



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“SISTEMA DE EVALUACIÓN PARA LA MATERIA
CIBERNÉTICA Y COMPUTACIÓN EN EL CCH
ORIENTE”.**

**TRABAJO ESCRITO
EN LA MODALIDAD DE SEMINARIOS
Y CURSOS DE ACTUALIZACIÓN Y
CAPACITACIÓN PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE
INGENIERA EN COMPUTACIÓN
P R E S E N T A
MARÍA DEL SOCORRO ÁVILA NICOLÁS**

ASESOR: M. en C. JESÚS HERNÁNDEZ CABRERA



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A Dios:

Por permitirme llegar hasta este momento tan importante de mi vida y por mi familia.

A mis padres: Saúl y Liboria.

Por su trabajo, apoyo, dedicación, amor, comprensión y sabiduría que siempre me han brindado. Porque siempre me lo dijeron es la mejor herencia que me han podido dar. Mamá gracias por continuar.

A mis hermanos: Marce, Fer, Ric, Caro, Carmen, Chayo, Hilda y Lalo.

Por su apoyo invaluable, comprensión y consejos. Sé que cuento con ellos siempre.

A la UNAM:

Por permitirme formar parte de ella, y brindado las bases de mi profesión.

A mi Asesor: M. en C. Jesús Hernández Cabrera

Por su tiempo, apoyo y opiniones, que me brindó para llevar a buen término mi trabajo.

A todos mis amigos.

Por su apoyo y compañía para la culminación de este trabajo.

***Gracias a todos!!
Gracias por ayudarme a lograrlo.***



ÍNDICE

Introducción	5
---------------------------	----------

Capítulo 1. Introducción a las Tecnologías en Desarrollo WEB

1.1 Navegadores.....	8
1.2 Servidores WEB	9
1.2.1 Servidor HTTP ISS	10
1.2.2 Servidor WEB Apache	11
1.2.3 Servidor de Aplicación Tomcat	12
1.3 Lenguajes de Programación WEB	12
1.3.1 Hyper Text Pre-Processor - PHP	12
1.3.2 Active Server Pages - ASP	13
1.3.3 JavaServer Pages - JSP.....	13
1.3.4 Servlets.....	14
1.3.5 ASP .NET	15
1.4 Sistemas Manejadores de Bases de Datos (SMBD)	16
1.4.1 PostgreSQL	16
1.4.2 Microsoft Server SQL.....	17
1.4.3 MySQL.....	18
1.4.4 Oracle.....	19
1.4.5 SQL	19
1.5 Acceso a BD con Java (JDBC).....	22
1.6 Patrón de Acceso a Datos	24
1.6.1 Diagrama de clases del patrón DAO.....	25

Capítulo 2. Análisis de Sistemas de Administración Escolar

2.1 Educación en la WEB	28
2.1.1 Sistemas de Administración de Aprendizaje	30
2.1.2 Sistemas de Administración de contenidos de Aprendizaje	30
2.1.3 Estándares en E-learning	32



2.1.4 Ventajas de Uso de Estándares.....	33
2.2 Plataformas E-learning de Licencia Pública	34
2.2.1 Atutor	35
2.2.2 Claroline	37
2.2.3 Moodle.....	39
2.3 Evaluación de Plataformas E-learning de Licencia Pública.....	42
2.3.1 Evaluación Nivel Administrador	42
2.3.2 Evaluación Tipos de Reactivos	43
2.3.3 Administración de Reactivos	44
2.3.4 Abstracción de Datos.....	44
2.3.5 Tipos de cuestionarios	45
2.3.6 Conclusión.....	45

Capítulo 3. Análisis y Planteamiento del Sistema

3.1 Acta de Proyecto	47
3.1.1 Objetivo del Proyecto.....	47
3.1.2 Breve descripción del Proyecto.....	47
3.1.3 Alcance.....	47
3.2 Desarrollo	48
3.2.1 Consideraciones	48
3.2.2 Necesidades que se Plantean Cubrir con la Adquisición del Software	49
3.2.3 Consideraciones Normativas	49
3.2.4 Definición de Requerimientos	49
3.3 Definición de Audencia(Niveles de Usuario)	52
3.4 Seguridad	53
3.5 Diagramas de Caso de Uso	54
3.6 Narrativas	56
3.6.1 Administrador – Administrar Alumnos	56
Casos de Uso de Alto Nivel	56
Casos de Uso de Formato Expandido	57
Caso de Uso: Agregar Alumnos	59
Caso de Uso: Eliminar Alumnos.....	60



Caso de Uso: Modificar Alumnos	62
3.6.2 Administrador - Administrar Profesores.....	64
Casos de Uso de Alto Nivel	64
3.6.3 Administrador – Realizar Examen.....	65
Casos de Uso de Alto Nivel	65
Casos de Uso de Formato Expandido	66
3.6.4 Administrador – Generar Reporte de Resultados.....	68
Casos de Uso de Alto Nivel	69
Casos de Uso de Formato Expandido	69
3.6.5 Profesor – Administrar Preguntas	72
Casos de Uso de Alto Nivel	72
Casos de Uso de Formato Expandido	73
Casos de Uso: Modificar Preguntas	75
3.6.6 Profesor – Consultar Examen	76
Casos de Uso de Formato Expandido	76
3.6.7 Alumno – Resolver Examen.....	78
Casos de Uso de Alto Nivel	78
Casos de Uso de Formato Expandido	79
3.7 Organización del Proyecto.....	80
3.7.1 Interfaces Externas.....	80
3.8 Mapas del Sitio.....	80
3.8.1 Árbol Funcional.....	81
3.9 Modelo Conceptual.....	83

Capítulo 4. Desarrollo e Implementación

4.1 Diagrama de Actividades	85
4.2 Interfaces WEB	86
4.2.1 Interfaces de Contenido de Información.....	86
4.3 Inicio.....	88
4.4 Acceso a Usuario	90
4.5 Agregar Usuario	93
4.6 Consulta de Usuario.....	95



4.7 Elección de Tipo de Pregunta	101
4.8 Vista de Examen	104
Conclusiones.....	110
Índice de Tablas y Figuras.....	112
Bibliografía.....	113



INTRODUCCIÓN

El presente trabajo se basa en el Diplomado “Diseño de Sistemas en WEB” 6ª generación, impartido en la FES Aragón; y se enfoca en la implementación del Sistema de Evaluación para la materia Cibernética y Computación en el CCH Oriente, considerando todos los aprendizajes obtenidos; se utiliza para el análisis y diseño, el lenguaje unificado de modelado (UML) que permite crear el análisis y diseño orientado a objetos; para el desarrollo se emplea el lenguaje de programación Java; para la capa de usuario, se utilizan algunas de las herramientas que se dieron a conocer para que el entorno sea amigable, en este caso se aplican las hojas de estilo en cascada (CSS); para la capa de datos se hace uso del SDBD MySQL, se trabaja con un acceso a datos con java (JDBC) y un patrón de acceso a datos en java (DAO); para las reglas del negocio, se manejan servlets y JSP’s formando páginas WEB de forma dinámica; todo esto en conjunto será utilizado para llevar a cabo dicho sistema.

El desarrollo del siguiente trabajo se presenta de la siguiente forma:

En el primer capítulo, se hace una breve descripción de las tecnologías para el desarrollo de sistemas en WEB, esto es, navegadores, servidores, lenguajes de programación, sistemas manejadores de bases de datos y el acceso a base de datos.

El segundo capítulo, trata sobre las plataformas eLearning de licencia pública, en donde se proporcionan algunas de las características de Atutor, Claroline y Moodle, asimismo se realiza una comparación entre ellas.

En el tercer capítulo, se expone el análisis y diseño del sistema a realizar, dando a conocer la organización del proyecto, requerimientos mínimos, casos de uso, narrativas, mapas del sitio, y mapa conceptual.



En el cuarto capítulo, se da a conocer el desarrollo e implementación del sistema, mostrando las pantallas resultantes del proyecto y código que se utilizó para lograrlo. Después de lo anterior se dan a conocer las conclusiones obtenidas del resultado.

CAPÍTULO 1

INTRODUCCIÓN A LAS TECNOLOGÍAS EN DESARROLLO WEB



INTRODUCCIÓN A LAS TECNOLOGÍAS EN DESARROLLO WEB

1.1 NAVEGADORES

Un navegador es un intérprete de HTML¹, es un programa generalmente gratuito, que instalados en el ordenador permiten ver documentos almacenados en el disco duro, usb, cd's, o a través de Internet, acceder a documentos alojados en servidores Web. Existe una variedad de navegadores, los cuales comparten características, funciones y prestaciones fundamentales. Contienen una barra de direcciones para escribir la dirección de la página Web (URL)² y una serie de botones básicos para ir hacia adelante, hacia atrás, actualizar la página, etc. También todos incluyen gestión de 'Favoritos' y del 'Historial'.

Los navegadores son utilizados para moverse por Internet, con los que se pueden ver los documentos (normalmente páginas en HTML) de los sitios Web.

Entre los diferentes navegadores se encuentran: Internet Explorer³ (IE) de Microsoft y está integrado en el sistema operativo Windows; Firefox⁴, el navegador de software libre de la Fundación Mozilla; Opera, Netscape⁵. Dependiendo del sistema operativo, el usuario puede elegir entre los navegadores.

Sin duda, lo fundamental es que el navegador sea capaz de cargar cualquier tipo de página Web sin problemas. Los navegadores importantes están adaptados a los estándares más comunes, algunas páginas diseñadas sólo para IE presentan

¹ HTML- HyperText Markup Language (Lenguaje de Marcas de Hipertexto. Lenguaje de programación para la creación de páginas Web

² URL - Uniform Resource Locator (Localizador Uniforme de Recurso). Dirección del recurso (página, imagen, etc.) en Internet.

³ Internet Explorer. <http://www.microsoft.com/spain/windows/products/winfamily/ie/default.mspx>

⁴ FireFox. <http://www.mozilla-europe.org/es/firefox/>

⁵ Netscape. <http://browser.netscape.com/releases>



problemas al cargarse en un navegador distinto. Sin embargo, aunque en ocasiones no vengan integrados, los navegadores están preparados para añadir los *plug-ins* (accesorios) necesarios para visualizar casi cualquier tipo de página (en Flash, PDF, Java, etc.).

Algunas de las prestaciones más interesantes que incluyen los navegadores:

- Navegación con pestañas.
- Publicidad.
- Buscadores.
- Personalización (añadir nuevos botones o barras de herramientas).
- Gestor de descargas.
- Seguridad y Privacidad (Gestionar contraseñas).

Todos los navegadores están preparados para visualizar páginas Web seguras, aquellas que protegen los datos enviados a través de Internet (utilizadas en compras y transacciones electrónicas).

1.2 SERVIDORES WEB

Un servidor Web es un programa que se ejecuta de forma continua en una computadora, manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que contesta a estas peticiones de forma adecuada, sirviendo una página Web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

Un servidor Web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

1. Espera peticiones en el puerto TCP⁶ indicado (el estándar por defecto para HTTP⁷ es el 80).

⁶ TCP – Transmission Control Protocol (Protocolo de Control de Transmisión). El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron



2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió petición.
5. Vuelve al segundo punto.

Un servidor Web que siga el esquema anterior cumplirá todos los requisitos básicos de los servidores HTTP, aunque sólo podrá proporcionar archivos estáticos.

Entre los servidores Web más utilizados se incluyen Microsoft Internet Information Server (IIS⁸), Apache⁹ HTTP Server, Tomcat¹⁰, entre otros.

1.2.1 Servidor HTTP IIS

Internet Information Server (IIS), es una serie de servicios para los ordenadores que funcionan con Windows.

Este servicio convierte a una computadora en un servidor de Internet o Intranet, es decir, que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente.

IIS es el servidor de páginas Web avanzado de la plataforma Windows. Se distribuye gratuitamente junto con las versiones de Windows basadas en NT, como pueden ser Windows 2000 Profesional o Windows 2000 Server, así como Windows XP, también en sus versiones Profesional y Server.

El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages

⁷ HTTP - HyperText Transfer Protocol (Protocolo de Transferencia de HiperTexto). Es el protocolo usado en cada transacción de la Web WWW.

⁸ Servidor ISS. <http://www.microsoft.com/DOWNLOADS/details.aspx?familyid=9EC9DBB9-82AD-4D34-9267-76A0126A8F18&displaylang=en>

⁹ Servidor Apache. <http://www.apache.org/dyn/closer.cgi>

¹⁰ Servidor Tomcat. <http://tomcat.apache.org/>



(ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

1.2.3 Servidor WEB Apache

Apache es uno de los servidores Web más populares del mercado, y el más utilizado actualmente, de código abierto y gratuito, disponible para Windows y Linux, entre otros.

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- Módulos Base: Módulo con las funciones básicas del Apache.
- Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software. Este servidor sirve solo para páginas de hipertexto (páginas estáticas).



1.2.4 Servidor de Aplicación Tomcat

Tomcat puede funcionar como un servidor independiente o como un plugin¹¹ de Apache por medio de conectores.

Este servidor es útil para las páginas dinámicas.

Utilizando la potencia del Servidor Apache y Tomcat como contenedor Web para la generación de páginas Web dinámicas, se obtiene una mejor optimización; para esto se añaden múltiples instancias de Tomcat para dotar de alta disponibilidad al servidor de aplicaciones, haciendo uso del conector `mod_jk`¹² como balanceador de carga y failover¹³.

En este caso, tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

1.3 LENGUAJES DE PROGRAMACIÓN EN WEB

1.3.1 PHP - Hyper Text Pre-Processor

PHP es un lenguaje de programación usado normalmente para la creación de páginas Web dinámicas, que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado. Últimamente también puede ser utilizado para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando las librerías Qt o GTK+.

No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, pero a diferencia de Java o JavaScript que se ejecutan en el

¹¹ Plugin. Pequeño programa que añade alguna función a otro programa. Son utilizados en los programas navegadores para ampliar sus funcionalidades

¹² Mod_jk. Módulo utilizado por Apache para comunicarse con Tomcat.
<http://tomcat.apache.org/connectors-doc/>

¹³ Failover. Rescate de la conexión. Realiza conexión de emergencia después de una falla.



navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

1.3.2 Active Server Pages – ASP

Active Server Pages (ASP) es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente. La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida ASP, con limitaciones ya que es una plataforma que no se ha desarrollado como lo esperaba Microsoft.

ASP, tiene objetos integrados disponibles para el programador, Application, ASPError, Request, Response, Server y Session. Cada objeto tiene un grupo de funcionalidades frecuentemente usadas y útiles para crear páginas Web dinámicas.

Las páginas pueden ser generadas mezclando código de scripts del lado del servidor (incluyendo acceso a base de datos) con HTML.

1.3.3 JavaServer Pages – JSP

JavaServer Pages (JSP) es una tecnología Java¹⁴ que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo.

Esta tecnología es un desarrollo de la compañía Sun Microsystems.

¹⁴ Java. <http://www.java.com/es/download/manual.jsp>



Las JSP's permiten la utilización de código Java mediante scriptlets. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

Microsoft, la más directa competencia de Sun, ha visto en esta estrategia una amenaza, lo que le ha llevado a que su plataforma .NET incluya su lenguaje de scriptlets ASP.NET que permite ser integrado con clases .NET (ya estén hechas en C++, VisualBasic o C#) del mismo modo que JSP se integra con clases Java.

1.3.4 Servlets

La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador Web. Por contraposición, un servlet es un programa que se ejecuta en un servidor.

Los servlets son objetos que corren dentro del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones que además de contenedor para servlet tendrá contenedor para objetos más avanzados (ej. OC4J Oracle) como son los EJB¹⁵ (Tomcat sólo es un contenedor de servlets).

El uso más común de los servlets es generar páginas Web de forma dinámica a partir de los parámetros de la petición que envíe el navegador Web.

Un servlet es un objeto que se ejecuta en un servidor o contenedor J2EE (Java Enterprise Edition), fue especialmente diseñado para ofrecer contenido dinámico desde un servidor Web, generalmente es HTML.

¹⁵ EJB. Enterprise Java Bean. Agrupa funcionalidades para una aplicación, en un ambiente de ejecución.



Los servlets forman parte de J2EE, que es una ampliación de JSE (Java Standard Edition).

Un servlet es un objeto Java que implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ej. `javax.servlet.HttpServlet`). Al implementar esta interfaz el servlet es capaz de interpretar los objetos de tipo `HttpServletRequest` y `HttpServletResponse` quienes contienen la información de la página que invocó al servlet.

Entre el servidor de aplicaciones (o Web content) y el servlet existe un contrato que determina cómo han de interactuar. La especificación de éste se encuentra en los JSR (Java Specification Requests) del JCP (Java Community Process).

1.3.5 ASP .NET

Es una plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones. Basado en esta plataforma, Microsoft intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de mercado.

.NET¹⁶ podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Sun Microsystems.

La plataforma .NET pretende proveer un conjunto único y expandible con facilidad, de bloques interconectados, diseñados de forma uniforme y bien documentados, que permitan a los desarrolladores tener a mano todo lo que necesitan para producir aplicaciones sólidas.

¹⁶ .NET. <http://www.microsoft.com/NET/>



Con esta plataforma Microsoft incursiona de lleno en el campo de los Servicios Web y establece el XML como norma en el transporte de información en sus productos y lo promociona como tal en los sistemas desarrollados utilizando sus herramientas.

.NET intenta ofrecer una manera rápida y económica pero a la vez segura y robusta de desarrollar aplicaciones - o como la misma plataforma las denomina, soluciones - permitiendo a su vez una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

1.4 SISTEMAS MANEJADORES DE BASES DE DATOS (SMBD)

Una base de datos es un conjunto de datos organizados, estructurados y relacionados entre sí, con el objetivo de estar disponibles para ser consultados. Además proporciona la infraestructura requerida para los sistemas de apoyo a la toma de decisiones, y a los usuarios una visión abstracta de los datos.

Un SMBD es Conjunto de programas organizados cuyo objetivo principal es apoyar a la creación, manipulación y recuperación de bases de datos, de una manera práctica y eficiente. Éste abstrae la capa física.

1.4.1 PostgreSQL

PostgreSQL¹⁷ es un servidor de base de datos objeto relacional libre, liberado bajo la licencia BSD¹⁸. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su

¹⁷ PostgreSQL. <http://www.postgresql.org/>

¹⁸ Licencia BSD. Berkeley System Distribution (Distribución del Sistema Berkeley).



desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

A continuación se enumeran las principales características de este gestor de bases de datos:

- Aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.
- Soporta operadores, funciona métodos de acceso y tipos de datos definidos por el usuario.
- Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- Usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

La característica de PostgreSQL conocida como *Write Ahead Logging* (Escribir delante del registro) incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podrá restaurar la base de datos.

1.4.2 MICROSOFT SQL SERVER

Microsoft SQL Server¹⁹ es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

¹⁹ <http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>



Entre sus características figuran:

- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

1.4.3 MySql

MySQL²⁰ es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL como software libre en un esquema de licenciamiento dual. Está desarrollado en su mayor parte en lenguaje C.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Entre sus características figuran:

- Portable gracias a su código.
- Administración de usuarios
- Soporta hasta 32 índices por tabla.
- Soporta varios tipos de datos.
- Soporta SQL Standard (parcialmente)

²⁰ MySql. <http://www.mysql.com/>



1.4.4 Oracle

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

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

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Las desventajas que tiene es su precio, la seguridad de la plataforma y las políticas de suministro de parches de seguridad, que incrementan el nivel de exposición de los usuarios.

Oracle, sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros SGBD con licencia libre como PostgreSQL o MySQL. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

1.4.5 SQL

El Lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.



El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet. SQL se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordSet y como la propiedad RecordSource del control de datos. También se puede utilizar con el método Execute para crear y manipular directamente las bases de datos Jet y crear consultas SQL de paso a través para manipular bases de datos remotas cliente - servidor.

Las sentencias SQL pertenecen a dos categorías principales: Lenguaje de Definición de Datos, DDL y Lenguaje de Manipulación de Datos, DML. Estos dos lenguajes no son lenguajes en sí mismos, sino que es una forma de clasificar las sentencias de lenguaje SQL en función de su cometido. La diferencia principal reside en que el DDL crea objetos en la base de datos y sus efectos se pueden ver en el diccionario de la base de datos; mientras que el DML es el que permite consultar, insertar, modificar y eliminar la información almacenada en los objetos de la base de datos.

Cuando se ejecutan las sentencias DDL de SQL, el SGBD confirma la transacción actual antes y después de cada una de las sentencias DDL. En cambio, las sentencias DML no llevan implícito el commit y se pueden deshacer. Existe un problema al mezclar sentencias DML con DDL, ya que estas últimas pueden confirmar las primeras de manera involuntaria e implícita.

A continuación se presenta una tabla con las sentencias SQL más comunes, clasificadas según el lenguaje al que pertenecen.



Sentencia DDL	Objetivo
Alter procedure	Recompilar un procedimiento almacenado.
Alter Table	Añadir o redefinir una columna, modificar la asignación de almacenamiento.
Analyze	Recoger estadísticas de rendimiento sobre los objetos de la BD para utilizarlas en el optimizador basado en costos.
Create Table	Crear una tabla.
Create Index	Crear un índice.
Drop Table	Eliminar una tabla.
Drop Index	Eliminar un índice.
Grant	Conceder privilegios o papeles, roles, a un usuario o a otro rol.
Truncate	Eliminar todas las filas de una tabla.
Revoke	Retirar los privilegios de un usuario o rol de la base de datos.

Tabla 1.4.1 Sentencias DDL

Sentencia DML	Objetivo
Insert	Añadir filas de datos a una tabla.
Delete	Eliminar filas de datos de una tabla.
Update	Modificar los datos de una tabla.
Select	Recuperar datos de una tabla.
Commit	Confirmar como permanentes las modificaciones realizadas.
Rollback	Deshacer todas las modificaciones realizadas desde la última confirmación.

Tabla 1.4.2 Sentencias DML



1.5 ACCESO A BD CON JAVA (JDBC).

La conectividad de la base de datos de Java (JDBC , Java Database Connectivity) es un marco de programación para los desarrolladores de Java en los programas que tienen acceso a la información guardada en bases de datos, hojas de calculo, y archivos "planos". JDBC se utiliza comúnmente para conectar un programa del usuario con una base de datos por "detrás de la escena", sin importar qué software de administración o manejo de base de datos se utilice para controlarlo. De esta manera, JDBC es una plataforma-cruzada.

JDBC es un conjunto de interfaces Java que estandarizan:

- Una forma de obtener conexión a la base de datos.
- Una aproximación a consultas SQL.
- Métodos para crear consultas parametrizadas.
- Estructuras de datos para los resultados de las consultas (Metadatos).

Clases de JDBC

<i>Clase/Interface</i>	<i>Descripción</i>
Driver	Permite conectarse a una base de datos: cada gestor de base de datos requiere un driver distinto.
DriverManager	Permite gestionar todos los drivers instalados en el sistema.
DriverPropertyInfo	Proporciona diversa información acerca de un driver.
Connection	Representa una conexión con una base de datos. Una aplicación puede tener más de una conexión a más de una base de datos.
DatabaseMetadata	Proporciona información acerca de una Base de Datos, como las tablas que contiene, etc.
Statement	Permite ejecutar sentencias SQL sin parámetros.



<i>Clase/Interface</i>	<i>Descripción</i>
PreparedStatement	Permite ejecutar sentencias SQL con parámetros de entrada.
CallableStatement	Permite ejecutar sentencias SQL con parámetros de entrada y salida, típicamente procedimientos almacenados.
ResultSet	Contiene las filas o registros obtenidos al ejecutar un SELECT.
ResultSetMetadata	Permite obtener información sobre un ResultSet , como el número de columnas, sus nombres, etc.

Tabla 1.5.3 Clases JDBC

Métodos más usados:

1. **deregisterDriver(Driver driver):** elimina un driver de la lista de registros.
2. **registerDriver(Driver driver):** Registra un driver.
3. **getConnection(String url):** Trata de establecer una conexión a URL, login y password.
4. **getConnection(String url, Properties info):** URL y la clase Property info que nos empaqueta la información de usuario.
5. **getDriver(String url):** Busca un driver en una URL específica.
6. **getDrivers():** Regresa un enumerado de los drivers registrados.
7. **Connection()**
 - ✓ Representa la conexión con la base de datos.
 - ✓ Es el objeto que permite realizar las consultas SQL y obtener los resultados de dichas consultas
 - ✓ Es el objeto base para la creación de los objetos de acceso a la base de datos.
 - ✓ Permite crear objetos de las clases: **Statement**, **PreparedStatement** y **CallableStatement** para ejecutar consultas.
 - ✓ Controla las transacciones.
 - ✓ Nos permite obtener Meta información con la clase **DatabaseMetaData**.
 - ✓ Ayuda a cerrar una conexión y a averiguar el estado de la misma **Connection.isClosed()** y **Connection.close()**.



1.6 PATRÓN DE ACCESO A DATOS.

DAO (Data Access Object). Es una solución al problema diferencial impedancia entre un programa de aplicación orientado a objetos y una base de datos relacional, empleando únicamente la interfaz de programación (API) nativa del manejador de la base de datos, o algún otro sustituto como el ODBC²¹, DBI, etc.

El DAO maneja la conexión con la fuente de datos para obtener y almacenar datos.

El uso de este patrón ofrece varios beneficios para la persistencia de datos:

- Sirve para separar el acceso a datos de la lógica de negocio. Recomendable en sistemas medianos o grandes, o que manejen lógica de negocio compleja.
- Encapsula la fuente de datos. Esto es beneficioso en sistemas con acceso a múltiples entradas.
- Oculta la API²² con la que se accede a los datos.
- Centraliza todos los accesos a datos en una capa independiente.

Cuando se trabaja con el patrón DAO, los datos se deben persistir en objetos. Por lo tanto, cuando piden realizar una operación, se abre la conexión a la base, se ejecuta el comando, si es una operación de lectura, se vuelca el contenido hacia una estructura de datos y se cierra la conexión.

1.6.1 Transacciones

Existen 2 modelos diferentes que aplican según la necesidad de la operación transaccional que se va a efectuar.

Uno hace al objeto DAO responsable de la transacción, el otro desplaza la transacción al objeto que esta llamando el método DAO.

²¹ ODBC - Open Database Connectivity. Estándar de acceso a base de datos.

²² API - Application Programming Interface (Interfaz de programación de aplicaciones).



1.6.2 Diagrama de clases del patrón DAO

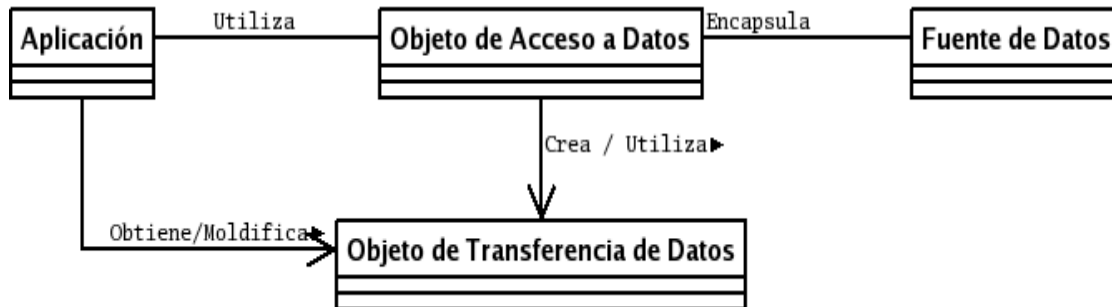


Fig. 1.6.2 Diagrama de clases DAO

La primera clase que se observa es Aplicación, es la clase cliente que necesita los datos almacenados en la fuente de datos para lograr su objetivo. Hay que hacer notar que ésta no le pide los datos directamente a la fuente de datos, sino que lo hace a través de un tercer objeto (el de acceso a datos), utilizando un cuarto objeto (el de transporte de datos) quien almacenará todos los datos que intercambiarán la aplicación con la fuente datos.

Luego esta el Objeto de Acceso a datos, quien se encargará de conectarse directamente con la fuente de datos utilizando su API nativa y ejecutará las transacciones necesarias sobre ella. Por ejemplo, si la operación es una consulta, el objeto hará la transacción indicada y devolverá a la aplicación un conjunto de objetos de transporte de datos conteniendo el resultado; por otro lado, si la transacción es una inserción, entonces la aplicación pasará como parámetro un objeto de transporte de datos que será almacenada en la fuente de datos, y así con las demás tareas como borrado y actualización.

El Objeto de Transferencia de Datos encapsula todos los datos que contiene operación sobre la fuente de datos. Podemos pensar en él como la representación de un registro (fila) de una tabla en una base de datos relacional.



Por último esta la fuente de datos, es quien almacena los datos con los que se va a trabajar. Puede ser la interfaz de programación de un manejador de base datos, de un servidor de LDAP²³, de un servicio Web de datos, etcétera.

El propósito del patrón DAO es, abstraer y encapsular todos los accesos a la fuente de datos. Con esto se obtiene varias ventajas:

1. Se tiene un paliativo al problema del diferencial de impedancia (transparencia).
2. Se baja en nivel de acoplamiento entre clases, reduciendo la complejidad de realizar cambios.
3. Se aíslan las conexiones a la fuente de datos en una capa fácilmente identificable y mantenible.

²³ LDAP - Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios).

CAPÍTULO 2

ANÁLISIS DE SISTEMAS DE ADMINISTRACIÓN ESCOLAR



2. ANÁLISIS DE SISTEMAS DE ADMINISTRACIÓN ESCOLAR

2.1 EDUCACIÓN EN LA WEB

El sector educativo ha encontrado en la Web diferentes procesos de enseñanza-aprendizaje, lo que ha permitido un desarrollo rápido y consistente. Como medio para la difusión de información ha facilitado la creación y el acceso a más contenidos y como infraestructura para impartir educación a distancia, la Web ha dado origen a un modelo conocido como *e-learning*,

La Web se ha introducido rápidamente como un recurso de apoyo, de extensión o de sustitución a los esquemas tradicionales de enseñanza presencial o a distancia. La aplicación de las tecnologías Web ha dado origen incluso a aplicaciones dinámicas que permiten la generación de cursos apegados a las necesidades particulares de cada estudiante y a las posibilidades tecnológicas y de contenidos de la Web. Existen tres formas en las que la Web participa en un proceso de enseñanza-aprendizaje:

1. *Web para el almacenamiento, diseminación y recuperación de información.* La Web es un importante apoyo para el acceso a contenidos de los sistemas de enseñanza tradicional. Los alumnos pueden consultar información o tener acceso a los contenidos de clase en la Web aunque toda la instrucción se lleva a cabo de manera presencial.
2. *Web para enseñanza en dos medios (mixta).* Muchas instituciones educativas están optando por impartir una parte de la instrucción presencial y otra parte a través de la Web, esto es lo que se conoce como aprendizaje mixto (*blended learning*).
3. *Aprendizaje Basado en Web.* Consiste en llevar a cabo todo el proceso educativo a través de la Web, prescindiendo por completo de cualquier componente de la enseñanza presencial tradicional.



Con mayor o menor penetración en cada uno de los modelos, la tecnología Web se impone como la tendencia en el desarrollo de sistemas educativos apoyados por las TIC.

E-learning (*electronic learning*): Término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en Web, aprendizaje basado en computadora, aulas virtuales y colaboración digital. Incluye entrega de contenidos vía Internet, intranet/extranet, audio y videograbaciones, transmisiones satelitales, TV interactiva, CD-ROM y más.

Ésta es una definición que abre el rango del *e-learning* prácticamente a cualquier proceso relacionado con educación y tecnologías. Sin embargo, hay autores que acotan más el alcance del *e-learning* reduciéndolo exclusivamente al ámbito de Internet, como Rosenberg que lo define como:

“... el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento. Está basado en tres criterios fundamentales:

- El *e-Learning* trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.
- Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.
- Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación”.

En la práctica, para llevar a cabo un programa de formación basado en *e-learning*, se hace uso de plataformas o sistemas de software que permiten la comunicación e interacción entre profesores, alumnos y contenidos. Se tienen principalmente dos tipos de plataformas: las que se utilizan para impartir y dar seguimiento administrativo a los cursos en línea y, por otro lado, las que se utilizan para la gestión de los contenidos digitales.



2.1.1 Sistemas de Administración de Aprendizaje

Entre las herramientas más utilizadas para los ambientes o sistemas *e-learning* están los Sistemas de Administración de Aprendizaje o LMS (*Learning Management Systems*), también ampliamente conocidos como plataformas de aprendizaje. Un LMS es un software basado en un servidor Web que provee módulos para los procesos administrativos y de seguimiento que se requieren para un sistema de enseñanza-aprendizaje, simplificando el control de estas tareas. Se distinguen como grupos funcionales de los sistemas de aprendizaje: gestión de cursos, gestión de clases, herramientas de comunicación, herramientas para los estudiantes, gestión del contenido, herramientas de evaluación y gestión de institución educativa.

En los LMS los módulos con funciones administrativas permiten, por ejemplo, configurar cursos, matricular alumnos, registrar profesores, asignar cursos a un alumno, llevar reportes de progreso y calificaciones. Los LMS también facilitan el aprendizaje distribuido y colaborativo a partir de actividades y contenidos pre-elaborados, de forma síncrona o asíncrona, utilizando los servicios de comunicación de Internet como el correo, los foros, las videoconferencias y el *chat*.

La complejidad y las capacidades de las plataformas varían de un sistema a otro, pero en general todas cuentan con funciones básicas como las que se han mencionado. Es importante resaltar que un LMS no se involucra con la creación, reutilización, administración o mantenimiento de los contenidos. Entre las plataformas comerciales más comunes se encuentran Blackboard, WebCT, EduStance de software libre las más reconocidas son Moodle, Atutor y Claroline.

2.1.2 Sistemas de administración de contenidos de aprendizaje

Los Sistemas de Administración de Contenidos de Aprendizaje o LCMS (*Learning Content Management Systems*) tienen su origen en los CMS (*Content Management*



System) cuyo objetivo es simplificar la creación y la administración de los contenidos en línea, y han sido utilizados principalmente en publicaciones periódicas (artículos, informes, fotografías, etcétera). En la mayoría de los casos lo que hacen los CMS es separar los contenidos de su presentación o estilo en pantalla y también facilitar un mecanismo de trabajo para la gestión de una publicación web. Los LCMS siguen el concepto básico de los CMS, que es la administración de contenidos, pero enfocados al ámbito educativo, administrando y concentrando únicamente recursos educativos y no todo tipo de información.

Se define un LCMS como un sistema basado en web que es utilizado para crear, aprobar, publicar, administrar y almacenar recursos educativos (como los objetos de aprendizaje) y cursos en línea.

El mercado de los LCMS es mucho más reducido que el de los LMS, entre los productos más comunes se encuentra ATutor, distribuido como software libre y compatible con estándares *e-learning* para la exportación e importación de contenidos con otras plataformas y con Repositorios de Objetos de Aprendizaje.

Principales características de una plataforma e-learning

- Interfaz sencilla e intuitiva.
- Registro y seguimiento completo de los accesos del alumno.
- Completo el sistema de evaluación y seguimiento. Dispone de informe de actividades de cada estudiante con gráficos y detalles sobre su paso por cada módulo.
- Permite mostrar cualquier contenido digital, Word, pdf, powerpoint, flash, vídeo, sonidos, etc.



2.1.3 Estándares en E-learning

Un estándar es un patrón, una tipificación o una norma sobre cómo realizar algo; hay de dos tipos: estándares de jure²⁴, los que provienen de una organización acreditada que certifica una especificación, y estándares de facto²⁵, cuando la especificación se adopta por un grupo mayoritario de individuos. Entonces, un estándar regularmente proviene de una especificación, esto es, de un conjunto de declaraciones detalladas y exactas de los requisitos funcionales y particularidades de algo que quiere construirse, instalarse o manufacturarse.

En el área educativa, las especificaciones son formalmente remitidas al LTSC (Learning Technology Standards Committee), Comité especializado en e-learning del IEEE, única organización acreditada de estandarización. Hasta el momento, el único estándar emitido es LOM (Learning Object Metadata), un esquema de metadatos para la descripción de recursos educativos.

Existen diferentes organismos que trabajan en el desarrollo de propuestas para la estandarización del e-learning, entre las cuales se encuentran y describen a continuación:

- AICC (Aviation Industry Computer-Based Training Committee). Es una asociación de entrenamiento profesional basado en tecnología. Se reconoce como una de las precursoras de la estandarización de materiales para entrenamiento profesional.
- IMS Global Consortium Inc. Cuenta con miembros de organizaciones comerciales, educativas y gubernamentales dedicadas a definir y distribuir arquitecturas abiertas para actividades de educación en línea. Uno de sus resultados es lo que se conoce como el estándar IMS.
- ADL (Advanced Distributed Learning). La misión de ADL es la de proveer acceso de la más alta calidad en educación y entrenamiento, en cualquier lugar y

²⁴ Estándar de jure. Es un estándar formal, legitimado por organizaciones especializadas.

²⁵ Estándar de facto. Estándar no legitimado. Norma aceptada y utilizada por un gran número de interesados.



momento, para lo cual crean el modelo SCORM (Shareable Content Object Referente Model).

- ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe). Patrocinado por la Unión Europea, este proyecto está enfocado al desarrollo de herramientas y metodologías para producir, administrar y reutilizar elementos pedagógicos basados en computadora, así como la currícula de entrenamiento a distancia.
- IEEE/LTSC (Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee). Es una asociación internacional que dentro de su organización cuenta con el Comité de Estándares para Tecnología del Aprendizaje o LTSC, encargado de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa.
- W3C. (World Wide Web Consortium). Se encarga del desarrollo de tecnologías interoperables (especificaciones, normas, software y herramientas) para aprovechar todo el potencial del web. Aunque no está directamente vinculado con el desarrollo del e-learning es importante mencionarlo, ya que de la interoperabilidad de la web dependen muchas de las funciones de la educación en línea.

De estos grupos de desarrollo, las propuestas más ampliamente adoptadas, en gran número de herramientas, han sido las de IMS Learning Consortium y la de ADL, que han tenido como resultado las especificaciones IMS y el modelo SCORM²⁶, respectivamente.

2.1.4 Ventajas del uso de estándares

Los estándares han sido un elemento indispensable para la masificación de cualquier tecnología, puesto que favorecen el crecimiento, la expansión y la generalización. En el campo del e-learning están en proceso de comprensión y de adopción, y conforme se vayan introduciendo cada vez más, habrá beneficios en interoperabilidad, reutilización, manejabilidad, accesibilidad, durabilidad, escalabilidad y confiabilidad, tanto en contenidos como en infraestructura y funcionalidad.

²⁶ SCORM. Es una especificación que permite crear objetos pedagógicos estructurados



La utilización de estándares amplía las opciones de los usuarios finales y reduce las restricciones de los sistemas propietarios y de soluciones aisladas. Las instituciones, los docentes, los estudiantes y los proveedores se verán beneficiados al contar con contenidos flexibles, plataformas homogéneas y bases de datos compartidas y distribuidas.

2.2. PLATAFORMAS E-LEARNING DE LICENCIA PÚBLICA

Existen diversas plataformas E-Learning de Licencia Pública, como: Atutor²⁷, Moodle²⁸, Claroline²⁹, Dokeos³⁰, MIT doLRN, Ilias, entre otras; a continuación se muestra una gráfica comparativa en el uso de las plataformas de licencia pública, que da a conocer la comunidad eLearning - eLearning WORKSHOPS especializada en la medida del uso de estas plataformas

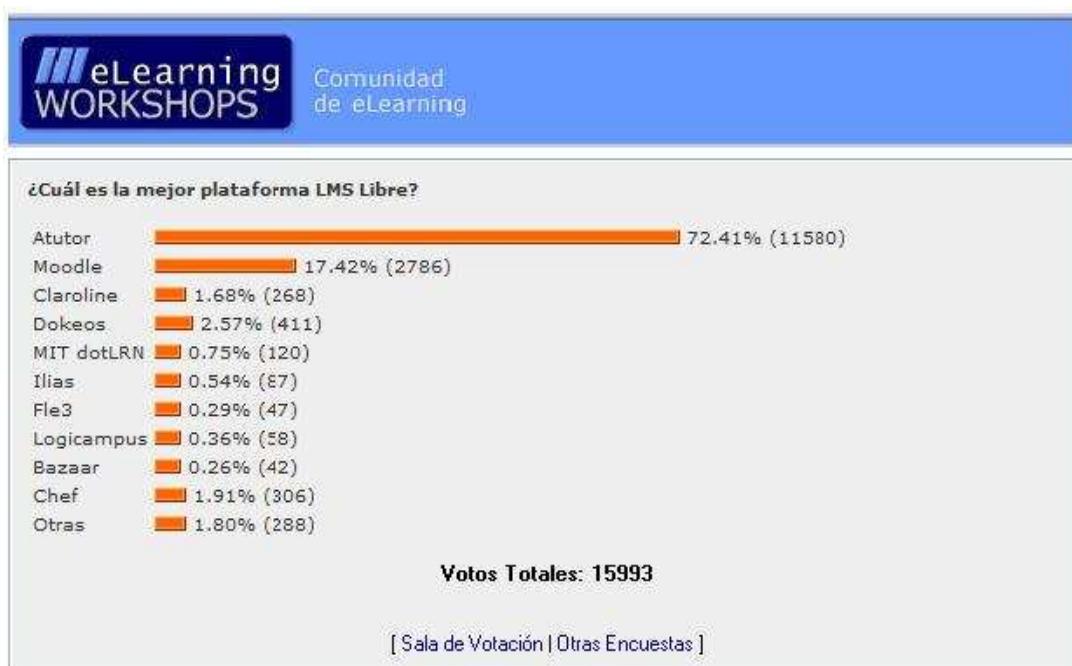


Fig. 2.2.1 Estadísticas de plataformas eLearning

Por la gráfica anterior, en este documento se contempla evaluar las tres primeras.

²⁷ Atutor. <http://www.atutor.ca/>

²⁸ Moodle. <http://moodle.org/login/index.php>

²⁹ Claroline. <http://www.claroline.net/>

³⁰ Dokes. <http://www.dokeos.com/es>



2.2.1 Atutor

El proyecto Atutor inició como un producto de Adaptive Technology Resource Centre (ATRC) de la Toronto University, el proyecto se mantiene en constante desarrollo gracias a los aportes de programadores principalmente de PHP y donaciones de organizaciones externas.

El código fuente de Atutor y se mantiene con Subversion, sistema de control de versiones. La última versión es ATutor 1.6.2 (11 de Diciembre de 2008); trabaja sobre las plataformas Windows, Unix, Linux y Solaris; se encuentra en 32 idiomas entre ellos el español.

ATutor es un Sistema de Gestión de Contenidos de Aprendizaje, Learning Content Management System de Código abierto basado en la Web y diseñado con el objetivo de lograr accesibilidad y adaptabilidad.



Fig. 2.2.2 Entorno de la plataforma Atutor



Especificaciones técnicas

En términos de arquitectura, Atutor es una aplicación web que se ejecuta sin modificaciones solo en Linux, y Windows.

ATutor es un programa diseñado en PHP, Apache, MySQL, contiene herramienta de Gerencia y administra alumnos, tutores, cursos y evaluaciones en línea, herramienta de Autoría incorporada, herramienta de Colaboración incorporada. La incorporación de las especificaciones de empaquetado de contenido IMS/SCORM, permitiendo que los diseñadores de contenidos creen contenido reutilizable que se puede intercambiar entre diversos sistemas de aprendizaje.

Características de Atutor

- Los administradores pueden instalar o actualizar ATutor en poco tiempo.
- Los educadores pueden rápidamente ensamblar, empaquetar y redistribuir contenido educativo, y llevar a cabo sus clases online.
- Los estudiantes pueden aprender en un entorno de aprendizaje adaptativo.

ATutor es el primer LCMS completamente conforme a las especificaciones de la accesibilidad, permitiendo el acceso a todos los estudiantes potenciales, instructores, y administradores, incluyendo a esos con problemas de acceso usando tecnologías asistidas. La conformidad con especificaciones de W3C (World Wide Web Consortium) XHTML 1.0 se asegura de que ATutor esté presentado constantemente en cualquier tecnología compatible con los estándares.



2.2.2 Claroline

Claroline es un groupware³¹ asíncrono y colaborativo, considerado por sus características un sistema de gestión de contenidos (CMS). Proyecto de software libre que se distribuye con licencia GNU/GPL³². Está escrito en el lenguaje de programación PHP, utiliza como SGBD, MySQL. Sigue las especificaciones de SCORM e IMS. Está disponible para plataformas (Linux) y navegadores libres (Mozilla, Netscape), y plataformas (Unix, Mac OS X y Windows) y navegadores propietarios (Internet Explorer).

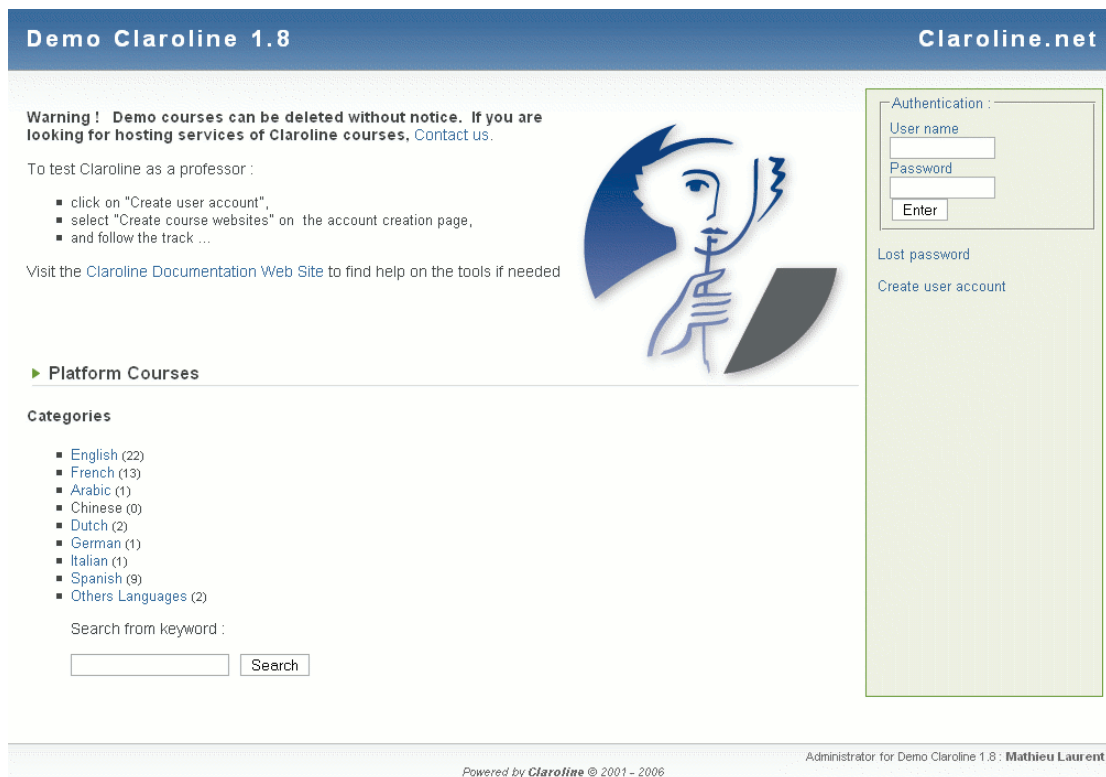


Fig. 2.2.3 Entorno de la plataforma Claroline

³¹ Groupware. Software de aplicación construido con programación que permite a un grupo de personas, desde lugares geográficos distantes, trabajar conjuntamente en una tarea colectiva, para satisfacer necesidades de organización, comunicación, coordinación y flujos de trabajo de un equipo.

³² GNU/GPL. Licencia Pública General de GNU. Licencia creada por la Free Software Foundation orientada principalmente a proteger la libre distribución, modificación y uso de software. Licencia de software libre.



La última versión disponible estable es: Claroline1.8.11 (23 de Julio 2008) y esta disponible en español, permitiendo:

- Publicar documentos en cualquier formato: Word, pdf, html, vídeo, etc.
- Administrar foros de discusión tanto públicos como privados.
- Administrar listas de enlaces.
- Crear grupos de estudiantes.
- Confeccionar ejercicios.
- Estructurar una agenda con tareas y plazos.
- Hacer anuncios, vía correo electrónico por ejemplo.
- Gestionar los envíos de los estudiantes: documentos, tareas, trabajos, etc.
- Editor WYSIWYG³³
 - Con manejo de notación matemática.
- Zona de comunicación asíncrona (foro) y síncrona (chat).
- Posibilidad de almacenar archivos.
- Editor de lecciones.
- Posibilidad de crear pruebas de opción múltiple, relación y llenado de espacio en blanco.
- Agenda de eventos y notificación .
- Escalable.

Sobre las evaluaciones:

- 1) Posibilidad de crear preguntas de desarrollo
- 2) Posibilidad de restringir el acceso a los exámenes online por la IP del computador del cliente.
- 3) Tener preguntas al azar desde un banco de preguntas.
- 4) Preguntas calculadas (un tipo especial de preguntas)

³³ WYSIWYG. What You See Is What You Get (Lo que ves es lo que obtienes). Editor que permite ingresar un documento, y al mismo tiempo visualizar el resultado final de vista o impresión.



5) Posibilidad de integrarse con un Editor de Ecuaciones para la creación de preguntas matemáticas.

En el proyecto Claroline, además de un área de descargas actualizada cada noche, se utiliza el sistema de control de versiones (CVS - Concurrent Versions System), el correo electrónico y foros para hacer sugerencias e informar sobre los errores encontrados.

2.2.3 Moodle

Moodle fue creado por Martin Dougiamas, basó su diseño en las ideas del constructivismo en pedagogía que afirma que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el aprendizaje colaborativo; la plataforma se considera un Sistema de Administración de cursos (LMS).

La primera versión de la herramienta apareció el 20 de agosto de 2002 y, a partir de allí han aparecido nuevas versiones de forma regular. Hasta julio de 2008, última versión 1.9.3+ (11 de julio 2008), es multiplataforma y está traducido en más de 75 idiomas, entre ellos el español.



Fig. 2.2.3 Entorno de la plataforma Moodle



Especificaciones técnicas

En términos de arquitectura, Moodle es una aplicación web que se ejecuta sin modificaciones en Unix, Linux, FreeBSD, Windows, Mac OS X, NetWare y otros sistemas que soportan PHP, incluyendo la mayoría de proveedores de hosting web.

Administración del sitio

Las características de administración que ofrece Moodle son:

- Administración general por un usuario administrador, definido durante la instalación.
- Personalización del sitio utilizando "temas" que redefinen los estilos, los colores del sitio, la tipografía, presentación, etc.
- Pueden añadirse nuevos módulos de actividades a los ya instalados en Moodle.
- Los paquetes de idiomas permiten una localización completa de cualquier idioma. Estos paquetes pueden editarse usando un editor integrado. Actualmente hay paquetes de idiomas para 35 idiomas.
- El código está escrito en PHP bajo GNU GPL.

Administración de los usuarios

Moodle soporta un rango de mecanismos de autenticación a través de módulos, que permiten una integración sencilla con los sistemas existentes.

Las características principales incluyen:

- Método estándar de alta por correo electrónico: los estudiantes pueden crear sus propias cuentas de acceso. La dirección de correo electrónico se verifica mediante confirmación.
- Base de datos externa. Cualquier base de datos que contenga al menos dos campos puede usarse como fuente externa de autenticación.
- Cada persona necesita sólo una cuenta para todo el servidor. Por otra parte, cada cuenta puede tener diferentes tipos de acceso. Con una cuenta de administrador que controla la creación de cursos y determina los profesores, asignando usuarios a los cursos.



- Seguridad: los profesores pueden añadir una "clave de acceso" para sus cursos, con el fin de impedir el acceso de quienes no sean sus estudiantes. Pueden transmitir esta clave personalmente o a través del correo electrónico personal, etc. Los profesores pueden dar de baja a los estudiantes manualmente si lo desean, aunque también existe una forma automática de dar de baja a los estudiantes que permanezcan inactivos durante un determinado período de tiempo (establecido por el administrador).

Módulo Cuestionario

- Los profesores pueden definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios.
- Las preguntas pueden ser almacenadas en categorías de fácil acceso, y estas categorías pueden ser "publicadas" para hacerlas accesibles desde cualquier curso del sitio.
- Los cuestionarios se califican automáticamente, y pueden ser recalificados si se modifican las preguntas.
- Los cuestionarios pueden tener un límite de tiempo a partir del cual no estarán disponibles.
- El profesor puede determinar si los cuestionarios pueden ser resueltos varias veces y si se mostrarán o no las respuestas correctas y los comentarios
- Las preguntas y las respuestas de los cuestionarios pueden ser mezcladas (aleatoriamente) para disminuir las copias entre los alumnos.
- Las preguntas pueden crearse en HTML y con imágenes.
- Las preguntas pueden importarse desde archivos de texto externos.
- Las preguntas pueden tener diferentes métricas y tipos de captura.

Interoperabilidad

Hay muchas dimensiones de interoperabilidad para sistemas de aprendizaje electrónico. Las características de interoperabilidad de Moodle incluyen:



- Autenticación, usando LDAP³⁴, Shibboleth³⁵, o varios métodos estándar.
- Inscripción, usando IMS Enterprise entre otros métodos, o por interacción directa con una base de datos externa.
- Recursos, usando IMS Content Packaging, SCORM, AICC, LAMS.
- Moodle también tiene características de importación para uso con otros sistemas específicos, como la importación de preguntas o cursos enteros.

2.3 EVALUACIÓN DE PLATAFORMAS E-LEARNING DE LICENCIA PÚBLICA

2.3.1 Evaluación nivel administrador.

En la siguiente tabla se muestra servicios que brindan las plataformas en forma general:

Plataforma	Interfaz	Autenticación	Usuarios	Registro
ATutor	Sencilla	Ante la BD	Modificar	Por curso
Claroline	Amigable	Ante la BD	Crear, Modificar	Por curso
Moodle	Sencilla	Múltiple método	Crear, Modificar	Siempre activo

Tabla 2.3.1 Comparación nivel administrador – Plataformas eLearning

Sobre la interfaz: La interfaz de acceso para las tres plataformas es similar, sin embargo en su navegación de cada uno hay diferencias; Atutor, muestra una interfaz sencilla por la cual el usuario puede navegar sin mayor dificultad; en la interfaz de Claroline se puede navegar de manera natural, por esto se considera que es amigable; en cambio la interfaz de Moodle para su navegación es rebuscada para los servicios.

³⁴ LDAP - Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios). Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

³⁵ Shibboleth. Software para inicio de sesión en una aplicación Web.
<http://shibboleth.internet2.edu/shib-which-version.html>.



En la autenticación: Atutor y Claroline, sólo se puede realizar ante la base de datos para cualquier tipo de usuario, en cambio Moodle realiza la autenticación de manera diversa, en uso específico de un servidor, correo electrónico, ante la base de datos, entre otros.

En la administración de usuarios, Claroline y Moodle permiten crear y modificar los tipos de usuario, mientras que Atutor solo permite modificar.

Para el registro de usuarios, Atutor y Claroline solo lo permiten realizar mientras el curso este activo, y en Moodle el usuario se puede registrar en cualquier momento.

Por lo anterior se considera mejor Moodle a nivel administrador.

2.3.2 Evaluación tipos de reactivos.

Las diferentes plataformas permiten, utilizar diferentes modalidades para obtener diversos tipos de reactivos, se consideran los tipos que se utilizan más:

Plataforma	Opción múltiple	Relacionar columnas	Respuesta corta	Palabras ocultas
Moodle	X	X	X	x
Claroline	X	X		
ATutor	x		x	

Tabla 2.3.2 Comparación tipos de reactivos – Plataformas eLearning

Como se muestra en la tabla, Moodle ofrece más diversos tipos de reactivos, sin embargo no son del todo funcionales, por ejemplo, Moodle y Claroline, ofrecen el tipo de pregunta relacionar columnas, en las cuales de un lado se muestra el concepto y se tiene que seleccionar la respuesta de una lista desplegable (select list) y no permite ingresar imagen; con la respuesta corta se deben considerar todas las posibilidades de ingreso de respuesta, y Moodle limita.



2.3.3 Administración de reactivos

En la siguiente tabla se señala, que plataformas nos permiten cambiar el orden de preguntas y respuestas, para que se presenten de forma aleatoria cada vez que se tenga acceso a resolver un examen.

Plataforma	Barajar preguntas	Barajar respuestas
Moodle	X	X
Claroline	X	X
Atutor	X	

Tabla 2.3.3 Comparación administración de reactivos – Plataformas eLearning

2.3.4 Abstracción de datos

Plataforma	Abstracción de datos
Moodle	X
Claroline	
Atutor	

Tabla 2.3.4 Comparación Abstracción de datos – Plataformas eLearning

La plataforma que permite la abstracción de datos es Moodle, la puede devolver en general del curso o por cuestionario, sin embargo no es suficiente, pues no es la esperada y necesaria, para analizar la calidad de reactivos. Por ejemplo, solo devuelve si la respuesta del alumno es buena o mala, y además se requiere la opción que eligió, para evaluar la calidad de los distractores.



2.3.5 Tipos de Cuestionarios

En las tres plataformas se hace una autoevaluación, sin embargo, en cualquiera de ellas, al mezclar varios tipos de preguntas, la evaluación que devuelve del cuestionario no es correcta, no importando la ponderación que se le ponga al reactivo, en algunos casos los toma por cuestionarios separados.

2.3.6 Conclusión.

Después de las comparaciones de diferentes plataformas de administración escolar, se concluye; realizar un nuevo sistema de evaluación, para satisfacer las necesidades de la materia; en nuestro caso se requieren varios tipos de preguntas, con evaluación correcta; con una adecuada abstracción de datos, general, por grupos, por turnos, por profesor, para que sea manejable la información y así procesarla por los diferentes programas de estadísticas, obteniendo los resultados convenientes. Además de considerar que es más barato, que agregar o modificar módulos a cualquiera de las plataformas, por capacitación y tiempo.

CAPÍTULO 3

ANÁLISIS Y PLANTEAMIENTO DEL SISTEMA



3. ANÁLISIS Y PLANTEAMIENTO DEL SISTEMA

3.1 Acta de Proyecto

3.1.1 Objetivo del Proyecto

Implementar una herramienta que sirva para una evaluación automática de los alumnos de la materia de Cibernética y Computación del Colegio de Ciencias y Humanidades Plantel Oriente, para proporcionar al profesor, fluidez en los datos, e identificar el nivel de aprendizaje de cada alumno en el plan de estudios de la materia y así realizar, cambiar o modificar las estrategias de enseñanza.

3.1.2 Breve Descripción del Proyecto

Desarrollar un sistema de evaluación automática para los alumnos, en el que puedan acceder a través de su número de cuenta y contraseña, obteniendo así un examen, el cual contará con distintas maneras de resolver cada segmento, al finalizar las respuestas dadas se cotejarán con las respuestas correctas y así se podrá asignar una calificación al alumno en la base de datos. Con la evaluación del examen el profesor podrá observar el aprendizaje de los alumnos en cada tema y así saber cuáles estrategias de enseñanza continuar, modificar o cambiar.

3.1.3 Alcance

Establecer un sistema de evaluación automática para los alumnos, con el fin de que el profesor integre la herramienta, y así evidenciar los aprendizajes obtenidos, lo cual le ayudara a saber cuáles son las estrategias de enseñanza que deben continuar, modificar o cambiar.



Las herramientas constarían de los siguientes módulos:

- Pantalla de inicio.
- Módulo de autenticación. Se pide el usuario y la contraseña para identificar el rol y los permisos.
- Módulo de administración. Sólo el administrador tiene acceso.
 - Submódulo administración de usuarios. Administrador, profesor, y alumno.
 - Submódulo de importación y exportación. Captura de preguntas y respuestas. Generación de examen.
- Módulo de publicación del examen. Pantalla en donde se podrá visualizar y responder el examen dado.
- Módulo de validación. Se hará la correspondencia de respuesta correcta con la respuesta usuario, para saber el número de reactivos que contestó correctamente.
- Módulo de almacenamiento de datos. Se tomará el número de aciertos obtenidos del usuario y se almacenaran en la base de datos para que quede registrado.
- Módulo de publicación de resultados. Al finalizar el almacenamiento se le indicará al alumno su número de aciertos, así como su calificación.

3.2 Desarrollo

3.2.1 Consideraciones

- Se recomienda generar una serie de lineamientos para establecer cuáles serían los criterios para aplicar un examen.
- Indicar el perfil para el profesor para la aportación de preguntas, y así se evalúe con un examen de buena calidad.



3.2.2 Necesidades que se Plantean Cubrir con la Adquisición del Software

- ✓ Mejor control en su evaluación.
- ✓ Evaluación de estrategias de aprendizajes.
- ✓ Contar con evaluaciones al término del examen.
- ✓ Reducir el número de reuniones, en donde se ponen en consideración reactivos.
- ✓ Disminuir el tiempo en reuniones generales.
- ✓ Rendición de bitácoras.
- ✓ Eliminar captura de respuestas.
- ✓ Análisis de datos.

3.2.3 Consideraciones Normativas

- La herramienta no sustituye ningún procedimiento generado, por alguna otra instancia.
- Se debe tener en cuenta que sólo es una evaluación parcial y no final.
- La página de inicio podrá ser cambiada o modificada según las necesidades de la dependencia.
- La instalación, personalización, mantenimiento, operación y administración de la herramienta será responsabilidad de la dependencia que la ocupe.

3.2.4 Definición de Requerimientos

El sistema que se quiere realizar es para contar con una evaluación automática, por lo que se solicita:

1. En el sistema podrán ingresar: el administrador y usuarios (alumnos y profesores).
2. El administrador, tiene que contar con su número de trabajador, RFC, un nombre de usuario y contraseña, para ingresar al sistema.



3. El profesor debe estar en servicio en la especialidad, contar con su número de trabajador, RFC, nombre de usuario y contraseña, para que el administrador lo de alta en el sistema.
4. El alumno debe contar con No. de cuenta, nombre de usuario y contraseña, además de estar inscrito en la materia para poder ser registrado.
5. El administrador podrá dar de alta, modificar y eliminar, registros de usuarios.
6. El profesor podrá agregar, modificar y eliminar preguntas, para la elaboración del examen.
7. El administrador realizará el examen de acuerdo al programa de estudios de la materia, además hará uso de las preguntas que ingresen los profesores.
8. El profesor podrá consultar el examen propuesto por el administrador y si lo requiere imprimir.
9. El alumno, podrá solucionar el examen y consultar resultados.
10. El administrador realizará una distribución de resultados, generando bitácoras de actividades de los profesores, reporte de resultados de los exámenes y estadísticas de ello.

Es indispensable que en el sistema exista una administración de usuarios, para controlar accesos y modificaciones de cada uno de ellos.

Administrador – Administrar alumnos.

Administrador – Administrar profesores.

Para dar inicio a la elaboración del examen se necesita que se vayan integrando reactivos, al mismo tiempo revisando las preguntas existentes.

Profesor – Administrar preguntas.

Se requiere además que alguien observe y verifique la información que se va generando, para que se vaya construyendo el examen, así como dar el visto bueno de él.

Administrador – Elaborar examen.

Profesor – Consultar examen.

Ahora, con el examen elaborado, es necesario que se publique y resuelva.

Administrador – Publicación de examen.



Alumno – Resolver examen.

Después, de haber contestado el examen, debe ser evaluado, así el alumno podrá consultar los resultados y éstos enviados a los profesores.

Evaluar examen.

Alumno – Consultar resultados.

Administrador – Distribuir resultados.



3.3 Definición de Audiencia (Niveles de Usuario)

La audiencia en el sistema se estima que será fluida de acuerdo al número de grupos que haya inscritos en la materia.

El sistema que evaluará parcialmente la materia, requiere tres niveles de usuarios, los cuales son el administrador de cuentas y resultados, administrador de examen, los profesores y los alumnos.

El administrador de examen, se encargará de la consulta de preguntas, realización de examen y publicación de examen.

El profesor, se encargará de la administración de preguntas y realización del examen como también consulta de resultados.

Los alumnos, ocuparán el sistema para resolver el examen y consultar sus resultados.

Cruce entre Requerimientos funcionales y Audiencia (Niveles de Usuario)

Nivel de acceso	Gestionar el registro a usuarios	Permitir denegar el acceso a usuarios	Distribuir resultados y Generar informes	Consultar preguntas	Realizar Examen	Publicación de Examen	Resolver Examen	Evaluar Examen	Consultar resultados
Acceso Administrador	X	X	X						
Acceso Administrativo				X	X	X			
Acceso Usuario Profesor				X	X		X		X
Acceso Usuario Alumno							X		X

Tabla 3.3.5 Cruce de requerimientos funcionales y audiencia



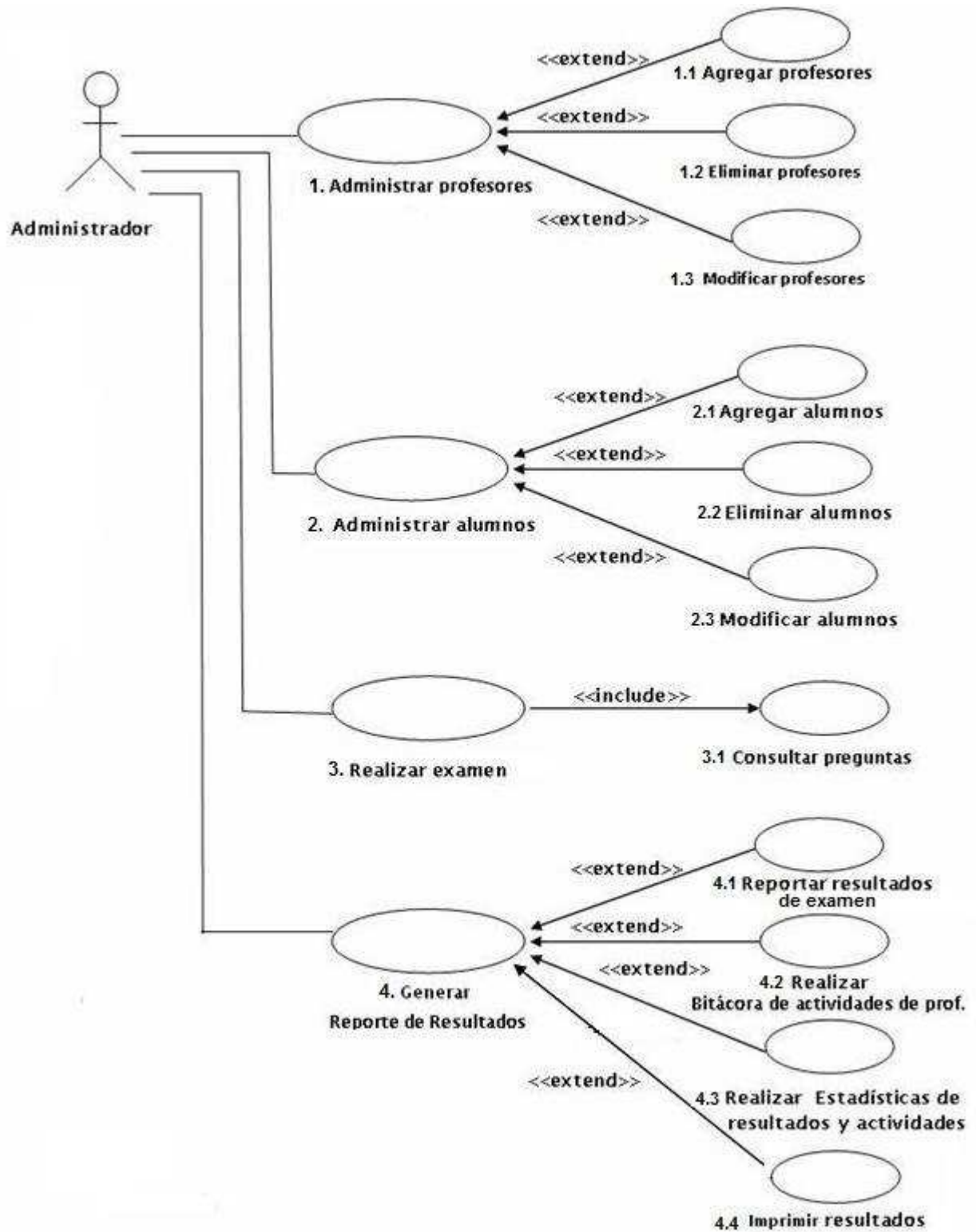
3.4 Seguridad

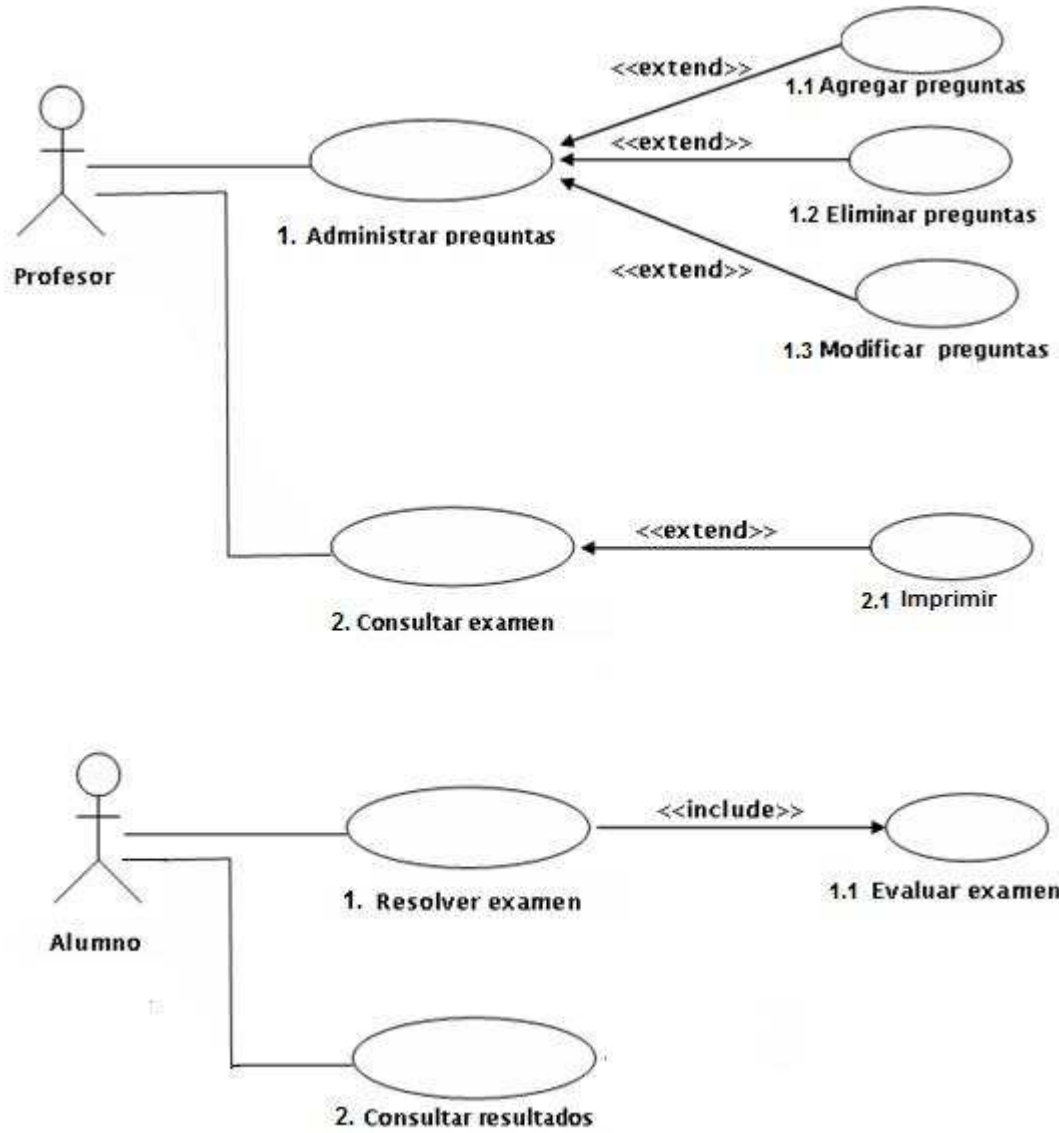
Elemento	Descripción	Necesario	¿Por qué?
Certificado digital	Documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública.	En consideración	Para que se haga más confiable
Humancheck	Código de protección para registros y evitar robots.	No	Hará la gestión de registros únicamente el administrador del sistema apoyándose de los usuarios registrados en la base de datos actualmente
Validación de formularios.	Requerimientos de datos e información necesaria.	Si	Identidad y Seguridad para el sistema

Tabla 3.3.6 Elementos de seguridad



3.5 Diagrama de Casos de Uso





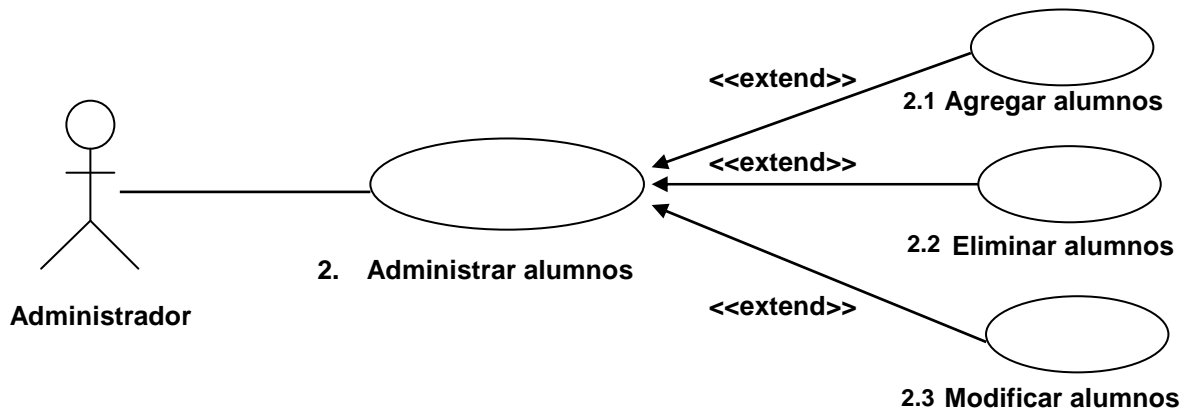


3.6 NARRATIVAS

3.6.1 Administrador – Administrar alumnos

CASOS DE USO DE ALTO NIVEL

CASO DE USO	2. Administrar alumnos
ACTOR	<i>Administrador</i>
TIPO:	PRIMARIO, ESENCIAL
DESCRIPCIÓN	El administrador puede agregar, eliminar o modificar información de los alumnos que lo requieran.

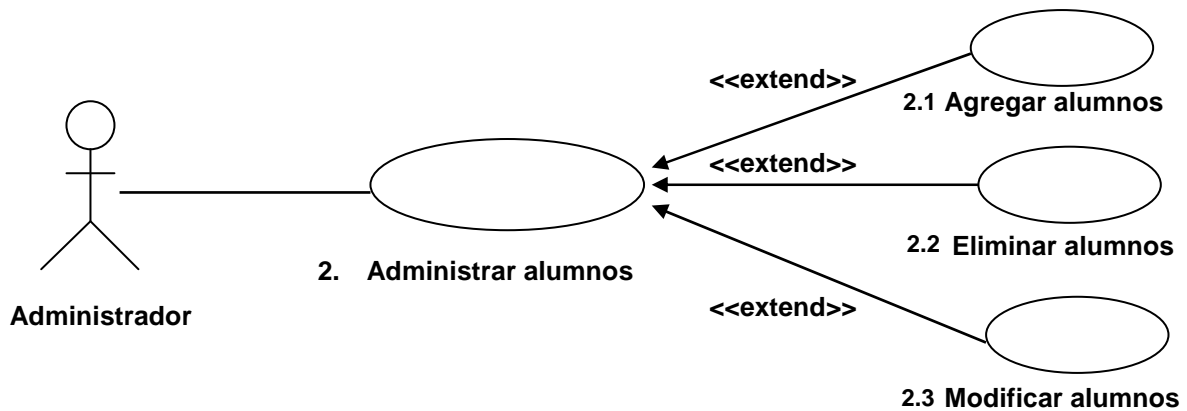




CASOS DE USO DE FORMATO EXPANDIDO

Caso de uso: **2. Administrar alumnos**

Actor: Administrador



Descripción: El administrador puede agregar, eliminar o modificar información de alumnos.

Precondiciones:

- El administrador debe de haber ingresado al sistema.

Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador escoge un control a realizar para un alumno.	2	El sistema solicita los datos del alumno	
3	El administrador captura y envía los datos del alumno.	4	El sistema muestra la información del alumno	E1



ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
5	El administrador acepta la información proporcionada	6	El sistema <i>guarda</i> la información enviada	E2
		7	El sistema informa que la información ha sido guardada satisfactoriamente.	E3

Excepciones:

Id	Nombre	Acción
E1	Datos del alumno no existen.	El sistema informa que el alumno no está registrado
E2	Salir sin guardar cambios	El sistema envía un mensaje para ver si se desean guardar los cambios hechos.
E3	No guardar satisfactoriamente.	El sistema envía un mensaje que la información no se pudo guardar, que se genere el movimiento nuevamente.

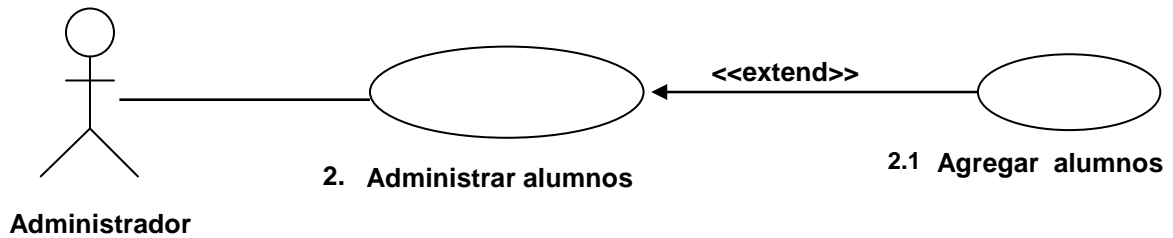
Postcondiciones:

- El sistema ha llevado a cabo un movimiento de un alumno.
- El alumno obtiene el movimiento requerido.



Caso de uso: 2.1 Agregar alumnos

Actor: Administrador



Descripción: El administrador puede realizar el registro de alumnos.

Precondiciones:

- El administrador debe ingresar al sistema.

Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador selecciona el botón agregar alumno.	2	El sistema solicita los datos del alumno.	
3	El administrador captura los datos del alumno.	4	El sistema muestra la información del alumno.	E1
5	El administrador acepta la información proporcionada	6	El sistema guarda la información enviada.	
		7	El sistema informa que la información ha sido guardada satisfactoriamente.	E2



Excepciones:

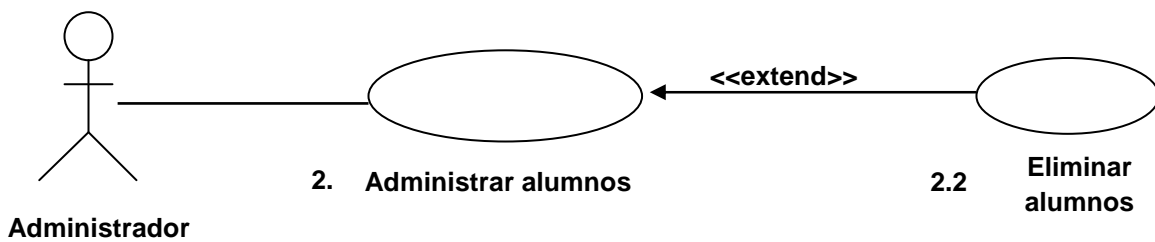
Id	Nombre	Acción
E1	Datos inválidos.	El sistema informa que el alumno no puede ser registrado.
E2	La información no fue guardada satisfactoriamente.	El sistema envía un mensaje que la información no se pudo guardar.

Postcondiciones:

- El sistema ha llevado a cabo un registro de un alumno.
- El alumno obtiene su registro al sistema.

Caso de uso: 2.2 Eliminar alumnos

Actor: Administrador



Descripción: El administrador puede dar de baja a alumnos.

Precondiciones:

- El administrador debe ingresar al sistema.
- El alumno debe estar registrado en el sistema

**Flujo :**

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador selecciona eliminar a un alumno.	2	El sistema solicita los datos del alumno	
3	El administrador captura los datos del alumno.	4	El sistema muestra la información del alumno	E1
5	El administrador confirma la eliminación del alumno.	6	El sistema elimina el registro del alumno.	E2, E3
		7	El sistema informa que el alumno ha sido dado de baja.	E4

Excepciones:

Id	Nombre	Acción
E1	Datos del alumno no existen.	El sistema informa que el alumno no está registrado
E2	Cancelar acción.	El sistema envía un mensaje para ver si se en verdad se desea cancelar.
E3	Si la información es errónea.	El sistema indica que la información no es válida.
E4	No eliminar registro satisfactoriamente.	El sistema envía un mensaje que la información no se pudo eliminar el registro.

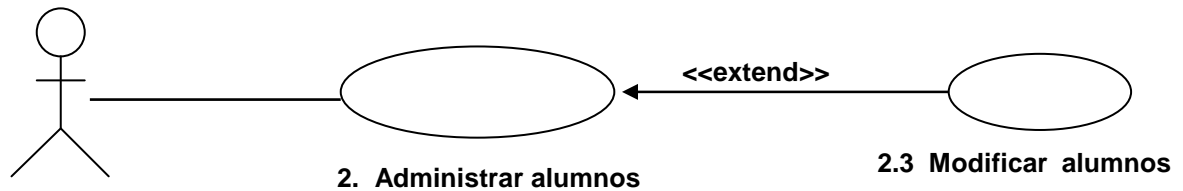
Postcondiciones:

- El sistema ha llevado a cabo una modificación de un alumno.
- El alumno obtiene el cambio requerido.



Caso de uso: 2.3 Modificar alumno

Actor: Administrador



Administrador

Descripción: El administrador puede modificar información de alumnos.

- El administrador debe ingresar al sistema.
- El alumno debe estar registrado en el sistema

Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador escoge la opción modificar alumno.	2	El sistema solicita los datos del alumno.	
3	El administrador captura los datos del alumno.	4	El sistema muestra la información del alumno.	E1
5	El administrador modifica la información requerida.			E2
6	El administrador acepta la información proporcionada.	7	El sistema guarda la información enviada.	E3
		8	El sistema informa que la información ha sido guardada satisfactoriamente.	E4



excepciones:

Id	Nombre	Acción
E1	Datos del alumno no existen.	El sistema informa que el alumno no está registrado.
E2	Salir sin guardar cambios	El sistema envía un mensaje para ver si se desean guardar los cambios hechos.
E3	Si la información es errónea.	El sistema indica que la información no es válida.
E4	No guardar satisfactoriamente.	El sistema envía un mensaje que la información no fue guardada satisfactoriamente.

Postcondiciones:

- El sistema ha llevado a cabo una modificación de un alumno.
- El alumno obtiene el cambio requerido.

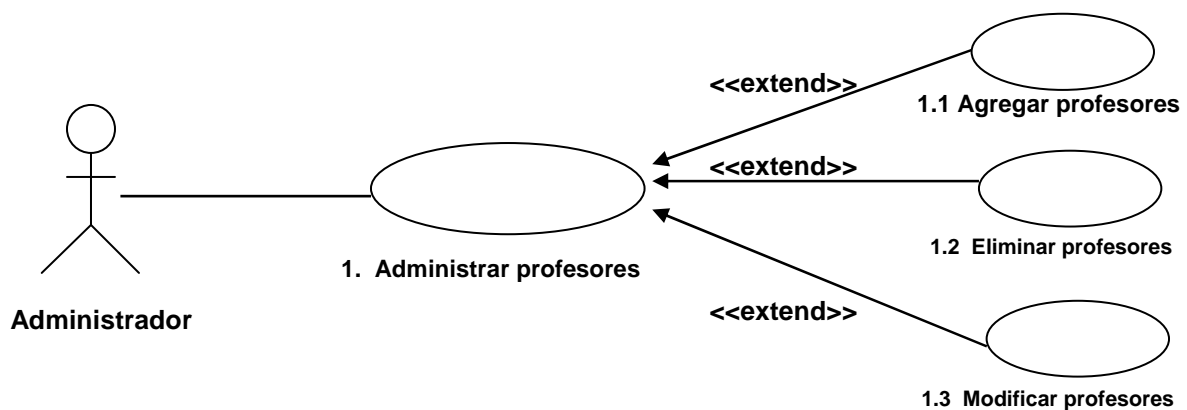


NARRATIVAS

3.6.2 Administrador - Administrar Profesores

CASOS DE USO DE ALTO NIVEL

CASO DE USO	1. Administrar profesores
ACTOR	Administrador
TIPO:	PRIMARIO, ESENCIAL
DESCRIPCIÓN	El administrador puede agregar, eliminar o modificar información de los profesores que lo requieran.

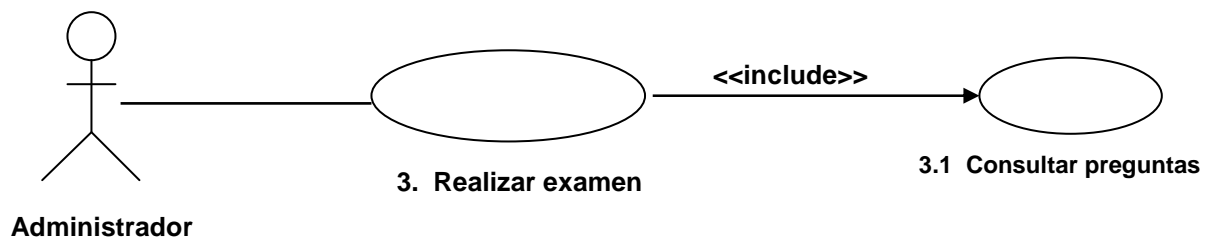




3.6.3 Administrador - Realizar Examen

CASOS DE USO DE ALTO NIVEL

CASO DE USO	Realizar examen.
ACTOR	Administrador
TIPO:	PRIMARIO, ESENCIAL
DESCRIPCIÓN	El administrador puede consultar las preguntas sugeridas por los profesores, para realizar el examen.

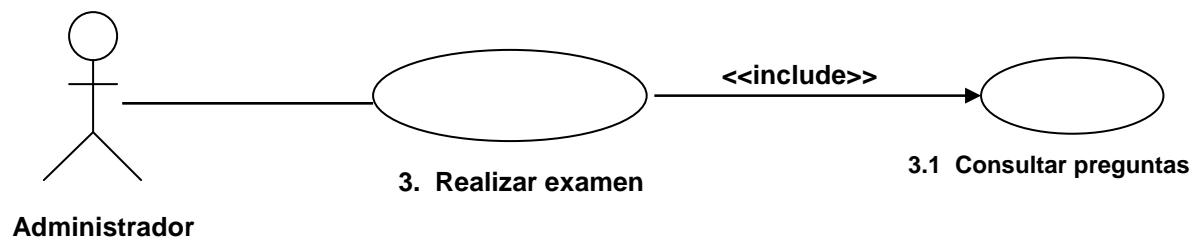




CASOS DE USO DE FORMATO EXPANDIDO

Caso de uso: 3. Realizar Examen

Actor: Administrador



Descripción: El administrador realizará el examen a partir de las preguntas que hayan propuestos los profesores.

Precondiciones:

- El administrador debe de haber ingresado al sistema.

Caso de uso: Consultar preguntas.

Descripción: El administrador consultará las preguntas sugeridas, para la elaboración del examen.

Precondiciones:

- El administrador debe ingresar al sistema.



Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador elige realizar el examen.			
2	El administrador escoge consulta de preguntas.	3	El sistema muestra las preguntas registradas.	E1
4	El administrador revisa y selecciona las preguntas convenientes.			
5	El administrador registra las preguntas seleccionadas.	6	El sistema guarda la información enviada.	
		7	El sistema informa que la información ha sido guardada satisfactoriamente.	E2

Excepciones:

Id	Nombre	Acción
E1	Consulta inválida.	El sistema informa que la consulta es inválida.
E2	La información no fue guardada satisfactoriamente.	El sistema envía un mensaje que la información no se pudo guardar.

Postcondiciones:

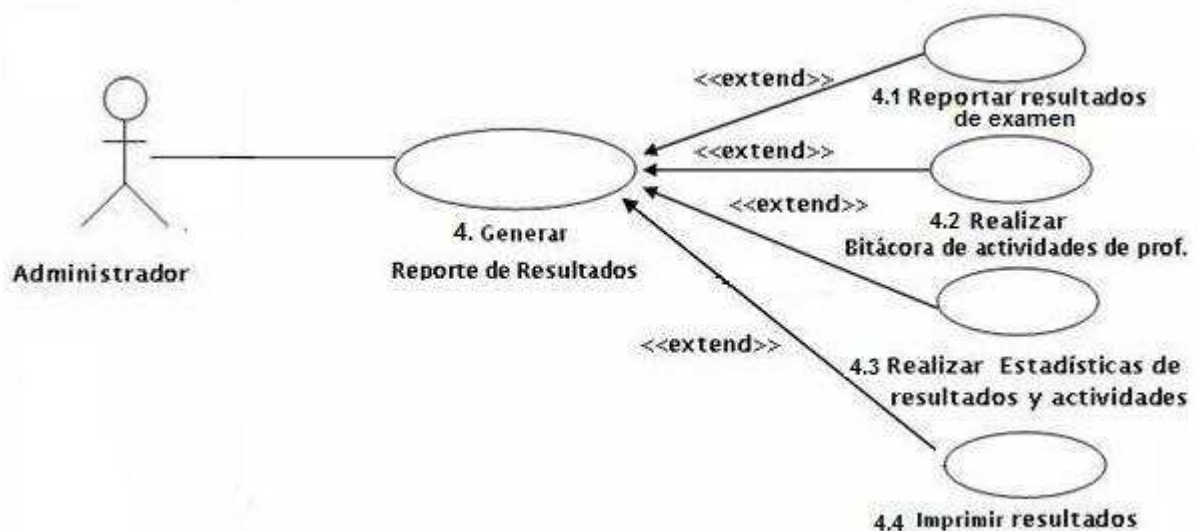
- El sistema ha llevado a generado el archivo examen.
- El administrador ha realizado el examen.



3.6.4 Administrador – Generar Reporte de Resultados

CASOS DE USO DE ALTO NIVEL

CASO DE USO	4. Generar reporte de resultados
ACTOR	<i>Administrador</i>
TIPO:	PRIMARIO, ESENCIAL
DESCRIPCIÓN	El administrador puede generar reportes de resultados de examen, bitácoras de actividades de profesores, estadísticas de resultados y actividades, para realizar la distribución a los usuarios que lo soliciten.

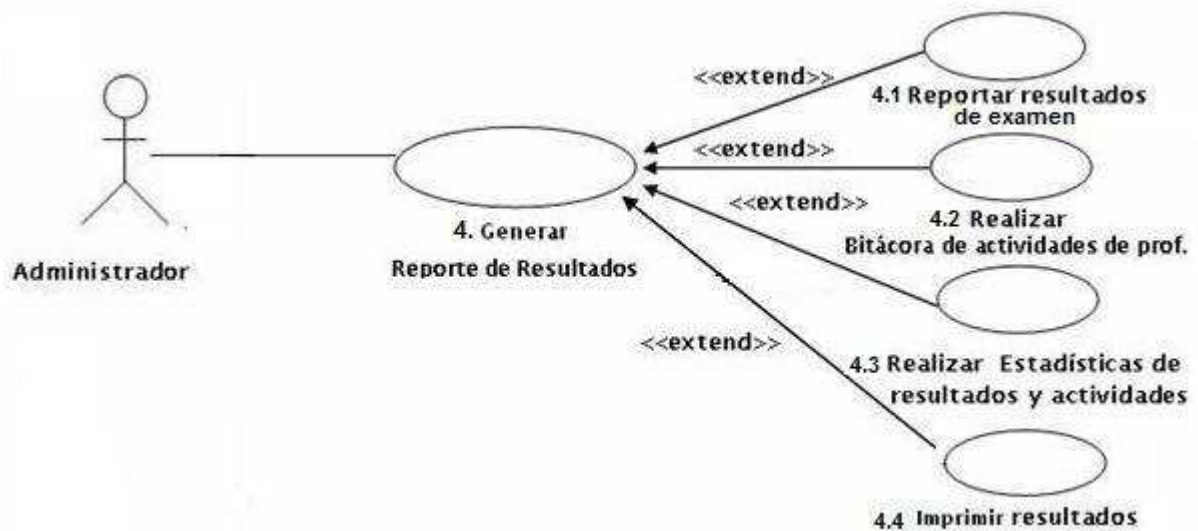




CASOS DE USO DE FORMATO EXPANDIDO

Caso de uso: Distribuir resultados

Actor: Administrador

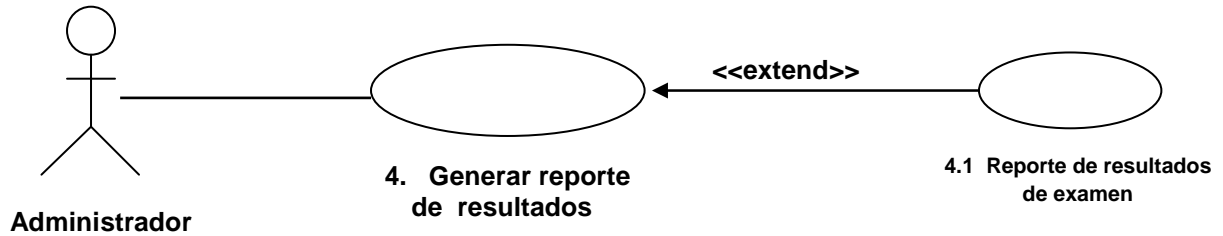


Descripción: El administrador puede generar reportes de resultados de examen, bitácoras de actividades de profesores, estadísticas de resultados y actividades, para realizar la distribución a los usuarios que lo soliciten.



Caso de uso: 4. **Generar reporte de resultados de examen.**

Actor: Administrador



Descripción: El administrador puede generar el reporte de los resultados obtenidos en el examen.

Precondiciones:

- El administrador debe ingresar al sistema.
- Los alumnos deben resolver el examen.
- Los exámenes deben estar evaluados.

Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El administrador solicita la evaluación de los exámenes.	2	El sistema muestra las opciones de las evaluaciones (Grupo, lista, registro).	
3	El administrador selecciona el tipo de evaluación requerida.	4	El sistema muestra la información de las evaluaciones seleccionadas.	
5	El administrador genera y envía el reporte de los resultados.		.	



ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
		6	El sistema guarda el reporte enviado, e informa que ha sido guardado satisfactoriamente.	E1

Excepciones:

Id	Nombre	Acción
E1	El reporte no fue guardado satisfactoriamente.	El sistema envía un mensaje que no se a completo el guardado del reporte.

Postcondiciones:

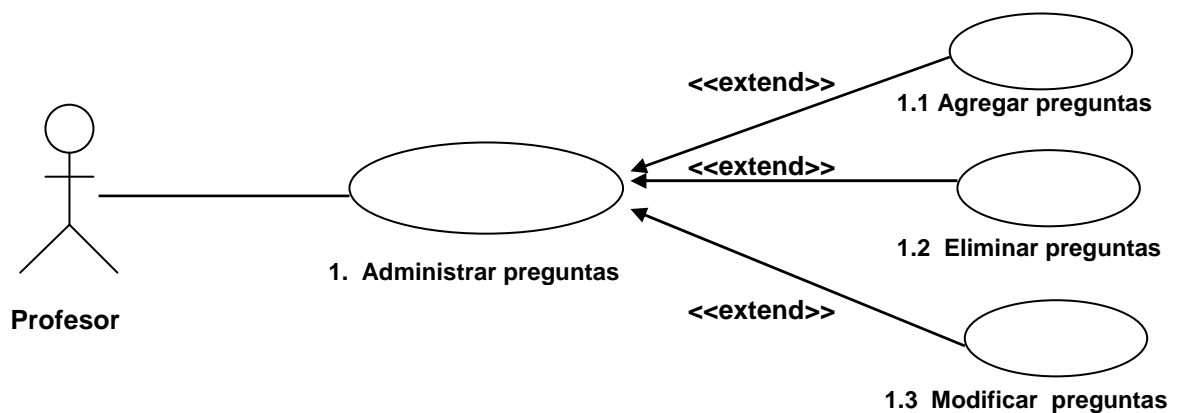
- El sistema contiene el reporte de los resultados del examen.
- El administrador ha realizado el reporte de los resultados del examen.



3.6.5 Profesor – Administrar Preguntas

CASOS DE USO DE ALTO NIVEL

CASO DE USO	1. Administrar preguntas
ACTOR	Profesor
TIPO:	PRIMARIO, ESENCIAL
DESCRIPCIÓN	El profesor puede agregar, eliminar o modificar preguntas del banco de preguntas que se genera

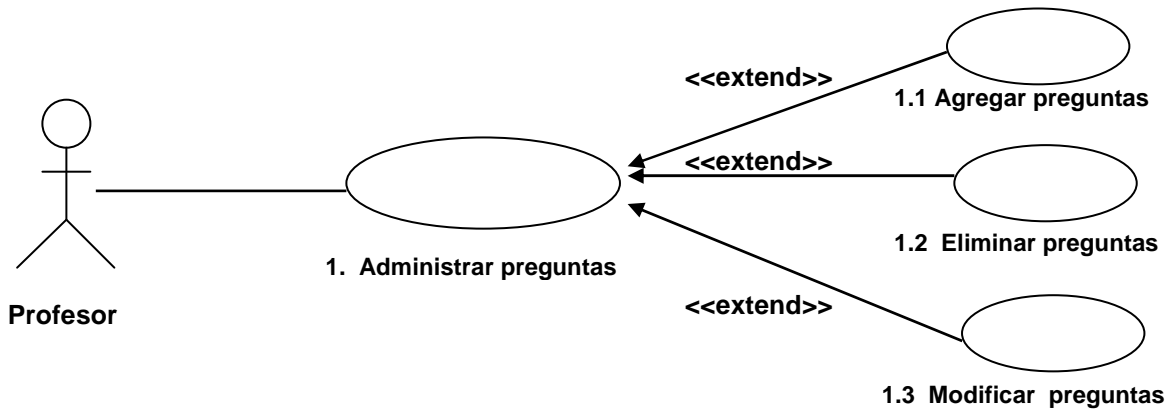




CASOS DE USO DE FORMATO EXPANDIDO

Caso de uso: 1. Administrar preguntas

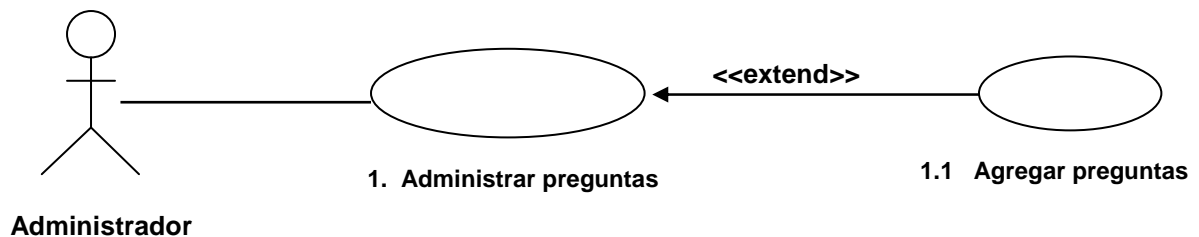
Actor: Profesor



Descripción: El profesor puede agregar, eliminar o modificar preguntas, con las que se elaborará el examen.

Caso de uso: 1.1 Agregar preguntas

Actor: Profesor



Descripción: El profesor puede realizar el registro de preguntas.

Precondiciones:

- El profesor debe ingresar al sistema.



Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El profesor elige agregar preguntas.	2	El sistema muestra la pantalla para la captura de preguntas.	
3	El profesor captura las preguntas que sugiere.			
4	El profesor envía las preguntas capturadas.	5	El sistema guarda la información enviada.	E1
		6	El sistema informa que la información ha sido guardada satisfactoriamente.	
7	El profesor acepta el mensaje.			

Excepciones:

Id	Nombre	Acción
E1	La información no fue guardada satisfactoriamente.	El sistema envía un mensaje que la información no se pudo guardar.

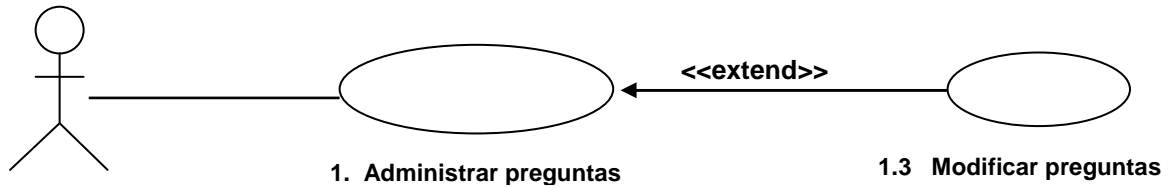
Postcondiciones:

- El profesor ha llevado a cabo el registro de las preguntas que sugiere.
- El sistema aumenta su banco de preguntas.



Caso de uso: 1.3 Modificar preguntas

Actor: Profesor



Profesor

Descripción: El profesor puede modificar las preguntas sugeridas.

Precondiciones:

- El profesor debe ingresar al sistema.
- El profesor tiene que haber hecho un registro de preguntas.

Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El profesor escoge modificar preguntas.	2	El sistema muestra el banco de preguntas	
3	El profesor modifica las preguntas que cree convenientes.			E1
4	El profesor acepta las modificaciones realizadas.	5	El sistema guarda la información enviada.	E2
		6	El sistema informa que la información ha sido guardada satisfactoriamente.	



Excepciones:

Id	Nombre	Acción
E1	Salir sin guardar cambios	El sistema envía un mensaje para ver si se desean guardar los cambios hechos.
E2	No guardar satisfactoriamente.	El sistema envía un mensaje que la información no fue guardada satisfactoriamente.

Postcondiciones:

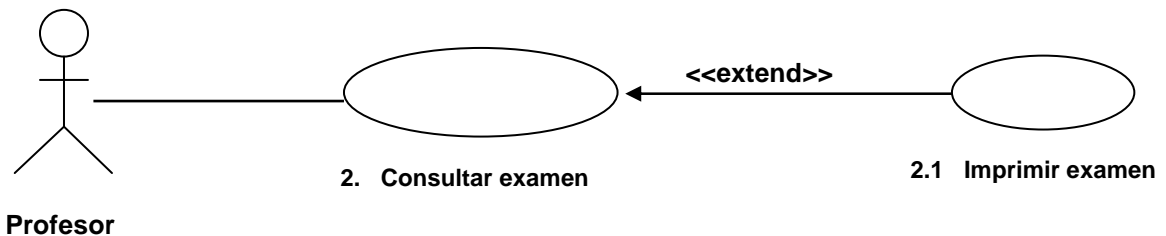
- El profesor ha modificado preguntas.

3.6.6 Profesor – Consultar Examen

CASOS DE USO DE FORMATO EXPANDIDO

Caso de uso: **Consultar Examen**

Actor: Profesor



Descripción: El profesor puede consultar el examen propuesto por el administrador.



Precondiciones:

- El profesor debe ingresar al sistema.
- El administrador debe elaborar el examen.

Caso de uso: **Imprimir examen.**

Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El profesor opta consultar el examen.	2	El sistema muestra la vista del examen.	
3	El profesor elige imprimir el examen.	4	El sistema muestra la ventana de impresión.	
5	El profesor selecciona y acepta las preferencias de impresión.	5	El sistema imprime el examen generado.	E1

Excepciones:

Id	Nombre	Acción
E1	Impresión cancelada.	El sistema envía un aviso indicando que la impresión es cancelada.

Postcondiciones:

- El profesor ha consultado e impreso el examen.



3.6.7 Alumno – Resolver Examen

CASOS DE USO DE ALTO NIVEL

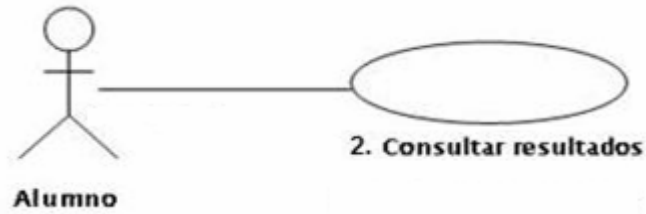
CASO DE USO	1. Resolver examen.
ACTOR	Alumno
TIPO:	PRIMARIO, ESENCIAL
DESCRIPCIÓN	El alumno podrá consultar el resultado de su examen después de haberlo resuelto.





CASOS DE USO DE FORMATO EXPANDIDO

Caso de uso: **2. Consultar Resultados.**



Descripción: El alumno podrá consultar el resultado de su examen después de haberlo resuelto.

Precondiciones:

- El alumno entra al sistema.
- El alumno resuelve el examen.

Flujo :

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	El alumno elige consulta de resultados de examen.	2	El sistema muestra el resultado de examen	
3	El alumno acepta el resultado del examen			

Postcondiciones:

- El alumno obtiene el resultado de su examen.



3.7 Organización del Proyecto

3.7.1 Interfaces Externas

El sistema contará con varias interfaces las cuales nos brindaran las operaciones que podrán realizar los diferentes usuarios.

La principal es la bienvenida e introducción al sistema, en ella se podrá elegir el tipo de usuario con el que se quiere acceder.

De la interfaz anterior se derivan las siguientes, depende el tipo de usuario la solicitud del nombre de usuario y contraseña, se indica a que corresponde cada uno de ellos en los casos respectivos.

Una vez ingresado el tipo de usuario se muestra la interfaz que nos sugieren las acciones de cada uno de ellos.

En la interfaz del administrador, se tendrá un menú con las siguientes acciones: administrar profesores, administrar alumnos, realizar examen, reporte de resultados.

En la interfaz de profesor, se verá como menú: Agregar preguntas, modificar preguntas, eliminar, preguntas, consultar examen y consultar resultados.

En la interfaz del alumno se tienen dos alternativas, primero resolver el examen y después consultar resultados.

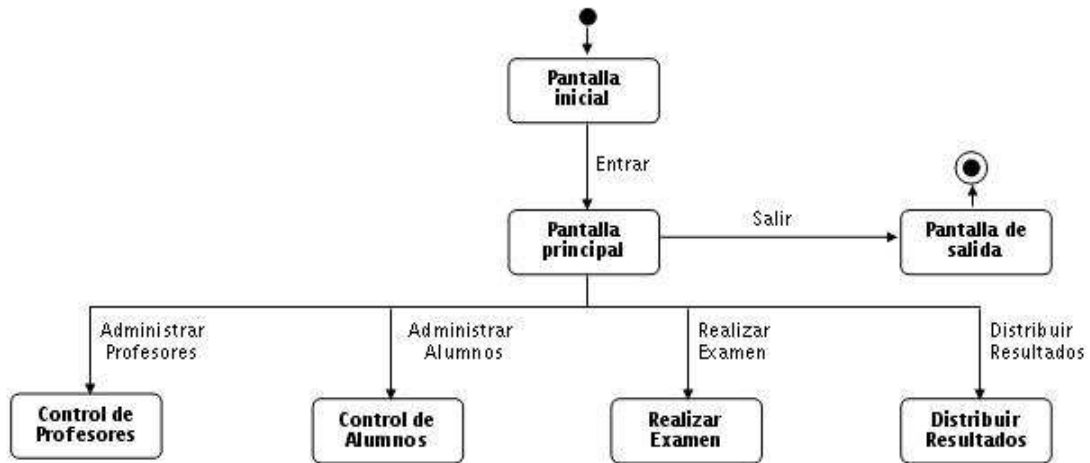
3.8 Mapas del Sitio

3.8.1 Árbol Funcional

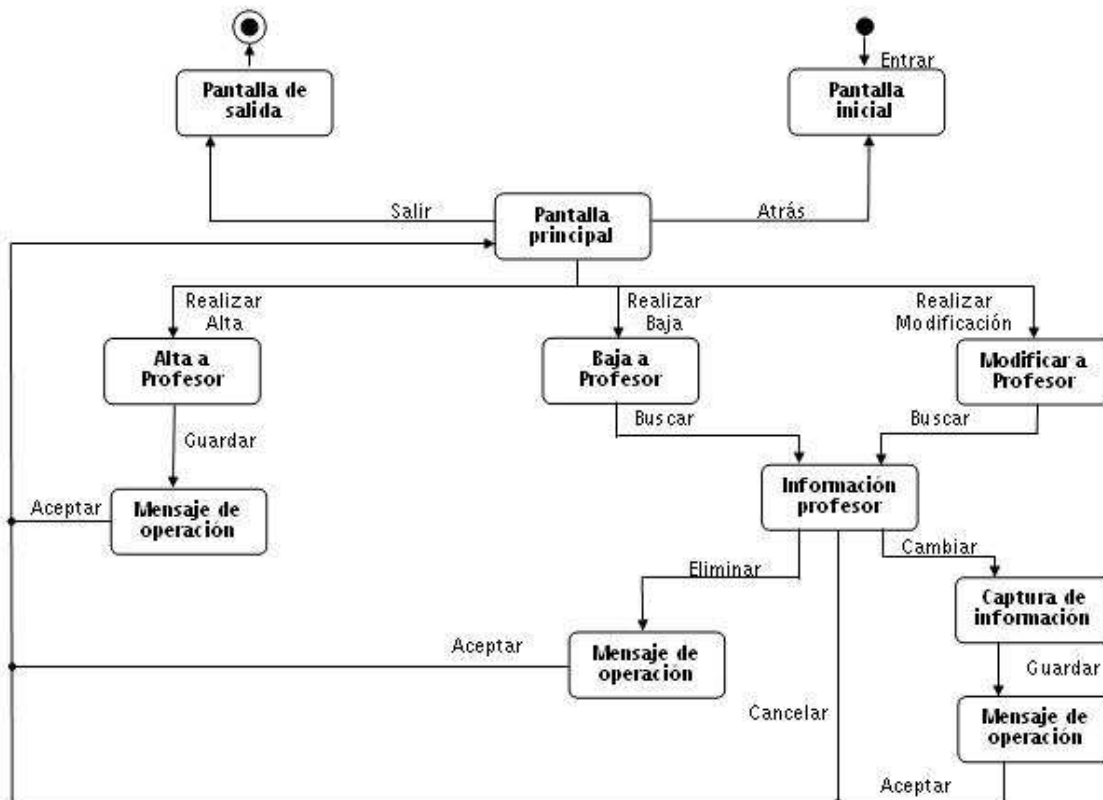
El árbol funcional, agrupa los contenidos de acuerdo a las tareas que se puedan realizar dentro del sitio. A continuación, se muestran los árboles funcionales para el administrador, indicando que cada rectángulo es una interfaz y la palabra de conexión es la liga.



PANTALLAS PARA LA NAVEGACIÓN DEL ADMINISTRADOR

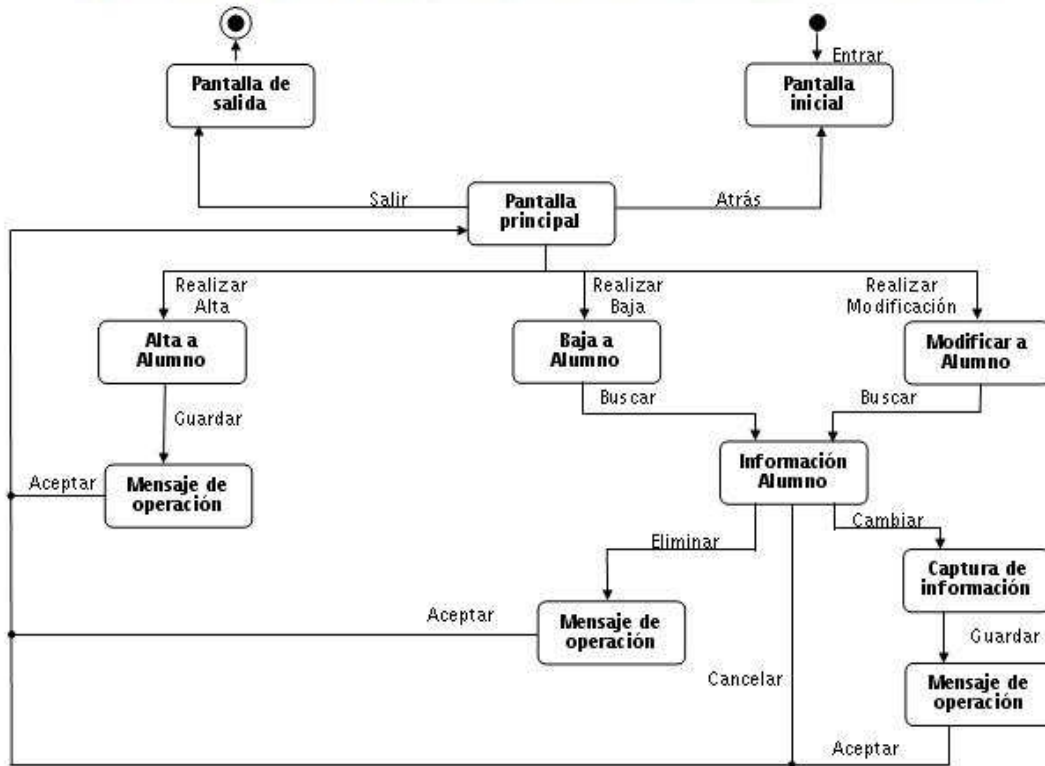


Pantallas para la navegación del Administrador con el Control de Profesores

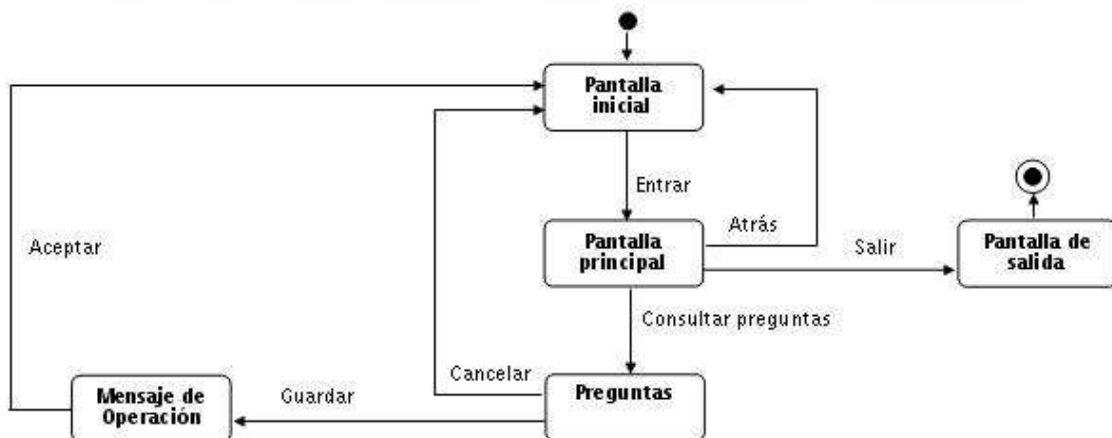




PANTALLAS DE NAVEGACIÓN DEL ADMINISTRADOR PARA EL CONTROL DE ALUMNOS

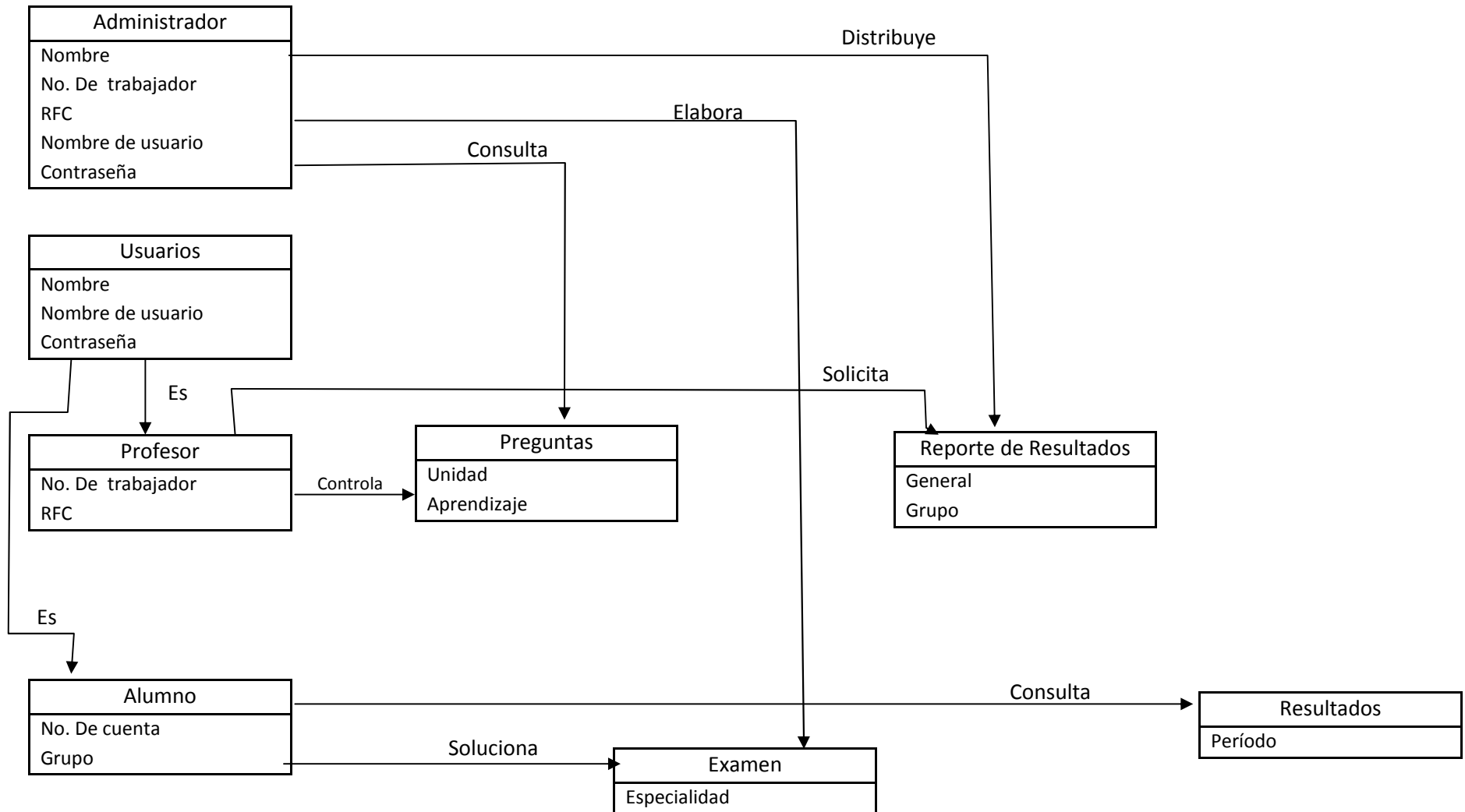


PANTALLAS DE NAVEGACIÓN DEL ADMINISTRADOR PARA REALIZAR EXAMEN





3.10 MODELO CONCEPTUAL



CAPÍTULO 4

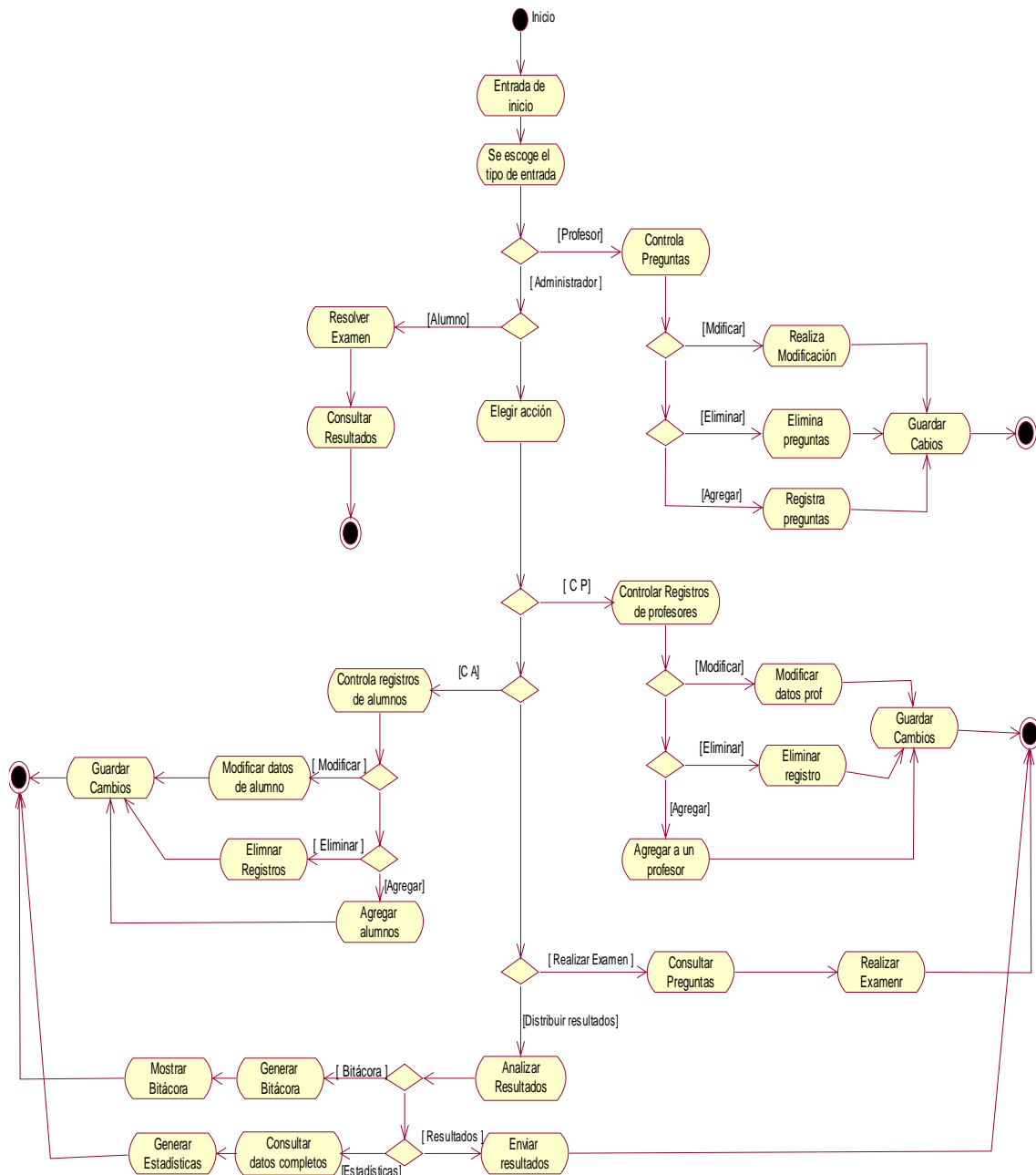
DESARROLLO E IMPLEMENTACIÓN



4. DESARROLLO E IMPLEMENTACIÓN

4.1 Diagrama de Actividades

Se presenta el diagrama de actividades de los usuarios, para conocer los roles en los que podrá interactuar.





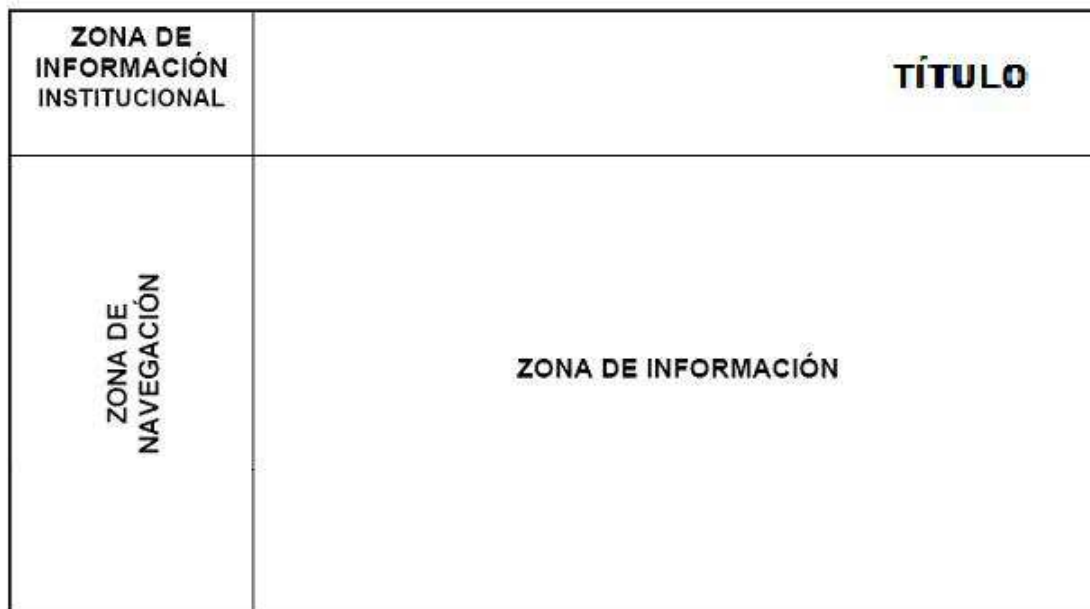
4.2 Interfaces Web

En lo particular, se describirá el funcionamiento de interfaces Web de algunas actividades más significativas para el Sistema de Evaluación; considerando las siguientes: administración de usuarios, acceso de usuarios al sistema, agregar preguntas y resolver examen.

4.2.1 Interfaces de Contenido de Información

Las interfaces deben proporcionar la información necesaria para que el usuario tenga un fácil acceso y haga un buen uso del sistema.

En lo general, el esquema que se utilizó es el siguiente:



Derechos Reservados

Para obtener el esquema anterior, se utilizó una hoja de estilo, generándose el siguiente código:

```
/* CSS Document */  
body{  
height:auto;
```



```
width:800px;
}
.menu_iz
{
text-color:#ff0990;
background-repeat:no repeat;
height:20px;
width:192px;
vertical-align:middle;
padding-bottom:20px;
}
a{
text-decoration:none;
font-size:middle;
color:white;
}

div{

width:100%;
text-align:left;
}

td.menu{
vertical-align:center;
font-size:middle;

color:white;
}

h3{
padding-top:10px;
padding-left:10px;
}
img{

position:relative;
left:0px;
}
span{
border:solid;
}
th{
text-align:center;
}
}
td.no{
text-align:center;
}
td.dos{
padding-left:10px;
}
tr.uno{
```



```
background-color:#ffffff;  
font-size:big;  
style:italic;  
color:black;  
family:Lucida Sans;  
}  
tr.dos{  
background-color:#FFCC33;  
}  
p.pie{  
text-align:center;  
font-size:small;  
style:italic;  
color:blue;  
}
```

Cabe mencionar, que en las diferentes actividades que se describirán por tipo de usuario, se realizaron ligeros cambios, como: combinar columnas en la zona de navegación e información, cambiar fondo de encabezado para distinguir en que sesión se esta trabajando y la actividad a realizar.

4.3 Inicio

Una de las páginas encargadas de proporcionarnos información es la página de inicio, la cual brinda una breve introducción del sistema.



Para crear la página anterior se utiliza el código siguiente:



```
<html>
<head>
<title>

</title>
<link href="estilo.css" type="text/css" rel="stylesheet" >
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>

<div id="encabezado">
  <table width="100%" >
    <tr>
      <td></td>
    </tr>
  </table>
</div>

<div id="Cuerpo" >
  <table width="100%">
    <tr >
      <td class="menu" width="20%" > <table>
        <tr ><td class="menu_iz" background="fondoC.jpg"><br><br>
          <a href="administrador.html" >Administrador</a><br><br>
          <a href="profesor(2).html" >Profesores</a><br><br>
          <a href="alumno.html" >Alumnos</a><br><br>
          <a href="salir.html" >Salir</a><br></td></tr>
        </table>
      </td>
      <td style="(padding-bottom:2%) ">
        <h1>Bienvenida</h1>
        <table align="center" width="90%" border="1" >

          <tr class="unc">
            <td class="no">Sistema de Evaluación parcial para la materia de Cibernética y Computación.<br>
              Elige el tipo de usuario <br>
              En este ejemplo, elige profesores</td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
</div>

<div id="Pie">
  <p class="pie"><br><br>
    Derechos Reservados<br>
    Cielo 2009 - 1</p>
</div>

</body>
</html>
```



Sin embargo, cada una de las páginas debe estar bien descrita para un mejor uso y funcionalidad del sistema.

4.4 Acceso a Usuario

Pulsando, el usuario deseado de la interfaz de inicio, aparece la pantalla que sigue, nótese el cambio en el encabezado según el tipo de usuario.



En esta pantalla se ingresan los datos nombre de usuario y contraseña. Ingresados y validados los datos del usuario, también cambia la zona de navegación según las actividades de cada usuario.

Para llegar al funcionamiento de esta pantalla, el administrador debe encargarse de gestionar los datos de los usuarios; para esto, se realiza la conexión a la base de datos usando el código que sigue:



//Conexión a la base de datos

```
//Conexión a la base de datos
package data;
import java.sql.*;

public class Conexion {

    public Conexion() {
    }

    public static Connection getConnection(String usuario, String passwd)
    Connection cn = null;
    try{
        Class.forName("com.mysql.jdbc.Driver");

        cn=DriverManager.getConnection("jdbc:mysql://192.249.221.731:3306/usuario", usuario, passwd);
    }catch(ClassNotFoundException e ){
        System.out.println("No se encuentra el driver de la Base de Datos");
    }catch(SQLException e){
        System.out.println("Error en la conexión"+e.toString());
    }

    return cn;
}

public static Connection getConnection()
    Connection cn = null;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        cn=DriverManager.getConnection("jdbc:mysql://192.249.221.731:3306/usuario", "root", "ciber");
    }catch(ClassNotFoundException e ){
        System.out.println("No se encuentra el driver de la Base de Datos");
    }catch(SQLException e){
        System.out.println("Error en la conexión"+e.toString());
    }

    return cn;
}
```



```
public static void cerrarDBObject(Object obj) {  
    try {  
        if(obj instanceof Connection) {  
            ((Connection)obj).close();  
        }  
        if(obj instanceof Statement) {  
            ((Statement)obj).close();  
        }  
        if(obj instanceof ResultSet) {  
            ((ResultSet)obj).close();  
        }  
        if(obj instanceof PreparedStatement) {  
            ((PreparedStatement)obj).close();  
        }  
        if(obj instanceof CallableStatement) {  
            ((CallableStatement)obj).close();  
        }  
    }  
}  
  
} catch(SQLException e) {  
    System.out.println("Error !" + e.getMessage());  
}  
}  
  
public static void main(String args[]) {  
    Connection con=null;  
    try {  
        con = Conexion.getConnection("root", "alber");  
        System.out.println("con: "+con.getConnection());  
        Conexion.cerrarDBObject(con);  
    } catch(Exception e) {  
    }  
}
```

Con la conexión de la base de datos se pueden agregar, modificar o eliminar a usuarios, para estas actividades solo el administrador tiene privilegios. El código que se utiliza para la gestión de usuarios varía según el tipo de usuario.



4.5 Agregar Usuario

Para el módulo de agregar usuario se utiliza el siguiente código, considerando que aquí se agrega el tipo de usuario alumno, sin embargo, es de manera similar, para otro tipo de usuario.

```
//Agregar datos
package prgEvaluacion;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.*;
import javax.servlet.http.*;

public class datosAgregar extends HttpServlet {
    private static final String CONTEXT_TYPE = "text/html; charset=windows-1251";

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        response.setContentType(CONTEXT_TYPE);
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Agregar</title></head>");
        out.println("<body>");
        out.println("<form action='<url>agregar</url>'><table border='1'><tr>");
        out.println("<td>Número de Cuenta</td><td><input type='text' name='txtNumCuenta'</td></tr>");
        out.println("<tr><td>Grupo</td><td><input type='text' name='txtGrupo'</td></tr>");
        out.println("<tr><td>Nombre de usuario</td><td><input type='text' name='txtUsuario'</td></tr>");
        out.println("<tr><td>Contraseña</td><td><input type='text' name='txtPasswd'</td></tr>");
        out.println("<tr><td><input type='submit' value='Agregar'</td><td><input type='button' value='Cancelar' *");
        out.println("<td colspan='2'><a href='<url>inicio.html'></td></tr>");
        out.println("</table></form>");
        out.println("</body></html>");
        out.close();
    }
}
```




Para visualizar el alumno agregado, se hace uso del siguiente código:

//Ver agregado – Se omiten librerías

```
public class VistaAgregar extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    Connection objCn;
    String strUsuario;
    String strPassword;
    String strHost;
    String strBDN;
    String strUrl;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        try {
            //Inicializa variables
            strUsuario="usuario";
            strPassword="123";
            strHost="192.248.221.173";
            strBDN="alumno";

            objCn=object.Connection(strUsuario ,strPassword);
        }

    }

    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();

        String strNombre=request.getParameter("txtNombre");
        String strNumCuenta=request.getParameter("txtNumCuenta");

        String strGrupo=request.getParameter("txtGrupo");
        String strUsuario=request.getParameter("txtUsuario");
        String strPassword=request.getParameter("txtPassword");

        String strQuery = new String();
        strQuery = "insert into Evaluacion values(?,?)";

        try {
            //Creamos un objeto Statement
            PreparedStatement objPst = this.objCn.prepareStatement(strQuery);
            objPst.setString(1,strNombre);
            objPst.setString(2,strNumCta);

            objPst.executeUpdate();

        } catch (Exception e) {
            e.printStackTrace();
        }

        out.println("<html>");
        out.println("<head><title>Almacenar/</title></head>");
        out.println("<body>");

        out.println("<h2>Linea datos agregados son:</h2><br>");
        out.println("<table><tr><td>Nombre:</td><td>"+strNombre+"</td></tr>"+
            "<tr><td>Numero de Cuenta:</td><td>"+strNumCta+"</td></tr>"+
            "<tr><td><input type=button value=Aceptar onclick=javascript:location.href='inicio.html'</td>"+
            "<td><input type=button value=Inicio onclick=javascript:location.href='detalleagregar'</td>"+
            "<td><input type=button value=Cancelar onclick=window.close()</td></tr>"+
            "</table>");
        out.println("</body></html>");
        out.close();
    }
}
```



El código anterior permite agregar un alumno, con el cual se introducen los datos; el sistema devuelve la información para corroborar si es correcta, si es así se oprime en aceptar; sino, se oprime en cancelar o inicio, la pantalla se muestra a continuación:

Datos:

Nombre:	Martinez Pablo Jhonatan
No. Cta.:	307278702
Grupo:	501

Aceptar Cancelar Inicio

Derechos Reservados
Ciclo 2009 - 1

4.6 Consulta de Usuario

Una vez, teniendo la información de los usuarios en la base de datos se puede realizar una consulta de ellos, para corroborarlos o actualizarlos, para lo anterior se utiliza el siguiente código:



//Datos alumno – Consulta

```
package pag07alumno01;

import java.io.IOException;
import java.io.PrintWriter;

public class DatosAlumno extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";

    Connection objCn;
    String strUsuario;
    String strPasswd;
    String strHost;
    String strSQL;
    String strSQL;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);

        try {
            //Inicializar variables
            strUsuario="usuario";
            strPasswd="123";
            strHost="192.168.121.173";
            strSQL="alumno";

            objCn=objet.Coneccion(strUsuario ,strPasswd);
        }

    }

    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        String strNumero;
        String strNumCta;
        strNombre = request.getParameter("CboNombre");

        StringTokenizer tokens=new StringTokenizer(strNombre,"_");
        String strQuery = new String();

        String Nombre;
        Nombre = tokens.nextToken();
        strNumCta = tokens.nextToken();

        java.sql.PreparedStatement stmt;
```



```
strQuery = "DELETE FROM Evaluation where Nombre = ?" ;
try {
    stmt = this.cobjCn.prepareStatement(strQuery);
    stmt.setString(1,Nombref);

    stmt.execute();
} catch (Exception ex) {
    ex.printStackTrace();
}
out.println("<html>");
out.println("<head<title>DatosAlumno</title></head>");
out.println("<body>");
out.println("<div>El Alumno es: </div><br>");
out.println("<table>");
out.println("<tr>");
out.println("<td>");
out.println("Nombre");
out.println("</td>");
out.println("<td>");
out.println("NumCta");
out.println("</td>");

out.println("</tr>");
out.println("<tr>");
out.println("<td>");
out.println(Nombref);
out.println("</td>");
out.println("<td>");
out.println(strNumCta);
out.println("</td>");
out.println("</tr>");
out.println("</table>");
out.println("<input type='button' value='Aceptar' * *
        *onclick=javascript:location.href='inicio.html' *");
//se envoca al servicio
out.println("<input type='button' value='Nueva Consulta' * *
        *onclick=javascript:location.href='listadoalumnos' *");
out.println("</body></html>");
out.close();
}
```



Para actualizar datos se utiliza el código siguiente:

```
public class DatosActualizar extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";

    Connection objCn;
    String strUsuario;
    String strPasswd;
    String strHost;
    String strBDN;
    String strUrl;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);

        try{
            this.strUsuario="usuario";
            this.strPasswd="123";
            this.strHost="132.245.221.173";
            this.strBDN="alumnos";
            this.strUrl="jdbc:oracle://"+strHost+"/"+this.strBDN;

            objCn=object.Conexion(strUsuario ,strPasswd);
        }

    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head<title>DatosActualizar</title></head>");
        out.println("<body>");

        out.print("<form action='vistaactualizacion/' method='get/'>");
        out.println("<p>Datos del Alumno:</p>");
        out.println("<p>Datos del Alumno:</p>");

        HttpSession session = request.getSession(true); //Iniciando session.
        session.setAttribute("Actualizar", request.getParameter("txtNombre"));

        PreparedStatement st;
        ResultSet crr;

        try {
            st = this.objCn.prepareStatement("select nombre, NumCta from Evaluacion where Nombre = ?");
            st.setString(1, (String)session.getAttribute("Actualizar"));
            crr = st.executeQuery();

            while(crr.next()){
                out.print("Nombre: <input type='text' name='txtNombre' value='"+crr.getString("nombre")+"'><br><br>");
                out.print("NumCta: <input type='text' name='txtNumCta' value='"+crr.getString("NumCta")+"'><br><br>");
            }

        }catch(Exception e){
            e.printStackTrace();
        }

        out.print("<input type='submit' value='Aceptar'>");
        out.print("<input type='button' value='Cancelar' onclick='\"location.href='\"+\"inicio.html\"+'>");
        out.print("<input type='button' value='Salir' onclick='\"window.close()\"+'>");
        out.println("</form>");
        out.println("</body></html>");
        out.close();
    }
}
```



Se actualiza la lista, con el siguiente código:

```
public class ListadoActualizar extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";

    Connection objCn;
    String strUsuario;
    String strPassword;
    String strHost;
    String strBDN;
    String strUrl;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);

        try {
            this.strUsuario="usuario";
            this.strPassword="123";
            this.strHost="132.248.221.173";
            this.strBDN="alumno";
            this.strUrl="jdbc:mysql://"+strHost+"/"+this.strBDN;

            objCn=object.Connection(strUsuario ,strPassword);
        }

        public void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {response.setContentType(CONTENT_TYPE);
            PrintWriter out = response.getWriter();

            Statement st;
            ResultSet crs;

            String qry = new String();
            qry = "SELECT NOMBRE FROM EVALUACION";

            out.println("<html>");
            out.println("<head><title>ListadoActualizar</title></head>");
            out.println("<body>");
            out.println("<form action='<code>datosActualizar</code>' method='<code>get</code>'>");
            out.println("<p>Listado de Alumnos:</p>");

            try {
                st = this.objCn.createStatement();
                crs = st.executeQuery(qry);

                out.print("<select name='cdoNombre'>");
                while(crs.next()){
                    out.print("<option value='<code>"+crs.getString("nombre")+"</code>'><code>"+crs.getString(1)+"</code></option>");
                }
                out.print("</select>");

            }catch(Exception e){
                e.printStackTrace();
            }

            out.print("<br><br>");
            out.print("<input type='submit' value='Aceptar'>");
            out.print("<input type='button' value='Cancelar' onclick='<code>location.href='\""+this.getServletContext().getRealPath(\"/index.html\")+"\"</code>'>");
            out.print("<input type='button' value='Salir' onclick='<code>window.close()</code>'>");
            out.print("</form>");
            out.println("</body></html>");
            out.close();
        }
    }
}
```



Se pueden mostrar los datos actualizados, utilizando el siguiente código:

```
public class VistaActualizacion extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";

    Connection objCn;
    String strUsuario;
    String strPassword;
    String strHost;
    String strBDN;
    String strUrl;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);

        try {
            this.strUsuario="usuario";
            this.strPassword="123";
            this.strHost="192.249.221.178";
            this.strBDN="alumno";
            this.strUrl="jdbc:mysql://"+strHost+"/"+this.strBDN;

            objCn=object.Connection(strUsuario , strPassword);
        }

    }

    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>VistaActualizacion</title></head>");
        out.println("<body>");

        out.println("<p>Los datos Actualizados Son:</p>");
        HttpSession session = request.getSession(true); //Iniciamos session.
        PreparedStatement st;

        try {
            st = this.objCn.prepareStatement("UPDATE Evaluacion SET nombre=?, NumCta=? where Nombre=?");
            st.setString(1, (String) request.getParameter("txtNombre"));
            st.setString(2, (String) request.getParameter("txtNumCta"));
            st.setString(3, (String) session.getAttribute("Actualizar"));

            out.print("Nombre: "+request.getParameter("txtNombre")+"<br><br>");
            out.print("NumCta: "+request.getParameter("txtNumCta")+"<br><br>");

            st.executeUpdate();

        } catch (Exception e) {
            e.printStackTrace();
        }

        out.print("<input type=button value=Aceptar onclick='\""+location.href+"menu.html'\">");
        out.print("<input type=button value='\""+location.href+"\">");
        out.print("<input type=button value=Salir onclick='\""+location.href+"\">");
        out.println("</body></html>");
        out.close();
    }
}
```




Una vez validado, el nombre de usuario y contraseña se le brinda el menú para que elija la acción a seguir.

4.7 Elección de tipo de pregunta

Se presenta la pantalla de bienvenida al ingreso de un profesor, se reitera, que el acceso de usuarios es similar, sólo se cambian los parámetros de los datos de entrada.





Para continuar un caso significativo, el profesor elige Agregar preguntas, y aparece la siguiente pantalla, donde se sugiere el tipo de reactivo.



Derechos Reservados
Ciclo 2009 - 1

En este caso, se elige la opción relacionar columnas, porque se considera una de las diferencias de las plataformas e-Learning, ya que se puede hacer la relación entre imágenes.

La pantalla que sigue, aparece al seleccionar la opción de Relacionar Columnas, que sirve para capturar la información y generar el reactivo.



Agregar preguntas

Pregunta	12. Relaciona las columnas de forma correcta. Arrastra el elemento de la derecha, junto al que corresponde de la izquierda.	
	Elementos ordenados	Elementos desordenados
1	Símbolo de la compuerta NAND	<code></code>
2	Símbolo de la compuerta NOT	<code></code>
3	Símbolo de la compuerta NOR	<code></code>
4	Símbolo de la compuerta AND	<code></code>
5	Símbolo de la compuerta OR	<code></code>

Derechos Reservados
Ciclo 2009 - 1

En la parte superior, se escribe el texto de la pregunta; en la columna de la izquierda se colocan los elementos ordenados, estos quedan fijos y en la columna de la derecha se coloca su correspondiente, estos elementos son los que varían en la presentación de la pregunta. Al terminar de capturar la información de la pregunta se oprime el botón guardar para que se almacene en el banco de reactivos.



4.8 Vista de Examen

El código que hace actuar una parte del resultado (examen), es el que se muestra, éste sigue formando parte del funcionamiento del tipo de respuesta de relacionar columnas con imágenes.

Funciones que se utilizan para que las imágenes se puedan arrastrar.

```
//Arreglo de objeto

function Array_IndexOf(Input){
    var Result = -1;
    for (var i=0; i<this.length; i++){
        if (this[i] == Input){
            Result = i;
        }
    }
    return Result;
}
Array.prototype.indexOf = Array_IndexOf;

function RemoveBottomNavBarForIE(){
    if ((C.ie)&&(document.getElementById('Reading') != null)){
        if (document.getElementById('BottomNavBar') != null){
            document.getElementById('TheBody').removeChild(document.getElementById('BottomNavBar'));
        }
    }
}

//Fijar y arrastrar

FC = new Array();
DC = new Array();

function onEndDrag(){
    DC[CurrDrag].DockToR(FC[DropTarget]);
    D[CurrDrag][2] = F[DropTarget][1];
    DC[CurrDrag].tag = DropTarget+1;
    Docked = true;
}
```



```
    }

    if (Docked == false){
        DC[CurrDrag].GoHome();
        DC[CurrDrag].tag = 0;
        D[CurrDrag][2] = 0;
    }
}

function DroppedOnFixed(DNum) {
    var Result = -1;
    var OverlapArea = 0;
    var Temp = 0;
    for (var i=0; i<FC.length; i++){

        Temp = DC[DNum].Overlap(FC[i]);
        if (Temp > OverlapArea){
            OverlapArea = Temp;
            Result = i;
        }
    }
    return Result;
}

//Colocar
for (var i=0; i<F.length; i++){
    CardContent = F[i][0];
    FC[i] = new Card('F' + i, 10);
    FC[i].elm.innerHTML = CardContent;
    if (FC[i].GetW() > Widest){
        Widest = FC[i].GetW();
    }
}

if (Widest > DragWidth){Widest = DragWidth;}

CurrTop = DragTop;

DragWidth = Rc.floor((DivWidth-Widest)/2) - 24;
RightColPos = DivWidth + LeftColPos - (DragWidth + 14);
var Highest = 0;
var WidestRight = 0;

for (i=0; i<D.length; i++){
    DC[i] = new Card('D' + i, 10);
    CardContent = D[i][0];
    DC[i].elm.innerHTML = CardContent;
    if (DC[i].GetW() > DragWidth){DC[i].SetW(DragWidth);}
    DC[i].css.cursor = 'move';
    DC[i].css.backgroundColor = '#0000ff';
    DC[i].css.color = '#000000';
    TempInt = DC[i].GetH();
    if (TempInt > Highest){Highest = TempInt;}
    TempInt = DC[i].GetW();
    if (TempInt > WidestRight){WidestRight = TempInt;}
}

var HeightToSet = Highest-12;
var WidthToSet = WidestRight-12;

for (i=0; i<D.length; i++){
    DC[i].SetT(CurrTop);
    DC[i].SetL(RightColPos);
```



```
        if (DC[i].GetH() < Highest){
            DC[i].SetH(HeightToSet);
        }
        if (DC[i].GetW() < WidestRight){
            DC[i].SetW(WidthToSet);
        }
        DC[i].SetHome();
        DC[i].tag = -1;
        CurrTop = CurrTop + DC[i].GetH() + 5;
    }

    CurrTop = DragTop;

    for (var i=0; i<F.length; i++){
        FC[i].SetW(Widest);
        if (FC[i].GetH() < Highest){
            FC[i].SetH(HeightToSet);
        }
        FC[i].SetT(CurrTop);
        FC[i].SetL(LeftColPos);
        FC[i].SetHome();
        TempInt = FC[i].GetH();
        CurrTop = CurrTop + TempInt + 5;
    }
}
```

```
F = new Array();
F[0] = new Array();
F[0][0]='S&#x00ED;mbolo de la compuerta NAND';
F[0][1] = 1;
F[1] = new Array();
F[1][0]='S&#x00ED;mbolo de la compuerta NOT';
F[1][1] = 2;
F[2] = new Array();
F[2][0]='S&#x00ED;mbolo de la compuerta NOR';
F[2][1] = 3;
F[3] = new Array();
F[3][0]='S&#x00ED;mbolo de la compuerta AND';
F[3][1] = 4;
F[4] = new Array();
F[4][0]='S&#x00ED;mbolo de la compuerta OR';
F[4][1] = 5;

D = new Array();
D[0] = new Array();
D[0][0]='</img>';
D[0][1] = 1;
D[0][2] = 0;
D[1] = new Array();
D[1][0]='</img>';
D[1][1] = 2;
D[1][2] = 0;
D[2] = new Array();
D[2][0]='</img>';
D[2][1] = 3;
D[2][2] = 0;
```



```
D[3][0]='</img>';
D[3][1] = 4;
D[3][2] = 0;
D[4] = new Array();
D[4][0]='</img>';
D[4][1] = 5;
D[4][2] = 0;

function ReduceItems2(){
    var ItemToDump=0;
    var j=0;
    while (F.length > QsToShow){
        ItemToDump = Rc.floor(F.length*Rc.random());
        for (j=ItemToDump; j<(F.length-1); j++){
            F[j] = F[j+1];
        }
        for (j=ItemToDump; j<(D.length-1); j++){
            D[j] = D[j+1];
        }
        F.length = F.length-1;
        D.length = D.length-1;
    }
}

function TimerStartUp(){
    setTimeout('StartUp()', 300);
}

function TimerStartUp(){
    setTimeout('StartUp()', 300);
}
```

Para cotejar entre las respuestas del usuario y las respuestas correctas se hace uso del siguiente código:



```
function CheckAnswers(){
    if (Locked == true){return;}
    // Establece el valor por defecto la puntuación y la respuesta

    var TotalCorrect = 0;
    Score = 0;
    var Feedback = '';

    // para cada fijo, compruebe si el valor de la etiqueta arrastrables es el mismo que el fijado

    if (AnswersTried.length > 0){AnswersTried += ' | '};
    var i, j;
    for (i=0; i<D.length; i++){
        if (i>0){AnswersTried += ','};
        AnswersTried += D[i][1] + '.' + D[i][2] + ' ';
        if ((D[i][2] == D[i][1])&&(D[i][2] > 0)){
            TotalCorrect++;
        }
        else{

            DC[i].SetL(DC[i].GetL() + 10);
            DC[i].Highlight();
        }
    }

    Score = Rc.floor((100*(TotalCorrect-Penalties))/F.length);

    var AllDone = false;

    if (TotalCorrect == F.length) {

        AllDone = true;
    }

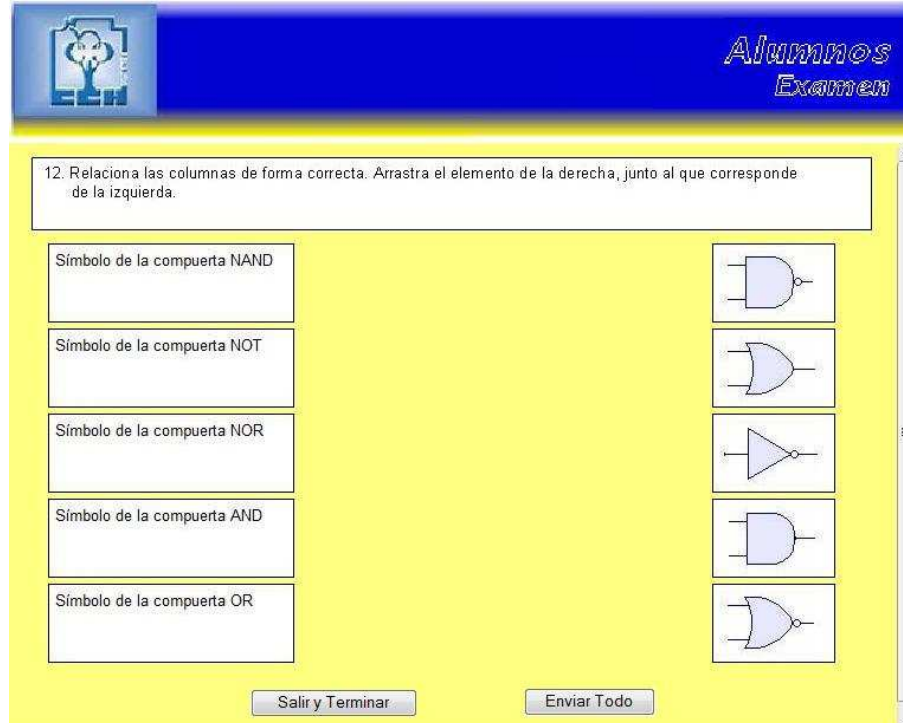
    if (AllDone == true){
        Feedback = YourScoreIs + ' ' + Score + '%.';
        ShowMessage(Feedback + '<br />' + CorrectResponse);
    }
    else {
        Feedback = IncorrectResponse + '<br />' + YourScoreIs + ' ' + Score + '%.';
        ShowMessage(Feedback);
        Penalties++;
    }

    if ((AllDone == true)|| (TimeOver == true)){

        TimeOver = true;
        Locked = true;
        Finished = true;
        setTimeout('Finish()', SubmissionTimeout);
        WriteToInstructions(Feedback);
    }
}
```



La vista del examen se muestra en la siguiente pantalla:



El alumno oprime el botón enviar todo, para que se registren sus resultados, y así terminar el proceso.



CONCLUSIONES

El presente trabajo muestra en lo general el panorama del Sistema de Evaluación para la materia de Cibernética y Computación en el CCH Oriente.

Los módulos que integran el Diplomado de Desarrollo de Sistemas en WEB, sirvieron para realizar el producto, pues con él, se conoció y reconoció algunos elementos que lo integran. El lenguaje unificado de modelado (UML), permite crear el análisis y diseño orientado a objetos, permitiendo esto al diseñador y programador, una mejor interacción. Al utilizar UML, se debe tener conocimiento del sistema sobre el cual se va a trabajar, esto es saber ¿Qué se va hacer? (etapa de análisis); ¿Cómo se le va a dar solución? (etapa de diseño), además de tener en cuenta que en la etapa de análisis se hace uso del lenguaje de usuario para saber lo que se va a requerir y en la etapa de diseño se hace uso de lenguaje técnico.

El sistema fue desarrollado con el lenguaje de programación java, y se utilizaron tecnologías como CSS (Hojas de estilo en cascada - Cascading Style Sheets), JavaScript, Servlets, JSP's, manejador de base de datos MySql, y JDBC con SQL. Las CSS consisten en definir, mediante una sintaxis especial, la forma de presentación que se le aplicará a toda una aplicación WEB o a solo una parte de ella. El código JavaScript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; esto permite que la carga en el proceso del lado del servidor sea menor.

En el desarrollo se procuró que fuera de fácil integración y adaptación de nuevos componentes para que en un futuro la aplicación fuera lo más flexible posible para mantenimiento perfecto. Se realizó la documentación adecuada, para hacer comprensible su funcionalidad y así facilitar la integración de futuros módulos.



Con la información expuesta en los capítulos anteriores, se demuestra que el sistema de Evaluación para la materia de Cibernética y Computación del CCH Oriente se implementó, logrando una evaluación automática parcial para los alumnos de esta materia, también se le proporcionó al profesor, los datos suficientes para identificar el nivel de aprendizaje de cada alumno, y así realizar, cambiar o modificar las estrategias de enseñanza.

Asimismo, cabe señalar que se concluyó con el desarrollo del sistema y que el usuario quedó satisfecho con la funcionalidad de éste, reconociendo la necesidad de nuevas aplicaciones para que pueda cubrir otras necesidades para la materia, como análisis estadísticos específicos.

La aplicación fue construida en el lenguaje de programación java, considerando que ese lenguaje es el que se maneja, no haciendo menos cualquier otro, conociendo que también se pudo realizar en un lenguaje diferente. Se piensa que no es el lenguaje de programación lo que hace la diferencia, ni le da calidad al producto, sino, que tan eficiente sé es en el uso del lenguaje y construcción de una aplicación.

También, el diplomado me ayudo a concluir una de mis etapas académicas, pues cerrando la etapa de licenciatura me permitirá avanzar en otras áreas de mi crecimiento profesional y personal.

Cursar el bachillerato y licenciatura en la Universidad Nacional Autónoma de México, es de gran satisfacción, pues como persona me permitió reafirmar mis valores de equipo y compañerismo; en lo Profesional, me preparó para aprovechar los recursos y estar a la vanguardia de la tecnología y de los cambios constantes que se dan dentro de mi profesión, estar actualizado y desarrollarme dentro de área laboral, aplicando los conocimientos transmitidos eficazmente por los profesores para el bien de la sociedad.



ÍNDICE DE TABLAS

Sentencias DDL	21
Sentencias DML	21
Clases JDBC	23
Comparación nivel administrador – Plataformas eLearning	42
Comparación tipos de reactivos – Plataformas eLearning	43
Comparación administración de reactivos – Plataformas eLearning	44
Comparación Abstracción de datos – Plataformas eLearning	44

ÍNDICE DE FIGURAS

Diagrama de clases DAO	25
Estadísticas de plataformas eLearning	34
Entorno de la plataforma Atutor	35
Entorno de la plataforma Claroline	37
Entorno de la plataforma Moodle	39



BIBLIOGRAFÍA

LIBROS

- ✓ Rosenberg Matc J. *E-Learning Estrategias para transmitir conocimiento en la era digital*, Ed. Mc Graw Hill. Argentina, 2001.
- ✓ Bou i Bauza, Guillem; Trinidad Cascudo, Carme; Huguet Borén, Llorenc. *E-Learning*, Ed. Anaya, Madrid, España, 2002.
- ✓ Díaz Martín José Manuele. *Creación de sitios WEB*, Ed. Pearson Addison Wesley, México, D. F., 2006.
- ✓ Booch G.; Rumbaugh J.; Jacobson I. *El lenguaje unificado de modelado*, Ed. Addison Wesley , México, D. F., 2000.
- ✓ Robert Cecil Martín. *UML para programadores Java*, Ed. Prentice Hall, México, D. F., 2004.
- ✓ Fowler Martin, Kendall Scott. *UML gota a gota*, Ed. Pearson Addison Wesley , México, D. F., 2000.
- ✓ Oros Cabello, Juan Carlos. *Navegar en Internet Diseño de Página WEB interactivas con JavaScript y CSS*, Ed. Rama, México, D. F., 2002.
- ✓ Schildt Herb. *Java soluciones de programación*, Ed. Mc Graw Hill, México, D. F., 2008.
- ✓ Patrick Naughton, Herbert Schildt. *Patrones de diseño aplicados a Java*, Ed. Addison Wesley , México, D. F., 2003.
- ✓ Chan C. Mark, Griffith W, Steven, Iasí F. Anthony. *1001 Tips para programar en Java*, Ed. Mc Graw Hill, Madrid, España, 2005.
- ✓ Reese George. *Programación de bases de datos con JDBC y Java*, Ed. Anaya, Madrid, España, 2003.



REFERENCIAS ELECTRÓNICAS

Plataformas de licencia pública

- ✓ <http://www.elearningworkshops.com/modules.php?name=Surveys&op=results&pollID=17&mode=&order=&thold=>
- ✓ <http://empresas.sence.cl/documentos/elearning/INTEC%20%20Estandares%20e-learning.pdf>
- ✓ http://mipagina.cantv.net/luzvirtual/modelo_e_learning/beneficios.htm

Documentación Atutor

- ✓ <http://www.atutor.ca/atutor/docs/index.php>

Documentación Claroline

- ✓ <http://www.claroline.net/online-documentation/online-documentation.html>

Documentación Moodle

- ✓ <http://docs.moodle.org/es/Portada>
- ✓ <http://es.wikipedia.org/wiki/Moodle>

Programación en Java

- ✓ <http://www.webteacher.com/javatour/>
- ✓ <http://www.javahispano.org>

Hojas de Estilo

- ✓ <http://www.ingenierosoftware.com>
- ✓ <http://www.w3.org/Style/CSS/>