrolladas de la misma manera, no debería contabilizarse el hecho de que existan 50 páginas ASPx y 30 JSP, pues lo que realmente interesa es saber cuántos sitios utilizan estas tecnologías, y no la cantidad de páginas sobre éstas, al corregir esto se podrá saber exactamente cuántos sitios manejan un tipo de tecnología en particular.

Además de estas conclusiones se menciona el hecho de la nula interacción de los sitios accedidos para con contenidos ASP con JSP, o sea, el existir una página tipo JSP que envíe datos a una ASP, pues ambas no cuentan con el mismo Servidor Web que las pueda desplegar, esto en ambientes Windows con Servidores Web realizados por esta empresa hace difícil que sea cierto, no ocurre lo mismo con otros software de este tipo comerciales realizados por otras empresas que si ofrecen esta capacidad.

En ambiente Linux esto se podría simular accediendo a la instalación de todos los paquetes de Catalina-Apache-Tomcat para la ejecución de páginas dinámicas JSP y ASP, incluso con el soporte a páginas PHP.








EVALUACIÓN	EXPLICACIÓN	PLATAFORMA SOBRESALIENTE
Curva de Explotación	<p>Las dos plataformas presentan un nivel de explotación semejante al momento de salir al mercado. Pero en la actualidad sobresale la utilización de de páginas ASP y Web Forms ASPx.</p>	
Explotación	<p>Por su tecnología .NET es muy superior, dejando a J2EE en un mercado global de aplicaciones robustas de servicios Web.</p>	
Distribución	<p>Se puede observar la utilización de ambas tecnologías para sitios de importancia y relevancia en el medio.</p>	 
Cantidad de Servidores Web	<p>Para este punto no hay ningún tipo de variable que destaque una plataforma sobre la otra.</p>	 
Capacidad de Servidores Web	<p>A través de distintos servidores Web, TOMCAT instalado en un sistema Operativo Linux, da a J2EE una ventaja sobre .NET al poder ejecutar correctamente un mayor tipo de páginas dinámicas (ASP, JSP y PHP).</p>	

Tabla 24: Tabla de Resultados Explotación y Uso J2EE v/s .NET

4.5.4. CONCLUSIONES PRUEBAS DE DESARROLLO.

Para realizar conclusiones dentro del área de evaluación

Desarrollo, se debe notar el hecho que algunas consecuencias se generan a partir de los resultados obtenidos en las pruebas definidas en esta área, y a la experiencia propia desarrollando aplicaciones Web bajo ambas tecnologías.

El considerar las líneas de código como un factor influyente se debe a que en el ambiente de programación, este es un punto analizado y considerado (revisar bibliografía). Los resultados obtenidos reflejan que la utilización de TAG tanto JSP como ASP, hacen ver que a mayor utilización de funciones en ambas tecnologías, los códigos HTML son mínimos, y que éste solo debe darle la interpretación al Browser para que pueda desplegar la información, de lo demás se encargará el Servidor Web.

Para las aplicaciones de saludo y cargar datos de una base de datos, las líneas de código arrojadas no devuelven datos concluyentes para hacer de este cotejo o una característica que marque superioridad. Ambas plataformas presentan un nivel aceptado y semejante a la hora de comparar aplicaciones similares en ambos IDE's de desarrollo.

Con respecto a los IDE's analizados puntualmente se puede

concluir que Netbeans a pesar de ser un entorno de desarrollo excelente para aplicaciones JAVA, dista mucho de su semejante Visual Studio.NET, para el desarrollo de aplicaciones Web. Otras herramientas J2EE más parecidas a Visual Studio.NET, distan en precios y objetividad.

A pesar de esto, ambos IDE's presentan las funcionalidades necesarias para el desarrollo de aplicaciones Web, aunque para .NET se presenta de una mejor manera, lo que hace que la creación y desarrollo de éstas puedan ser realizadas por un usuario común y no necesariamente un programador o desarrollador Web.

Directamente en el desarrollo de aplicaciones, el manejo de errores es llevado con mayor objetividad por J2EE, que por .NET, las respuestas a estos errores se presentan de manera más clara y concisa lo que posibilita una mejor y más rápida corrección de los códigos. La accesibilidad del código fuente para ambas tipos de páginas dinámicas Web, se diferencia por el hecho de al acceder a páginas ASP, estas presentan un mayor nivel de información indeseada que JSP.

Otro hecho a considerar como conclusión son las ventajas extras que estos IDE's de desarrollo ofrecen, como el

hecho particular de presentar los códigos en otros formatos como Netbeans, el cual presenta los códigos JSP como páginas HTML, lo que brinda un buen elemento extra de desarrollo.

Para terminar se puede decir que ambas plataformas presentan funcionalidades extras superiores a la otra en el área de evaluación de Desarrollo, por un lado J2EE destaca por su simplicidad de códigos y herramientas extras, .NET sobresale por la facilidad de diseño y desarrollo de aplicaciones Web.








EVALUACIÓN	EXPLICACIÓN	PLATAFORMA SOBRESALIENTE
Líneas de Código	Las dos tecnologías arrojan un número semejante y considerado para el desarrollo de aplicaciones Web. Ambos tipos de TAG se consideran a un nivel mayor por parte de .NET en cuanto a diseño. Y mayor número de líneas de código más que TAG por parte de JSP.	
Herramientas Extras	.NET facilita el diseño y desarrollo de aplicaciones Web, J2EE presenta un número de herramientas extras que facilitan el trabajo del programador.	
Facilidad de Desarrollo	En lo que respecta a los IDE's seleccionados, .NET presenta uno de los mejores diseños para la facilidad de aplicaciones Web que J2EE, facilitando el desarrollo, mejorando tiempos de creación y ejecución.	
Claridad de los Códigos	Al acceder a los códigos J2EE da menos información a personas ajenas a éste.	
Manejo de Errores	Se puede considerar a J2EE con su IDE de desarrollo NetBeans como un mejor manejo de errores, al indicar claramente cual vendría a ser éste.	
Atractivo del IDE de Desarrollo	.NET ofrece un poderoso lenguaje de codificación y un gran atractivo visual.	
Necesidad de Componentes	Para el desarrollo de aplicaciones Web con manejo de datos, las dos aplicaciones necesitan de componentes extras	

Tabla 25: Tabla de Resultados 4: Desarrollo J2EE v/s .NET

4.5.5. CONCLUSIONES PRUEBAS DE USO DE LICENCIAS.

Al comenzar a definir las conclusiones obtenidas bajo esta área de evaluación, se debe hacer notoriedad en los dos hechos más fundamentales abarcados bajo estos criterios de evaluación. Primero separar el uso de productos libres de licencias, incluyendo los permisos otorgados sobre los códigos y claridad de estas licencias. Posteriormente el hecho de los costos de acceso a herramientas profesionales bajo estos dos estándares de desarrollo.

Al referirse a las licencias otorgadas tanto sobre las tecnologías J2EE y .NET y sus respectivos IDE's de desarrollo utilizados en este estudio, se puede mencionar que para la mayoría de los productos de licencia libre para J2EE tienen cláusulas sobre qué pasa con programas desarrollados bajo sus herramientas, que al ser estos comerciales, e incluir a Netbeans, por ejemplo, como herramienta de desarrollo, debe accederse a un tipo especial de licencia, hecho parecido a lo que ocurre con el motor de base de datos MySql. Lo que hace que al querer lucrar con herramientas de licencia libre, éstas pongan sus restricciones para evitar el uso indebido de la tecnología. Por parte de .NET el adquirir una licencia

pagada sobre sus productos, no impide que con éstas herramientas pueda generarse nuevas aplicaciones con nuevas licencias, un hecho a favor para la tecnología .NET.

El otro punto fundamental dentro de esta área de evaluación es, la consideración de los costos monetarios adquiridos al acceder a herramientas profesionales de desarrollo. Para J2EE, a pesar de presentar IDE's libres de costo con Netbeans mencionado anteriormente, al querer acceder a herramientas visuales, semejantes a entornos de desarrollo ofrecidos por Microsoft u otras empresas que realizan software de este tipo, los precios se "disparan", elevándose a cifras que difícilmente un usuario normal o mediana o pequeña empresa pueda acceder, lo que hace que la orientación este claramente hacia empresas y corporaciones importantes las cuales están dispuestas a otorgar todas las funcionalidades necesarias para que el usuario potencial pueda acceder sin problemas a todo lo soportado por dichos productos.

Sin embargo .NET, pese a ser un producto pagado en todas sus líneas, éste al ser comparado con herramientas semejantes J2EE lo posicionan como una mejor alternativa para el usuario común.





EVALUACIÓN	EXPLICACIÓN	PLATAFORMA SOBRESALIENTE
Claridad de Uso de Licencias	Ambas licencias no presentan consideraciones engorrosas que generen algún tipo de interpretación ajena a las que éstas presentan.	
Consideraciones paquetes educativos	Solo J2EE considera la descarga gratuita de muchos de sus productos e IDE's, de desarrollo incluso de máquinas virtuales de acceso	
Costos de herramientas simples	Por contar con paquetes educativos o considerados de licencia libre J2EE ofrece sus productos gratis, mismos a los que se puede acceder desde internet.	
Costos por aplicaciones creadas	La licencias de .NET permiten al usuario desarrollar sus propias aplicaciones comerciales sin necesidad de acceder a una licencia especial. Este hecho contrasta con J2EE que para cierto tipo de aplicaciones se debe adquirir licencias extras para incluir a sus productos y puedan formar parte de una aplicación comercial.	

Tabla 26: Tabla de Resultados Uso de Licencias J2EE v/s
.NET

4.5.6. CONCLUSIONES DE PRUEBAS DE COMPATIBILIDAD.

Para dar conclusiones en cuanto a las pruebas desarrolladas para evaluar la compatibilidad se comentará sobre una de las áreas de evaluación más analizadas el hecho de poder migrar aplicaciones, la Portabilidad.

En cuanto a la portabilidad, un concepto importante dentro de las aplicaciones Web, en J2EE se considera el lema de "escribelo una vez, ejecútalo en cualquier parte", haciendo referencia a lo independiente del lenguaje en cuanto a la plataforma de hardware o sistema operativo utilizado, se puede comentar que esto es particularmente cierto, luego de poder migrar una aplicación de un sistema operativo Windows con el servidor Web Internet Information Server y ser migrado a OpenBSD con Apache - Tomcat como servidor Web y, que ésta, siguiera ejecutando correctamente luego de pequeñas modificaciones hace que en este sentido J2EE supere en este ámbito a la plataforma de desarrollo de Microsoft. En cuanto a ésta, continua con su voluntad de apoyar solo a Windows, por lo que, por el momento Microsoft .NET funcionará únicamente en plataformas basadas en Win32.

Apoyándose en el estudio de Ricardo Chamorro, se concluye además que a pesar de la gran portabilidad de J2EE, existe el problema éste es un estándar y no un producto en sí. Este hecho, que facilita la adopción de esta tecnología por parte de varios fabricantes, también conlleva que las implementaciones de J2EE no son 100% compatibles entre sí, ya que cada vendedor ha realizado su propia interpretación del estándar y ha añadido nuevas características que no tienen por qué incluir el resto de competidores. A su vez este documento entrega 3 posturas en cuanto a dicha portabilidad:

PRIMERO: Si el software o aplicación Web a desarrollar que una empresa madura, especialmente en una de ámbitos de negocios o desarrolladora de software de este tipo, no puede cerrarse solo a una arquitectura, restringiendo a los usuarios a desarrollar aplicaciones J2EE en ambientes Linux/Unix o pensar que los clientes solo acepten a Windows y .NET como solución.

SEGUNDO: Si los interesados en desarrollar y/o utilizar aplicaciones Web utilizan sistemas operativos Windows, estos tienen la ventaja de que pueden desarrollar tanto en J2EE como .NET, ya que ambas soluciones se ejecutan en estos sistemas operativos sin problemas.

TERCERO: si la empresa, cliente u usuario dispone de los recursos, tanto técnicos, económicos, de entorno y capacidades, la elección de una de estas plataformas de desarrollo no debería generar algún problema ya que ellos manejan sus necesidades y sabrán tomar una decisión adecuada.

También se puede incluir dentro de la Compatibilidad, que más que ser un problema de migración, puede ser considerado un problema monetario, por el amplio costo que genera el dedicarse por completo a una de estas plataformas, es por esto que J2EE siempre será discriminada a pesar de su ventaja de portabilidad, por sus muchas veces excesiva necesidad de recursos, tanto de hardware como humanos, y en ámbitos comparativos de empresas grandes, el coste monetario se hace presente con fuerza. Por el lado de Microsoft a pesar de que puede consumir menos recursos que J2EE su incapacidad de migración la hacen inferior en creces en el ámbito de la compatibilidad. A su vez debe considerarse este hecho como importante dependiendo del cliente que este manejando dicha tecnología, pues si sus intereses son otros, esta ventaja sería tomada con menos importancia.

A pesar de esto se debe mencionar el proyecto de grandes proporciones y bastante ambicioso de .NET, que pretende generar MSIL o IL "Microsoft Intermediate Language" (un nuevo código Intermedio), para todo tipo de lenguaje (inclusive JAVA). Esto posibilitaría el poder desarrollar una aplicación en varios lenguajes y, todos estos, inter operar juntos una vez convertidos a MSIL. Obviamente el poder mezclar varios lenguajes en un solo programa y/o aplicación sería algo sin precedente, con esto Microsoft pretende lograr la interoperabilidad ofrecida por Java y llevarla acabo para a través de MSIL a todo lenguaje.



EVALUACIÓN	EXPLICACIÓN	PLATAFORMA SOBRESALIENTE
Visualización	Las dos plataformas no presentan grandes problemas con respecto a la visualización de páginas dinámicas.	
Portabilidad	J2EE presenta portabilidad, dependiendo del tipo de usuario considerado	

Tabla 27: Tabla de Resultados Compatibilidad J2EE v/s .NET

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES.

- Las dos plataformas brindan lo necesario para desarrollar de forma satisfactoria Aplicaciones Web y de esta manera crear soluciones empresariales bajo cada tecnología. Sin embargo se destacan algunos aspectos importantes que se pueden tener en cuenta, a la hora de elegir o escoger entre algunas de las dos.
- Ambas plataformas se basan en especificaciones de seguridades comunes a nivel de mensaje lo que hace posible que la interoperabilidad se mantenga aún al emplearlas. Así que la decisión de utilizar una plataforma en vez de la otra radica al final en los desarrolladores o en la organización que pretenda implementar al momento de desarrollar una Aplicación Web, aunque se cree que existe una ventaja de seguridad de J2EE frente a .NET, ya que, desde

un comienzo Java se fundamentó en un estricto modelo de seguridad a diferencia de .NET, y que por ende la experiencia Java en cuanto a seguridad es mucho mayor.

- Cada una se fundamenta en una arquitectura de código intermedio y de máquinas virtuales para permitir la portabilidad, pero J2EE posee un mayor grado de portabilidad. Si la portabilidad es un factor importante y fundamental es mejor decidirse por una solución J2EE, donde ya esta portabilidad ha sido demostrada a lo largo de la historia. Por otro lado si lo que se pretende desarrollar exige varios lenguajes de propósito programacional diferente, es mejor una solución .NET debido a su independencia de lenguaje, proporcionada por el CLR (Common Language Runtime).
- Tanto J2EE como Visual Studio .Net, son herramientas potentes que ayudan a crear aplicaciones por capas, y a que los desarrolladores se concentren únicamente en la lógica de negocio.

- En el desarrollo de Aplicaciones Web cada una demostró tener los instrumentos necesarios para construir servicios y clientes web. La ventaja obtenida al comparar los dos IDE se encuentra a favor de Visual Studio.
- Aspectos como el IDE marca la diferencia cuando el factor tiempo de desarrollo, es de suma importancia en lo que se pretende realizar, aun cuando se trata de Aplicaciones Web o de cualquier clase de servicio, decidiéndose así por optar por una solución .NET cuando se da esta clase necesidad. Por otro se resalta el hecho que NetBeans es una de tantas herramientas para construir servicios en J2EE. Además cuando se cuenta con aplicaciones con diferentes sistemas operativos, la utilización de Visual Studio se convierte en un problema. Entonces en un escenario como este la mejor solución es J2EE, ya que, la mayoría de las herramientas para construir servicios se pueden instalar en cualquier sistema operativo.
- Al desarrollar Aplicaciones Web J2EE y .NET y consumir los servicios desarrollados para clientes en ambas

plataformas, se comprobó en efecto que los servicios web permiten la interoperabilidad.

Por otro lado, Sun y Microsoft están trabajando mancomunado en algunos aspectos específicos, para aumentar el nivel de interoperabilidad entre las dos.

Si los factores de fiabilidad, transaccionabilidad y seguridad son de suma importancia; se pueden utilizar cualquiera de las dos plataformas, ya que, ambas poseen soporte e interoperabilidad en estos aspectos.

Sin embargo se puede perder cierto grado de interoperabilidad al utilizar servicios web .NET, ya que, estos a veces devuelven datos específicos de la misma plataforma. Si se requiere un máximo nivel de interoperabilidad en los servicios web, que se pretenden desarrollar es mejor optar por una solución J2EE.

Diferentemente, si se sabe que los servicios web únicamente serán consumidos por clientes .NET (teniendo en cuenta también el criterio de desempeño), es una buena opción construir estos

servicios en .NET utilizando DataSet en caso que así se requiera, y ahorrar tiempo en el desarrollo de los mismos.

- Tanto J2EE como .NET ofrecen un buen grado de desempeño óptimo, para realizar soluciones empresariales, y aunque en las pruebas realizadas se nota una superioridad de .NET, sería bueno realizar otras pruebas con otras características, como se menciona en las recomendaciones de este documento.

No obstante de acuerdo a los resultados obtenidos de las pruebas realizadas de desempeño, se recomienda utilizar servicios web .NET para lograr el máximo desempeño.

5.2. RECOMENDACIONES

- Este documento puede ser utilizado para investigaciones futuras donde se estudie o se analicen criterios de comparación en un área diferente a las Aplicaciones Web como por ejemplo componentes o servicios web, entre otros.
- Ampliar este proyecto escogiendo otros criterios de comparación. De igual forma investigar sobre el estado del arte de los servicios web semánticos y su posible implementación e interoperabilidad en cada plataforma.
- Las pruebas realizadas para comparar el desempeño, se hicieron bajo el sistema operativo Windows XP profesional Service Pack 3 y utilizando la herramienta QEngine, se recomienda realizar nuevas pruebas de desempeño bajo otros sistemas operativos en el caso de J2EE para su posterior verificación con los datos arrojados en el sistema operativo Windows. Así mismo utilizar una herramienta diferente a QEngine y utilizar otros servidores aplicaciones en J2EE como: WebSphere (IBM), Oracle Application

Server (Oracle Corporation), WebLogic (BEA) entre otros.

- Estudiar e Implementar otras especificaciones de seguridad a nivel de mensaje para el desarrollo de Aplicaciones Web en aras de fortalecer más los conocimientos en esta área y estar así mejor preparado para enfrentar los retos de seguridad que se originen.
- Implementar laboratorios con los requerimientos necesarios de hardware para la realización de prácticas y pruebas con J2EE y .NET para los estudiantes de la Universidad, mismo que ayudará en la preparación profesional y desarrollo de la investigación en esta área.

BIBLIOGRAFÍA.

- [1] Romero Masis, Edgardo Alberto. Estado del Arte de la Tecnología de .Net. 9na Ed. MCGRAW, Mexico 2007.

- [2] Carpe García, Francisco. Estudio de la plataforma .NET. 2da ed. REYMO, México 2006.

- [3] Garrido Pino Miguel Ángel. Evaluación Comparativa de aplicaciones Web entre J2EE y Microsoft.NET. 3ra Ed. MCGRAW, Mexico 2009.

- [4] Lía Molinari. Arquitecturas Orientadas a Aplicaciones Web. 2da Ed. EDUTEKA, Colombia 2008.

- [5] Samper, José G. Integración del Sistema de Gestión Administrativa y el Sistema de Comercio Electrónico mediante XML. 4ta Ed. Ediciones DMC, Cartagena - Colombia 2002.

- [6] Chávez Cárdenas, Mario Alberto. Introducción a Visual Studio 2008 y .NET. 2da Ed. LIMUSA. Cali 2007.

- [7] Guest Simón. Interoperabilidad de Servicios y Aplicaciones Web. 4 Ed. MCGRAW, Mexico 2008

- [8] Skonnard Aaron. Mejorar la interoperabilidad de los servicios Web. 2da Ed. MCGRAW, Mexico 2005

- [9] Geir Bostad. Infrastructure Development for Scalability Assessment of J2EE Systems. 3ra Ed. Supporting SAM, Florida 2008.
- [10] Geir Bostad. Load Testing and Functional Testing Tool. 5ta Ed. Supporting SAM, Florida 2004.
J. J. Domínguez Jiménez, A. Estero Botaro, I. Medina Bulo, M. Palomo Duarte y F. Palomo Lozano. El Reto de Las Aplicaciones Web para el Software Libre. 2da Ed. MCGRAW, España 2006.
- [11] Maneesh Rampally. Performance Comparison Of Interoperability Between J2EE And .NET Web Services. 2da Ed. MCGRAW, Mexico 2008.
- [12] William Vásquez Romero y Juan Guillermo Rojas. Mecanismos de Control de Acceso en Aplicaciones Web. 4ta Ed. Oveja Negra, Colombia 2006.
- [13] Adrián Delfino, Sebastián Rivero, Marcelo San Martín. Ingeniería de Tráfico en Redes. 6ta Ed. MCGRAW, España 2007.

LIBROS WEB

- [14] Rolando Menchaca Méndez y Félix García Carballeira. Arquitectura de la Máquina Virtual Java.
<http://www.revista.unam.mx/vol.1/num2/art4/>
Mayo 2009
- [15] Daniel Fernández Garrido. Desmitificando Java.
<http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=56>
Mayo 2009
- [16] Nathalie M. Aquino Salvioni. Fundamentos de la Máquina Virtual Java y el Entorno .NET.
http://www.jeuazarru.com/docs/Java_y_PuntoNET.pdf

Junio 2009

- [17] Juan Carlos Frutos Acosta. Análisis Comparativo entre Microsoft.NET y SUN J2EE.
<http://www.ciberteca.NET/articulos/programacion\net/index.html>
 Junio 2009

- [18] Juan Méndez. Proyecto Mono.
http://www.mono-project.com/Main_Page
 Marzo 2009

- [19] Mauricio Zurita Especial Java v/s .NET.
<http://www.comunidadjava.cl>
 Marzo 2009

- [20] Cientec. Análisis Cimientos de las Aplicaciones Web
<http://www.cientec.com>
 Marzo 2009

- [21] Caton Salazar. Diseño de Aplicaciones Web con J2EE.
http://java.ciberaula.com/articulo/disenio_patrones_j2ee/
 Abril 2009

DOCUMENTOS (Varios autores en diferentes medios impresos y/o electrónicos):

- [22] Rodrigo Bustos. PcMagazine. "Java Versus .NET",
 Abril del 2008:32-33.
- [23] Carlos Del Collado. PcMagazine. "Comentarios generales sobre J2EE y .NET y análisis sobre Implementing Sun Microsystems' Java Pet Store J2EE Blueprint Applications using Microsoft .NET", Mayo 2008:20-21-22.

- [24] Francisco Gamboa. Microsoft Framework.NET y los Servicios Web. Universidad Católica d Temuco. [Http://biblioteca.uct.cl/tesis/miguel-garrido](http://biblioteca.uct.cl/tesis/miguel-garrido)
Julio 2009
- [25] Joel Francia. Desarrollo de una Aplicación en tres Capas con VS .NET. Microsoft. <http://www.abcdatos.com/tutoriales/tutorial/18959.html>
Julio 2009
- [26] Escuela Superior de Ingenieros. Aprende Java como si estuvieses en primero. Universidad de Navarra. <http://www.unav.es/biblioteca>
Febrero 2009
- [27] Grupo EIDOS.Luís Miguel Blanco. Programación en Visual Basic.NET
<http://www.educaweb.com/.../enlace-grupo-eidos-0019853.html>
Mayo 2009
- 28 Richard Anderson, Brian Francis, Alex Homer, Rob Howard. Grupo EIDOS.Profesional ASP.NET 1.0. <http://www.gennio.com/.../p2p.wrox.com>
Mayo 2009
- 29 Ricardo Chamorro. Análisis comparativo entre Microsoft.NET y SUN J2EE. Universidad Nacional de Costa Rica. <http://www.unavirtual.una.ac.cr/aulavirtual/>
Junio 2009

GLOSARIO DE TÉRMINOS Y SÍMBOLOS

A

- **Algoritmo:** Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.
- **API:** Interfaz de Programación de Aplicaciones.
- **ASP:** Active Server Pages (ASP) es una tecnología de Microsoft.

B

- **BD:** Base de Datos.

C

- **CLASS:** Extensión generado al compilar una clase Java en Bytecode.
- **CGI:** Common Gateway Interface.

- **Código Gestionado:** Código de un programa de computador que se ejecuta bajo la gestión de una máquina virtual.
- **Código No Gestionado:** Código de un programa de computador que se ejecuta directamente por la CPU del computador.
- **Componente de Software:** son aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otro/s, un entorno funcional requerido por cualquier proceso predefinido. Son independientes entre ellos, y tienen su propia estructura e implementación.
- **Containers:** Contenedores.
- **CPU:** Unidad de Proceso Central.

D

- **Desencriptar:** Proceso inverso de encriptación que permite ver una información que se había protegido.
- **DLLs:** Dynamic Linking Librarys.

E

- **EAR:** Enterprise Archive file.
- **ECMA:** Ecma International es una organización internacional basada en membrecías de estándares para la comunicación y la información. Adquirió el nombre *Ecma International* en 1994, cuando la *European Computer Manufacturers Association* (ECMA), cambió su nombre para expresar su alcance internacional.
- **ENCRIPITAR:** Acción de proteger información para que no pueda ser leída sin una clave.

F

- **FIREWALL:** Corta Fuegos, es un elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas según las políticas de red que haya definido la organización responsable de la red.
- **FTP:** Protocolo de transferencia de Archivos.

G

- **GET:** Comando del Protocolo HTTP que solicita el recurso ubicado en la URL específica.
- **GF:** Servidor GlassFish de J2EE.

H

- **HTML:** Lenguaje de Marcas de Hipertexto.
- **HTTP:** Protocolo de transferencia de Hipertexto.
- **HTTP/S:** Protocolo de transferencia de Hipertexto seguro.

I

- **IDE:** Entorno de Desarrollo Integrado.
- **IU:** Interfaz usuario.

J

- **JAXP:** API de Java para el Procesamiento de XML.
- **JDBC:** Conectividad de Bases de Datos de Java.

- **JDK:** Java Deployment Kit.
- **J2EE:** Java 2 Enterprise Edition.
- **J2SE:** Java 2 Standard Edition.
- **J2ME:** Java 2 Micro Edition.
- **JSP:** Java Server Pages, que en español sería Páginas de Servidor Java. Es una tecnología orientada a crear páginas web con programación en Java.
- **JVM:** Máquina Virtual de Java.
- **JMS:** Java Message Service.
- **JTA:** Java Transaction API.
- **JavaMail API:** API para envío de notificaciones email en Internet.
- **JAF:** JavaBeans Activation Framework.
- **JAXP:** Java API for XML Processing.
- **JAX-RPC:** API for XML-Based RPC.
- **JAXR:** Java API for XML Registries.
- **JDBC:** Java Data Base Connection.

- **JNDI:** Java Naming and Directory Interface.
- **JAAS:** Java Authentication and Authorization Servic.
- **JSTL:** Java Server Pages Standard Tag Library.
- **JAR:** Java archive.

L

- **LP:** Lenguaje de programación.
- **Llave Pública:** Clave que se utiliza para desencriptar información.
- **Llave Privada:** Clave que se utiliza para encriptar una información.

N

- **NNTP:** Network News Transfer Protocol.

P

- **POST:** Comando del Protocolo HTTP que Envía datos al programa ubicado en la URL especificada.

P

- **RAR**: Resource Adapter Archive.

R

- **ROUTER**: Enrutador.

S

- **SAAJ**: SOAP with Attachments API for Java. Suministra una manera estándar de enviar documentos XML sobre Internet desde la plataforma Java.
- **SCHEMA**: Es un vocabulario basado en XML para describir instancias de documentos.
- **SI**: Sistemas de Información.
- **SMTP**: Simple Mail Transfer Protocol (SMTP), Protocolo Simple de Transferencia de Correo.
- **SQL**: Lenguaje de Consulta Estándar o Universal.
- **StAX**: Streaming API for XML. Para tratar los documentos XML de forma más rápida.

- **SUN:** Sun Microsystems, empresa productora de software.

U

- **URI:** Identificador Uniforme de Recursos.
- **URL:** Localizador Uniforme de Recursos.
- **UML:** Unified Modeling Language.

V

- **VS:** Visual Studio.

W

- **W3C:** World Wide Web Consortium.
- **WAR:** Web Application archive.
- **WS:** Web Service(s). Servicio(s) Web.
- **WSS:** WS-Security.

X

- **XML.** Un "schema" describe una clase de documentos, del cual pueden haber varias instancias. Esta relación es parecida a la que hay entre clases y objetos en los sistemas orientados a objetos.

ANEXOS

ANEXO 1.-

MARCO LEGAL

En la realización de este trabajo se utilizaron una serie de herramientas, las cuales ayudaron tanto para realizar el comparativo entre las dos plataformas como para desarrollar el prototipo, cada una de estas herramientas cuenta con una licencia con la cual se rige el desarrollo y la comercialización de todo aquello que se desarrolle en estas.

A continuación se presenta cada una de las licencias de las herramientas o programas utilizados en el desarrollo de este proyecto.

- **NetBeans:** es un producto de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.

- **Java DB:** Java DB se basa en el Apache Derby proyecto de código abierto, el cual se puede descargar es su última versión, pero para la realización de prototipo se utilizo el Java DB que viene integrado con el IDE NetBeans.
- **Visual Studio .NET 2008 Profesional:** Es un producto cuya licencia es privativa y comercial.
- **IIS 6:** No es un producto software que se puede comprar o adquirir como tal en forma individual, es un componente del Sistema Operativo Windows el cual se debe añadir al momento de la instalación del Windows o agregarlo como un componente del SO Windows utilizando el Cd de instalación.
- **AdventNet QEngine:** Es un producto cuya licencia es privativa y comercial, pero se utilizo una versión preview.

ANEXO 2.-**INFLUENCIA AMBIENTAL**

En el desarrollo de este trabajo de grado no se presentó ningún impacto negativo al medio ambiente, ya que los dispositivos, equipos y tecnologías que se utilizaron en el desarrollo tanto del proyecto investigativo, se hizo en forma segura evitando afectar y perjudicar el medio ambiente y al ser humano; de lo cual se pudo hacer buen uso de los mismos, sin dañar el medio que nos rodea.

ANEXO 3.-

COTIZACIÓN VISUAL STUDIO PROFESIONAL



Consultoría en Tecnología Informática
Maestros en el arte y el uso de sus herramientas para
implementar soluciones de sistemas integrales de
Software & Hardware para sus clientes
Con pasión, seriedad y calidad



Guayaquil, Mayo 07 2009

Suficiente : TECNAMIGO
Atención : INGJORGE RUBIO

Excmo cliente
Reciba un cordial saludo, nos es un gusto haberle llegado nuestra cotización

COTIZACIÓN N° C07050910459RC

TECNAMIGO

Cantidad	Descripción	Valor Unitario	Valor Total
1	<p>LICENCIA DE VISUAL STUDIO PROFESIONAL 2008</p> <p>MSDN BENEFITTS SA</p> <p>*Visual Studio Profesional 2008</p> <p>*Visual Studio Tool for Office</p> <p>*Visual Studio 6.0 -2008 Enterprise</p> <p>*Visual Studio 6.0 -2008 Professional</p> <p>*Visual FoxPro 9.0</p> <p>*Virtual PC</p> <p>OPERATING SYSTEMS (s)</p> <p>*SQL Server Developer Edition</p> <p>MSDN Library</p> <p>MSDN Online/Condaarge</p> <p>MSDN Managed Navgroups</p> <p>*Visual Studio .Net</p> <p>*Visual Fox Pro 7.0</p> <p>*Visual Source Safe 6.0</p> <p>*Desarrollo de aplicaciones Inteligencia</p> <p>*Visual Studio .NET</p>	594.00	594.00

HAGANOS CONOCER SUS SUGERENCIAS, QUEJAS y/o COMENTARIOS
A NUESTRO BUZON: paraservir@ingelsistem.com.ec

Forma de Pago: CONTADO Válida/Oferente 8 días	SUBTOTAL	594.00
	IVA	71.28
Asumimos,	TOTAL :	665.28

BITACALJA
SERVICIO AL CLIENTE
edilano@ingelsistem.com.ec



Telefax : 2386692 - 2884257 - 2884084 - 2883928 - 2885487 - 2880381
Email: ingelsistem@ingelsistem.com.ec www.ingelsistem.com.ec
GUAYAQUIL ECUADOR

ANEXO 4 . -**COTIZACIÓN VISUAL STUDIO PROFESIONAL**

Bogotá D.C.,

Señor **Ingeniero Jorge Rubio**

AudiSoft Ltda es una empresa 100% colombiana cuya misión es brindar productos y servicios de la mejor calidad destinados a las empresas y profesionales del desarrollo de software.

Somos un equipo comprometido en satisfacer las necesidades de nuestros clientes, mediante servicios de consultoría, soporte, capacitación y soluciones de software de última tecnología.

Contamos con trece años de experiencia ofreciendo capacitación en diversos productos de software y contamos con instructores certificados en tecnologías Microsoft, Sun Microsystems, Borland y Oracle entre otros.

Nuestra compañía desea agradecer a ustedes la oportunidad de presentar nuestros productos, entre los que se encuentra la reconocida marca Borland, de la cual somos Distribuidores Autorizados en Colombia.

Esperamos que esta propuesta satisfaga sus expectativas, y quedamos a su entera disposición para facilitarle cualquier información adicional.

Cordialmente,

Alejandro Echeverri

Ejecutivo de cuenta

Tels: (1) 2487027 - 2487096

alejandroe@audisoft.com

www.audisoft.com

Bogotá, Colombia

Valor de la Inversión

Cant.	Descripción	Vr. Unitario	Valor Total
	Precios JBuilder 2008		
1	JBuilder 2008 professional-(Academic)	US\$ 109	US\$ 1090
1	JBuilder 2008 professional	US\$35	US\$35
	Nota: todas las licencias (academic)se venden con un minimo de 10, el valor unitario es de 109 por 10 igual a 1090 usd .		
	JBuilder 2008 professional-(Academic)		

Jbuilder 2008 que es la última versión, en su presentación más económica.

Ahora la licencia es electrónica, lo que significa que apenas lo hayas licenciado, te entregamos un URL y unas credenciales para que lo puedas bajar de internet, e instalarlo y registrarlo en tu máquina.

T 004.678 Rubio Peñaherrera, Jorge Bla 005294
R825e Estudio comparativo práctico

BIBLIOTECA

T 004.678 Rubio Peñaherrera, Jorge Bla 005294
R825e Estudio comparativo práctico

