



**Universidad Autónoma del Estado de
Hidalgo**

Instituto de Ciencias Básicas e Ingeniería
Área Académica en Computación



Licenciatura en Sistemas Computacionales

“Tecnologías Java y MySQL para la Implementación
de ítems multimedia en un sistema de evaluación
educativa en línea”

T E S I S

**QUE PARA OBTENER EL GRADO DE
LICENCIADOS EN SISTEMAS COMPUTACIONALES**

PRESENTAN:

Cano Amador Guillermo

Pedraza Vazquez Mario

Asesores:

M. en C. Félix Agustín Castro Espinoza

Dr. Joel Suárez Cansino

Pachuca de Soto Hidalgo, Marzo 2011.



**”Tecnologías Java y MySQL para la
Implementación de Ítems Multimedia en
un Sistema de Evaluación Educativa en**

Línea

por

Guillermo Cano Amador y Mario Pedraza Vazquez

Tesis

Propuesta a la Universidad Autónoma del Estado de

Hidalgo

para obtener el grado del programa

Licenciatura en Sistemas Computacionales

Asesores:

Félix Agustín Castro Espinoza y Joel Suárez Cansino

Área Académica de Computación

Pachuca de Soto, Hidalgo, México, Enero 2011





UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
LIC. EN SISTEMAS COMPUTACIONALES

P.D.L.S.C. **GUILLERMO CANO AMADOR**
P.D.L.S.C. **MARIO PEDRAZA VAZQUEZ**
PRESENTES

Por este conducto les comunico que el Jurado que les fue asignado a su trabajo de titulación denominado **“Tecnologías Java y MySQL para la Implementación de ítems multimedia en un sistema de evaluación educativa en línea”**, y que después de revisarlo en reunión de sinodales han decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación se anotan las firmas de conformidad de los integrantes del Jurado:

PRESIDENTE: M. en C. MARIANO JAVIER POZAS CÁRDENAS

PRIMER VOCAL: M. en C. FÉLIX AGUSTÍN CASTRO ESPINOZA

SEGUNDO VOCAL: M. en C. EFRAÍN FRANCO FLORES

TERCER VOCAL: M. en C. ILIANA CASTILLO PÉREZ

SECRETARIO: L.S.C. JAIR JONATHAN MONROY CEDILLO

PRIMER SUPLENTE: DR. JOEL SUÁREZ CANSINO

SEGUNDO SUPLENTE: L.S.C. ARTURO OCAMPO LÓPEZ

Sin otro particular, reitero a ustedes la seguridad de mi atenta consideración.

ATENTAMENTE
“AMOR, ÓRDEN Y PROGRESO”
Mineral de la Reforma, Hgo., a 18 de febrero de 2011


M. en C. Iliana Castillo Pérez
Coordinadora Adjunta
Licenciatura en Sistemas Computacionales

Instituto de Ciencias Básicas e Ingeniería,
Carretera Pachuca - Tulancingo Km. 4.5, Ciudad Universitaria,
Colonia Carboneras, Mineral de la Reforma, Hidalgo, México, C.P. 42184
Tel. 771 7172000 exts 6103, 6104 fax 2109
ilianac@uah.edu.mx



Índice general

Agradecimientos	VI
Declaraciones	VIII
Resumen	IX
Abreviaciones	XI
1. Introducción	1
1.1. Planteamiento del Problema	2
1.2. Conjetura	2
1.3. Justificación	3
1.4. Objetivos	3
1.4.1. General	4
1.4.2. Específicos	4
1.5. Metodología	5
2. Marco Teórico	7
2.1. Java	7
2.1.1. Características de Java	8
2.2. J2EE y CGI	9
2.3. Servlets y JSP's	10
2.4. Servidor Web	11
2.5. IDE	11
2.6. Diferencias entre lenguajes para desarrollo Web	11
3. Estado del Arte	13
3.1. Formatos de ítems en aplicaciones Web (QuestionMark)	13
3.2. Sistemas de evaluación	16
3.3. Sistemas de evaluación basados en Web	20

3.3.1.	Aplicaciones bajo Open Source sin JSP's	20
3.3.2.	Aplicaciones bajo Open Source con JSP's	21
3.3.3.	Aplicaciones comerciales	22
4.	Herramientas utilizadas para el Desarrollo de las Interfaces	23
4.1.	Lenguaje para el desarrollo de páginas web dinámicas	23
4.2.	Inserción de archivos multimedia en páginas Web	24
4.2.1.	Imágenes	24
4.2.2.	Videos	25
4.2.3.	Sonidos	26
4.3.	Insertar objetos con object	27
4.3.1.	Atributos opcionales	28
4.4.	Archivos a utilizar	29
4.5.	Direccionamiento del archivo multimedia	29
4.6.	Rastreo de sesiones en un sistema web	30
4.6.1.	¿Qué es una sesión?	30
4.6.2.	Establecer y manejar una sesión	31
4.6.3.	Guardar objetos en una sesión	33
4.6.4.	Recuperar objetos de una sesión	34
4.6.5.	Destrucción de una sesión	35
5.	Desarrollo	36
5.1.	Alojar el proyecto Ariya en el servidor Tomcat	36
5.2.	Interface Reactivo Multimedia	41
5.3.	Interface Seleccionador	45
5.4.	Interface de Login	46
5.5.	Interface Evaluador	47
5.6.	Interface Captura Reactivo	47
5.7.	Interface Captura Usuario	49
6.	Conclusiones y Trabajo Futuro	50
6.1.	Ventajas	51
6.2.	Desventajas	52
6.3.	Alcances o Limitaciones	52
6.3.1.	Alcances	52
6.3.2.	Limitaciones	54
6.4.	Trabajo Futuro	54

A. Código para insertar un Blob	58
B. Instalación y Configuración de Plataformas	61
B.1. Instalación de Netbeans	61
B.1.1. Descargar e Instalar JDK 6 Update 14	62
B.1.2. Instalacion de Netbeans 6.7	66
B.2. Instalación de Mysql	74
C. Modelo de la Base de Datos	80

Índice de figuras

3.1. Ejemplo de reactivo en QuestionMark	16
3.2. Prototipo multimedia	18
3.3. Prototipo multimedia de Irvin y Walker	19
3.4. Evaluador de magnetismo, luz y circuitos	20
5.1. Copiado de archivos	36
5.2. Librerías	37
5.3. Archivos copiados dentro del servidor	37
5.4. Ver privilegios	38
5.5. Cambio de atributos	39
5.6. Cambiando atributos de seguridad de Tomcat6	40
5.7. Login del sistema	40
5.8. Interface reactivo multimedia 1	41
5.9. Interface reactivo multimedia 2	42
5.10. Interface reactivo multimedia 3	43
5.11. Interface reactivo multimedia 4	44
5.12. Interface Seleccionador	45
5.13. Interface de Login	46
5.14. Login de acuerdo al tipo de usuario	46
5.15. Interface Evaluador	47
5.16. Interface Captura Reactivo	48
5.17. Interface Captura Usuario	49
B.1. Página de descarga	62
B.2. Ventana para identificarse en la página	63
B.3. Icono para instalación de JDK	63
B.4. Ventana para ejecutar o cancelar la instalación	64
B.5. Ventana de terminos de licencia	64
B.6. Ventana para elegir los componentes a instalar	65

B.7. Estado de la instalación	65
B.8. Fin de la instalación	66
B.9. Pantalla de registro de productos Sun	66
B.10. Pantalla principal de Netbeans	67
B.11. Complementos para descargar con Netbeans	67
B.12. Icono de instalación de Netbeans	68
B.13. Inicio del Asistente de instalación de Netbeans	68
B.14. Ventana para personalizar la instalación de Netbeans	69
B.15. Inicio del Asistente de instalación de Netbeans personalizado	69
B.16. Términos de Licencia para instalación de Netbeans	70
B.17. Selección de Ruta para instalar Netbeans y elegir la ruta de JDK	70
B.18. Selección de Ruta y Usuario para instalar GlashFish	71
B.19. Selección de Ruta para instalar GlashFish v3 Prelude	72
B.20. Selección de Ruta para instalar Tomcat	72
B.21. Características a Instalar	73
B.22. Término de la instalación	73
B.23. Registro del producto	74
B.24. Icono de Netbeans	74
B.25. IDE Netbeans	75
B.26. Página de descarga	75
B.27. Inicio Asistente de MySQL	76
B.28. Fin de la instalación de MySQL	77
B.29. Asistente de configuración de MySQL	77
B.30. Configuración de seguridad	78
B.31. Inicio de instalación de configuración	78
B.32. Mensaje de error	79

Agradecimientos

Principalmente agradezco el apoyo incondicional que mis padres me han brindado a lo largo de mi desarrollo, tanto personal como profesional, ya que es gracias a ellos que he tenido las herramientas necesarias para salir adelante por mí mismo. Estoy en deuda con ellos por todos aquellos consejos que me brindaron y que en su momento me permitieron enfrentar obstáculos difíciles en mi vida. Agradezco a mi hermana por ser tolerante conmigo y brindarme su amistad y apoyo en momentos difíciles, así como en momentos de felicidad y alegría.

A toda mi familia les doy las gracias por creer en mí y ser una fuente de motivación para poder alcanzar todos y cada uno de mis objetivos. La realización de este trabajo no estaría completa sin la ayuda y supervisión de mis asesores Félix Agustín Castro Espinoza y Joel Suárez Cansino, ya que gracias a sus consejos, críticas y observaciones sobre el trabajo lo mejoraron notablemente. Asimismo, agradezco a todos aquellos maestros de la UAEH que me impartieron clases a lo largo de toda mi estancia en esta institución, ya que gracias a sus enseñanzas hoy cuento con los conocimientos necesarios para enfrentarme el mundo laboral.

Por último deseo agradecer a los proyectos FOMIX 2006 CTFHSO-04/04 y FOMIX número 2008/970701, que me permitió participar de tiempo casi completo en el desarrollo de este tema de tesis dentro del marco del proyecto ARIYA ("Framework para Evaluación y Extracción del Conocimiento").

GUILLERMO CANO AMADOR

Quiero agradecer profundamente a mis padres por la oportunidad de vida que me han dado, por todo el amor que he recibido a cada momento, ya que ese sentimiento ha sido un gran estímulo para ir alcanzando muchos momentos de éxito en mi vida. Quiero decirles por medio de estas palabras que cada consejo y esfuerzo que realizan en beneficio de mis hermanos y mío, nos han dejado la mejor herencia que podíamos recibir, que es el conocimiento, y estoy plenamente convencido que todo lo que nos han otorgado nos seguirán llevando por la senda de un porvenir prometedor y lleno de felicidad, por ello siempre estaré sumamente agradecido y quiero decirles que toda esa lucha ha valido la pena y sin duda la llevaré presente con mucho cariño. Gracias por darme esos momentos de amor y alegría, los quiero mucho.

Quiero agradecer a mis hermanos por sus consejos, sugerencias y por apoyarme en todo momento, ustedes han sido de gran ayuda para alcanzar esta etapa en mi vida.

Gracias a los profesores Félix Agustín Castro Espinoza y Joel Suarez Cansino por guiarme en este trabajo.

Agradezco a los proyectos FOMIX 2006 CTFHSO-04/04 y FOMIX número 2008/970701, que me permitió participar de tiempo casi completo en el desarrollo de este tema de tesis dentro del marco del proyecto ARIYA ("Framework para Evaluación y Extracción del Conocimiento").

MARIO PEDRAZA VAZQUEZ

Declaraciones

Este trabajo es una extensión de un proyecto más general llamado ARIYA (“Framework para Evaluación y Extracción del Conocimiento”).

GUILLERMO CANO AMADOR Y MARIO PEDRAZA VAZQUEZ

Resumen

Este trabajo trata acerca de la creación de interfaces gráficas para evaluación académica en línea, desarrolladas utilizando Java para brindar a dichas interfaces un mejor rendimiento y utilidad.

En el presente trabajo, se describe el proceso con el que se logra la creación de las interfaces de un evaluador multimedia en línea. Es importante mencionar que esta labor es parte de un proyecto de mayores dimensiones donde se establece un sitio Web que ofrece un servicio de evaluación de conocimiento, que incluye evaluación clásica y adaptable. Una parte del proyecto general llamado Ariya es el elaborado en la tesis denominada *Análisis, Diseño e Implementación de una base de datos para la administración de reactivos en un evaluador educativo* en la cual se creó el diseño de la base de datos que utiliza este sistema, dicho modelo se presenta en el Apéndice C.

El objetivo de este trabajo está enfocado básicamente al diseño e implementación de las interfaces de un sistema de evaluación clásico en línea. En primer lugar, se presenta una introducción en la que se describen temas tales como: Planteamiento de problema, Justificación, Conjetura, Objetivos General y Específicos, así como la metodología seguida para la culminación del proyecto, que son los elementos que permitirán entender la razón de ser del trabajo. Posterior a esto, se presenta una serie de conceptos útiles para la comprensión de los términos utilizados dentro del escrito, haciendo una comparación entre alternativas para la creación de páginas Web dinámicas y lo que implica cada una de ellas, todo esto en el capítulo denominado Marco Teórico.

Continuando con el trabajo, se encuentra la parte del Estado del Arte, donde se comenta acerca de trabajos realizados que tienen que ver con el desarrollo de interfaces de sistemas de evaluación, tanto monousuario como sistemas en línea, desarrollados ya sea con software propietario o con licencia libre, haciendo un análisis entre las características con las que cuentan, comparándolas con las desarrolladas en la presente tesis. Para la realización del trabajo, se hizo una investigación en la cual se evaluaron las alternativas, para la inserción de multimedia en las páginas Web y el manejo de sesiones dentro de un sitio Web con Java. El punto importante en

este capítulo fue la elección de la mejor alternativa para la inserción de multimedia dentro de las interfaces de evaluación.

Los siguientes capítulos muestran, en primer lugar, la descripción de cada una de las interfaces creadas para la integración del sistema y finalmente se hacen una serie de conclusiones resultantes del desarrollo de este trabajo de tesis, así también se muestran posibles opciones para desarrollos futuros relacionados con el proyecto.

Abreviaciones

API: Application Programming Interface.

ASP: Active Server Pages.

BD: Base de Datos.

CGI: Common Gateway Interface.

ETS: Educational Testing Service.

GNU: General Public License.

HTML: HyperText Markup Language.

IDE: Integrated Development Environment.

JDK: Java Development Kit.

JSP: Java Server Pages.

J2EE: Java 2 Platform, Enterprise Edition.

JRE: Java Runtime Enviroment.

JVM: Java Virtual Machine.

LMS: Learning Management System

MySQL: My Structured Query Language.

PC: Computadora Personal.

PHP: Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools).

SQL: Structured Query Language.

SO: Sistema Operativo.

SOAP: Simple Object Access Protocol.

Capítulo 1

Introducción

La evaluación en línea es una herramienta muy empleada actualmente en sistemas basados en educación a distancia, se puede encontrar una gran diversidad de ejemplos de ella en Internet. Una de las características importantes de este tipo de sistemas es la que definen las interfaces con las que interactúa el usuario. Se sabe que estas interfaces deben cumplir con características que las hagan sobre todo usables. Cuando la evaluación considera formatos de reactivos relativamente simples, por ejemplo que solamente contemplen descripciones textuales de ítems, estas interfaces no resultan complicadas de elaborar, y suelen ser muy fáciles de interpretar, además de que la transferencia de los datos contenidos en el ítem se realiza de una forma relativamente rápida desde el servidor hasta el cliente o usuario, quien se puede localizar en un lugar completamente diferente al de alojamiento del sitio.

Si se trata de construir ítems con un formato más complejo que incluya no solamente texto, sino también recursos multimedia como por ejemplo, audio, video e imagen, entonces el problema de transferencia se hace más difícil. Esto es así por la enorme cantidad de recursos de cómputo que se necesitan para hacer el envío de esta información desde el sitio hasta el usuario. La cuestión aquí radica en encontrar la forma de mejorar este proceso para que el sistema de evaluación de que se trate funcione óptimamente (tiempos de transferencia, que la información esté disponible y libre de errores).

Afortunadamente, la tecnología de desarrollo de sistemas de información ofrece actualmente posibilidades de mejorar el funcionamiento descrito a través de formas diversas, entre las que se pueden mencionar el uso de lenguajes de alto nivel propios para el desarrollo de sistemas Web con páginas dinámicas, un claro ejemplo es la tecnología Java J2EE, el lenguaje PHP, los lenguajes ASP y .NET, etc.

Aunado a esto, estos sistemas pueden interactuar colectivamente con otras tecnologías que permiten hacer diversas acciones, un ejemplo de esto es la interacción con bases de datos, que es lo que principalmente se busca en ambientes Web dinámicos. En este sentido, existe una gran variedad de manejadores de bases de datos con funcionalidades suficientes para cumplir con los objetivos que un sistema Web complejo requiere. Ejemplos de estos manejadores son MySQL, SQL Server, ORACLE, entre otros.

1.1. Planteamiento del Problema

El aprendizaje en línea (e-learning) es un paradigma actual que se sustenta en diversas metodologías, algunas de ellas más eficientes que otras. Un componente esencial de este tipo de sistemas es el evaluador, con el cual se asigna una certificación del conocimiento del usuario. Estos componentes requieren de interfaces sofisticadas que incluyen el manejo de gráficos, audio y video, además de texto. Estas interfaces se construyen con un diseño apropiado que cumpla con características mínimas de calidad, además de que con éstas sea posible crear un enlace vía Web para que se acceda a los recursos del sistema y de esa manera se puedan ejecutar acciones del usuario.

Actualmente se cuenta con un prototipo de sistema evaluador denominado “Framework para Evaluación y Extracción del Conocimiento”, desarrollado bajo el lenguaje PHP con el objeto de controlar la evaluación en línea. La implantación de este sistema deja una serie de deficiencias en la calidad del mismo, debido a la forma no tan dinámica con que trabaja PHP, como lo es la comunicación con el servidor y el cliente, exceso de trabajo al servidor que contiene la base de datos, entre otras. Es por ello que se formaliza en este punto la necesidad de una tecnología que solucione la forma en la cual se presentará dicho sistema ante el usuario, que incluya características de accesibilidad, seguridad y eficiencia.

1.2. Conjetura

Se conjetura que las herramientas basadas en tecnología Java (JSP's y Servlets) permiten que el sistema cumpla con las necesidades de accesibilidad, seguridad, eficiencia y eficacia. La tecnología JSP permite al servidor del sitio del sistema Web responder a las peticiones del cliente de manera rápida, sin ocupar muchos recursos de cómputo. Adicionalmente, estas herramientas facilitan el desarrollo, implantación,

administración y mantenimiento del sistema, facilitando el manejo de recursos multimedia.

1.3. Justificación

La tendencia actual en el desarrollo de sistemas de información hace uso de la tecnología Java en sus diferentes vertientes, un desarrollador debe considerar esto si no quiere quedar fuera del juego; es decir, debe contemplar utilizarla para construir sus aplicaciones. Sin embargo, existen otras alternativas que pueden brindar las mismas funcionalidades, como por ejemplo .NET, PHP, etc. Aún así, Java posee características que la convierten en una mejor opción para el desarrollo de sistemas, algunas de estas características son que Java es multiplataforma y de licencia libre (licencia GNU).

En el aspecto de desarrollo Web, Java es un lenguaje que incluye una librería bastante rica que facilita la programación y el mantenimiento del sistema, ofreciendo ventajas en el rendimiento, velocidad y eficiencia del sitio Web.

Java permite conectividad con diferentes entornos de aplicación tales como manejadores de bases de datos, diseñadores de sitios Web, etc. Java es un sistema multiplataforma que se puede alojar en cualquier sistema operativo, siempre y cuando se cuente con los componentes necesarios para su ejecución. Además de establecer un sistema de seguridad que ofrece tanto al administrador como al usuario del sistema la confianza de que la información que se maneje no será víctima de robo o extravío por parte de terceros, ésto se logra con el diseño y análisis de un esquema de seguridad, el cual es desarrollado en otra tesis del proyecto Ariya. Adicionalmente, no se paga licencia por el uso de la tecnología Java, por lo que los sistemas desarrollados se adquieren a bajo costo.

Asimismo la elección de Java radica que el Framework de proyecto Ariya contará con herramientas de post-análisis como Minería de Datos, que están desarrollados en el mismo lenguaje.

1.4. Objetivos

El trabajo desarrollado ofrece diferentes aspectos propios de un buen sistema, entre los cuales se encuentran la portabilidad, la seguridad, el acceso a base de datos, la

usabilidad, la flexibilidad, la robustez y el ser implantado en servidores con características Hardware y Software deseables para un buen desempeño. Muchos de estos aspectos son cubiertos empleando lenguajes de programación orientados a la Web, como por ejemplo Java (J2EE). Resumiendo, el trabajo está orientado a satisfacer los siguientes objetivos general y específicos.

1.4.1. General

Desarrollar un prototipo de sistema de evaluación en línea utilizando la tecnología Java (JSP's, Servlets), que debe incluir interfaces con contenido multimedia para evaluación e interfaces para la administración del sistema (usuarios, reactivos, presentación de resultados, etc.).

1.4.2. Específicos

Una vez presentado el eje principal de este trabajo, es razonable desmenuzarlo en actividades que permitan alcanzarlo exitosamente, aplicando en este sentido la máxima “divide y vencerás”. En los siguientes puntos, se enuncian gradualmente objetivos que a medida del avance de su ejecución permitirán ir conformando el prototipo del sistema de evaluación en línea que se realiza en el Área Académica de Computación de la Universidad Autónoma del Estado de Hidalgo y que es un pilar importante para trabajos futuros:

1. Desarrollar interfaces para un sitio Web con las que sea posible acceder a una base de datos e ítems y a través de ellas procesar convenientemente la información contenida en dicha base.
2. Desarrollar interfaces que propicien que el usuario tenga una navegación fácil en el sistema; es decir, que lo haga de una forma intuitiva y no confusa.
3. Crear interfaces relacionadas con un esquema de seguridad que garantice que usuarios autorizados acceden a los servicios del sitio, siempre y cuando éstos hayan sido dados de alta previamente en la base de datos otorgándoles una cuenta de acceso.
4. Permitir a las interfaces ofrecer eficiencia en el rendimiento del prototipo del sistema, tomando en cuenta que es de suma importancia propiciarla, no ocupando muchos recursos e incluso cerrar éstos una vez que dejen de usarse, un ejemplo de esto son las conexiones con la base de datos.

5. Documentar el sistema de evaluación ya que este trabajo será de mucha utilidad en el futuro y es la base para la construcción de diferentes tipos de evaluación.
6. Darle flexibilidad al sistema a través de las interfaces, queriendo decir con esto que con ellas se pueda ingresar a través de diferentes navegadores Web.
7. Proporcionar a las interfaces las características necesarias para que den portabilidad al prototipo de sistema de evaluación, dando oportunidad de que éste sea alojado en diferentes plataformas y sitios Web, o capaz de ser puesto a disposición de diferentes tecnologías (LMS's por ejemplo).
8. Construir un prototipo de sistema en línea que contenga una estructura robusta dirigida a satisfacer las peticiones hechas por los líderes del proyecto, esto es que el sistema permita un login, evaluar a una persona y emitir un resultado, administrar ítems o reactivos, administrar usuarios, etc.
9. Implantar este prototipo de sistema en línea dentro de un servidor de aplicaciones. En particular, el servidor está ubicado en el laboratorio VipLab perteneciente al Cuerpo Académico de Computación Inteligente del Área Académica de Computación de la Universidad Autónoma del Estado de Hidalgo.

1.5. Metodología

La obtención de los objetivos se hace a través de diferentes etapas o actividades que incluyen la investigación a fondo de varias aplicaciones que permiten crear sitios Web dinámicos, considerando también las diferencias, ventajas y desventajas de estas aplicaciones. Así, la metodología que se sigue en este trabajo consiste en conocer detalladamente la tecnología Java para el desarrollo de sitios o sistemas Web (J2EE, que incluye Servlets y JSP's, entre otros), instalación de componentes necesarios para el correcto funcionamiento de esta tecnología (servidor Apache Tomcat, IDE Netbeans, JDK, etc.), validación de interfaces y código fuente, entre otros.

Específicamente, los pasos que se siguen en este trabajo de tesis se apegan a los requerimientos solicitados para el proyecto ARIYA y se dividen en consulta o búsqueda de información, instalación, desarrollo, implementación y validación. En lo que respecta a la consulta, se consideran manuales y páginas Web acerca de las definiciones del lenguaje de programación Java, particularmente información acerca de la creación de páginas Web dinámicas con JSP's y Servlets, así como la interrelación de éstos en el manejo de base de datos.

Asimismo, se considera la búsqueda y análisis de alternativas para la inserción de archivos multimedia (imágenes, videos, sonidos) en páginas Web y la investigación y pruebas en el manejo de sesiones dentro de sitios Web creados con Servlets. Finalmente, se investiga detalladamente acerca de los requerimientos para el alojamiento de un sitio o sistema Web en un servidor Apache Tomcat teniendo como plataforma el sistema operativo Linux.

La instalación de componentes es el siguiente paso y aquí se toman en cuenta la instalación del IDE Netbeans y los módulos necesarios a utilizar, la instalación del manejador de base de datos (MySQL) y del servidor de aplicaciones (Apache Tomcat).

Como tercer paso, se realiza el desarrollo de páginas con JSP's y Servlets, así como la creación de base de datos que ayudan a tomar una mejor decisión acerca del diseño de las interfaces Web. Algunas de estas interfaces son páginas en las que se realiza inserción, eliminación y modificación de datos en una base de datos, mientras que en algunas otras se emplean Servlets para mostrar imágenes, audio y video. En este caso, la ruta de la imagen se encuentra almacenada en un registro de una base de datos, aunque existe la posibilidad de emplear Blob's dejándose esta alternativa para un trabajo futuro. Un Blob es un tipo de dato en MySQL que almacena datos en forma de bits, permitiendo almacenar archivos multimedia.

Como cuarto paso, se lleva a cabo el alojamiento del sitio Web en los servidores Ariya01 y Ariya02, localizados en el laboratorio VipLab del Cuerpo Académico de Computación Inteligente. Finalmente, en el quinto y último paso se valida el funcionamiento de las interfaces del sitio, incluyendo en esta validación la corrección de posibles fallas, se valida también el correcto funcionamiento del servidor Apache Tomcat y la conectividad de las interfaces con la base de datos.

Capítulo 2

Marco Teórico

En el desarrollo de las interfaces que conforman el sistema de evaluación en línea, se establecen una serie de tecnicismos que permiten comprender diferentes aspectos del proceso, estando primordialmente entre estos últimos la forma en que funciona un sitio Web y las razones que llevaron a elegir la tecnología Java como herramienta, además de que permiten aclarar el por qué de las ciencias aplicadas en la creación de las interfaces que se desarrollan en este trabajo.

La información que se presenta a continuación muestra el camino completo que se ha seguido desde la decisión misma de tomar Java como la herramienta principal de desarrollo, hasta la especificación de la forma en que los diferentes medios del entorno interactúan con el lenguaje para poder establecer un sitio Web, estando entre estos medios el servidor Apache Tomcat, el cual permite un manejo interactivo de páginas.

2.1. Java

Debido a que Java es un lenguaje potente (cuenta con una variedad de comandos de programación) y robusto, en este trabajo se opta por la elección de Java como lenguaje para programar interfaces que tengan características dinámicas. Su robustez se logra a través de la incorporación de diversos elementos que son de gran utilidad en la implementación de la interacción que el sistema tiene con otras diversas componentes, como por ejemplo una Base de datos principalmente. Toda referencia a Java obliga necesariamente a englobar un entorno sumamente extenso y de gran alcance, afortunadamente la integración de estos diversos componentes se logra en una forma relativamente fácil debido a que Java proporciona todo un panorama rico en programación[TP02].

2.1.1. Características de Java

Para empezar, Java es un lenguaje de programación que posee un sinnúmero de características que permiten realizar aplicaciones de alto desempeño. Inicialmente, su programación orientada a objetos es una característica que diferencia a Java de otros lenguajes de programación de alto nivel; esta forma de programar se refleja en los datos que maneja el sistema, así como en los métodos que utiliza el lenguaje y en las diversas técnicas que el lenguaje tiene para implementar la encapsulación, el polimorfismo, la herencia, clases, etc. Adicionalmente, Java posee un gran caudal de clases ya preestablecidas (API) que ofrece muchas facilidades para el desarrollo de aplicaciones en red, como sockets, además a procesamientos remotos, Servlets, JSP's, entre otros.

Hablando particularmente de las API's de Java que se han empleado en la creación de interfaces, se mencionan especialmente todas aquellas API's que son necesarias durante la elaboración de Servlets. Cuando se emplean estas clases se establecen particularmente acciones muy específicas como puede ser la conexión hacia las bases de datos, o bien la acción en la que se emplea la clase `HttpSession` con la que se rastrea la sesión que generó un usuario de una interfaz muy específica, desde que este usuario ingresa al sitio hasta que se retira de él.

Como preámbulo, se tiene como objetivo implantar el sistema de evaluación en línea usando como plataforma al sistema operativo Linux, pero cabe destacar que la conformación del sitio Web también se puede establecer en cualquier otro sistema operativo debido a que ésta es una particularidad que Java presenta, esta propiedad del lenguaje se ve reflejada en la utilización del formato de su código en archivos que son llamados Bytecodes, que son independientes de la plataforma donde se trabaja. Los Bytecodes son interpretados por una máquina virtual que interactúa sobre el Hardware y Software específico de cada SO. Esta característica permite a Java ser multiplataforma.

Existen más características en Java que por el momento no se utilizan en el desarrollo de las interfaces de este trabajo, pero se considera que en un futuro próximo pueden ser de gran utilidad debido a que permiten incrementar los alcances de una aplicación en comparación con otros lenguajes. Java cuenta con el elemento llamado multihilo, que en esencia se refiere a la capacidad de Java para hacer varias acciones a la vez, lo cual se lleva a cabo por hilos que son creados durante la ejecución. Cada hilo tiene una función en específico, y cada uno de ellos puede hacer un procedimiento por separado o independiente de los demás.

Los 4 párrafos anteriores son fundamentados con lo contenido en la referencia [TP02].

Un ejemplo de la característica multihilo se presenta cuando se contempla una aplicación de red, un hilo se encarga de la comunicación con otro equipo o una base de datos, otro hilo realiza la interacción con el usuario, otro hilo presenta una animación y otro realiza cálculos, todas estas acciones se ejecutan a la vez por diferentes caminos y, como ya se comentó, estos caminos son vías simultáneas llamadas hilos [CSH09].

En la aplicación realizada en este trabajo se manejan varios datos y, por consiguiente, es importante el cuidado de ellos, hablando a nivel de memoria primaria, pero con el lenguaje Java establecemos la seguridad en la ejecución de las aplicaciones y el cuidado de los datos, es por ello que Java omite la administración y acceso a la memoria física por parte del programador o del usuario, para impedir el desbordamiento de datos y la utilización de punteros que generan gran problemática.

2.2. J2EE y CGI

Lo comúnmente realizado en un ambiente Web, particularmente hablando de los CGI, es que generan procesos que se realizan por parte del servidor Web, se puede decir que cada vez que un CGI realiza una petición de información, la devolución de ésta, una ejecución de una estructura de control, etc., realiza ejecuciones de instrucciones en el sistema operativo. Lo anterior motivó aun más la utilización de Java en el proyecto, debido a que la ejecución de código no se establece en instrucciones del sistema operativo sino a través de un intermediario.

Como se mencionó líneas arriba, algo que es muy notorio en Java es que existe una capa lógica que hace creer que la aplicación Java se ejecuta en una computadora real, donde se tiene acceso a sus recursos, esta máquina recibe las instrucciones de la aplicación y genera instrucciones máquina para realizar acciones en el hardware y software [Tur02].

Así, cuando esté puesto en marcha el sitio Web, se presume que no hará uso del sistema operativo, esto permite tener grandes ventajas, como la utilización de poca memoria en el servidor, una interacción más eficiente con el servidor de base de datos y servidor de aplicaciones Web. Esto es más entendible si se explica de la siguiente forma, cuando se utilice por ejemplo un Servlet, éste se queda alojado en

memoria para su posterior utilización y sólo hace la petición una sola vez en la capa intermedia que se denomina Java Virtual Machine, y con un CGI cuando se hace una petición de una página tendrá que realizarse un nuevo proceso en el sistema operativo. Sólo resta hacer una observación y ésta consiste en dejar claro que las acciones mencionadas anteriormente son del lado del servidor.

2.3. Servlets y JSP's

La estructura principal de las interfaces del sistema está dada por dos elementos principales; los Servlets y JSP. Los Servlets son programas que se ejecutan del lado del servidor y actúan como el elemento intermedio entre un buscador y aplicaciones, bases de datos etc. Los Servlets realizan varias funciones entre las cuales se cuentan el recibir las peticiones de un buscador, hacer procesos con la información tomada, dar una respuesta al usuario con un cierto formato para disposición en el buscador [MZ06].

Para hacer esto más entendible al caso práctico de las interfaces que se desarrollan en este trabajo, los Servlets y JSP's se emplean para hacer conexiones a una base de datos de MySQL que aloja reactivos principalmente. Aunado a esto, los Servlets y JSP's se utilizan para obtener datos de la base anteriormente mencionada, estos datos son preguntas, respuestas, complementos multimedia de éstas principalmente, y en un futuro se manejarán datos como usuarios, contraseñas etc.

Una razón más del uso que se le da también a los JSP's consiste en que permiten hacer las mismas acciones que los Servlets, pero de una forma no tan lógica sino que se prestan mucho para cuestiones de diseño y utilización del lenguaje de modelado de Hipertexto. Entrando en una parte más formal, se puede decir que un JSP es una tecnología que se ejecuta del lado del servidor y permite conjuntar otras tecnologías como HTML que principalmente hacen un contenido estático y Servlets que permiten estructurar el contenido dinámico. En esencia un JSP termina siendo un Servlet. La tecnología JSP utiliza etiquetas para manifestar que hará uso de instrucciones del lenguaje Java que permiten realizar acciones en casos específicos como son la conexión a una base de datos, etc., logrando de esta forma una interacción dinámica[Tur02].

2.4. Servidor Web

El servidor característico que se utiliza para hacer aplicaciones Web con Java es Apache Tomcat aunque también se puede hacer uso de Glassfish, WebLogic. En este proyecto se opta por el servidor Apache Tomcat, el cual no ocasiona ninguna problemática y además por su fácil uso. Formalmente se establece que Apache Tomcat es un software de código abierto para desarrollar aplicación con las tecnologías Servlets y JSP de Java, creados por la fundación Apache. Éste programa tiene la función de crear un sitio Web, para alojar páginas Web y BD [VCW04].

2.5. IDE

El IDE utilizado para el desarrollo de las interfaces es Netbeans que permite crear aplicaciones Java, éstas pueden ser simples, de escritorio ó Web; así mismo, aunque Netbeans está enfocado a Java puede servir para cualquier otro lenguaje de programación, tal es el caso de PHP y C++. Netbeans fue de mucha utilidad en la creación de las interfaces para el proyecto de evaluación en línea ya que posee un sin fin de características que conceden programar una aplicación Java Web de manera fácil y sencilla, dentro de este IDE es posible escribir, compilar, depurar y ejecutar programas [Cora].

2.6. Diferencias entre lenguajes para desarrollo Web

Se pueden citar muchas comparaciones con los lenguajes de desarrollo de aplicaciones Web, algunas de ellas ya se han mencionado en líneas anteriores, pero no basta con eso porque también se ha comprendido el funcionamiento de un CGI, además de que contamos con la experiencia de la realización de una aplicación Web sobre esta interfaz, específicamente se habla de PHP. Un CGI que es una interfaz de entrada común, es el elemento intermedio entre el servidor Web y las aplicaciones que se ejecutan en el servidor, estos programas CGI están escritos en diversos lenguajes como C, Perl, PHP, entre otros.

Se presenta un ejemplo de un funcionamiento del CGI en un servidor cuando en una página Web existe un formulario al que se le ingresan datos que posteriormente son enviados por el usuario con el objeto de hacer una invocación y ejecutar un programa, esta invocación la hace el servidor HTTP y de la misma manera pasa los datos que introdujo el cliente al programa. Consecutivamente el programa CGI

genera una respuesta o salida al cliente que por lo general es una página HTML. El típico uso de un CGI es el de manipular a una BD por medio de una aplicación Web donde se manejan métodos GET y POST [Reg].

Enfocándose a un CGI tradicional, se escribirá acerca de PHP que se ha comprobado en este trabajo su gran utilidad. PHP es un lenguaje interpretado de alto nivel, que se encuentra incrustado en páginas HTML y su código se ejecuta del lado del servidor. El código de PHP se encuentra integrado en las páginas HTML, a través de etiquetas que hacen anuncio del inicio y final de las instrucciones PHP, manifestándose de esta manera acciones ante el servidor que son generadas por el código PHP. Con PHP pueden hacerse procesamiento de formularios, contenidos dinámicos en páginas HTML, un ejemplo es una consulta, modificación y otras acciones a base de datos[JM08].

Se puede decir que PHP permite consolidar un sitio Web, pero eso no significa que su funcionamiento sea tan eficiente como lo es con Java debido a que depende de otros elementos como ya se ha comentado anteriormente, cuando se habló del sistema operativo y la interacción con los procesos que genera PHP. Hasta aquí, estas líneas proporcionan un panorama del porqué el uso de Java se toma como herramienta principal en la programación y en la realización de interfaces que conforman el sistema de evaluación en línea.

Capítulo 3

Estado del Arte

En la actualidad existe gran cantidad de programas capaces de administrar cursos acerca de algún tema o tópicos en específico. Para medir el rendimiento de cada alumno que sea usuario del sistema, estos programas contienen sus propios métodos de evaluación. Los sistemas de este tipo son conocidos como LMS (Learning Management System), cuya función es administrar y controlar cursos en línea, así como a los alumnos, profesores o instructores, administradores e instituciones autorizadas para usarlos, de igual manera coordinan y distribuyen las actividades que en general se realizan dentro del sitio Web.

En este trabajo de tesis, resulta importante el componente dedicado a la administración del proceso de evaluación ya que generalmente se construyen incluyendo solamente formatos de ítems bastante específicos que carecen de interactividad con el usuario, que adolecen de falta de dinamismo y que restringen la posibilidad de medir habilidades complejas.

Algunos autores han realizado esfuerzos para integrar mecanismos computacionales sofisticados que resuelvan la problemática planteada en el párrafo previo [Chr]. A continuación, se hace un análisis de las ventajas, desventajas, coincidencias y discrepancias entre algunos de estos trabajos y el tópico que se desarrolla en esta tesis.

3.1. Formatos de ítems en aplicaciones Web (Question-Mark)

Dentro de las pruebas y evaluaciones se puede encontrar con un software muy popular a nivel mundial llamado Questionmark, este software es comercial, desarrollado

en lenguaje ASP, un lenguaje de desarrollo independiente que presume de ofrecer servicios de la más alta calidad, y el protocolo de acceso a objetos simple (SOAP). Dentro de sus evaluaciones se presentan una serie de ítems, que utilizan imágenes, audio y video en la evaluación. Entre los ítems que maneja este software se tiene:

1. **ARRASTRAR Y SOLTAR:** el participante hace clic y arrastra imágenes a una ubicación.
2. **PREGUNTA DE ENSAYO:** el participante contesta escribiendo hasta 30.000 caracteres de texto. Este tipo de pregunta también se utiliza para solicitar opiniones o sugerencias sobre un tema particular.
3. **PANTALLAS CON EXPLICACIONES:** el participante inserta texto o gráficos para que los vea antes de contestar una serie de preguntas.
4. **CARGA DE ARCHIVO:** en ocasiones se les exige a los participantes que completen una asignación que requiere la creación de un documento en forma de archivo de computadora.
5. **COMPLETAR ESPACIOS EN BLANCO:** al participante se le presenta un enunciado donde faltan una o más palabras y completa los espacios con las palabras que faltan.
6. **PUNTO ACTIVO:** un participante hace clic en una imagen para indicar su opción.
7. **MATRIZ DE CONOCIMIENTO:** este tipo de pregunta presenta un conjunto de preguntas de opción múltiple en las que el participante selecciona una opción para cada enunciado o pregunta presentada.
8. **ESCALA LIKERT:** el participante selecciona una de varias opciones, como por ejemplo “totalmente de acuerdo”, a “totalmente en desacuerdo” que se ponderan con números para ayudar con el análisis de los resultados.
9. **DE RELACIÓN:** se presentan dos series de enunciados o palabras y el participante debe unir los elementos de una lista con los elementos de la otra lista.
10. **OPCIÓN MÚLTIPLE:** el participante selecciona una opción de una lista de hasta 40 posibles respuestas. No hay límite para la longitud de cada respuesta.

11. **MÚLTIPLE RESPUESTA:** similar a la pregunta de opción múltiple excepto que el participante no está limitado a elegir una respuesta, sino que puede seleccionar una o más de las opciones ofrecidas o ninguna.
12. **PREGUNTAS NUMÉRICAS:** se le solicita a un participante que ingrese un valor numérico y éste se puede calificar como un valor para una respuesta exacta y otra puntuación si la respuesta está dentro de una gama de posibilidades.
13. **LISTA DESPLEGABLE (pregunta de selección):** se presentan una serie de enunciados y el participante puede hacer coincidir estos enunciados con la lista desplegable.
14. **CLASIFICACIÓN (clasificar en orden):** una lista de opciones que debe ser clasificada numéricamente sin que se permitan coincidencias duplicadas.
15. **SELECCIONAR UNA PALABRA PARA UN ESPACIO EN BLANCO:** al participante se le presenta un enunciado donde falta una palabra; las palabras pueden ser seleccionadas de una lista desplegable para indicar su respuesta.
16. **VERDADERO/FALSO:** el participante selecciona “verdadero”, o “falso”, como respuesta a la pregunta.
17. **RESPUESTA CON PALABRAS (correspondencia de texto):** el participante escribe una o varias palabras para indicar su respuesta.
18. **SÍ/NO:** el participante selecciona “Sí”, o “No”, como respuesta a la pregunta.
19. **SIMULACIONES EN ADOBE CAPTIVATE (Macromedia):** QuestionMark es compatible con una interfaz que cuenta con Macromedia Captivate, y permite a expertos en contenido crear simulaciones que pueden aplicar información sobre puntajes a múltiples interacciones y registrar los resultados dentro de la base de datos de respuestas.
20. **RESPUESTA ORAL:** Es posible procesar el puntaje de las respuestas orales junto con otros puntajes de pruebas utilizando las herramientas de confección de informes de QuestionMark.
21. **MATRIZ DE ENCUESTA:** Este tipo de pregunta le permite incluir múltiples filas de preguntas Likert dentro de una tabla con encabezados de columnas incluidos.
22. **ADOBE (Macromedia) Flash:** QuestionMark es compatible con una interfaz con Macromedia Flash que permite a los programadores desarrollar elementos

personalizados mediante el uso de Flash y registrar los resultados dentro de la base de datos de respuestas.

Una ventaja grande de QuestionMark es que continuamente genera nuevas mejoras que van integrando diversas tecnologías informáticas para ofrecer un producto rentable enfocado a la evaluación, en México se mencionan a las universidades Anáhuac e Iberoamérica que hacen uso de esta tecnología y a nivel mundial es muy popular en el continente Europeo[Corb].

Este software sin duda es uno de los líderes en cuanto a los sistemas de evaluación y realiza funciones similares a las que se presentan en este proyecto, con ésto se deja en claro que en otras partes del mundo, como en Europa, se trabaja sobre evaluadores que implementan tecnologías multimedia, pero cabe destacar que no existen muchas propuestas que implementen estos recursos. Ejemplo en la Figura 3.1:

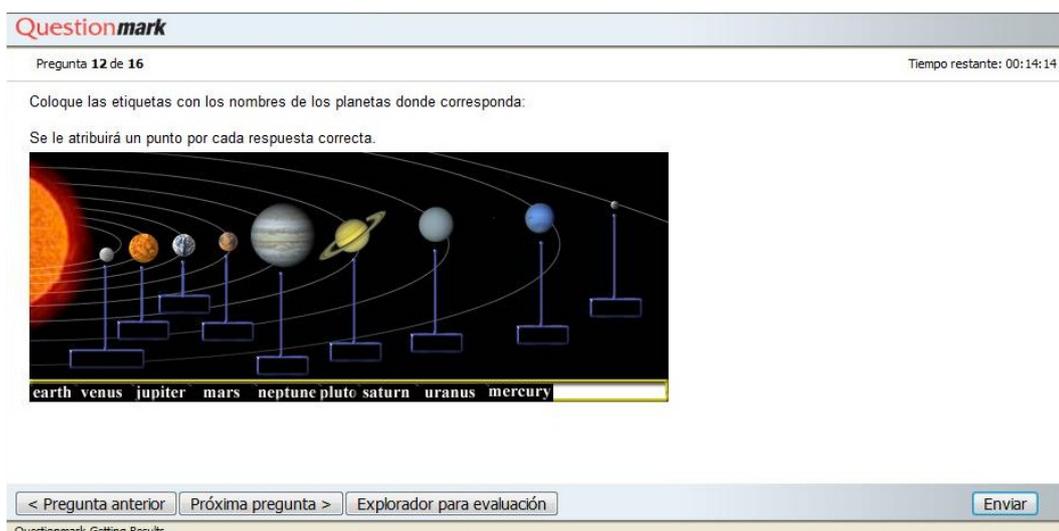


Figura 3.1: Ejemplo de reactivo en QuestionMark

3.2. Sistemas de evaluación

Existen múltiples medios para medir los conocimientos adquiridos por un alumno dentro de un curso, estos medios muchas veces no miden los conocimientos reales que los alumnos tienen o simplemente no proporcionan los suficientes recursos para lograr una comprensión real de lo que se pregunta o evalúa.

Un sistema de evaluación es un conjunto de elementos que se interrelacionan para obtener una estimación real del conocimiento adquirido por un participante en algún curso, esta estimación debe ser verídica y objetiva para lograr una apreciación real del conocimiento [RH].

Los múltiples sistemas de evaluación centran sus recursos basándose en el tipo de ítems que tendrán. Por lo cual pueden existir sistemas basados en ítems escritos, en evaluaciones orales, así como hoy en día utilizar la Web para hacer la medición del conocimiento. Es aquí donde pueden crearse ítems apoyados en recursos multimedia para proporcionar al alumno una forma más sencilla de entender el reactivo.

Los recursos multimedia son una buena alternativa para la elaboración de sistemas cuyo propósito sea la evaluación de conocimientos, proporcionando tanto al alumno como al maestro una alternativa viable para la medición de lo aprendido o lo enseñado respectivamente. Así mismo, la utilización de un sistema computarizado da la oportunidad de proveer un ambiente más rico a los estudiantes para demostrar sus conocimientos y habilidades, ser más auténtico, con mayor contextualización y propiciar un potencial individual [KB]. Así los autores citados con anterioridad proponen un prototipo, el cual se presenta en la Figura 3.2.

La innovación en los ítems utilizando recursos multimedia en ambientes computarizados incluye el correcto uso de sonido, video, animación o mezcla de ellos, ya sea en el área de pregunta, en el de respuesta o una combinación de éstas. Esto provoca un grado más alto de interactividad, debido a que en la evaluación es posible incluir reactivos de relacionar, complementar, opción múltiple, etc.

En este trabajo de tesis se incluye ese tipo de reactivos apoyados por videos, imágenes y sonidos para una mejor comprensión de la pregunta, dando mayor interactividad para crear mejores evaluaciones. El área de respuesta no incluye recursos multimedia, puesto que la atención se centra en proporcionar al alumno preguntas de fácil comprensión y entendimiento.

La utilización de los recursos multimedia dentro de las evaluaciones resulta bastante útil cuando no se miden habilidades de lectura o escritura. Este tipo de recursos son de bastante ayuda cuando el ítem necesita de algún apoyo visual o de audio para sustentar o complementar el contenido de la pregunta.

Irvin y Walker crearon un examen computarizado basado en video para medir las habilidades para la competencia social en niños con pocas habilidades. Esta

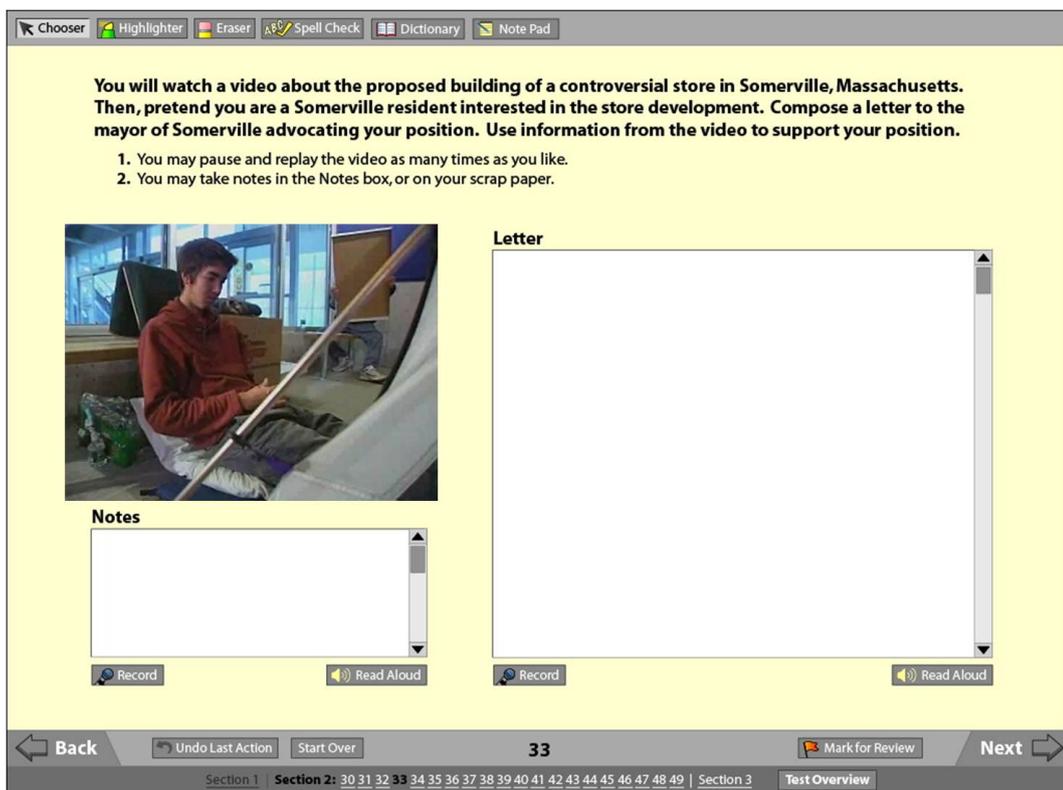


Figura 3.2: Prototipo multimedia

evaluación mostraba un video para visualizar la información y una computadora con touch screen para examinar la respuesta. Este sistema de evaluación permitía medir situaciones de discriminación en niños que mostraban competencia y aquellos que mostraban incompetencia [IW94], ver Figura 3.3.

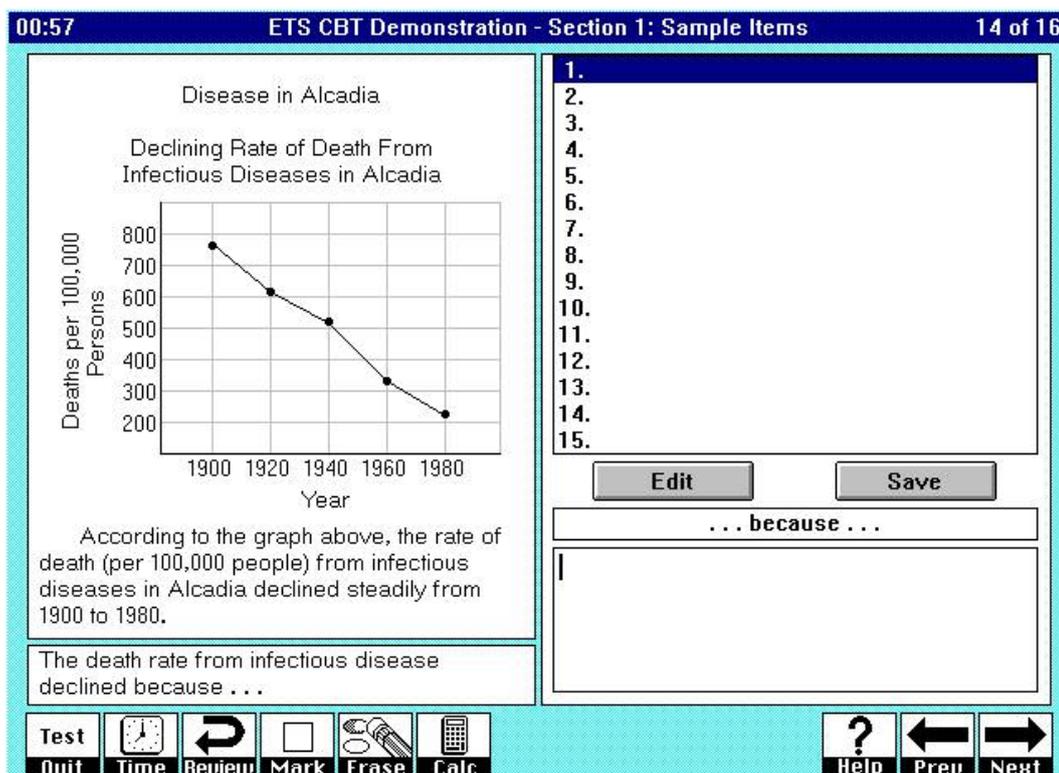


Figura 3.3: Prototipo multimedia de Irvin y Walker

French y Godwin crearon un examen que incluía innovadores ítems para medir conceptos en magnetismo, luz y circuitos. “Generating Explanations”, fue un prototipo creado por the Educational Testing Service (ETS). Las exanimaciones fueron presentadas con un conjunto de datos y preguntando sobre una lista de varias hipótesis para explicar las situaciones [Ser]. Esto es un buen ejemplo para la utilización de evaluaciones innovadores, las cuales en un principio se hacen en papel y son pasadas a un sistema computarizado que hace más fácil de administrar y de medir la estimación del conocimiento [J.], ver figura 3.4.

Los anteriores sistemas de evaluación son monousuario, a diferencia del que se desarrolló que es un sistema en línea, el cual presenta una gran ventaja sobre éstos,

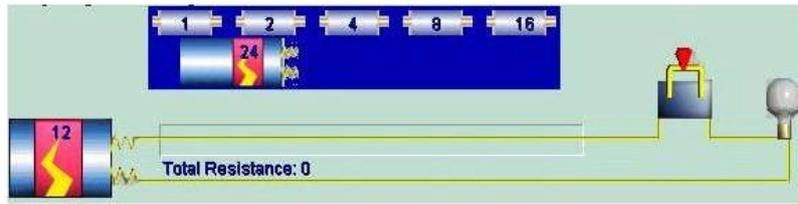


Figura 3.4: Evaluador de magnetismo, luz y circuitos

ya que puede llegar a ser utilizado por más personas de una manera más rápida, todo esto a través de Internet, proporcionando la comodidad de poder acceder desde su hogar a la aplicación sin tener que tener que instalar el sistema, sólo basta con acceder a la URL indicada y así poder efectuar la evaluación, teniendo en cuenta que el navegador cuente con los complementos actualizados.

3.3. Sistemas de evaluación basados en Web

Internet proporciona a toda persona recursos informativos, así como una gran cantidad de conocimientos a nivel mundial. A su vez permite la cooperación y colaboración entre muchas comunidades y grupos interesados en temas específicos. Como respuesta a esto gran cantidad de disciplinas ó empresas lo ven como un medio de ayuda para llegar a un gran número de personas.

Claro ejemplo de esto es la educación que a través de Internet llega a gran cantidad de personas, proporcionando conocimientos a aquellos individuos que ven por este medio un excelente recurso para continuar sus estudios, ya que por algún motivo (trabajo, por ejemplo) no pueden asistir de forma presencial. Este mecanismo es conocido como e-learning, y día a día tiene mayor influencia en la sociedad, claro ejemplo son las universidades que aprovechan este medio para ofertar sus planes de estudios.

Es aquí donde también los LMS hacen su aparición como herramienta para medir conocimientos a través de la Web.

3.3.1. Aplicaciones bajo Open Source sin JSP's

Algunos de estos LMS están desarrollados bajo lenguajes Open Source, tal es el caso de los desarrollados bajo el lenguaje de programación PHP, en estos sistemas la información se encuentra almacenada en una BD desarrollada con Sistema Gestor de Base de Datos MySQL. Un claro ejemplo de éstos es Moodle, que es un sistema útil

para crear y administrar cursos eficaces de aprendizaje en línea. Además, permite interactuar al alumno y profesor con el curso en desarrollo, de forma tal que siempre exista una comunicación adecuada entre éstos. Su forma de evaluación es: los profesores definen una serie de preguntas que serán almacenadas para posteriormente ser presentadas al alumno en diferentes cuestionarios [Moo].

Otro sistema desarrollado bajo este lenguaje de programación es Claroline la cual es una plataforma de e-learning y e-working que al igual que Moodle permite crear y gestionar cursos de manera online. Este sistema incorpora un módulo de evaluación, en éste el responsable del curso crea las preguntas, que de acuerdo al tipo de respuesta pueden ser de elección múltiple, relacional o de rellenar huecos. Estas preguntas son agrupadas y seleccionadas de manera que puedan ser utilizadas en diferentes ejercicios [Cla].

La forma de presentar la evaluación en los dos sistemas anteriores es similar a la que se utiliza en el presente trabajo, ya que aquí de igual manera, ya sea el profesor o los administradores del sistema, subirán la pregunta para después ser utilizada en las evaluaciones, la diferencia radica principalmente en que en el proyecto realizado se incluyen recursos multimedia, tales como videos y sonidos.

3.3.2. Aplicaciones bajo Open Source con JSP's

Existen de igual manera sistemas desarrollados bajo tecnología Java, tal es el caso de OpenUSS que aprovecha las ventajas de un lenguaje de programación moderno que pueden ser usadas para hacer todo en el contexto del desarrollo evolutivo y mantenimiento de los efectos positivos. De la misma manera el sistema también soporta el procesamiento de cargas pesadas. Incorpora a Freestyle de Aprendizaje (FSL) que es utilizado para crear y ejecutar contenidos multimedia [Ope].

El proyecto Sakai es otro sistema que contempla la tecnología Java, éste es un sistema para la administración y gestión de procesos de aprendizaje en línea. Al igual que los anteriores sistemas éste se encuentra almacenado en un servidor. La plataforma permite la creación de diferentes tipos de exámenes o evaluaciones que se ajusten a los criterios pedagógicos de cada responsable de una asignatura [UNE].

Aquí entran herramientas integradas como gradebook y test and quizzes. Gradebook se puede utilizar para mostrar los puntajes de las pruebas y cuestionarios hechos a los estudiantes, también se puede usar para grabar y mostrar los resultados

obtenidos en el trabajo realizado fuera de la plataforma de Sakai. La herramienta de test and quizzes muestra las pruebas y resultados que cada alumno ha obtenido durante el curso en el que está inscrito. Cabe destacar que los sistemas anteriores son OpenSource y de licencia libre.

Dichas herramientas proporcionan una gran alternativa a la hora de observar los resultados de cada evaluación, es por ello que dentro del sistema creado se cuenta con una interface útil para mostrar el resultado que cada alumno obtiene dentro de cada examen realizado.

3.3.3. Aplicaciones comerciales

Existen otros LMS que su uso es comercial, como BlackBoard que es un sistema que brinda herramientas tales como:

- Administración de cursos
- Herramientas instructivas
- Colaboración y Comunicación
- Estimación y Evaluación

Dentro de las herramientas de Estimación y Evaluación se pueden almacenar, usar y compartir preguntas a lo largo de las estimaciones y los cursos mediante Question Pools. Cuenta con opciones flexibles para calificaciones automáticas y manuales, con cálculos basados en la ponderación de calificaciones por categoría o por pregunta, todo esto a través de Gradebook [SYS].

Todos los anteriores sitios cuentan con herramientas o complementos de evaluación clásica. Las interfaces que se desarrollan en el presente proyecto, a diferencia de las que los anteriores sitios, presentan recursos multimedia que ayudan de gran manera a facilitar el entendimiento de la pregunta en cuestión.

En este trabajo se utiliza Java para generar o crear páginas dinámicas que permitan interactuar con los algoritmos necesarios para que el sistema de evaluación sea en su totalidad multimedia.

Capítulo 4

Herramientas utilizadas para el Desarrollo de las Interfaces

En este capítulo se abordan todos aquellos temas que fueron necesarios investigar y conocer a fondo para obtener un desarrollo óptimo de las páginas web útiles para construir las interfases requeridas.

4.1. Lenguaje para el desarrollo de páginas web dinámicas

En la actualidad el uso del internet tiene gran difusión y es de vital importancia para las empresas, tanto en aspectos comerciales como económicos. Dichas empresas utilizan el Internet como medio para realizar transacciones importantes para ellas, por ello, las páginas web que utilizan deben de ser seguras, rápidas y fáciles de utilizar, lo cual garantiza un mejor rendimiento.

Es por ello que día a día surgen nuevas tecnologías que permiten un desarrollo de páginas web que cumplan con dichos requisitos. De igual manera las empresas necesitan de cierta forma tener acceso a una base de datos y a recursos del servidor desde los sitios web, lo cual se hace mediante la creación de páginas dinámicas [Kie09].

Una página web dinámica es aquella que es desarrollada mediante un lenguaje de programación (Script), el cual es ejecutado del lado del servidor, a diferencia de los scripts de las páginas estáticas que se ejecutan en el propio navegador del cliente.

Los lenguajes más utilizados en estos días son PHP y páginas desarrolladas con tecnología Java (JSP, JSF, Servlets, etc). Para el desarrollo de las interfaces desarrolladas se utilizó la tecnología Java con base en los requerimientos del proyecto global. Ésto debido a que ya se contaba con un prototipo desarrollado bajo PHP y no cumplía con las necesidades ni especificaciones que el proyecto requiere.

4.2. Inserción de archivos multimedia en páginas Web

Existen varias alternativas para la inserción de archivos multimedia (imágenes, videos, sonidos, etc.) en páginas web en HTML, a continuación se muestran varias opciones.

4.2.1. Imágenes

La imagen aparecerá en el lugar donde se encuentre la etiqueta. Mediante el uso de otros parámetros se puede especificar la alineación del texto alrededor de la imagen, el tamaño de la misma y otras características. Se verán los más utilizados:

```
1 <IMG SRC="URL de la imagen">
```

ALIGN: permite especificar la alineación de la imagen y del texto alrededor de la imagen. Puede tomar los valores left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom. Ejemplos:

```
1 <IMG SRC="logo.gif" ALIGN="middle">
```

Texto alineado en el centro...

```
1 <IMG SRC="logo.gif" ALIGN="bottom">
```

Texto alineado abajo...

TITLE: este parámetro permite especificar un texto alternativo, que se muestra cuando el puntero del ratón pasa por encima de la imagen. Ejemplo:

```
1 <IMG SRC="logo.gif" TITLE="Tutorial de HTML">
```

ALT: el argumento de este parámetro se muestra mientras se está cargando la imagen, y en caso de que no pueda cargarse permanece visible (de modo que el usuario pueda saber a qué hace referencia esa imagen). En los navegadores que no soportan imágenes (por ejemplo, Lynx) aparece el texto especificado en este parámetro en lugar de la imagen.

WIDTH / HEIGHT: permiten especificar el tamaño de la imagen (ancho y alto). Si no se incluyen, la imagen se muestra a su tamaño real. Aunque se desee mostrar la imagen a tamaño real, es conveniente incluir estos parámetros para permitir que el navegador construya la página antes de haber finalizado la carga de todas las imágenes, mostrando por lo menos el texto del documento.

Ejemplo:

```
1 <IMG SRC='logo.gif' WIDTH='150' HEIGHT='100'>
```

BORDER: cuando una imagen es un enlace, por defecto, el navegador la incluye en una marco o borde. Si no se desea que aparezca este borde, debe especificarse `border="0"`. También puede tomar un valor distinto de 0, en cuyo caso representa el tamaño (grosor) del borde.

4.2.2. Videos

Se pueden ejecutar o incrustar videos dentro de nuestra página web utilizando el siguiente código HTML:

```
1 <EMBED SRC='colon.avi' HEIGHT=300 WIDTH=400 AUTOSTART=TRUE  
2 LOOP=FALSE>
```

SRC: Indica dónde está el fichero de vídeo.

HEIGHT: (alto) y WIDTH (ancho) no pueden omitirse, ya que sus valores por defecto no permiten ver ninguna imagen real. Los tamaños estándar de los ficheros .AVI son 90x120, 120x160, 180x240, 240x320, 300x400, etc., las medidas deben conservar las proporciones de la televisión, es decir 4:3. Hay que tener cuidado a la hora de poner el valor de estos atributos, ya que si no coinciden con los que realmente tiene la imagen, ésta será recortada (no redimensionada) de derecha a izquierda y de abajo a arriba.

El siguiente atributo es AUTOSTART, con dos valores posibles: FALSE (por defecto) y TRUE que como se puede suponer, sirve para que el vídeo arranque nada más cargarse el fichero. Si se establece FALSE, entonces el vídeo se activa al hacer clic con el ratón sobre él, lo mismo que para repetir el video una vez finalizado.

Para hacer que el vídeo se repita continuamente o se pare después de un pase, se tiene el atributo LOOP con dos valores: FALSE (por defecto) y TRUE.

Por último se menciona el atributo ALIGN y sus valores TOP, BOTTOM, CENTER, LEFT, RIGHT, MIDDLE y TEXTTOP que funcionan igual que sus respectivos del elemento IMG, es decir, que no tiene efectos sobre el video, sino sobre su alineación con el texto de la página.

4.2.3. Sonidos

<EMBED>: Este comando delimita un cuadro que permite la inserción de objetos directamente en la página HTML. Los objetos incluidos son manejados por aplicaciones externas a la página, algunas aplicaciones adicionales: WebFX, Paper Software para ver dos mundos VRML (Virtual Reality Modeling Language), Adobe Acrobat para documentos PDF, y Macromedia directo para multimedia, en este caso, se hará referencia a aplicaciones de audio exclusivamente [SC07].

Atributos:

SCR. Localización del archivo

ALIGN. Alinea el objeto en la página (top, middle, bottom, baseline)

HEIGHT. Altura del objeto

WIDTH. Ancho del objeto

AUTOSTART. Permite decidir la ejecución automática o manual.

Ejemplo de uso del formato del comando:

```
1 <EMBED SRC='BLUES.MID' height='60' width='144'  
2 autostart='true'>
```

SCR declara que el archivo de audio que se va a ejecutar se llama BLUES.MID y se encuentra en el directorio junto con el archivo HTML o dentro de la trayectoria de búsqueda (path) .

HEIGHT y WIDTH dan la dimensión del cuadro (objeto, frame), si se desea que no sea visible el cuadro asigne el valor de cero tanto al HEIGHT como al WIDTH.

AUTOSTART =“TRUE”. Indica que al consultar la página, automáticamente abre la aplicación y ejecuta, el archivo mencionado, si se omite la condición autostart por defecto asume el valor de “FALSE”.

4.3. Insertar objetos con object

En versiones anteriores de HTML cada tipo de objetos tenía su propia etiqueta para ser insertados en un documento, incluso aparecieron etiquetas propias de los distintos navegadores para permitir la inserción de determinados tipos, etiquetas que no funcionaban en los otros navegadores.

Además, continuamente aparecen nuevos tipos de objetos, como nuevos formatos de video, sonido, animaciones, etc., y por tanto era necesaria una etiqueta que permitiese la inserción de cualquier objeto actual y cualquiera que pueda aparecer en un futuro. Esta es la etiqueta <OBJECT>que desplaza a todas las demás, incluso a la que se acaba de ver, aunque aún se permita su uso.

El uso de esta etiqueta es el siguiente:

```
<OBJECT DATA=“objeto” TYPE=“tipo de contenido”>...</OBJECT>
```

Ejemplo:

```
1 <OBJECT DATA=‘ ‘Java.gif ’ ’ TYPE=‘ ‘image/gif ’ ’ >...</OBJECT>
```

Se observa que se necesita etiqueta de cierre, al contrario que .

DATA=“objeto”:

Aquí se pone el nombre del archivo que se quiere insertar.

TYPE=“tipo de contenido”

Aquí se pone el tipo de contenido del archivo, es decir, si se trata de una imagen, un sonido, un video, etc. Algunos de los tipos válidos más conocidos son: image/gif, audio/mpeg, video/quicktime, text/HTML, etc.

Si se desea saber el tipo de contenido de un archivo se puede encontrar en el explorador de Windows, seleccionando Ver - Opciones de carpeta - Tipos de archivo.

4.3.1. Atributos opcionales

DECLARE: Declara el objeto pero no lo crea, de modo que puede, por ejemplo, ser llamado con un vínculo mediante el nombre asignado con el atributo ID.

WIDTH="número" HEIGHT="número": Indican las dimensiones del objeto.

CLASSID="objeto": Indica el objeto que se quiere insertar. Dependiendo del tipo de objeto se utiliza este atributo o DATA. CLASSID se utiliza por ejemplo para insertar subprogramas en Java u otros lenguajes, DATA se utiliza para objetos de imagen, sonido o video.

CODETYPE="tipo de contenido": Aquí se pone el tipo de contenido del archivo cuando se utiliza classid. Por ejemplo, para insertar una aplicación en Java el tipo es application/Java-archive.

CODEBASE="dirección web": A menudo un objeto insertado utiliza otros archivos externos, con este atributo se indica la carpeta donde residen estos objetos, en caso de omitirlo los buscará en la carpeta donde se encuentra la página actual.

ARCHIVE="lista de archivos": En el caso anterior se puede especificar aquí esos archivos externos para que se descarguen acelerando la presentación del objeto.

STANDBY="texto": Se puede escribir aquí el texto que se mostrará mientras se carga el objeto, es el típico mensaje "Por favor, espere...".

USEMAP="nombre": Con esta etiqueta se puede hacer que la imagen se comporte como un mapa donde cada zona es un vínculo a una página distinta como vimos con la etiqueta .

4.4. Archivos a utilizar

Dentro del proyecto se manejarán varios formatos de archivos:

- PDF: este formato es utilizado para almacenar el contenido de las preguntas, así como para las imágenes utilizadas como apoyo multimedia.
- MPEG: Formato para los videos.
- WMA: Formato para los sonidos.

4.5. Direccionamiento del archivo multimedia

Se presentan dos formas a través de las cuales se pueden insertar los archivos multimedia dentro de las páginas web dinámicas.

La primera opción que se presenta es cuando los archivos multimedia se encuentran almacenados en el directorio del SO como archivos comunes y corrientes.

Para la inserción del archivo multimedia típicamente no presenta gran problemática, ya que solo basta con almacenar en la BD las rutas correspondientes a la ubicación física del archivo dentro del directorio. De esta manera sólo se obtiene dicha ruta y pasarla como parámetro a la etiqueta a utilizar.

```
1 out.println(“<center><embed src=t’‘+ruta_mult+’#toolbar=0  
2 &navpanes=0&scrollbar=0’ autosize=’1’></embed></center >’);
```

“+ruta_mult+”: ésta es una variable que obtiene la ruta física del archivo, dicha ruta está almacenada en la BD.

Lo complejo se presenta cuando los archivos son almacenados directamente en la BD, lo cual implica que el archivo es almacenado como ceros y unos en el servidor. Para la visualización de dicho archivo es necesaria la obtención de dichos ceros y unos para convertirlos en el archivo con el formato correspondiente, ésto se hace en JSP como se muestra en el apéndice A.

El formato de salida varía de acuerdo al tipo de archivo, es decir, si es imagen, video, sonido o PDF.

Ya obtenido el archivo sólo se llama el JSP que contiene los datos:

```
1 <embed src='Pdf.JSP#toolbar=0&navpanes=0&scrollbar=0'  
2 width='100' height='100' ></embed>
```

`#toolbar=0&navpanes=0&scrollbar=0`: esta parte del código obliga a visualizar un archivo PDF sin sus barras de herramientas, esto con el fin de sólo mostrar el contenido.

4.6. Rastreo de sesiones en un sistema web

La seguridad en un sistema web es básica y de vital importancia, porque así se garantiza el manejo adecuado de la información y evita diferentes problemáticas como; el robo de información, pérdida de la misma, colapsos en el sistema. Un punto importante en la seguridad en un sistema web es saber quien accede a dicho sitio, el horario de ingreso, la cuantificación de la permanencia y el conocimiento de las acciones que realizó un usuario en el sistema web.

Es por ello que se hará uso de un mecanismo de seguridad que permita seguir las acciones que realiza un usuario y sobre todo ir las guardando en una base de datos. Java, como se ha mencionado es un lenguaje muy robusto y permite hacer diferentes acciones, una de esas acciones es el rastreo de sesiones de la cual se hará uso en el sistema de evaluación que se realizó.

4.6.1. ¿Qué es una sesión?

Cuando se ingresa a Internet, las sesiones están presentes en el momento en el que se ingresa una URL en un navegador y se visualiza la página correspondiente, en ese momento se está realizando una sesión.

Para comprender este concepto se dirá que una sesión es la interacción que permite intercambiar información entre un cliente y un servidor. Una sesión se establece cuando un usuario ingresa por primera vez al sitio, y finaliza ya sea por abandonar el sitio, cuando se rebasa un tiempo de inactividad en el sitio o el servidor que aloja el sitio deja de funcionar en algún momento.

En las sesiones se establece a que interfaz se ingresó, que se realizó, el tiempo que permaneció en la interfaz y el número de veces que se ingresó a dicha página. En el evaluador existe obligatoriamente una identificación para poder ingresar al

sistema, estos usuarios tienen diferentes roles y permisos, con lo anterior se podrá rastrear a los usuarios registrados en la base de datos y poder guardar en un esquema de seguridad los atributos que implica una sesión, como ya se ha mencionado anteriormente.

4.6.2. Establecer y manejar una sesión

En Java, específicamente en los JSP, y en los Servlets puede manejarse una sesión mediante la interfaz `HttpSession` y los métodos que implementa, esta interfaz está incluida dentro del paquete `javax.servlet.http`.

Para poder tener acceso a un JSP mediante una sesión se debe poner el atributo `session` de la directiva `PAGE` con el valor `TRUE`.

```
1 < @page session='true' >
```

Para obtener la sesión de un usuario se utiliza el método `getSession()` que devuelve un objeto de tipo `HttpSession`.

```
1 HttpSession sesion=request.getSession();
```

Una vez creado el objeto `sesion` se puede hacer uso de sus diferentes métodos con los que cuenta, uno de esos métodos es `getSessionId()` que devuelve un identificador único asociado a una sesión, esta identificación es generada por el servidor.

```
1 HttpSession sesion=request.getSession();
2
3 out.println( "IdSesion: "+sesion.getSessionId());
```

Otra acción que se puede hacer es saber el momento en el que se realiza la sesión con el método `getCreationTime()`.

```
1 < @page import="Java.util.*" session="true" >
2
3     <%
4
```

```

5      HttpSession sesion=request.getSession();
6
7      out.println(“ Creación: ”+sesion.getCreationTime());
8
9      Date momento=new Date(sesion.getCreationTime());
10
11     out.println(“ <BR>Creación: ”+momento);
12
13     %>

```

En el código anterior se puede ver que se tiene en la línea 1 el atributo session con el valor true que permite hacer uso una sesión en un JSP. En la línea 3 se crea una instancia a la que se le ha llamado “sesion”, y se establece la sesión por medio del método “request.getSession()”. Una vez que se ha creado la instancia sesion se puede hacer uso de ella y de sus funciones de este objeto. En la línea 7 con la función sesion.getCreationTime() se obtiene la hora en que se realizó la sesión, la fecha y en la línea 9 se ha hecho una conversión para que se manejen los datos obtenidos de la sesión en un formato dato. En este contexto también es posible ver la fecha y la hora del último acceso, un ejemplo es el siguiente código.

```

1 <%
2
3     Date acceso=new Date(sesion.getLastAccessedTime());
4
5     out.println(“ Último acceso: ”+acceso+”<br >”);
6
7 %>

```

Al conocer la hora en la que ingresó el usuario y el último acceso que tuvo se puede establecer la duración que ha tenido la sesión, esta explicación se puede ver reflejada en el siguiente código:

```

1 <%
2
3     long longDuracion=sesion.getLastAccessedTime();
4
5     sesion.getCreationTime();

```

```

6
7         Date duracion=new Date(longDuracion);
8
9         out.println(“Duracion:”+duracion.getMinutes()+“min.”
10        +duracion.getSeconds()+”seg”);
11
12    %>

```

HttpSession brinda el método isNew() mediante el cual es posible saber si la sesión creada es nueva o se está tomando de una previamente creada, un ejemplo se muestra en el siguiente código:

```

1 <%
2
3     HttpSession sesion=request.getSession();
4
5     out.println(“nueva: ”+sesion.isNew());
6
7 %>

```

Si se ejecuta el ejemplo, la primera vez el método devolverá true, ya que previamente no había ninguna sesión y ha sido creada en ese instante. Si se recarga la página devolverá false ya que la sesión ya ha sido creada.

4.6.3. Guardar objetos en una sesión

Para guardar un objeto en una sesión se utiliza el método setAttribute(), que ha sustituido al método putValue(). Este método utiliza dos argumentos:

- El primero es el nombre que identificará a esa variable.
- El segundo es el dato que se va a guardar. SetAttribute(Java.lang.String name, Java.lang.Object value)

Un ejemplo de como guardar una cadena de texto en la sesión:

```

1 < @page import = “Java.util.*” session = “true” >
2

```

```

3 <%
4
5     HttpSession sesion=request.getSession();
6
7     sesion.setAttribute(“trabajo”,“Paginas de JSP”);
8
9 %>

```

4.6.4. Recuperar objetos de una sesión

Los datos que se guardan en la sesión permanecen ahí a la espera de ser utilizados. Para ello es necesario realizar el proceso contrario a cuando se graban, comenzando por la recuperación del objeto de la sesión para empezar a ser tratado.

Para poder realizar este paso se utiliza el método `getAttribute()` (anteriormente se utilizaba el método `getValue()`, pero este método se encuentra en desuso), utilizando como argumento el nombre que identifica al objeto que se quiere recuperar.

```

1     getAttribute(Java.lang,String nombre)

```

Un ejemplo de recuperación de objetos almacenados en la sesión:

```

1 <%
2
3     HttpSession sesion=request.getSession();
4
5     Sesion.getAttribute(“nombre”);
6
7 %>

```

Cuando este método devuelve el objeto no establece en ningún momento de qué tipo de objeto se trata (String, Vector,etc.)

Por ello si se conoce previamente el tipo de objeto que puede devolver tras ser recuperado de la sesión es necesario realizar un casting, para convertir el objeto de tipo genérico al objeto exacto que se va a usar. Para realizar esta operación se añade

el tipo de objeto al lado de tipo HttpSession que utiliza el método `getAttribute()` para obtener el objeto que devuelve:

```
1 <%
2
3     HttpSession sesion=request.getSession();
4
5     String nombre=(String)sesion.getAttribute("nombre");
6
7     out.println("Contenido de nombre: "+nombre);
8
9 %>
```

4.6.5. Destrucción de una sesión

Como se ha visto, los datos almacenados por las sesiones pueden destruirse en tres casos:

- El usuario abandona la aplicación web (cambia de web o cierra el navegador).
- Se alcanza el tiempo máximo permitido de inactividad de un usuario (time-out).
- El servidor se para o se reinicia.

Pero la situación más probable es querer iniciar las sesiones o dar por finalizada una si se ha cumplido una o varias condiciones. En este caso no es necesario esperar a que ocurra alguno de los tres casos citados anteriormente, ya que mediante el método `invalidate()` es posible destruir una sesión concreta.

En el siguiente caso la sesión "sesion", se destruye al invocar el método `invalidate()`; y por la tanto el valor u objeto que está asociado a la misma.

```
1 <%
2
3     sesion.invalidate();
4
5 %>
```

Capítulo 5

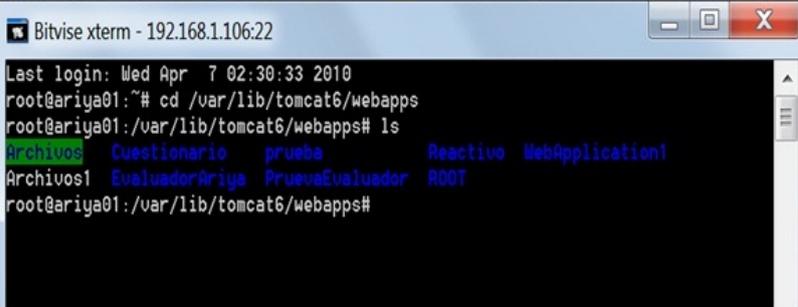
Desarrollo

En las líneas siguientes se explica cada una de las interfaces que integran el presente trabajo de tesis, así como lo que se necesita para implementarlo. Como se ha mencionado anteriormente, este proyecto sólo pretende presentar un prototipo de un sistema de evaluación en línea desarrollado con el lenguaje Java. Cada una de las interfaces del prototipo se fueron desarrollando con base en los requerimientos del proyecto.

5.1. Alojamiento del proyecto Ariya en el servidor Tomcat

1.- En primera instancia se debe contar con el proyecto Ariya que por nombre se ha llamado EvaluadorAriya, dicho proyecto cuenta con varias carpetas y archivos.

2.- Una vez contando con la carpeta EvaluadorAriya se procede a copiarla en el servidor Ubuntu server en la carpeta /var/lib/ Tomcat6/webapps, que es la carpeta indicada donde se alojan los proyectos web en el servidor Tomcat. Una vez realizada la acción anterior se debe presentar una pantalla similar a la Figura 5.1:



```
Bitwise xterm - 192.168.1.106:22
Last login: Wed Apr  7 02:30:33 2010
root@ariya01:~# cd /var/lib/tomcat6/webapps
root@ariya01:/var/lib/tomcat6/webapps# ls
Archivos  Cuestionario  prueba  Reactive WebApplication!
Archivos! EvaluadorAriya PruebaEvaluador ROOT
root@ariya01:/var/lib/tomcat6/webapps#
```

Figura 5.1: Copiado de archivos

3.- El siguiente paso es alojar las librerías que permitirán realizar a las interfaces acciones como; la conexión con la base de datos Ariya creada en el gestor de base de datos MySQL, el uso de clases y métodos que permitan subir archivos de una terminal cliente a un servidor por medio de un navegador, etc. Para lo anterior, se debe contar con los archivos que contienen esas librerías, que son archivos de Java con extensión .jar como se muestra en la Figura 5.2.

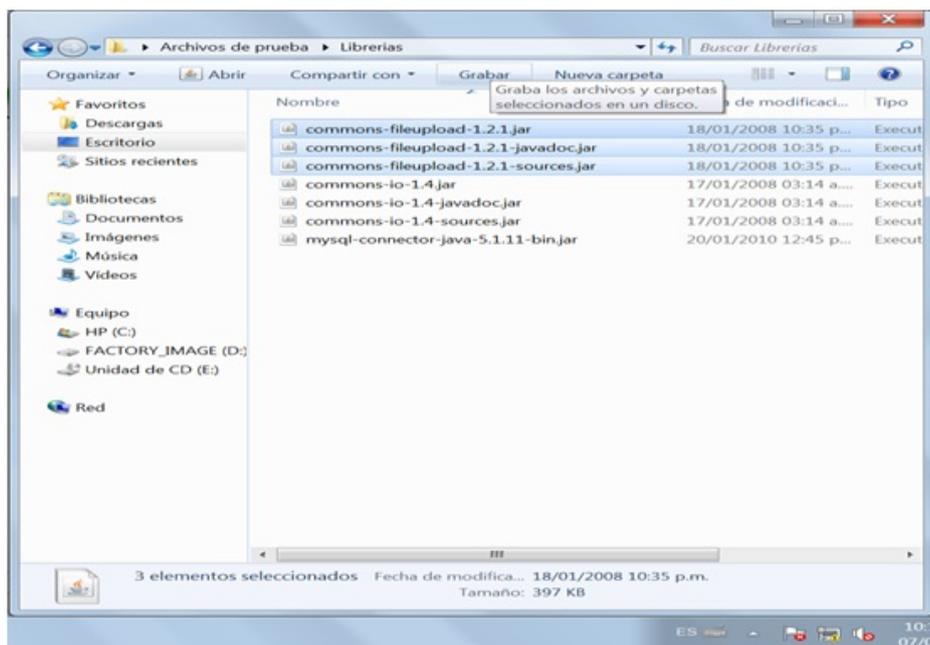


Figura 5.2: Librerias

4.- Los archivos de las librerías tendrán que ser copiados en el servidor Ubuntu server en la carpeta /usr/share/ Tomcat6/lib como se ve en la Figura 5.3.

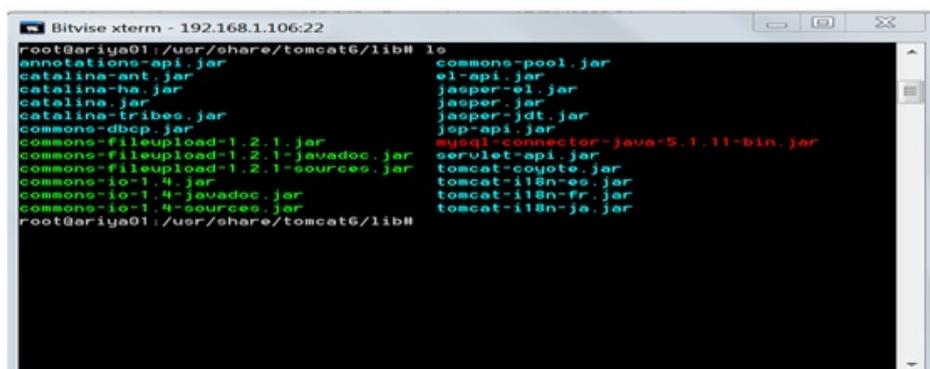


Figura 5.3: Archivos copiados dentro del servidor

5.- Una vez copiados, a cada archivo de las librerías, tendrán que cambiarse los privilegios para darle todos los permisos de escritura, lectura y ejecución con el comando `chmod`, un ejemplo seria:

Se cambiarán los permisos al archivo `commons-io-1.4.jar`, otorgándole todos los privilegios de escritura lectura y ejecución. Ésto se realiza tecleando:

```
1 chmod 777 /usr/share/ Tomcat6/lib/ commons-io -1.4.jar
```

Para visualizar los cambios en los privilegios de los archivos se ingresa el comando `ls -l`. Después se observa una pantalla donde al principio del archivo se verán todos los permisos con los que cuenta dicho archivo esto se determina a través de las letras `-rwxrwxrwx`.

De igual forma se cambiará, los privilegios de cada uno de los archivos a fin de que al consultar sus permisos, se vean como lo muestra la Figura 5.4.

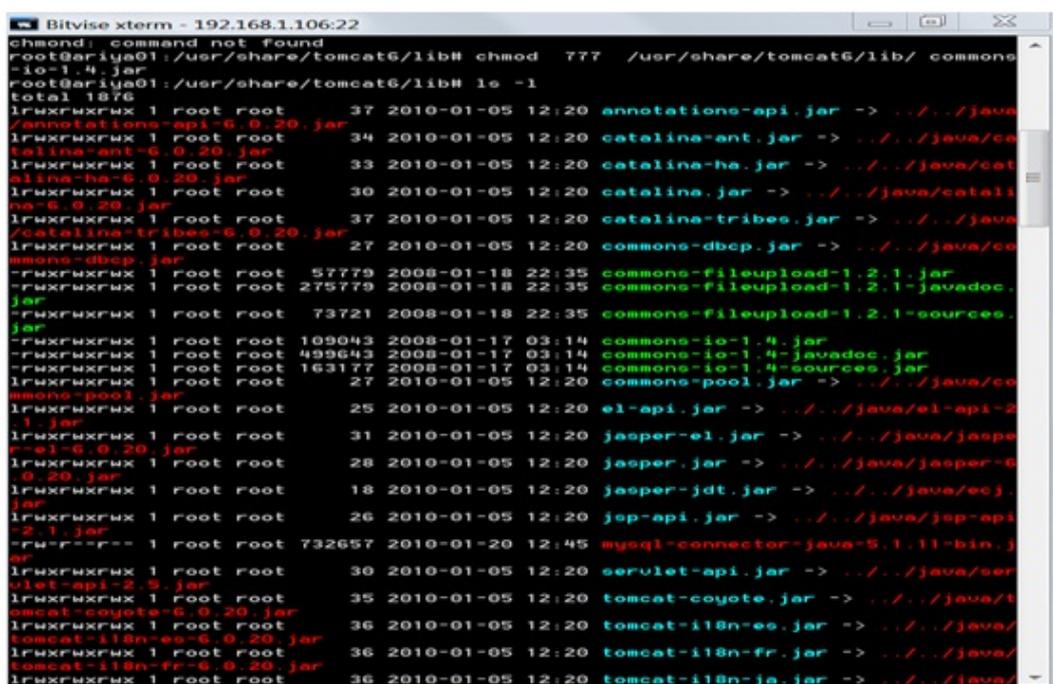


Figura 5.4: Ver privilegios

6.- Se tiene que cambiar un atributo de seguridad en el archivo Tomcat6 el cual se encuentra en la ruta `/etc/default` como se muestra en la Figura 5.5.

```
Bitvise xterm - 192.168.1.106:22
root@ariya01:/etc/default# ls
apache2  console-setup  halt          rcS          tmpfs
appport  cups           image-store-proxy  rsync       tomcat6
bind9    devpts        libvirt-bin   samba       ufw
bootlogd dhcp3-server   locale       saned       useradd
cacerts  grub          ntpdate      ssh         winbind
root@ariya01:/etc/default#
```

Figura 5.5: Cambio de atributos

Una vez identificado el archivo default, se tecleará nano /etc/default/ Tomcat6, para que el editor de textos (nano) pueda abrir el archivo y ahí se modifica la línea:

```
TOMCAT SECURITY = yes
```

Se reemplaza por esta:

```
#TOMCAT SECURITY = yes
```

Ésto es necesario para permitir la conexión de la BD, en el servidor Web, ver Figura 5.6.

Después se oprimen las teclas Ctrl + O (para guardar cambios) y después Ctrl + X (para cerrar el archivo).

7.- Una vez realizado lo anterior se podrá visualizar en un navegador el proyecto poniendo la ruta y el puerto en este caso es como se muestra en la Figura 5.7.

<http://192.168.1.106:8080/EvaluadorAriya/web/login.JSP> donde se ve que:

IP: 192.168.1.106

Puerto: 192.168.1.106

Carpetas: EvaluadorAriya/web/

Archivo: login.JSP

```
Bitvise xterm - 192.168.1.106:22
GNU nano 2.0.9 File: /etc/default/tomcat6 Modified

# Use the Java security manager? (yes/no, default: yes)
# WARNING: Do not disable the security manager unless you understand
# the consequences!
# TOMCAT6_SECURITY=yes

# Number of days to keep logfiles in /var/log/tomcat6. Default is 14 days.
#LOGFILE_DAYS=14

# Location of the JUM temporary directory
# WARNING: This directory will be destroyed and recreated at every startup !
#JUM_TMP=/tmp/tomcat6-temp

G Get Help  O WriteOut  R Read File  Y Preu Page  K Cut Text  C Cur Pos
X Exit      J Justify    W Where Is  U Next Page  U UnCut Text T To Spell
```

Figura 5.6: Cambiando atributos de seguridad de Tomcat6

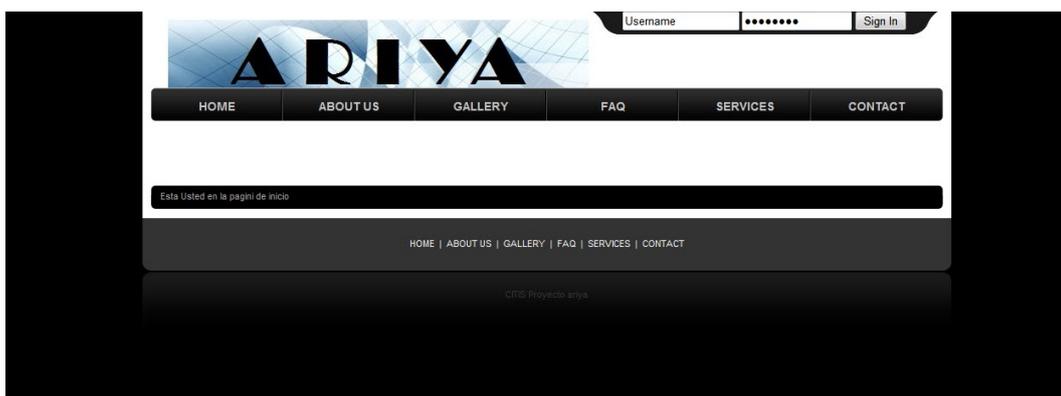


Figura 5.7: Login del sistema

5.2. Interface Reactivo Multimedia

Esta interfaz tiene como objetivo ir presentando una serie de reactivos para que el evaluado los conteste, a medida que vaya dando respuesta irán cambiando los reactivos, estos reactivos son presentados dependiendo del tema, tópico, subtópico y constructo que se eligió.

En esta interface se muestra el reactivo y si éste tiene algún soporte multimedia éste aparecerá en la parte de arriba del reactivo, este complemento multimedia facilita la comprensión de las preguntas que se formulan, ver Figura 5.8.

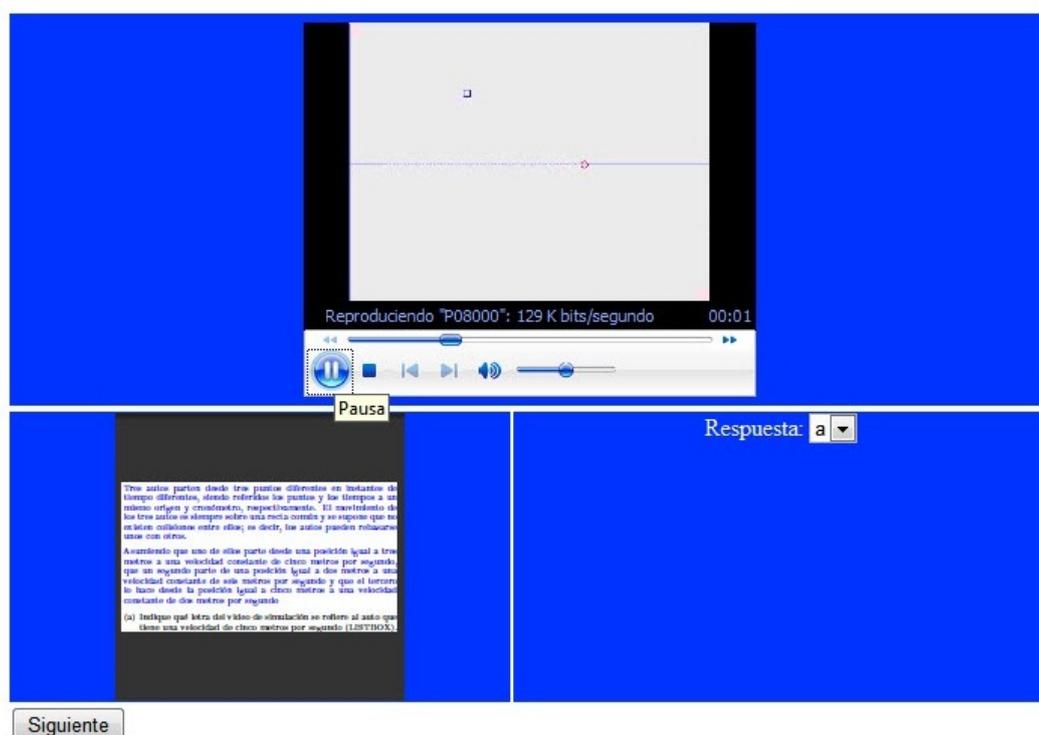


Figura 5.8: Interface reactivo multimedia 1

Los reactivos, imágenes, videos o sonidos son tomados de una consulta a la base de datos, para obtener estos datos el sistema cuenta con una conexión que se establece por el canal JDBC. Una vez obtenida la comunicación con la base de datos se realiza una sentencia Statement que ejecuta el Query para traer una tabla con registros, esta tabla es pasada a un ResultSet, posteriormente, se obtienen las columnas de consulta, en estas columnas se tienen los registros de las rutas. Después

Three cars start from three different points at different times, with their positions and times referred to a common origin and chronometer, respectively. The movement of the three cars is always along a straight line and it is assumed that no collisions occur between them; that is, the cars can pass each other.

Assuming that one of them starts from a position equal to three meters with a constant velocity of five meters per second, that another starts from a position equal to two meters with a constant velocity of one meter per second, and that the third starts from a position equal to five meters with a constant velocity of two meters per second.

(b) In the graph of position versus time, what is the intersection of the lines that refers to the meeting of the car with one meter per second with the car with five meters per second? (CLICK).

Respuesta:

Siguiente

Figura 5.9: Interface reactivo multimedia 2

se pasan las rutas a las etiquetas HTML correspondientes para que sean visualizadas en el navegador, ver Figura 5.9.

Si el reactivo tiene relacionado un archivo multimedia éste se mostrará en la parte de arriba de la pregunta, si tuviera dos archivos la parte de arriba se dividiría en dos, y si fueran tres sería proporcional, ver Figura 5.10.



Figura 5.10: Interface reactivo multimedia 3

Al lado del reactivo se muestra la área de respuesta, la cual varía de acuerdo al tipo de reactivo que se esté aplicando. Ésta puede ser con opción múltiple, de completar, de relación, etc. ver Figura 5.11.

Para iniciar la evaluación y presentar el primer reactivo se hace una consulta en la cual se incluyan preguntas que cumplan con los requisitos de materia, tópico, subtópico y constructo que fueron previamente elegidos. Cabe mencionar que existen preguntas de tipo base las cuales dentro del mismo reactivo se incluyen varias preguntas. En la interface se pueden identificar estas preguntas en la parte inferior de la pregunta ya que si el botón dice “siguiente” indica que pertenece a una pregunta tipo base y que faltan las preguntas complementarias, si el reactivo es normal el botón dirá “enviar”.

Complete con la palabra adecuada cada uno de los espacios mostrados, de tal forma que la proposición compuesta obtenida sea verdadera.

El _____ percibe las ondas físicas a través de los _____
Estos fenómenos son componente esencial del _____ que rodea al _____.

Respuesta:

a).-

b).-

c).-

d).-

Enviar

Figura 5.11: Interface reactivo multimedia 4

5.3. Interface Seleccionador

En esta parte del sistema el evaluado elegirá mediante combobox la materia, el t pico, subt pico y constructo,  sto con el fin de presentarle una evaluaci n con reactivos afines a su elecci n. Los datos que se muestran en los combos de esta pantalla son extra dos de la Base de datos. El proceso que se realiza para llenar estos combos consiste en realizar consultas a la Base de datos, posteriormente se obtienen los campos de la tabla que arroj  la consulta y son pasados a la etiqueta HTML que es la que crea el combo.

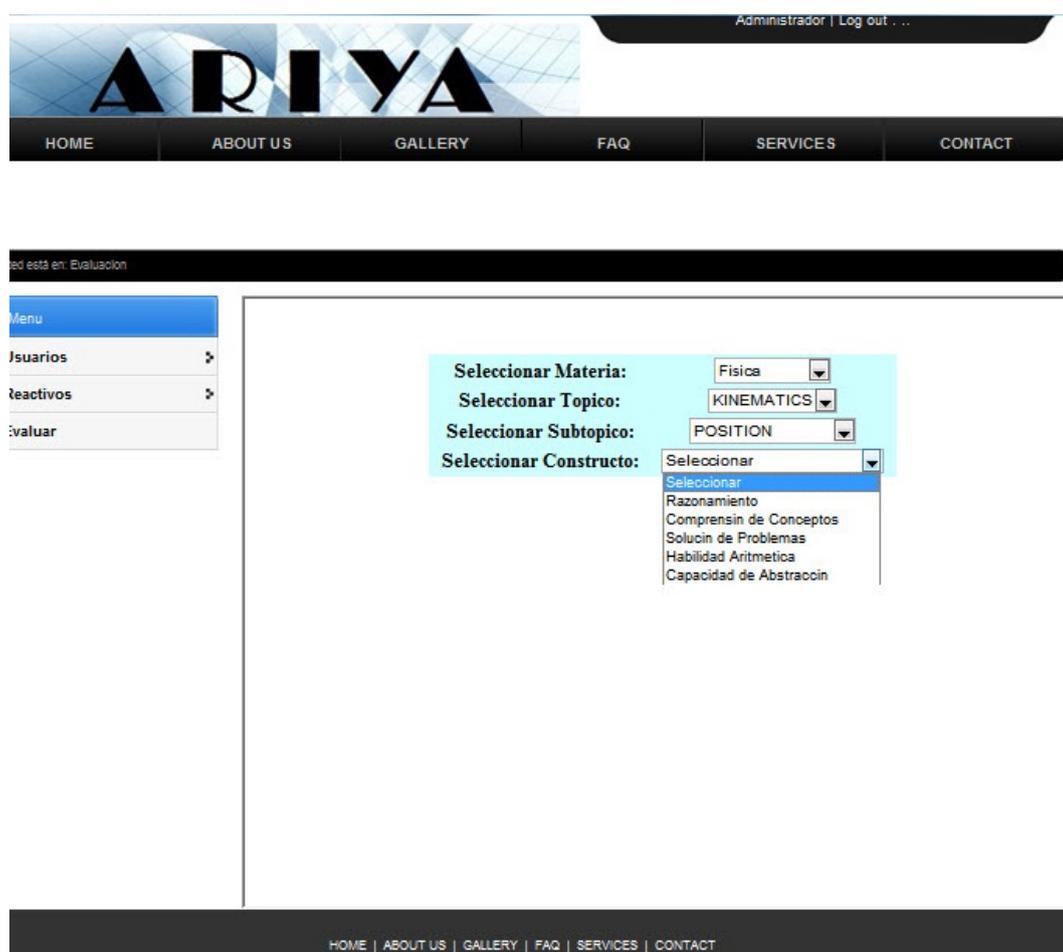


Figura 5.12: Interface Seleccionador

Cada materia se encuentra relacionada con varios t picos, es por ello que principalmente se elige la materia para poder cargar los t picos correspondientes,

así mismo, cada tópico se relaciona con subtópicos, en los que se pueden evaluar constructos, ver Figura 5.12.

5.4. Interface de Login

Aquí se determina quien tiene acceso o no al sistema. La consulta para poder acceder al sistema se hace directamente en la BD, con ello se sabe exactamente qué persona puede entrar al sistema y con qué privilegios lo hará, ver Figura 5.13.

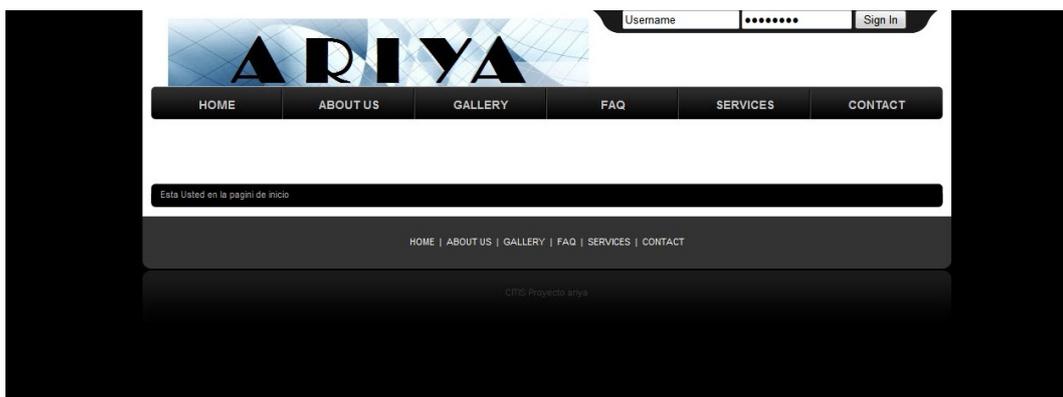


Figura 5.13: Interface de Login

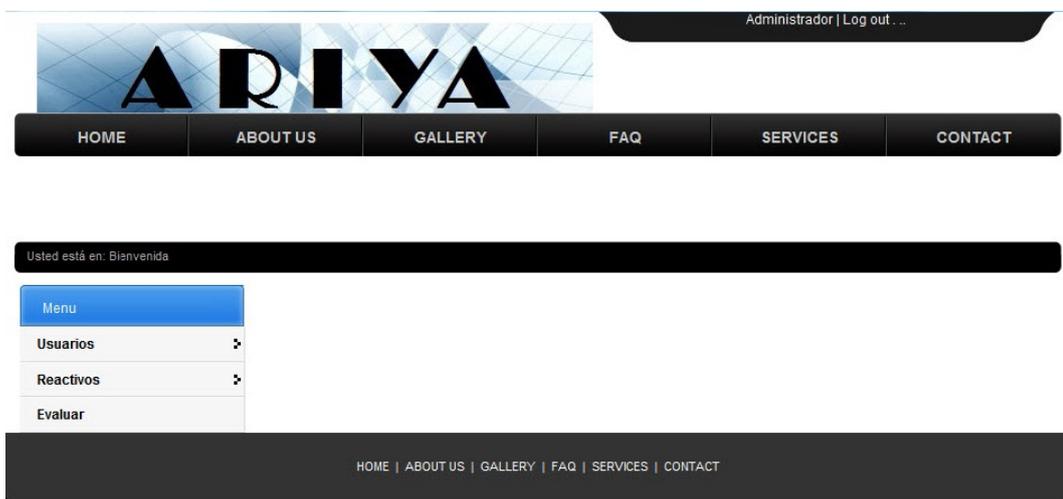


Figura 5.14: Login de acuerdo al tipo de usuario

Lo anterior implica que sólo bastará introducir el nombre de usuario y la contraseña correspondiente y el sistema determinará que opciones estarán disponibles

de acuerdo al nivel que se le haya asignado a dicho usuario, ver Figura 5.14.

El usuario podrá acceder en el nivel de alumno, profesor o administrador, aunque en un futuro se pueden determinar otros roles.

5.5. Interface Evaluador

En esta interface se muestra la conclusión de la evaluación donde aparece cada una de las respuestas dadas para cada pregunta por parte del evaluador y son comparadas con las respuestas correctas almacenadas en la BD, ver Figura 5.15.

No	Respuesta del Alumno	Respuesta Correcta
1	b	2
2	b	a
3	d	d
4	b	d
5	2	a
6	3	b
7	3	c
8	3	d

Figura 5.15: Interface Evaluador

Al final se muestra cuantas respuestas coinciden dando una calificación aproximada. Las respuestas dadas por el alumno a cada reactivo son almacenadas en la BD para su futura comparación.

5.6. Interface Captura Reactivo

En esta parte se llenan los campos correspondientes que incluirá el nuevo reactivo, se detectan que archivos multimedia están relacionados con el reactivo.

Se deben llenar los campos correspondientes para definir a que materia, tópic, subtópico y constructo pertenece el reactivo para hacer las inserciones correspondientes dentro de la BD. Una vez elegidos los elementos que identifican a donde pertenece el reactivo, éstos serán tomados en cuenta para guardarlos en las carpetas del servidor, si estas carpetas no existen el sistema las creará automáticamente, ver Figura 5.16.

Administrador | Log out ...

HOME ABOUT US GALLERY FAQ SERVICES CONTACT

Usted está en: Alta de reactivos

Menu

- Usuarios
- Reactivos
- Evaluar

INSERTAR REACTIVO

PREGUNTA

ARCHIVO PDF DE LA PREGUNTA Examinar...

TIPO SI/NO

COMPLEJIDAD 1

TAXONOMIA 1

CALIBRACION 1

COMPLEMENTO MULTIMEDIA

ARCHIVO COMPLEMENTO Examinar...

TIPO VIDEO

RESPUESTA

RESPUESTA A

HERRAMIENTA A

POSICION DE HERRAMIENTA 1

Enviar Restablecer

HOME | ABOUT US | GALLERY | FAQ | SERVICES | CONTACT

Figura 5.16: Interface Captura Reactivo

En esta sección el usuario debe indicar la ruta de la carpeta donde se encuentra el archivo pertenece al reactivo que da de alta y su ayuda multimedia, que puede ser un video, audio o PDF.

5.7. Interface Captura Usuario

Esta interface permite dar de alta a un nuevo usuario, los nuevos usuarios serán dados de alta ya sea por parte del administrador o por un profesor, ver Figura 5.17.

Administrador | Log out . . .

HOME ABOUT US GALLERY FAQ SERVICES CONTACT

Listed está en: Alta de usuarios

Menu

- Usuarios
- Reactivos
- Evaluar

ALTAS DE USUARIOS

Usuario:

Contraseña:

Nombre:

Apellido Paterno:

Apellido Materno:

Edad:

Estado Civil:

Sexo:

Direccion:

Correo:

HOME | ABOUT US | GALLERY | FAQ | SERVICES | CONTACT

Figura 5.17: Interface Captura Usuario

Capítulo 6

Conclusiones y Trabajo Futuro

En el presente proyecto se estableció como fin del mismo el desarrollo de interfaces para una evaluación en línea sobre un tema, tópico o subtópico en específico, permitiendo así calificar de forma precisa y veraz el conocimiento, y con esto poder arrojar una calificación tangible.

El evaluar el conocimiento de un individuo, como se mencionó en un principio en este documento, se torna complicado, si es que se quiere cuantificar de forma objetiva el conocimiento con el que cuenta dicha persona, y a través de la evaluación intervienen muchos factores como lo son la forma en que se presenta el examen, la duración estimada, las herramientas para hacer cálculos etc. Al darse cuenta de estas vertientes y elementos que intervienen en la evaluación, en el desarrollo del proyecto se pueden asumir las siguientes conclusiones, con respecto al desarrollo de las interfaces:

- Para el desarrollo de un sistema de evaluación en ambiente Web se debe hacer uso de una tecnología que sea robusta y cuente con una amplia gama de elementos que permitan realizar acciones en específico para que se pueda dar la evaluación; un ejemplo es el uso de ayuda multimedia en la presentación de reactivos al evaluado. Java es un excelente ambiente de programación para poder realizar aplicaciones Web debido a que cuenta con una extensa composición de clases que facilitan el desarrollo de algoritmos y usos de recursos.
- El entorno Web que se presenta al usuario final de un sistema Web debe ser de fácil acceso y tornarse *amigable* para que no influya sobre el proceso de evaluación que realiza un individuo, esto es debido al hecho que navegar en un sistema Web debe ser de una forma que no pierda la noción de donde se encuentra navegando.

- La seguridad a nivel aplicación debe hacerse presente en cada momento dentro del sistema Web de evaluación para garantizar la integridad de la información que se presenta y tener la consistencia de las acciones que realiza un usuario en una fecha y horario estimado por el sistema evaluador.

- El lenguaje con el que se crea el sistema de evaluación debe ser flexible para que permita la interacción con otras tecnologías y así poder disponer de más elementos que permitan más acciones, un ejemplo de ello es la carga de una misma página o sólo una parte de esa página en forma rápida y eficaz, otro ejemplo es poder efectuar una conexión y la interacción con una base de datos.

Para presentar la ayuda multimedia en el evaluador es más conveniente presentar un archivo multimedia mediante su ubicación en una carpeta, la llamada a esa ruta y no mediante el mecanismo de poder extraer archivos de tipo blob de una base de datos para posteriormente interpretar los bits y mostrar el archivo multimedia que generan los bits del dato tipo Blob. En este trabajo de tesis ha resultado más eficiente obtener la ruta del archivo multimedia de la BD, que utilizar datos de tipo Blob en el que se almacene el archivo, sin embargo, para proyectos futuros se puede considerar esta alternativa.

6.1. Ventajas

En el desarrollo de proyectos Web, Java es muy eficaz debido a que los procesos que se realizan al ejecutar un programa Web no son pesados ya que sólo se carga una sola vez el proceso en la maquina virtual de Java (JVM) y ésto hace más eficaz la interacción entre el cliente y el servidor, ya que cuando un cliente haga una petición al servidor donde se aloja el proyecto Ariya sólo se ejecutará un proceso de un JSP en la JVM y cuando otro usuario haga la misma petición, el servidor sólo creará una instancia de la clase que crea el JSP.

Se presenta un esquema de seguridad a través del uso de ciertas tablas de la base de datos donde existen usuarios con ciertos roles, contraseñas que deberán autenticarse para poder hacer uso del sistema.

El sistema de evaluación en línea hace uso de elementos multimedia como son sonidos, videos, imágenes, etc. Esto provee una herramienta de análisis y apoyo al evaluando.

Con Java en el desarrollo de las interfaces para el evaluador podemos hacer uso de otras tecnologías como son JavaScript, HTML, etc. Esto permite poner de

forma más fácil la construcción de las interfaces de evaluación con elementos como tablas, generación de métodos en JavaScript para hacer validaciones, etc.

Un elemento más que se tiene para garantizar la seguridad en el sistema de evaluación es el uso de rastreo de sesiones, donde al iniciar un usuario registrado en la base de datos, se empieza a registrar las acciones que realiza dicho usuario, como el ingreso a la evaluación, etc. También se obtiene la fecha y hora en que el usuario hizo la última acción.

Java permite la interacción que se realiza en el evaluador en línea con una base de datos hecha en MySQL y así con ello se provee de una gran cualidad que permite obtener, guardar y hacer uso de datos en el momento en que se requiera. En la mayoría de las interfaces se hace una conexión a la base de datos Ariya y así se disponen de los datos que se encuentran en diversas tablas y a nivel de aplicación se puede hacer consultas a dicha base de datos con tan sólo formular una sentencia SQL.

6.2. Desventajas

Una desventaja que se presenta en este proyecto es que aun no está concluido y que existen varios huecos; por ejemplo aun no se concluyen las interfaces de la evaluación adaptable que se pretende sean realizadas, esto se hará en la continuación de trabajos del proyecto Ariya.

No se utiliza algun Framework de Java que facilite el desarrollo y la programación por capas, de un sistema Web.

6.3. Alcances o Limitaciones

En las siguientes líneas se exponen una serie de procesos, muchos de los cuales cubre el sistema que se ha realizado durante este trabajo, pero al mismo tiempo se mencionan aquellos que no fueron realizados dentro de dicho proyecto.

6.3.1. Alcances

A continuación se enumeran cada uno de los alcances logrados en el trabajo presentado:

1. Las interfaces permiten evaluar a un individuo de forma clásica, aunque se deja abierta la posibilidad de incluir otras metodologías.
2. Al concluir con el proyecto, la unión de las interfaces conformará un sitio Web que permita evaluar el conocimiento a un individuo.
3. Los usuarios pueden ingresar al sitio Web mediante una cuenta de usuario y contraseña.
4. El sistema utiliza una base de datos como medio de almacenamiento, elaborada para el gestor de base de datos MySQL.
5. El sitio Web está alojado en un servidor Web Apache Tomcat.
6. El sistema operativo que contendrá al servidor Web y al sistema de evaluación es Linux en su distribución Ubuntu Server.
7. Las interfaces son desarrolladas en el lenguaje Java.
8. Los usuarios que pueden ingresar al sistema son: los alumnos, profesores y administradores.
9. El sistema cuenta con un mecanismo de seguridad que permitirá ver las acciones que realizaron los usuarios y los tiempos o instantes en que ingresaron.
10. El sistema Web permitirá al administrador la modificación, ingreso y eliminación de reactivos y cuentas de usuario.
11. El usuario podrá saber la calificación de su evaluación.
12. La presentación de la información será diversificada ante el usuario, pudiendo ser ésta en forma literal, audio, gráfica o video.

6.3.2. Limitaciones

Asimismo como con anterioridad se muestran los alcances de la tesis, a continuación se enumeran una a una las limitaciones:

1. El sistema no cuenta con herramientas de colaboración virtual como foros de discusión, chats etc., sin embargo a futuro las herramientas desarrolladas en el presente proyect Ariya, pueden ser incrustadas en LMS's, por lo que esta limitación esta cubierta.
2. No podrán ingresar usuarios ajenos al curso de evaluación.
3. Los alumnos no podrán subir información al sitio sólo serán evaluados.
4. Los profesores no podrán impartir sesiones educativas por medio de este sistema.
5. El sistema no proveerá de anexos complementarios a los tópicos de aprendizaje, como artículos de revista, secciones de libros, ligas a otros sitios etc.

6.4. Trabajo Futuro

Estas interfaces desarrolladas, son el comienzo de un sistema completo el cual será completado en las posteriores etapas del proyecto global. Será completado basándose en las interfaces ya desarrolladas, las cuales serán adaptadas y mejoradas para cumplir con los objetivos previstos en el sistema.

El sistema final deberá cumplir los objetivos propuestos en el proyecto Ariya, el cual se encuentra financiado por CONACYT, deberá realizar una evaluación adaptable ya que con lo que se realizó sólo se efectúa una evaluación clásica.

Deberán crearse más interfaces para completar el sistema, entre las cuales deberán considerarse: el diseño final del proyecto, interface para cursos, que en su caso se integrará con un LMS como Moodle o algún otro.

Bibliografía

- [Chr] Jarmes J. Christine. Computer-based testing: Toward the design and use of innovative items. Recuperado el 19 de Octubre de 2009, de la **página web** denominada Computer-Based Testing, url: <http://www.coedu.usf.edu/itphdsem/eme7938/ch899.pdf>.
- [Cla] Consorcio Claroline. Claroline.net. Recuperado el 11 de Octubre de 2009, de la **página web** denominada Claroline.Net, url: <http://www.claroline.net/>.
- [Cora] Oracle Corporation. Netbeans. Recuperado el 25 de Octubre de 2009, de la **página web** denominada Netbeans, url: http://netbeans.org/index_es.html.
- [Corb] Questionmark Corporation. Questionmark. Recuperado el 10 de Diciembre de 2009, de la **página web** denominada QuestionMark, url: <http://www.questionmark.com/esp/index.aspx>.
- [CSH09] Gary Cornell Cay S. Horstmann. *Core java volumen 1 fundamentals*. Prentice hall, 2009.
- [IW94] Larry K. Irvin and Hill M. Walker. Assessing children's social skills using video-based microcomputer technology. *Exceptional Children*, 1994. Recuperado el 19 de Octubre de 2010, de la **página web** con url: <http://www.highbeam.com/doc/1G1-15824015.html>.
- [J.] French A. & Godwin J. Using multimedia technology to create innovative items. Paper presented at the Annual Meeting of the American Educational Research Association (New York, NY, April 8-12 1996), Recuperado el 20 de Octubre de 2010, de la **página web**, url: <http://www.eric.ed.gov/PDFS/ED400321.pdf>.

- [JM08] Matt Telles Julie Meloni. *PHP 6 FAST and EASY WEB DEVELOPMENT*. Course Technology, 2008.
- [KB] Rebecca Beck Kelly Burling, Bob Dolan. Constructing innovative computer-administered itemsl. Recuperado el 19 de Octubre de 2010, de la **página web** denominada Pearson Educational Measurement url: education.pearsonassessments.com.
- [Kie09] Richard Kiessig. *Ultra-Fast ASP.NET*. Apress, 2009.
- [Moo] Moodle.org. Open-source community-based tools for learning. Recuperado el 10 de Octubre de 2009, de la **página web** denominada Moodle, url: <http://moodle.org/>.
- [MZ06] Kevin Mukhar and Chris Zelenak. *Beginning Java EE 5 Plataforma from Novice to Professional*. Apress, 2006.
- [Ope] OpenUSS. Openuss. Recuperado el 19 de Octubre de 2009, de la **página web** denominada OpenUSS, url: <http://openuss.sourceforge.net/openuss/index.html>.
- [Reg] Jorge López Reguera. Common gategay interface. Recuperado el 03 de Octubre de 2009, de la **página web** denominada Universidad de Concepcion de Chile, url: <http://www.inf.udec.cl/~jlopez/DSWR/cgi.html>.
- [RH] Laurence Wolff y Eduardo Vález Robin Horn. Sistemas de evaluación educacional en américa latina. Recuperado el 19 de Octubre de 2009, de la **página web** denominada UNESCO, url:<http://unesdoc.unesco.org/images/0009/000919/091968s.pdf>.
- [SC07] David Schultz and Craig Cook. *Beginning HTML with CSS and XHTML*. Apress, first edition, 2007.
- [Ser] Educational Testing Service. Educational testing service, generating explanations. Recuperado el 19 de Octubre de 2009, de la **página web** denominada Educational Testing Service, url: <http://www.ets.org>.
- [SYS] BLACKBOARD LEARNING SYSTEM. Blackboard learning system. Recuperado el 19 de Octubre de 2009, de la **página web** denominada BLACKBOARD LEARNING SYSTEM, url: http://library.blackboard.com/docs/brochures/Bb_Learning_System_Brochure_International_Spanish.pdf.

- [TP02] Pineda Rojas Eloy Tremblett Paul. *SUPE UTILIDADES PARA JAVA SERVER PAGES*. MCGRAW-HILL, 2002.
- [Tur02] James Turner. *MySQL and JSP Web Applications: Data-Driven Programming Using Tomcat and MySQL*. Sams Publishing, 2002.
- [UNE] UNED. Observatorio de tecnología en educación a distancia. Recuperado el 11 de Octubre de 2009, de la **página web** denominada Sakai, url: <http://observatoriouned.org/index.php/actualidad/plataformas-lms-y-similares/114-sakai.html>.
- [VCW04] Amit Bakore Vivek Chopra and Chanoch Wiggers. *Professional Apache Tomcat 5*. Wiley Publishing, Inc., 2004.

Apéndice A

Código para insertar un Blob

```
1 < @ page import="java.sql.*" >
2
3 < @ page import="java.io.*" >
4
5 < @ page import="java.net.*" >
6
7 < @ page import="javax.servlet.*" >
8
9 < @ page import="javax.servlet.http.*" >
10
11 <%
12
13     Connection conn=null;
14
15     ResultSet rs;
16
17     Statement sentencia;
18
19     String login = "usuario";
20
21     String password = "password";
22
23     String bd = "baseDeDatos";
24
25     String url = "jdbc:mysql://localhost:3306/"+bd;
```

```

26
27     conn = DriverManager.getConnection(url , login , password);
28
29     sentencia = null;
30
31     rs = null;
32
33     sentencia = conn.createStatement();
34
35     rs = sentencia.executeQuery("SELECT * FROM
36     tablaconcampoTipoBlob WHERE idTabla="1");
37
38     if (rs.next()){
39
40         response.setContentType("application/pdf");
41         //Formato de salida
42
43         InputStream in=rs.getBinaryStream("campoTipoBlob");
44
45         OutputStream aux=response.getOutputStream();
46
47         byte [] buffer = new byte[4096];
48
49         for (;;) {
50
51             int nBytes = in.read(buffer);
52
53             if (nBytes == -1)
54
55                 break;
56
57             aux.write(buffer ,0 ,nBytes);
58
59         }
60
61         in.close();
62

```

```
63         aux.flush ();
64
65         aux.close ();
66
67     }else {
68
69         throw new SQLException("image not found");
70         con.close ();
71     }
72 %>
```

Apéndice B

Instalación y Configuración de Plataformas

B.1. Instalación de Netbeans

Netbeans IDE es un entorno de desarrollo visual de código abierto para programar aplicaciones PHP, C++, etc., pero está orientado principalmente a Java, debido a que este lenguaje de programación es uno de los más poderosos y eficientes del momento.

Su aprendizaje se ha convertido en algo fundamental para quienes están interesados en el desarrollo de aplicaciones multiplataforma. No importa que la máquina donde se instale sea un Linux, un Mac o un Windows, pues el funcionamiento del programa creado será igual.

Mediante Netbeans es posible diseñar aplicaciones con sólo arrastrar y soltar objetos sobre la interfaz de un formulario. Con Netbeans IDE no solo es posible elaborar potentes aplicaciones para el Escritorio, también para la Web y para dispositivos portátiles, como móviles o Pocket PC, sin que cambie la forma de programar.

La programación mediante Netbeans se realiza a través de componentes de software modulares, también llamados módulos.

La instalación de Netbeans, en cualquiera de los sistemas operativos Windows, Linux o Solaris, requiere de la instalación previa del JDK (Java Development Kit). Actualmente la versión más reciente es JDK 6 update 14, mientras que en lo que se refiere al IDE, la versión más moderna es Netbeans 6.8

Existen dos formas básicas de instalar esta combinación en Windows:

A) De forma conjunta, es decir, el mismo instalador monta conjuntamente ambos productos. Esta opción presenta el inconveniente de que la instalación se realiza con un grupo de opciones y componentes elegidos por Sun, y por ello, pudiera ser que dichos componentes o plugins no sean exactamente los que se necesitan para la tarea prevista.

B) De forma separada, es decir, primero se instala independientemente JDK 6 update 14 y luego más tarde Netbeans en este caso será su versión 6.7 pero la acción de instalación es la misma para la versión 6.8 que es la más reciente. Esta opción, resulta más conveniente, ya que, como mas tarde se verá, en el momento de la descarga de Netbeans se pueden elegir los componentes, plugins, etc., de acuerdo a las necesidades del proyecto que se quiere realizar. Se elegirá aquí la opción B, para ello se necesita:

B.1.1. Descargar e Instalar JDK 6 Update 14

Para ello se debe ir la página: <http://Java.sun.com/Javase/downloads/widget/JDK6.JSP>, ver Figura B.1.



Figura B.1: Página de descarga

Se selecciona el sistema operativo Linux o Windows, y se elige la opción Downloads, consecuentemente, aparecerá una pantalla que pedirá que se identifique,

si se tiene alguna cuenta de usuario o si se desea adquirirla para recibir información acerca de Java, ésta es opcional y si se desea se puede hacer el registro o sólo cerrarla, ver Figura B.2.

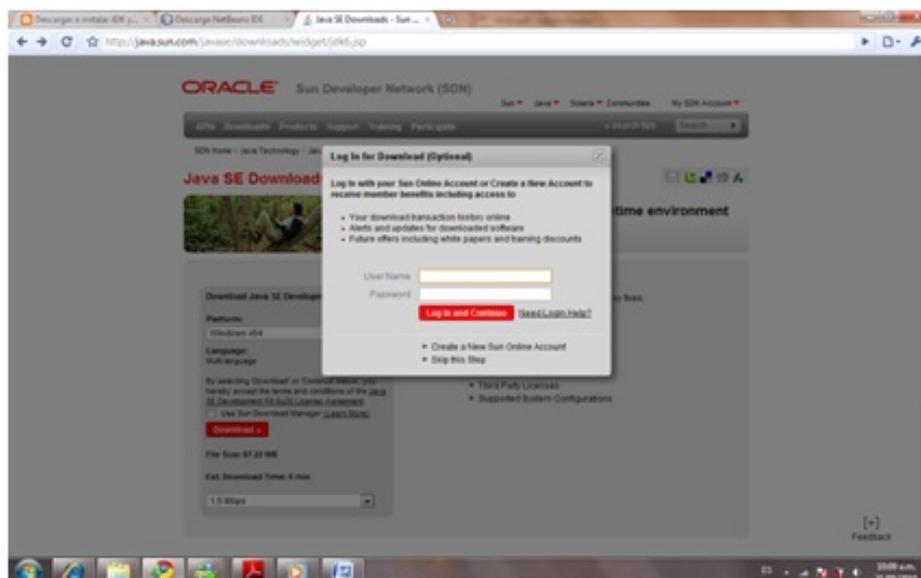


Figura B.2: Ventana para identificarse en la página

Posteriormente empieza la descarga del JDK en este caso como es a través del Navegador Chrome se realiza la descarga en la carpeta asignada para ello, en otros exploradores como Internet Explorer pide elegir la ruta.

Una vez realizada la descarga se procede a instalar el JDK. Para instalarlo se hace doble clic sobre el icono como se muestra en la Figura B.3.



Figura B.3: Icono para instalación de JDK

Aparecerá una ventana similar a la Figura B.4.

Se hace clic en "Ejecutar" y aparecerán los términos de la licencia. ver Figura B.5.

Se aceptan los términos de la licencia. Ver Figura B.6.



Figura B.4: Ventana para ejecutar o cancelar la instalación

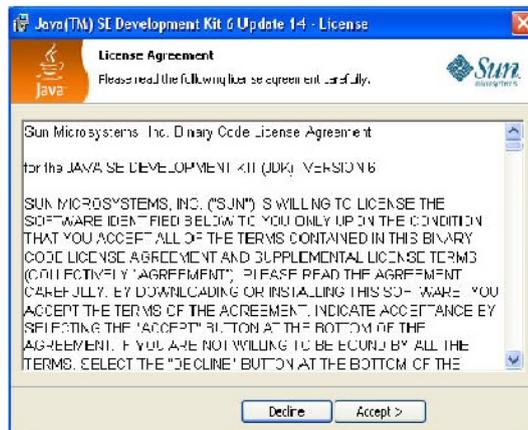


Figura B.5: Ventana de terminos de licencia

Tal como puede verse en la Figura B.6, existen 5 componentes: Development Tools, Demos and Samples, Source Code, Public JRE y Java DB.

De todos estos elementos, Development Tools es el único de instalación obligatoria. (Aunque lo recomendable es instalar *TODO*). Este elemento es el Java SE Development Kit 6 Update 14, que incluye JRE (Java Runtime Environment) a Update 14.

El paquete JRE proporciona la máquina virtual de Java ó JVM (Java Virtual Machine), para que el código sea interpretado para un determinado sistema operativo. Además, contiene todo el conjunto de API's indispensables para que el código Java pueda ejecutarse. Existe un JVM por cada sistema operativo (Windows, Linux, Solaris), e incluso plataforma, como por ejemplo dispositivos móviles, PDA y otros. El componente necesita 300 Mb en disco duro.

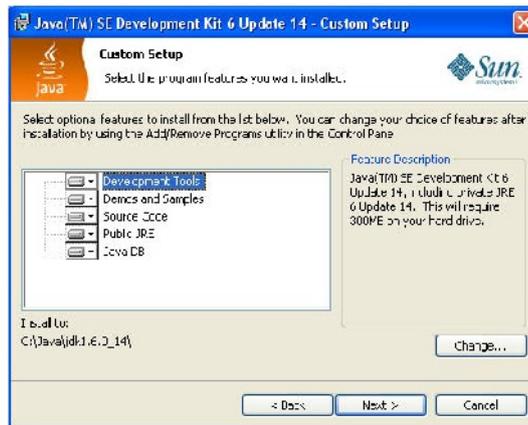


Figura B.6: Ventana para elegir los componentes a instalar

Demos and Samples: Este componente proporciona el código fuente de algunas Demos y Applets. El componente necesita 46 Mb de disco duro.

Source Code: Este componente contiene el código fuente de las clases que componen el API de Java. El componente necesita 49 Mb de disco duro.

Public JRE: El componente necesita 173 Mb de disco duro.

Java DB: Este componente es la base de datos desarrollada por Sun con un código 100% Java. El componente necesita 25 Mb de disco duro.

Si se presiona *Next* aparecerá una ventana igual a la Figura B.7.

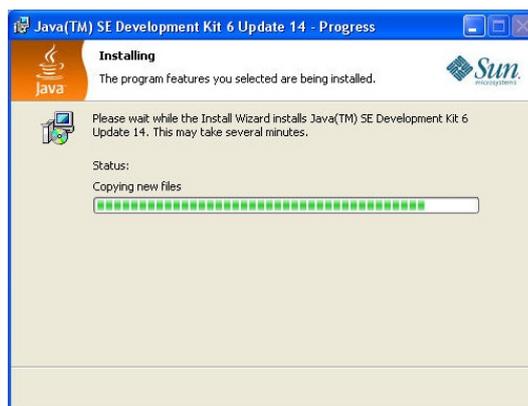


Figura B.7: Estado de la instalación

Al finalizar la instalación, aparecerá una ventana como la que se muestra en la Figura B.8.



Figura B.8: Fin de la instalación

Si se hace clic en *Finish* aparece la pantalla de registro, como se muestra en la Figura B.9.



Figura B.9: Pantalla de registro de productos Sun

Esta pantalla pretende servir de puente entre los "desarrolladores" y el equipo de Sun que desarrolla las herramientas para así establecer una adecuada política de uso y satisfacción de los desarrolladores. Aunque el registro no es obligatorio es interesante realizarlo por las ventajas que ello supone.

B.1.2. Instalacion de Netbeans 6.7

Se verá ahora la segunda parte de la instalación, en este caso para ejemplificar se hace la instalación de Netbeans 6.7 pero se encuentran comúnmente disponibles mas versiones de Netbeans a medida que transcurre el tiempo, así procedemos la descarga e instalación del propio Netbeans 6.7.

Para ello se visita a la página principal de Netbeans <http://www.netbeans.org/>. Ver Figura B.10.



Figura B.10: Pantalla principal de Netbeans

Si se hace clic en Download Netbeans IDE aparecerá la siguiente pantalla. Ver Figura B.11.

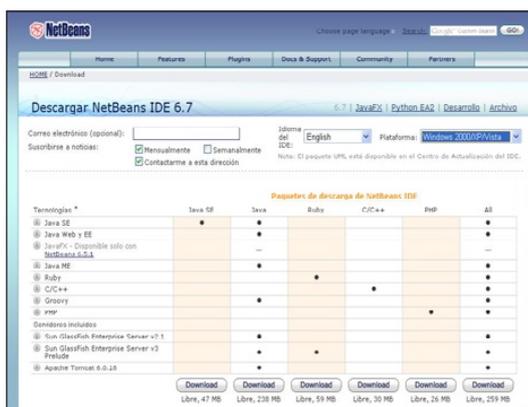


Figura B.11: Complementos para descargar con Netbeans

Tal como puede observarse, se puede elegir inicialmente tanto el lenguaje como el sistema operativo en que deseamos trabajar. Por otra parte se puede seleccionar las opciones que se ajusten a los proyectos con los que se desea trabajar, así como los servidores web que permitirán ejecutar las aplicaciones y finalmente también puede seleccionarse las distintas plataformas con las que deseamos trabajar según el proyecto: J2SE, J2EE y J2ME.

Si se elige la combinación *All* en Español y en Windows, finalmente se

descargará Netbeans 6.7 Instalador para Windows/Español (es) netbeans-6.7-ml-windows.exe (254.6 MB).

Una vez descargado, para instalar, se hace doble clic en el icono que se muestra en la Figura B.12.



Figura B.12: Icono de instalacion de Netbeans

y aparecerá algo similar a la Figura B.13



Figura B.13: Inicio del Asistente de instalacion de Netbeans

Antes de continuar y con objeto de disponer del servidor de aplicaciones Apache Tomcat que es el servidor con el cual se trabajó para realizar el proyecto de evaluación en línea, hacemos clic en "Personalizar", con lo que aparecerá una ventana similar a la Figura B.14

Se selecciona Apache Tomcat 6.0.18 y se acepta.

Ahora se hace clic en Siguiente, ver Figura B.15.

Se aceptan los términos de la licencia y se hace clic en Siguiente. Ver Figura B.16.

Se fija el directorio y subdirectorios donde se desea instalar y se presiona el botón Siguiente. Ver Figura B.17.

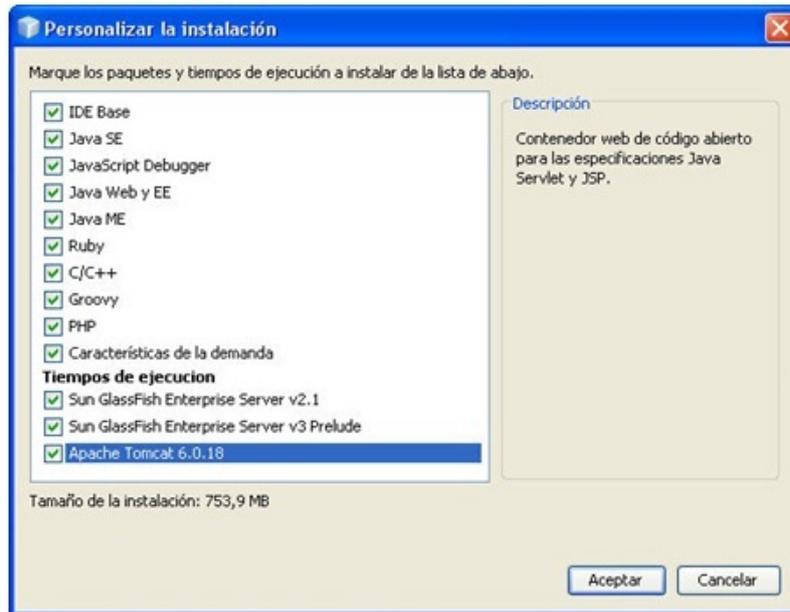


Figura B.14: Ventana para personalizar la instalación de Netbeans



Figura B.15: Inicio del Asistente de instalación de Netbeans personalizado

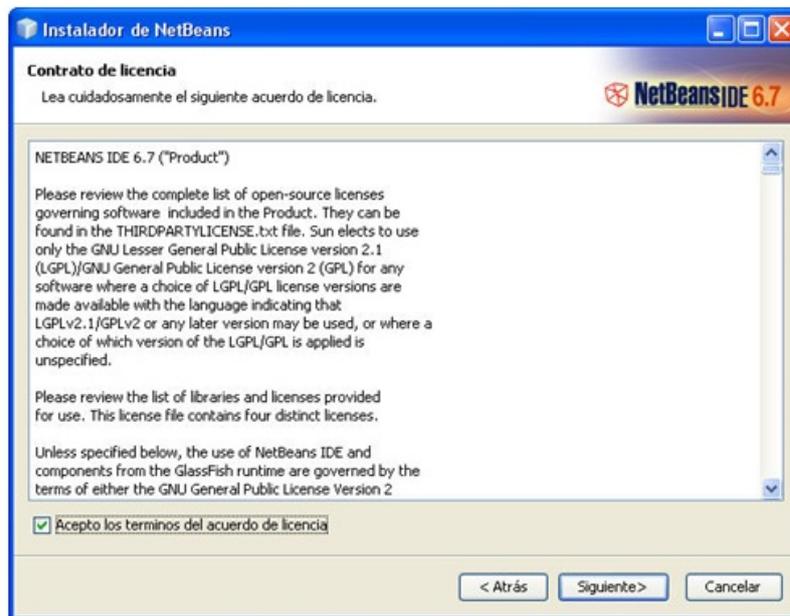


Figura B.16: Términos de Licencia para instalación de Netbeans

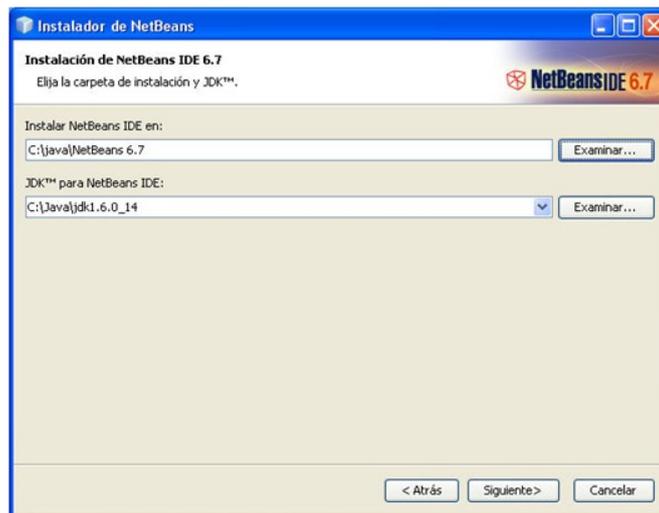


Figura B.17: Selección de Ruta para instalar Netbeans y elegir la ruta de JDK

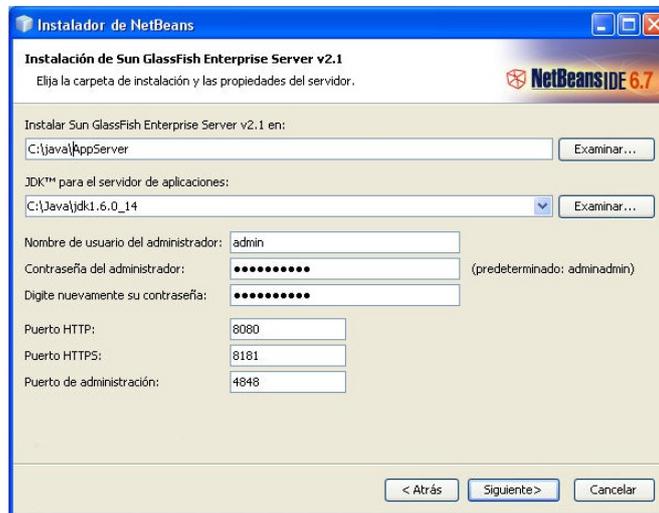


Figura B.18: Selección de Ruta y Usuario para instalar GlashFish

(Note aquí que el sistema ha detectado la presencia del JDK, de no existir éste previamente, será imposible instalar Netbeans).

Se fija el directorio y subdirectorios para el servidor de aplicaciones GlassFish Enterprise, así como usuario, contraseña y puertos. Se pulsa *Siguiente*. Ver Figura B.18.



Figura B.19: Selección de Ruta para instalar GlashFish v3 Prelude

Se fija ahora el directorio y subdirectorio del servidor GlassFish Enterprise v3 Prelude, se pulsa *Siguiente*. Ver Figura B.19.

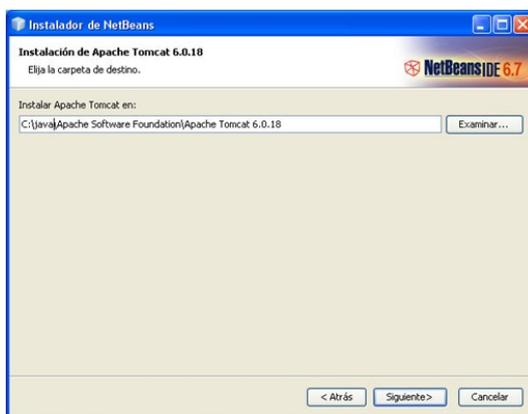


Figura B.20: Selección de Ruta para instalar Tomcat

Se fija ahora el directorio y subdirectorio de Apache Tomcat, y se oprime *Siguiente*. Ver Figura B.20.

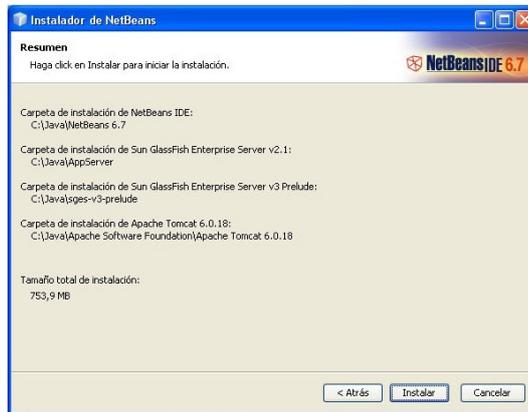


Figura B.21: Características a Instalar

La Figura B.21 presenta el resumen de todas las carpetas indicadas, así como, la cantidad de memoria de disco necesaria.

Si se está conforme, se hace clic en *Instalar*.

Si se desea se elige la opción para hacer el registro y finalmente se oprime *Terminar*. Ver Figura B.22.



Figura B.22: Término de la instalación

Si se eligió la opción para registro aparece una ventana similar a la Figura B.23.

Se crea una cuenta si se desea, y se cierra el navegador, y con esto queda terminada la instalación de Netbeans 6.7

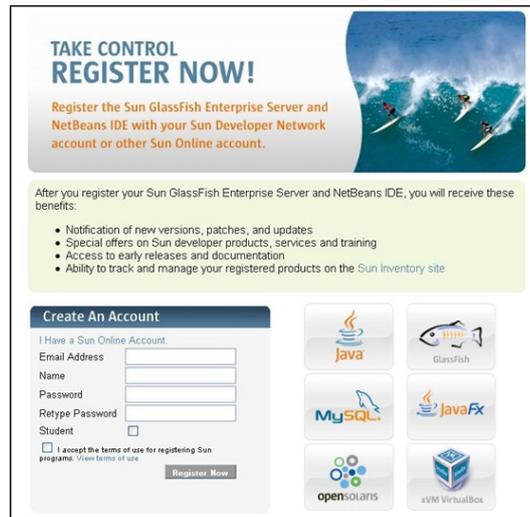


Figura B.23: Registro del producto

Se comprueba que en el escritorio se ha incluido un nuevo icono que permite ejecutar Netbeans IDE 6.7. Ver Figura B.24



Figura B.24: Icono de Netbeans

Si se hace doble clic aparecerá el IDE. Ver Figura B.25.

Con esto ha finalizado la instalación de Netbeans

B.2. Instalación de Mysql

Previamente, al igual que en la instalación de Netbeans, se debe contar con la paquetería en este caso MySQL, se ingresa a la página y se obtiene la descarga de MySQL, en este caso para Windows.

<http://dev.mysql.com/downloads/mirror.PHP?id=389367#mirrors>, ver Figura B.26



Figura B.25: IDE Netbeans

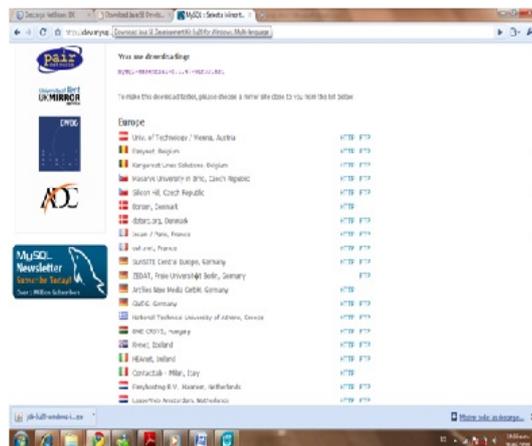


Figura B.26: Página de descarga

Inicialmente se debe dar doble clic en el archivo mysql-essential-5.0.51-win32.msi

El servidor MySQL Server 5.0 - Setup Wizard aparece con el mensaje Welcome to the Setup Wizard for MySQL Server 5.0. Ver Figura B.27

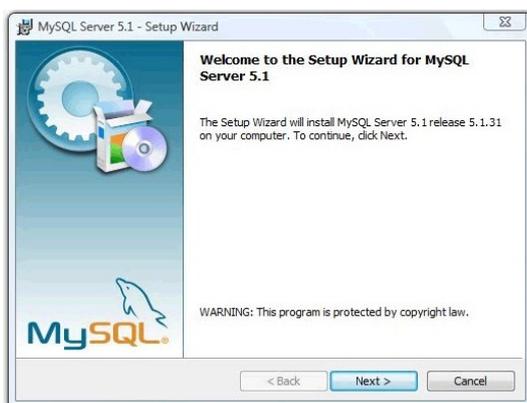


Figura B.27: Inicio Asistente de MySQL

Dar clic en siguiente y seleccionar el tipo de instalación

Verificar que la opción Típica esté seleccionada y continuar. Posterior a esto se debe dar clic en Instalar

Aparecerá el mensaje “Instalado MySQL Server 5.0”, se debe esperar mientras el asistente instala MySQL.

Verifique que Skip Sign-Up está seleccionado. Al terminar el asistente se debe verificar que “Configurar el Servidor MySQL Server ahora” está activado y dar clic en *Finalizar*. Ver Figura B.28.

El asistente de la instancia del Servidor MySQL Server aparece con el mensaje Bienvenido al asistente de la configuración de la instancia de MySQL Server. Dar clic en siguiente.

Verifique que la opción Configuración Estándar está seleccionada y dar clic en siguiente. Ver Figura B.29.

Establecer las opciones:

Verifique que *Install as a Windows Service* está seleccionada.



Figura B.28: Fin de la instalación de MySQL

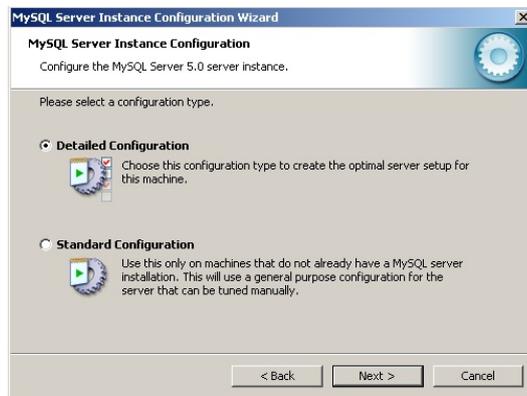


Figura B.29: Asistente de configuración de MySQL

Verifique que *MySQL* es el nombre del servicio de Windows a utilizar.

Verificar que *Launch the MySQL Server automatically* está marcada.

Verificar que *Include Bin Directory in Windows PATH* está seleccionado y dar clic en siguiente.

Configurar las opciones de seguridad, ingrese y confirme (reingrese) su contraseña y dar clic en Siguiente. Ver Figura B.30.

Aparecerá listo para ejecutar, dar clic en *Ejecutar*. Ver Figura B.31.

Aparecerá el mensaje Procesando configuración. Espere por favor mientras el asistente de configuración configure el Servidor MySQL. Dar clic en Finalizar.

Connection Error



Figura B.30: Configuración de seguridad



Figura B.31: Inicio de instalación de configuración

En algunos casos (usualmente cuando MySQL está siendo instalado sobre versiones previas), la instancia de configuración del Servidor MySQL dice un mensaje similar a *Connection Error*. Ver Figura B.32.

En este caso se debe ingresar al menú *Todos los programas*, reconfigurar la instancia de MySQL Server.

Fin de la instalación de MySQL.

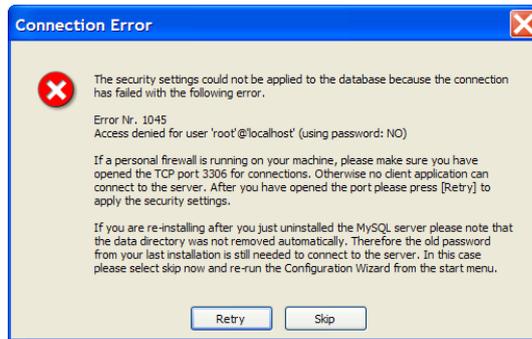


Figura B.32: Mensaje de error

Apéndice C

Modelo de la Base de Datos

