

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

“Desarrollo de aplicación web para la gestión de exámenes”

Autor: Gonzalo Bolaños López

**Tesina presentada para obtener el título de:
Ing. En Sistemas Computacionales**

**Nombre del asesor:
Jorge Salvador Regalado Meza**

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

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

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





UVAQ

M.R.

**UNIVERSIDAD
VASCO DE QUIROGA**

FACULTAD DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

“Desarrollo de Aplicación Web para la Gestión de Exámenes”

TESINA

PARA OBTENER EL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTA

GONZALO BOLAÑOS LÓPEZ

ASESOR

M.C. JORGE SALVADOR

REGALADO MEZA

CLAVE: 16PSU0049F (Ing)

ACUERDO: LIC000808 (ing)

MORELIA, MICHOACÁN

DICIEMBRE 2009

AGRADECIMIENTOS

- A DIOS:

Por permitirme llegar hasta este momento tan importante y lograr otra meta más en mi carrera, y llenar mi vida de dicha y bendiciones.

- A MIS PADRES:

Por darme estabilidad emocional, económica y sentimental; por enseñarme que todo se aprende y que todo esfuerzo es al final recompensa. Les agradezco de todo corazón por su amor, cariño y comprensión. En todo momento los llevo conmigo, los amo.

- A MIS HERMANOS:

Por su compañía y apoyo que me brindan. Por convivir cada etapa de mi vida, dándome consejos e infinidad de alegrías, y todo ello con una gran dosis de amor. Sé que cuento con ustedes siempre.

- A MIS PROFESORES:

Que participaron en mi desarrollo profesional durante mi carrera. Por su disposición y ayuda brindadas. Su persistencia, su paciencia y su motivación han sido fundamentales para mi formación.

“No intentes ser un hombre de éxito.

Intenta ser un hombre de valor.”

Albert Einstein

ÍNDICE GENERAL

RESUMEN	IV
ABSTRACT	V
PLANTEAMIENTO DEL PROBLEMA	VI
ANTECEDENTES	VIII
OBJETIVOS	X
ALCANCES Y LIMITACIONES	XII
JUSTIFICACIÓN	XIII
CAPÍTULO 1 INTRODUCCIÓN	1
CAPÍTULO 2 MARCO TEORÍCO	3
2.1 TIPOS DE EXAMENES.....	3
2.2 APLICACIONES DE GESTION DE EXAMENES.	7
2.2.1 EXAMSYSTEM® II	7
2.2.2 SAGRADER	9
2.2.3 PERCEPTION	11
CAPÍTULO 3 LENGUAJE JAVA Y JSP	14
3.1 GENERALIDADES DEL LENGUAJE JAVA.....	14
3.2 GENERALIDADES DEL LENGUAJE JSP.....	18
CAPÍTULO 4 REVISION TECNICA	21
4.1 TOMCAT	21
4.2 JDK	23
4.3 PHPMYADMIN	24
4.4 WEB HOSTING.....	25
CAPÍTULO 5 METODOLOGIA	29
5.1 INSTALACION DE TOMCAT	29
5.2 SISTEMA DE ARCHIVOS DE LA APLICACIÓN EN TOMCAT	31
5.3 JEDIT	32
CAPÍTULO 6 RESULTADOS	36
6.1 INTERFACES	36
6.2 BASE DE DATOS	67
6.3 POJO`S Y SERVLETS.....	73
CAPÍTULO 7 CONCLUSIONES Y TRABAJO FUTURO	85
BIBLIOGRAFÍA	87
ÍNDICE DE FIGURAS	89
GLOSARIO DE TÉRMINOS	90

RESUMEN

En el presente trabajo se llevó cabo investigación y desarrollo acerca de una aplicación *Web* basada en JSP la cual proporciona al usuario (en este caso al profesor) la capacidad de crear un examen de tipo digital con distintas modalidades. El audio, video e imágenes quedan excluidos para este tipo de aplicación.

Existen dos tipos de usuario:

*Alumnos. Ingresan a su cuenta y realizan un examen en específico, y se guarda para su futura evaluación.

*Profesores. Tienen la capacidad de crear exámenes para poder ser aplicados a los alumnos en línea, guardarlos y obtener un resultado de acuerdo a la evaluación del sistema una vez finalizado el examen.

Es una aplicación *Web* sencilla y fácil de utilizar por parte del usuario, contiene de igual manera una interfaz adecuada para la institución que representa.

ABSTRACT

This work is an investigation and development of a Web application based on JSP, it provides to the final user (in this case a teacher) the ability to create a digital exam of different categories. Audio, video and images are excluded for this application.

There are two user types:

*Students. Are able to sign in and make a specific exam, also save it for a future evaluation.

*Teachers. Are able to create exams and being apply to the students on line, save them and have the result according to the evaluation that the system shall do once the exam has been applied.

It is a simple and easy Web application with a practical use for the final user and contains also a proper interface for the institution that represents.

PLANTEAMIENTO DEL PROBLEMA

Con el avance de la tecnología se desarrollan día a día aplicaciones que facilitan el quehacer cotidiano ya sea solo para escribir una carta, jugar o construir una casa.

En el sector educativo se ha detectado una dificultad en la aplicación de las evaluaciones, pues los exámenes parciales y/o finales tienden a ser inconsistentes en los resultados, ya sea que un alumno tenga una respuesta similar a otro y su resultado sea más alto o también que la información requerida para contestar el examen no sea la adecuada o insuficiente.

De igual manera es difícil para el alumno tener un control sobre sus evaluaciones, pues los profesores en ocasiones tardan en calificar sus exámenes y tienen que consultar posteriormente en control escolar cuales fueron sus resultados. También hay ocasiones en las cuales los profesores por descuido u otras circunstancias pierden los exámenes y el alumno tiene que volverlos a realizar o ser evaluados de otra manera.

Es por ello que existía la necesidad de crear una aplicación la cual desarrollara y proporcionara exámenes en línea, y así poder obtener un resultado más confiable de acuerdo a las ponderaciones previamente establecidas por los profesores acerca de la calificación de cada pregunta así como el examen en su totalidad.

Para la realización de esta aplicación *Web* surgieron varias interrogantes:

- 1.- ¿Que recursos tecnológicos se requieren para realizar una aplicación *Web* basada en JSP?
- 2.- ¿Qué características esenciales deberá tener la aplicación *Web*?
- 3.- ¿Desarrollar dicha aplicación mejorará el rendimiento académico del alumno?
- 4.- ¿Qué métodos o funciones en JSP se pueden implementar para realizar la aplicación *Web*?

En el transcurso de esta investigación se respondieron dichas preguntas, dando como resultado una aplicación *Web* que puede ser utilizada por cualquier institución educativa para la realización de exámenes en línea.

ANTECEDENTES

En los últimos años las aplicaciones *Web* han mostrado un gran auge gracias a Internet y la proliferación de sitios *Web*, sobre todo con el fin de fomentar el comercio electrónico.

Su facilidad de administración centralizada las hace ideales tanto para su despliegue en Internet como en intranets corporativas.

La facilidad de uso de las interfaces *Web* y el hecho de que cada día más personas están acostumbradas a la navegación por Internet, hace que el tiempo de aprendizaje se reduzca considerablemente respecto a aplicaciones tradicionales.

El auge de multitud de soluciones *open source* hace que su desarrollo sea sencillo y que un gran número de desarrolladores tengan experiencia con ellos. Otro hecho a tener en cuenta es que una vez realizada una aplicación *Web* para uso interno de una empresa, por ejemplo en una intranet, el poner esa funcionalidad, o incluso funcionalidades nuevas, a disposición de empleados o el público general tiene un coste mínimo a la vez que una potencial proyección mundial.

Actualmente existen varias aplicaciones que nos ayudan a nuestras tareas cotidianas, en este proyecto se centra en la actividad educativa donde a través de exámenes en línea se realizan evaluaciones periódicas de los alumnos. Su tarea principal consiste en proporcionar al profesor una manera práctica y sencilla de generar y evaluar exámenes en el menor tiempo posible, obtener de igual manera una calificación exacta y satisfactoria para el alumno, la cual el propio sistema generará. Es a este respecto que existen ya varias aplicaciones que realizan dichas funciones como:

* El Sistema Profesional de Calificación de exámenes *examSYSTEM*[®] II. Esta herramienta poderosa y fácil de usar califica exámenes basados en objetivos o de respuesta múltiple utilizando Formularios *OMR scan* y escáner *OMR OpScan*[®].

* *SAGrader*, es un programa de software capaz de evaluar trabajos y preguntas de exámenes. El programa utiliza análisis semántico para realizar la evaluación y la tecnología del programa *qualrus* para reconocer estructuras dentro de los trabajos y compararlas con respuestas acertadas. De esta forma determina una calificación y ofrece información complementaria a los estudiantes.

La finalidad del proyecto consistió en generar una herramienta similar a las ya existentes, con la particularidad de que está desarrollada con JSP, por lo que puede ser utilizada a través de internet y en consecuencia en cualquier parte del mundo.

OBJETIVOS**Los objetivos del proyecto son:****OBJETIVO GENERAL**

*Desarrollar una aplicación *Web* para la gestión de exámenes, implementando JSP dirigido a los profesores de todos los niveles educativos para que puedan aplicar exámenes en línea y ser evaluados automáticamente por el sistema de la aplicación.

OBJETIVOS ESPECIFICOS

*Utilizar los métodos que brinda el lenguaje JSP para hacer la aplicación dinámica.

*Desarrollar la aplicación de exámenes en línea de manera que sea intuitiva, sin necesidad de proporcionar la función de ayuda que otras aplicaciones contienen por la dificultad del sistema para ser utilizado por el usuario.

*Hacer que el propio sistema califique los exámenes de acuerdo a las ponderaciones que se le otorguen a cada pregunta del examen en cuestión.

ALCANCES Y LIMITACIONES

Alcances

- *Todo alumno y profesor tiene acceso a la aplicación para realizar consultas respectivas a su área.
- *Se pueden crear exámenes con tres diferentes modalidades: Opción Múltiple, Falso / Verdadero y Mixto (combinación de las dos anteriores).
- *Una vez el examen contestado es evaluado por el sistema.
- *Cada usuario tiene opciones específicas de consulta para que su búsqueda e información resultante sea limitada en base a sus requerimientos.
- *Los alumnos tienen un tiempo límite para contestar el examen, cuando falta 1 minuto para su culminación se envía un mensaje mostrando el tiempo restante, posteriormente se lleva a cabo su evaluación de manera automática.

Limitaciones

- *La aplicación tiene únicamente dos tipos de usuario: alumnos y profesores. Solo podrán registrarse aquellos usuarios que tenga la clave correspondiente.
- *No contiene información general de la universidad, como actividades artísticas y/o deportivas. Se da la opción que consulte la página principal.
- * Solo son tres modalidades de examen: Opción Múltiple, Falso / Verdadero y Mixto (combinación de las dos anteriores). Las modalidades Correlacional, Orden Predeterminado no se comprenden para esta aplicación.
- * Es necesario tener activado JavaScript por parte del navegador para que todas las funciones de la aplicación se lleven a cabo de forma adecuada. En caso contrario se le pedirá al usuario que lo active.

JUSTIFICACIÓN

La información de cualquier empresa actualmente es uno de los activos más valiosos que se tienen, el hecho de que los usuarios ya sean: sus socios, asociados, clientes o proveedores puedan tener acceso a ella en el momento que la requieren, ofrece una gran ventaja competitiva.

Una aplicación *Web* es una interfaz entre un formulario diseñado específicamente para cubrir con las necesidades de todo tipo de negocio e información que actualmente se tenga hacia el interior de la empresa, como pueden ser sistemas administrativos, inventarios, facturación, cuentas por cobrar, productos, etc. (la información puede ser de dominio público o restringida a ciertas personas a través de un nombre de usuario y contraseña) con el objetivo de que cualquier persona pueda consultarla e interactuar con ella desde Internet.

El desarrollo de una aplicación *Web* permite publicar un catálogo electrónico de productos, manejo de inventarios, órdenes de compra, publicación de información con acceso restringido a ciertos usuarios, actualización y mantenimiento de su sitio *Web* y en general, permitirá publicar cualquier tipo de información que se pueda incorporar a una base de datos.

En el caso de empresas educativas se puede dar un mayor auge y publicidad, es por ello que se cree conveniente la realización de una página de creación y consulta de exámenes para alumnos y profesores en particular de la Universidad Vasco de Quiroga, generado de la necesidad del alumno principalmente de revisar su historial académico para tener un mayor control sobre sus resultados y materias que está cursando actualmente, de igual manera proporcionar una herramienta a los profesores para que puedan directamente generar sus exámenes en línea con sus distintas modalidades de aplicación, así como realizar la evaluación automáticamente por parte del sistema, evitando así, retrasos en fechas de entrega por situaciones a veces ajenas a ellos.

Se ofrece un mejor servicio al alumno al poder realizar exámenes en línea, sin fallo de error al momento de ser calificados. JSP es la herramienta en la cual se basa el desarrollo de este servicio, apoyando el proyecto en el lenguaje Java a través de sus métodos, brindando funcionalidad e interactividad a la aplicación.

Capítulo 1

INTRODUCCIÓN

Hoy en día la sociedad reclama una educación de calidad para los futuros profesionistas, en la búsqueda de dicha calidad se comprenden distintas variables capaces de modificarla.

La evaluación del aprendizaje se vale de varias herramientas que le ayudan a determinar el desempeño del estudiante, uno de ellos es el examen. El examen proporciona la ayuda necesaria para que el aprendizaje cumpla con sus objetivos de enseñanza.

La aplicación de exámenes dentro de la evaluación es fundamental ya que conforman un gran porcentaje de la calificación final de un alumno, es por ello que un examen bien estructurado, planeado y aplicado es fundamental.

Actualmente la introducción de la tecnología para el apoyo en el sector educativo da una pauta para implementar dichos exámenes que antes eran en papel, ahora se realicen de manera digital, y es donde la usabilidad toma gran importancia pues es uno de los aspectos a considerar cuando se realiza una aplicación determinada, en este caso para crear exámenes en línea, pues de ésta depende cómo van a interactuar los usuarios con ella y podría suponer un éxito o un fracaso de un proyecto.

Este proyecto es un sistema el cual está dirigido a los profesores, proponiendo ser una herramienta de fácil uso, gran comodidad y flexibilidad; por lo que no se requieren amplios conocimientos en Internet e informática para utilizarlo.

Estos exámenes son aplicados a los estudiantes a distancia desde la comodidad del hogar o de igual manera haciendo uso de un laboratorio ó sala de cómputo, los cuales son cada día más comunes en las instituciones educativas de nuestro país.

El proyecto fue generado con el lenguaje de programación JSP el cual permite a la aplicación ser totalmente funcional sobre Internet o Intranets.

En el siguiente capítulo se abordará el tema acerca de los tipos de exámenes, en donde se desarrolla una pequeña explicación sobre las distintas modalidades que se comprenden, es decir, los diferentes tipos de preguntas y la manera en que se deben responder. Por otra parte se describen también algunas aplicaciones por medio de las cuales se generan todo tipo de exámenes, se guardan y posteriormente se aplican a los alumnos. Son este tipo de aplicaciones una guía para el desarrollo de este proyecto.

En el capítulo 3 se desarrolla el tema sobre las generalidades del lenguaje Java y JSP, señalando algunas de sus características así como también ventajas y desventajas sobre su implementación.

El capítulo 4 hace referencia a describir las tecnologías que se van a implementar para esta aplicación como son: el servidor Tomcat, el JDK, *PhpMyAdmin* y las paginas que ofrecen servicios para subir y alojar las aplicaciones en la *Web*.

En el capítulo 5 se señalan los pasos para instalar el servidor Tomcat, se describe su sistema de archivos y el programa compilador de archivos java llamado *JEdit*.

En el capítulo 6 se aborda la parte principal del proyecto donde se describen las interfaces que contiene, las tablas de la base de datos y una breve explicación de las funciones de cada archivo Java y JSP.

Finalmente en el capítulo 7 se señalan las conclusiones y trabajo futuro que generó el proyecto, señalando los resultados obtenidos y proponiendo nuevas funciones y mejoras a la aplicación.

Capítulo 2

MARCO TEORICO

Un examen o prueba es una herramienta o técnica que se usa para medir los conocimientos y habilidades de los estudiantes.

Existen distintos tipos de examen, cada uno de ellos con sus ventajas e inconvenientes, y por supuesto también con la forma más indicada de realizarlo.

2.1 Tipos de Exámenes

* Orden predeterminado.

Requieren que el alumno realice un ordenamiento específico de las diferentes opciones que se les proporcionan. Dependerá del tipo de pregunta para saber de qué manera se ordenarán dichas opciones. Ver figura 2.1:

1.- Ordena los planetas de acuerdo a su localización en el sistema solar de adentro hacia afuera: a) Tierra b) Venus c) Marte

Respuesta: b) Venus a) Tierra c) Marte

Figura 2.1 Examen tipo Orden Predeterminado

VENTAJAS: estar familiarizados con la técnica, al observar las opciones disponibles agilizan la memoria y entendimiento de lo que se pretende obtener.

INCONVENIENTES: confundir el orden en que se deben de llevar las opciones disponibles. Si una opción no está en el orden correcto toda la respuesta es incorrecta.

REQUISITOS: capacidad de memorización, entendimiento total de lo que se pregunta y el orden en que se debe responder, es necesario tener cierto dominio de la materia.

NORMAS PRÁCTICAS:

- Dedicar un tiempo inicial a analizar las opciones.
- Leer detenidamente las preguntas.
- Comenzar por las preguntas que se dominen mejor.
- Contestar exactamente a lo que se pide.
- Hacerlo respetando el orden en que se debe contestar.
- No dejar preguntas sin respuesta.
- Dedicar un tiempo final para releer y corregir.

*** Opción múltiple.**

Los exámenes tipo test (opción múltiple) también son pertenecientes a la especie de amados por unos, odiados por otros. El autor del examen proporciona varias respuestas (generalmente cuatro o cinco) para que seleccione el estudiante. Sólo hay una respuesta correcta, generalmente es solo una opción, aunque la respuesta correcta puede consistir en seleccionar dos o más opciones.

VENTAJAS: la corrección es más objetiva y facilitan el recuerdo. Las preguntas de opciones múltiples son muy fáciles de calificar.

INCONVENIENTES: hay que trabajar deprisa y se suelen penalizar los errores.

REQUISITOS: capacidad para trabajar rápidamente y leer y comprender toda la materia.

NORMAS PRÁCTICAS:

- Leer las instrucciones con mucha atención.
- Leer cada pregunta localizando la palabra clave.

- Saber si se penalizan los errores, para no arriesgarse innecesariamente. En caso afirmativo, no contestar si no hay una relativa seguridad de acertar.
- Poner mucha atención a los calificativos que aparezcan en la pregunta (adverbios, adjetivos) porque:
 - Sirven para distinguir entre lo verdadero y falso.
 - Sirven para razonar lógicamente.
 - Los muy generales y absolutos (todo, nada, siempre...) suelen ser falsos, incorrectos o no los mejores.
- Al elegir entre varias opciones:
 - Tratar de adivinar la respuesta antes de leer las opciones.
 - Leer las opciones.
 - Si alguna coincide con nuestra respuesta puede ser la correcta.
 - Si no coincide, ir eliminando las que no tienen sentido hasta llegar a la más acertada.
 - Plantear la pregunta de otra forma.
 - Buscar pistas en las respuestas. [6]

* Preguntas correlacionales.

Se le pide al sujeto asociar o emparejar determinadas palabras, expresiones, conceptos, fórmulas, etc., ofrecidos en el enunciado, con un conjunto de respuestas. Para tener más claro el concepto se proporciona el siguiente ejemplo. Ver figura 2.2:

a) Irradia calor, es el centro de nuestro sistema solar a) Marte
b) Es el cuarto planeta del sistema solar b) Sol



Figura 2.2 Examen tipo Correlacional

VENTAJAS: son fáciles de calificar porque tienen respuestas previamente definidas. Evita construir opciones múltiples para cada alternativa. Permite evaluar bien asociaciones, definiciones, características o ejemplos de conceptos o principios.

Es un formato que ocupa poco espacio, requiere menos tiempo que el de opción múltiple y engloba más contenidos.

INCONVENIENTES: puede haber ocasiones en las que existan más respuestas que definiciones y confundan al estudiante.

REQUISITOS: capacidad de distinguir y razonar las respuestas pues no puede haber una respuesta para dos definiciones distintas.

NORMAS PRÁCTICAS

- Leer cuidadosamente cada definición
- Contar previamente cuantas definiciones y respuestas existen para comparar si el número de definiciones es el mismo que el de respuestas.
- Direccionar adecuadamente una línea a la respuesta correcta para no tener lugar a errores.

* Falso / Verdadero.

El alumno debe responder acerca de la verdad o falsedad de un enunciado.

VENTAJAS: son preguntas más fáciles de redactar que las de opción múltiple. Son fáciles de puntualizar.

INCONVENIENTES: suelen contener información trivial. Para resolver correctamente este tipo de ítems no se requiere un aprendizaje muy elaborado de la información. Se favorece un aprendizaje memorístico. A veces, si no se cuida mucho la redacción, los estudiantes que más saben encuentran matices que les hacen dudar sobre la verdad o falsedad del enunciado.

NORMAS PRÁCTICAS

- Contestar las primeras preguntas que sean fáciles de respuesta por parte del estudiante.

- Leer varias veces el enunciado para comprender su significado y así razonar la respuesta.
- No dejar sin contestar ninguna pregunta. [7]

* **Misceláneos o Mixto.**

Este tipo de exámenes comprenden todas las modalidades antes mencionadas es decir, la combinación de preguntas abiertas y solución de problemas con respuestas de opción múltiple, Falso / Verdadero y correlacionales. Como ejemplo tenemos, ver figura 2.3:

1. Ordena los planetas de acuerdo a su localización en el sistema solar de afuera hacia adentro. a)Neptuno b)Urano c)Plutón (Pregunta orden predeterminado)
2. La luna es un satélite de la tierra ___ (Falso / Verdadero)
3. La posición de la Tierra respecto al sol es: (Op. Múltiple)
a) Segunda b) Primera c) Tercera

Figura 2.3 Examen tipo Misceláneo

Estos exámenes hacen que el aprendizaje sea más dinámico y menos tedioso pues proporcionan al estudiante mayor información y opciones para su resolución.

2.2 Aplicaciones de Gestión de Exámenes.

A continuación se dará una pequeña descripción de los programas de gestión de exámenes existente actualmente, con la finalidad de obtener sus características esenciales para ser desarrolladas y adecuadas al proyecto.

2.2.1 Sistema Profesional de Calificación de exámenes *examSYSTEM II*.

El Sistema Profesional de Calificación de exámenes *examSYSTEM II*® le permite evaluar diversas aplicaciones, tales como:

Educación

- Exámenes de clase diseñados por los maestros
- Exámenes de final de capítulo
- Cuestionarios, exámenes de mitad de curso y exámenes finales
- Exámenes estatales estándar/práctica
- Exámenes de opción múltiple
- Exámenes de situación
- Exámenes de habilidades
- Exámenes de distrito
- Pre y post exámenes
- Exámenes de centro de evaluación

Organizaciones Comerciales y Oficinas Gubernamentales

- Exámenes de capacitación de nivel II
- Adquisición de habilidades específicas e inventario de habilidades
- Exámenes de evaluación pre-empleo
- Exámenes de pre y post capacitación
- Análisis de necesidades de capacitación
- Exámenes de certificaciones y licencias
- Certificaciones de conocimientos y habilidades

Propiedades clave y características

Es flexible.

- Califica exámenes de diversos formatos y tamaños de manera rápida
- Puede acomodar fácilmente hasta 360 preguntas por examen.
- Capacidad poderosa de análisis y calificación
- Soporta hasta 10 sub-exámenes de acuerdo a objetivos de aprendizaje, habilidades, competencias o unidades de conocimiento
- La opción de ensamble le permite fácilmente incluir resultados para exámenes recuperatorios.

- Interactúa con el software de respuestas *InteGrade Pro*[®]
- Capaz de exportar fácilmente datos a otros paquetes de software.

Es de fácil uso.

- Instrucciones directas y sencillas lo conducen a través del proceso de calificación.
- El sistema requiere una computadora estándar *PC* con *Windows*[®]
- Un *CD-ROM* interactivo de capacitación le permite un rápido aprendizaje de todas las funciones.
- 19 páginas de respuestas únicas le permite acomodar todas sus necesidades de exámenes.

Trabaja con los escáner *OMR OpScan*[®] de *Scantron*.

El software *ExamSYSTEM II* interactuar con todos los escáner *OMR OpScan* de *Scantron*. La tecnología de *Scantron AccuRead*, incorporada a cada escáner *OpScan*, le asegura que la información de cada test reproducido por escáner es exacta y fue marcada en el documento.

El Software *ExamSYSTEM II* produce los siguientes tipos de reportes fáciles de entender:

- Reporte por resultado individual de examen
- Reporte temático por respuesta individual
- Reporte de análisis por tema
- Reporte de distribución de frecuencia relativa
- Reporte de distribución de frecuencia absoluta
- Reporte de distribución de Calificaciones por examen [8]

2.2.2 SAGrader

SAGrader, es un programa de software capaz de evaluar trabajos y preguntas de exámenes. El programa utiliza análisis semántico (SA) para realizar la evaluación y la tecnología del programa *qualrus* para reconocer estructuras dentro de los trabajos y

compararlas con respuestas acertadas. De esta forma determina una calificación y ofrece información complementaria a los estudiantes.

Cómo determina las calificaciones *SAGrader*.

El programa *SAGrader* mide la capacidad de los estudiantes para identificar los elementos más importantes del conocimiento sustantivo y razonarlos de la manera adecuada para la asignatura en cuestión.

Primero es necesario que el profesor responsable de una signatura cree una base de conocimiento de la misma que represente el conocimiento sustantivo a través de una red semántica de conceptos, teorías, trabajos de investigación, descubrimientos e información complementaria. Dicha red semántica identifica las características claves, y especifica relaciones lógicas entre las mismas, tales como, resultados de cada investigación, características de cada concepto y conceptos relacionados con cada teoría.

A continuación el profesor debe especificar un trabajo, prueba o examen. El trabajo, más la base de conocimiento, nos permite especificar los argumentos deseados que se deberían incluir en cualquier buen trabajo y las notas que deberían ser asociadas con cada uno de los mismos (una rúbrica de evaluación).

Se utilizan varias estrategias informáticas inteligentes, incluyendo reconocimiento de estructuras aprendizaje con máquinas y conocimiento de procedimientos para detectar estructuras en los trabajos de los estudiantes. No solamente conceptos sino también formas de razonar y argumentar que resulten básicas para la asignatura en cuestión. Se comparan las características deseadas de la asignatura con las detectadas en el trabajo del estudiante. Finalmente se utilizan las similitudes y las diferencias entre los argumentos deseados y observados para razonar la evaluación final del trabajo y generar la respuesta. [9]

2.2.3 Software de Evaluación Online Perception.

Este software es una aplicación poderosa especial para crear exámenes y encuestas en Internet. El software permite fácil y rápidamente crear y administrar evaluaciones online con un esfuerzo mínimo, proporcionando más tiempo para invertir con los alumnos en la clase o en la oficina.

Preguntas de los exámenes

Los autores de los exámenes no requieren experiencia de programación para crear una base de datos de preguntas utilizando el software *Perception*. La creación de exámenes y preguntas es flexible, con un amplio rango de estilos disponibles, incluyendo:

- Opción múltiple (incluye si-no y verdadero-falso)
- Respuesta múltiple
- Complete en el espacio
- Preguntas a desarrollar tipo ensayo
- Respuesta por palabra (coincidencia de texto)
- Respuestas numéricas
- Ordenar / numerar (pregunta de selección)
- Punto clave
- Arrastre y suelte
- Matriz

Test y Encuestas

Los administradores pueden definir parámetros para los test y evaluaciones de acuerdo a:

- Restricciones de tiempo
- Suministro de opinión (inmediato)
- Estilo de prototipos
- Examinación adaptable
- Preguntas y Respuestas al azar

- Procedimientos de final de examen
- Selección aleatoria por cada participante

Los autores pueden acceder a una vista preliminar de sus evaluaciones o encuestas previa a la administración, luego enviar las calificaciones vía una computadora *PC* con *Microsoft® Windows®*, un *CD-ROM*, un red de área local (*LAN*), red organizacional Intranet o Internet.

Opciones de Reporte

Varios reportes preseleccionados están disponibles con el software *Perception*, incluyendo los siguientes:

- Reporte por Participante
- Reporte estadístico por pregunta
- Reporte general de la sesión
- Reporte por encuesta
- Reporte por lista
- Reporte por boletín de calificaciones
- Reporte estadístico por tema
- Reporte de comparación por grupos

Aplicaciones

El software *Perception* puede también ser usado para enviar exámenes, cuestionarios y encuestas a estudiantes de todas las edades. Los instructores utilizan el software *Perception* para crear y administrar exámenes de práctica, exámenes de diagnóstico, y exámenes de pre-evaluación. Otras aplicaciones incluyen evaluación de habilidades pre-curso, exámenes de final de capítulo, exámenes de mitad de curso o finales (para incluir a estudiantes en lugares remotos) y test de conocimiento de productos.

Opciones de Seguridad

El software *Perception* incluye las siguientes opciones de seguridad:

- Nombre de usuario y palabras clave
- *Time Windows*
- Restricciones en el número de veces un estudiante puede tomar un examen
- Una función que evita que los estudiantes cierren el examen cuando están usando otras aplicaciones de la computadora o navegando en la Red mientras la evaluación está en curso.
- Botones de impresión inhabilitados
- *Browser* de seguridad, que remueve las herramientas estándar de los *browser* e inhabilita teclas de *Windows* clave. [10]

Capítulo 3

LENGUAJE JAVA Y JSP

Para la realización de este proyecto se emplearán dos tipos de tecnologías: el lenguaje de programación Java y el lenguaje de programación basado en *Web* JSP. Es por ello que se considera dar una pequeña introducción a estas dos ramas de la programación para conocer sus características, ventajas y desventajas que conlleva sus respectivas implementaciones.

3.1 Generalidades del lenguaje Java.

Java es un lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems* a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por *Sun Microsystems* en 1995. Desde entonces, *Sun* ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del *Java Community Process*, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de *Sun*, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, *Sun Microsystems* liberó la mayor parte de sus tecnologías Java bajo la licencia *GNU GPL*, de acuerdo con las especificaciones del *Java Community Process*, de tal forma que prácticamente todo

el Java de *Sun* es ahora software libre (aunque la biblioteca de clases de *Sun* que se requiere para ejecutar los programas Java todavía no es software libre).

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.[1]

Características del lenguaje Java

*Lenguaje simple.

Java posee una curva de aprendizaje muy rápida. Resulta relativamente sencillo escribir *applets* interesantes desde el principio. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, y dado que la mayoría de la gente los conoce aunque sea de forma elemental, resulta muy fácil aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.

*Orientado a objetos

Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

***Distribuido**

Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

***Interpretado y compilado a la vez**

Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los *bytecodes*, semejantes a las instrucciones de ensamblador.

Por otra parte, es interpretado, ya que los *bytecodes* se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (*run-time*).

***Robusto**

Java fue diseñado para crear *software* altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

***Indiferente a la arquitectura**

Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde *Unix* a *Windows NT* pasando por *Mac* y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. Para acomodar requisitos de ejecución, el compilador de Java genera *bytecodes*: un formato intermedio indiferente a la arquitectura diseñada para transportar el código eficientemente a múltiples plataformas *hardware* y *software*. El resto de problemas los soluciona el intérprete de Java.

*Portable

La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas. Estas dos últimas características se conocen como la Máquina Virtual Java (JVM).

*Alto rendimiento

Multihebra

Hoy en día ya se ven como terriblemente limitadas las aplicaciones que sólo pueden ejecutar una acción a la vez. Java soporta sincronización de múltiples hilos de ejecución (*multithreading*) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas. Así, mientras un hilo se encarga de la comunicación, otro puede interactuar con el usuario mientras otro presenta una animación en pantalla y otro realiza cálculos.

*Dinámico

El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.

*Produce *applets*

Java puede ser usado para crear una gran cantidad de diversos programas.

Las aplicaciones independientes se comportan como cualquier otro programa escrito en cualquier lenguaje, como por ejemplo el navegador de *Web HotJava*, escrito íntegramente en Java.

Por su parte, los *applets* son pequeños programas que aparecen embebidos en las páginas *Web*, como aparecen los gráficos o el texto, pero con la capacidad de

ejecutar acciones muy complejas, como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo para luego emprender acciones, etc. [2]

3.2 Generalidades del lenguaje JSP.

JSP es un acrónimo de *Java Server Pages*, (Páginas de Servidor Java). Es una tecnología orientada a crear páginas *Web* con programación en Java.

Con JSP se pueden crear aplicaciones *Web* que se ejecuten en variados servidores *Web*, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar *scripts* de servidor en sintaxis Java. Por tanto, las JSP's podremos escribirlas con nuestro editor HTML/XML habitual.

Motor JSP

El motor de las páginas JSP está basado en los *servlets* de Java -programas en Java destinados a ejecutarse en el servidor-, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo aprender que los *servlets*.

En JSP se crean páginas de manera parecida a como se desarrollan en ASP o PHP (otras dos tecnologías de servidor). Generamos archivos con extensión .jsp que incluyen, dentro de la estructura de etiquetas HTML, las sentencias Java a ejecutar en el servidor. Ejemplo:

```
<%
    String numS1 = request.getParameter("num1");
    int x = Integer.parseInt(numS1);
    if ( x<0 )
        out.print(x+"<0");
    if(0<=x&&x<=100)
        out.print ("0<="+x+"<=100");
    if(x>100)
        out.print(x+">100");
%>
```

En este programa se realiza una comparación de un número que es introducido por el usuario, se ubica en los parámetros $x < 0$, $0 \leq x \leq 100$ y $x > 100$ e imprime el resultado. Antes de que sean funcionales los archivos, el motor *JSP* lleva a cabo una fase de traducción de esa página en un *servlet*, implementado en un archivo *class* (*Byte codes* de Java). Esta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de la página *.jsp*, aunque existe la opción de precompilar en código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

JSP puede considerarse como una manera alternativa, y simplificada, de construir *servlets*. Es por esto que una página puede hacer todo lo que un *servlet* puede hacer, y viceversa. Cada versión de la especificación de JSP está fuertemente vinculada a una versión en particular de la especificación de *servlets*.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del *servlet* a generar. Este *servlet* será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario. [3]
JSP -> Servidor Aplicaciones (*Servlets*) -> Cliente (Navegador)

Sintaxis:

```
<%= new java.util.Date() %>
```

Características

- *Código separado de la lógica del programa.
- *Las páginas son compiladas en la primera petición.
- *Permite separar la parte dinámica de la estática en las páginas *Web*.
- *Los archivos se encuentran con la extensión *.jsp*.
- *El código JSP puede ser incrustado en código HTML.

Elementos de JSP

Los elementos que pueden ser insertados en las páginas JSP son los siguientes:

- **Código:** se puede incrustar código “Java”.
- **Directivas:** permite controlar parámetros del *servlet*.
- **Acciones:** permite alterar el flujo normal de ejecución de una página.

Ventajas:

- Ejecución rápida del *servlet*.
- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.
- La parte dinámica está escrita en Java.
- Permite la utilización de *servlets*.

Desventajas:

- Complejidad de aprendizaje. [5]

Capítulo 4

REVISION TECNICA

A continuación se realiza una breve explicación acerca de las tecnologías que se aplicarán para el desarrollo del proyecto es decir, el tipo de servidor, la base de datos y los tipos de páginas que proporcionan servicios para implementar las aplicaciones en la *Web*.

4.1 Tomcat

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

Es un servidor *Web* el cual incluye el compilador *Jasper*, que lleva a cabo la compilación de JSP`s convirtiéndolas en *servlets*. El motor de *servlets* de Tomcat a menudo se presenta en combinación con el servidor *Web* Apache. Puede funcionar como servidor *Web* por sí mismo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de su máquina virtual (JVM).

Estructura de directorios

La jerarquía de directorios de instalación de Tomcat incluye:

- *bin* - arranque, cierre, y otros scripts y ejecutables.
- *common* - clases comunes que pueden utilizar Tomcat y las aplicaciones *Web*.
- *conf* - archivos XML y los correspondientes DTD para la configuración de Tomcat.
- *logs* - *logs* del servidor Tomcat y de las aplicaciones.

- *server* - clases utilizadas solamente por Tomcat.
- *shared* - clases compartidas por todas las aplicaciones *Web*.
- *webapps* - directorio que contiene las aplicaciones *Web*.
- *work* - almacenamiento temporal de archivos y directorios.

Características

Tomcat 3.x (distribución inicial)

- Implementado a partir de las especificaciones *Servlet 2.2* y *JSP 1.1*
- Recarga de *servlets*
- Funciones básicas HTTP

Tomcat 4.x

- Implementado a partir de las especificaciones *Servlet 2.3* y *JSP 1.2*
- Contenedor de *servlets* rediseñado como Catalina
- Motor JSP rediseñado con *Jasper*
- Conector Coyote
- *Java Management Extensions (JMX)*, JSP y administración basada en *Struts*

Tomcat 5.x

- Implementado a partir de las especificaciones *Servlet 2.4* y *JSP 2.0*
- Recolección de basura reducida
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas
- Análisis rápido JSP

Tomcat 6.x

- Implementado de *Servlet 2.5* y *JSP 2.1*
- Soporte para *Unified Expression Language 2.1*
- Diseñado para funcionar en Java *SE 5.0* y posteriores
- Soporte para *Comet* a través de la interfaz *CometProcessor*

*La versión utilizada para este proyecto es apache-tomcat-6.0.18. [11]

4.2 JDK

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java. Puede instalarse en una computadora local o en una unidad de red.

En la unidad de red se puede tener la aplicación distribuida en varias computadoras y trabajar como una sola aplicación. En los sistemas Windows sus variables de entorno son:

- *JAVAPATH*: es un *path* completo del directorio donde está instalado el JDK.
- *CLASSPATH*: variable de entorno para ubicar las librerías o clases de usuario.
- *PATH*: variable donde se agrega la ubicación de JDK

Los programas más importantes que se incluyen son:

- *Appletviewer*: es un visor de *applet* para generar sus vistas previas, ya que un *applet* carece de método *main* y no se puede ejecutar con el programa java.
- *javac*: es el compilador de JAVA.
- *java*: es el intérprete de JAVA.
- *javadoc*: genera la documentación de las clases java de un programa. [12]

Intérprete en tiempo de ejecución (JRE)

Permite la ejecución de los programas Java (*.class). La sintaxis para su utilización es la siguiente:

```
java [Opciones] ClaseAEjecutar [Argumentos]
```

- *Opciones*: Especifica opciones relacionadas con la forma en que el intérprete Java ejecuta el programa.
- *ClaseAEjecutar*: Especifica el nombre de la clase cuyo método *main()* se desea ejecutar como programa. Si la clase reside en un paquete se deberá especificar su ruta mediante en forma *paquete.subpaquete.clase_a_ejecutar*.

- *Argumentos*: Especifica los argumentos que se recibirán en el parámetro `s` del método `main(String s)`, por si el programa necesita de parámetros de ejecución. Si por ejemplo el programa realiza el filtrado de un archivo, probablemente nos interese recibir como argumento la ruta del fichero a filtrar, y una ruta destino.

Depurador

Es una utilidad de línea de comandos que permite depurar aplicaciones Java. No es un entorno de características visuales, pero permite encontrar y eliminar los errores de los programas Java con mucha exactitud. Es parecido en su funcionamiento al depurador `gdb` que se incluye con las distribuciones del compilador `gcc/g++` para C/C++.

Se activa con la sintaxis:

`jdb [Opciones]`

- *Opciones*: Se utiliza para especificar ajustes diferentes dentro de una sesión de depuración. [13]

*La versión utilizada para este proyecto es `jdk-6u13` (*JDK 6 Update 13*).

4.3 PhpMyAdmin

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL.

Este proyecto se encuentra vigente desde el año 1998, siendo el mejor evaluado en la comunidad de descargas de SourceForge.net como la descarga del mes de

diciembre del 2002. Como esta herramienta corre en máquinas con Servidores *Web* y Soporte de PHP y MySQL, la tecnología utilizada ha ido variando durante su desarrollo.

Entre sus características destacan:

- Capacidad para crear, modificar y borrar tablas.
- Mantenimiento de tablas.
- Borrar, editar y añadir campos a tablas.
- Ejecutar cualquier comando SQL.
- Exportar e Importar archivos en formato XML, CSV, SQL.
- Crear gráficos PDF con el contenido de las tablas.
- Mas de 50 lenguajes.
- Multiplataforma

PhpMyAdmin, dispone de diferentes métodos de seguridad: HTTP o Cookies. El programa dispone de una amplia documentación para desarrolladores, para mejorar el programa. [21]

* La versión utilizada para este proyecto es la 3.2.2.1

4.4 Web Hosting

Es un negocio que consiste en alojar, servir, y mantener archivos para uno o más sitios *Web*. Más importante que el espacio del ordenador que se proporciona para los archivos del sitio web es la conexión rápida a Internet.

La mayoría de los servicios de *hosting* ofrecen conexiones que para una persona individual resultarían muy costosas. Usar un servicio de *hosting* permite que muchas compañías compartan el coste de una conexión rápida a Internet para el acceso a los archivos de sus sitios *Web*.

Geocities ofrece a sus visitantes registrados espacio para un sitio *Web* gratis, aunque este tipo de *hosting Web* gratuito suele ser muy básico.

Algunas compañías de *hosting* describen sus servicios como *hosting* virtual. *Hosting* virtual generalmente implica que sus servicios serán transparentes y que cada sitio *Web* tendrá su propio alojamiento de dominio y sus propias direcciones de email. En la mayoría de los casos, el *hosting* y el *hosting* virtual son sinónimos. Algunas compañías de *hosting* permiten a sus usuarios tener su propio servidor virtual, con la apariencia de que el usuario está controlando un servidor dedicado enteramente a su sitio *Web*.

Los alojamientos dedicados consisten en el uso exclusivo de todo un servidor por un único cliente, mientras que en los alojamientos compartidos varios clientes comparten un servidor. Solamente los sitios *Web* con mucho tráfico requieren el *hosting* dedicado. Muchas compañías compran sus propios servidores y los colocan en un sitio que proporcione acceso rápido a Internet. Esta práctica se llama colocación.

Servicios de hosting

Hay una amplia variedad de servicios de *hosting*. El más básico es el *hosting* de archivos (alojamiento *Web* u hospedaje *Web*), donde se pueden alojar las páginas de los sitios web y otros archivos vía ftp o una interfaz *Web*. Los archivos se muestran en la *Web* tal cual o sin mucho procesado. Muchos proveedores de Internet ofrecen este servicio de forma gratuita a sus clientes. El alojamiento *Web* es normalmente gratuito, patrocinado por anunciantes, o barato.

Normalmente el alojamiento *Web* sólo es suficiente para páginas *Web* personales. Un sitio *Web* complejo necesita un paquete más avanzado que proporcione soporte para bases de datos y plataformas de desarrollo de aplicaciones (ej. PHP, Java, y ASP. NET). Estas permiten que los clientes escriban o instalen scripts para aplicaciones como foros y formularios. Para el comercio electrónico también se requiere SSL.

El proveedor de *hosting* puede también proporcionar un interfaz *Web* (ej. panel de control) para manejar el servidor web e instalar los scripts así como otros servicios como e-mail.

Tipos de hosting

El hosting se puede dividir en varios tipos generales:

***Hosting gratuito:** el *hosting* gratuito es extremadamente limitado comparado al *hosting* de pago. Los proveedores *hosting* gratuito normalmente requieren sus propios anuncios en el sitio alojado de forma gratuita y tienen límites muy grandes de espacio y de tráfico. No obstante, la mayoría de la gente empieza en la *Web* con *hosting* gratuito.

***Hosting de imágenes:** limitado a algunos formatos de imágenes. Este tipo de *hosting* normalmente es gratuito y la mayoría requieren que el usuario se registre. La mayoría de los proveedores de *hosting* de imágenes permiten el *hotlinking*, de modo que el usuario pueda subir imágenes al servidor del proveedor para ahorrar espacio y ancho de banda.

***Hosting compartido:** es cuando un mismo servidor aloja a varios cientos de sitios *Web* de clientes distintos. Un problema en uno de los sitios en el servidor puede traer abajo al resto de los sitios. El *hosting* compartido también tiene algunas restricciones con respecto a qué se puede hacer exactamente, aunque estas restricciones no son en ninguna manera tan restrictivas como en el *hosting* gratuito.

***Hosting dedicado:** existe un servidor individual. No tienen ninguna restricción, a excepción de las diseñadas para mantener la integridad del proveedor (por ejemplo, prohibiendo sitios con contenido para adultos debido al riesgo del aumento de ataques por los hackers y las cuestiones legales). A menos que se pague una tarifa aparte al proveedor, el usuario tiene que hacerlo todo por sí mismo. Esto puede ser costoso, pues la contratación del servidor dedicado en sí es generalmente más costosa comparada al alojamiento compartido.

Web hosting virtual a menudo se utiliza en gran escala en las empresas cuyo modelo de negocio consiste en proporcionar alojamiento de sitios *Web* de bajo costo para los clientes. La gran mayoría de los servicios de alojamiento web de clientes en todo el mundo son los sitios web alojados en servidores compartidos, con el alojamiento virtual de tecnología.

Muchas empresas utilizan servidores virtuales para uso interno, donde existe una tecnología especial para mantener varios sitios *Web* independientes, tales como sitio *Web* de la extranet del cliente, empleado extranet, intranet interna, intranet para los diferentes departamentos. Si no hay problemas de seguridad en el sitio *Web* de arquitecturas, pueden ser fusionados en un único servidor, lo que reduce la gestión y gastos generales de administración y el número de servidores independientes necesarios para apoyar el negocio. [15]

Beneficios de usar un *Web hosting*

1. No se requiere comprar equipos.
2. No se requiere actualizar equipos.
3. No se requiere conexión a Internet 24 horas.
4. No se requiere personal especializado. [17]

Capítulo 5

METODOLOGIA

A continuación se describirán las fases en las cuales se desarrolló el proyecto, se enfocará principalmente en la instalación de las herramientas de apoyo para la aplicación así como la programación en su totalidad.

5.1 Instalación de Tomcat.

Como ya se señaló anteriormente, Tomcat es un servidor *Web* con soporte de *servlets* y *JSP*'s el cual ayudará a la aplicación para ser desarrollada e implementada a través de la *Web*, es por ello que es importante su correcta instalación. Se instala de la siguiente manera (para Windows únicamente):

1. Descargar el apache Tomcat de la página <http://tomcat.apache.org/>, seleccionado la versión más reciente, en estos momentos es Tomcat 6x, navegar a las Distribuciones binarias y seleccionar la versión *Core* con extensión zip. Descomprimir el apache Tomcat, puede ser en cualquier lugar pero preferentemente en el disco duro de C: \. Renombra el directorio a únicamente Tomcat.
2. Descargar el JDK de Java de la página <http://java.sun.com/>, entrar en Productos y seleccionar *Java SE* (es la versión estándar de Java), navegar a *downloads*, dentro de esta página descargar *JDK 6 Update 13*. Instalar el JDK dando doble click sobre el icono y seguir las instrucciones del asistente de instalación.
3. En la carpeta lib de Tomcat copiar los siguientes archivos: el-api.jar, servlet-api.jar, jsp-api.jar y tomcat-dbcp.jar. Con la finalidad de que sean accesibles para el JDK. Incluirlos en el directorio del JDK, es decir C:\ Archivos de programa\Java\jdk1.6.0_13\jre\lib\ext.
4. Establecer la variable de entorno JAVA_HOME la cual debe apuntar al directorio donde se instaló el JDK. Para ello navegar al directorio bin de Tomcat y editar el archivo Catalina.bat. dentro del archivo encontrará líneas

que empiecen con la palabra *rem* después de la última línea que contenga dicha palabra agregar set JAVA_HOME=C:\Archivos de programa\Java\jdk1.6.0_13 (la última carpeta dependerá del JDK que esté instalado en este caso es la versión 13).

5. Posteriormente dar doble click en el archivo *starup.bat* del directorio bin de Tomcat, puede o no aparecer un mensaje de Windows dar en el botón Desbloquear, a continuación aparecerá una ventana de MS DOS donde mostrará que se a inicializado correctamente el servidor con el siguiente mensaje: *INFO: Server startup in 2298 ms*. Entrar al navegador e introducir la dirección <http://127.0.0.1:8080/>, si aparece una página del servidor Tomcat quiere decir que se ha instalado correctamente el servidor. Para parar el servidor Tomcat dar doble click en el archivo *shutdown.bat* del directorio bin de Tomcat.
6. Apagado el servidor Tomcat se realizaron algunas configuraciones extras. Entra a la carpeta conf de Tomcat editar el archivo *tomcat-users* para tener un usuario y un rol como se muestra a continuación:

```
<tomcat-users>
  <role rolename="manager"/>
  <user username="admin" password="admin" roles="manager"/>
</tomcat-users>
```

Esta configuración es con la finalidad de inicializar y terminar una aplicación sin tener que finalizar el servidor Tomcat.

7. Inicializar de nuevo el servidor Tomcat y entrar a Tomcat *Manager*, aparecerá una ventana pidiendo el nombre de usuario y contraseña, introducir la información contenida entre las etiquetas *<tomcat-users>* del archivo anteriormente mencionado.
8. Posteriormente se enlistan todas las aplicaciones que se están ejecutando en ese momento, cada una tiene las opciones de: Arrancar, Parar, Recargar y Replegar (Borrar).

9. Realizar también la configuración para cuando se modifique un *servlet* en el disco duro, Tomcat sea capaz de detectar este cambio y lo vuelva cargar en memoria. Entrar en el directorio *conf* y abrir el archivo *context*, dentro de la etiqueta `<context>` introducir `<context reloadable="true">`. Ahora de nueva cuenta parar el servidor Tomcat y reinicializarlo. [24]

5.2 Sistema de archivos de la aplicación en Tomcat

Para tener un correcto funcionamiento, estructura y ubicación de cada archivo de la aplicación, se optó por la estructura de archivos MVC (Modelo – Vista - Controlador), como se muestra en la figura 5.1:



Figura 5.1 Estructura de archivos MVC

Es por ello que se tiene una jerarquía de archivos dentro del directorio Tomcat en la carpeta *webapps* se encuentra la carpeta del “Gestión de Exámenes en Línea” (es el nombre de la aplicación), y dentro de ella se tiene contenida la siguiente estructura de archivos:

- a) Archivos con extensión *jsp*, *css*, *js*, *html* y una carpeta llamada *images*. Los cuales proporcionarán las interfaces y comunicaciones con los *servlets*.
- b) Carpeta WEB-INF. Aquí se tienen diferentes archivos y carpetas:
 1. Web.xml. Donde se nombran y mapean los *servlets*.

2. Carpeta desarrollo. Donde se encuentra el archivo `build.xml` para compilar todos los archivos `.java` y `servlets` de la aplicación. También existe una carpeta llamada `src`, que contiene la carpeta `diplomado` y ésta a su vez los archivos `java` de la aplicación.
3. Carpeta `classes`. Contiene otra carpeta llamada `diplomado` donde tiene todos los archivos `.class` resultado de la compilación de los archivos `.java` de la aplicación antes mencionados. [24]

5.3 JEdit

JEdit es un editor de texto para programadores distribuido bajo los términos de la Licencia Pública General de *GNU*. Está escrito en Java y se ejecuta en *Windows*, *Linux*, *Mac OS X* y otros sistemas operativos que dispongan de la máquina virtual Java.

Dispone de docenas de *plugins* para diferentes áreas de aplicaciones. Soporta de forma nativa el coloreado de la sintaxis para más de 130 formatos de archivos. También se puede incluir nuevos formatos de forma manual utilizando archivos *XML*. *JEdit* soporta UTF-8 y otros formatos de codificación del texto.

JEdit es mejor que muchas herramientas caras de desarrollo por su amplitud de características y de fácil uso, fue lanzado como software libre, siempre bajo los términos de la *GPL 2.0*.

Algunas de las características incluyen:

- * Escrito en Java, por lo que se ejecuta en *Mac OS X*, *OS / 2*, *Unix*, *VMS* y *Windows*.
- * Construido en lenguaje de macros; arquitectura extendible de *plugin*. Docenas de macros y complementos disponibles.
- * *Plugins* pueden ser descargados e instalados desde *JEdit* utilizando el "*plugin manager*".

- * Auto guión, y resalto de sintaxis para más de 130 idiomas.
- * Soporta un gran número de codificaciones de caracteres incluyendo UTF8 y Unicode.
- * *Word wrap*. El ajuste de línea.
- * Altamente configurable y personalizable.

Éste es el editor de texto que se implementará para esta aplicación y podrá ser descargado de la página <http://www.jedit.org/>. [18]

Configuración de JEdit.

Para la configuración de *JEdit* se necesita descargar el *apache ant* de la pagina <http://ant.apache.org/>, la cual es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción (*build*).

Esta herramienta, hecha en Java, tiene la ventaja de no depender de las órdenes de *shell* de cada sistema operativo, sino que se basa en archivos de configuración *XML* y clases Java para la realización de las distintas tareas. La versión que actualmente se maneja es la *Ant 1.7.1*. Ya descargada descomprimirla en cualquier lugar de *Windows*, en este caso se descomprimió en *C:\Documents and Settings\GONS\Escritorio*.

Posteriormente ya descargados e instalados tanto el *JEdit* como el *apache ant*, iniciamos dando doble click al icono de *JEdit*. Damos click en el menú *Plugins*, seleccionamos la opción *Plugin manager*, abrimos la pestaña *Install* y seleccionamos los siguientes *Plugins*:

- *Ant Farm*. Permite a los desarrolladores construir proyectos dentro de *jEdit*. Proporciona una agradable interfaz gráfica para administrar múltiples archivos de construcción(*build*).

-
- *AntPlugin*. Es una herramienta Java basada en la construcción de archivos.
 - *Background*. Herramienta para poner un fondo ya sea foto o cualquier imagen en *JEdit*.
 - *Beauty*. Herramienta para diferentes formatos de archivo, consolida los formatos en un API estándar de manera que se pueden modificar de un modo general.
 - *BufferTabs*. Añade un conjunto de pestañas en el panel del área de texto.
 - *Clipper*. Proporciona acceso a las bibliotecas de clips de texto (por ejemplo, las etiquetas HTML, programación construcciones, etc.).
 - *Code2HTML*. Transforma el archivo fuente actual de *JEdit* en un archivo HTML equivalente.
 - *Common Controls*. Proporciona un conjunto común de componentes swing destinados a ser utilizados por otros *JEdit plugins*.
 - *Console*. Ofrece dos *shells* de línea de comandos, uno para el sistema y otra para *Beanshell*. Asimismo, proporciona una interfaz para otros *plugins* para añadir arbitraria *shells* a la consola de *plugin*.
 - *ErrorList*. Proporciona acceso con un solo click, a los errores generados por *EditBus*.
 - *JavaSideKick*. Código Java del navegador, provee apoyo completo de sintaxis para todos los Java 1.5 y anteriores.
 - *JTidyPlugin*. Integra el HTML en *JEdit*, comprueba si existen errores en el código HTML / XML y los corrige.
 - *LatestVersionCheck*. Revisa cual es la última versión de *JEdit* y te pregunta si la quieres descargar.
 - *Project Viewer*. Proporciona una interfaz para la gestión de un "proyecto", es decir, relacionados con un grupo de archivos fuente.
 - *QuickNotepad*. Proporciona un lugar para escribir notas.
 - *SideKick*. Proporciona un área de trabajo con características de lenguaje de alto nivel.
 - *Spell Check*. Corrector ortográfico. Realiza la corrección ortográfica de texto en un buffer de *JEdit*.

- *VoxSpell*. Comprueba la ortografía del idioma Inglés mientras se está escribiendo, subrayando palabras erróneas cuando se encuentran.
- *XercesPlugin*. Este *plugin* empaqueta la documentación de Xerces-J versión 2, como un *plugin JEdit*.
- *XML* Establece una estructura de árbol del navegador para editar XML, HTML, CSS y archivos Javascript , y la finalización de archivos XML, HTML y CSS.

Posteriormente dar click en el botón *Install* para instalar los *Plugins* seleccionados. Ya instalados los *Plugins*, se selecciona nuevamente en el menú *Plugins* y se da click en *Plugin Options*. En la pestaña de *Ant Farm* se selecciona *Build Options*, en la opción *Build execution method* se selecciona *Run Ant targets using an external script/build file* y se tecléa la dirección donde se encuentra el archivo ant.bat, es decir (en este caso) C:\Documents and Settings\GONS\Escritorio\apache-ant-1.7.1\bin\ant.bat, damos click en *Apply*.

Y así se da por terminada la configuración de *JEdit* para que se desarrolle y compile el proyecto en sus carpetas respectivas anteriormente mencionadas, basadas en la estructura de archivos MVC.

Capítulo 6

RESULTADOS

Ya instalado el servidor Tomcat y los programas mencionados anteriormente, se desarrolló la aplicación *Web* empezando por las interfaces y posteriormente con la funcionalidad del sistema, como se detalla a continuación.

6.1 Interfaces

Las interfaces que se implementan en este proyecto son tres principales: Página principal, Profesores y Alumnos.

* Interfaz Página Principal.

En esta interfaz en su parte superior comprende un menú con diferentes opciones: Inicio (es la Pagina Principal), Profesores (donde existe una forma de registro para el profesor), Alumnos (donde existe una forma de registro para el alumno), Noticias (noticias relevantes sobre nuevos exámenes creados) y Acerca de (es una descripción generalizada de la aplicación). Como se muestra en la figura 6.1:

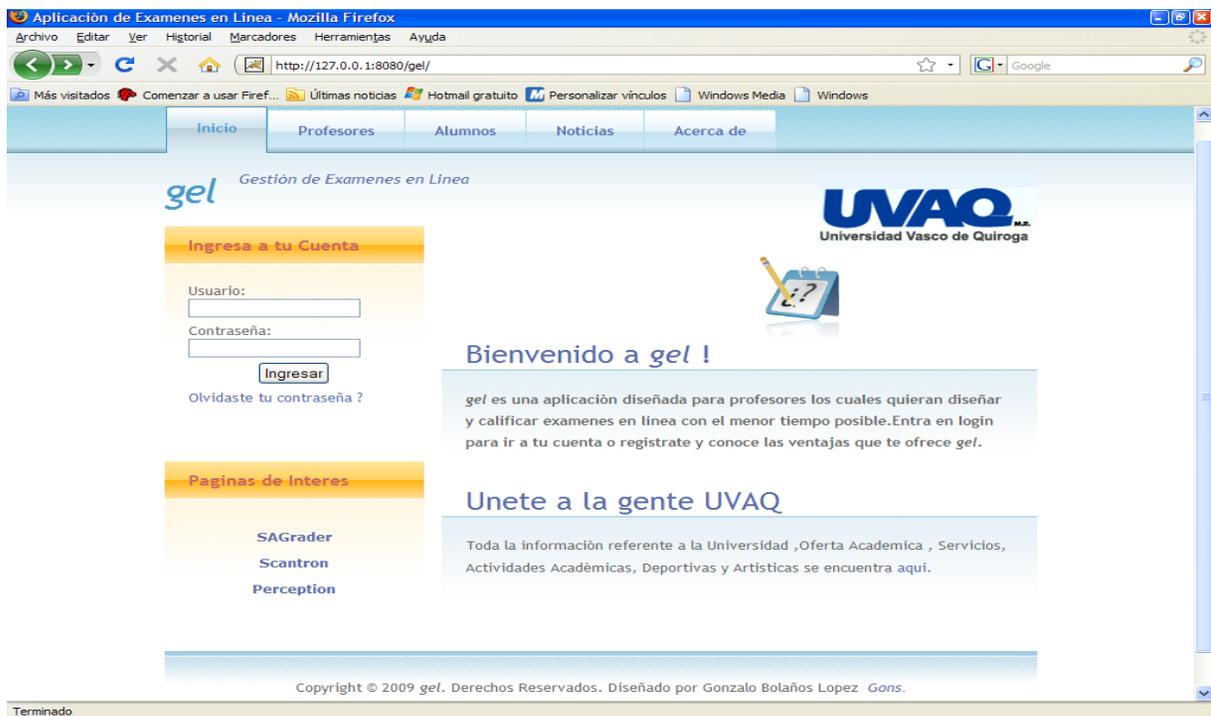


Figura 6.1 Interfaz Pagina Principal

En la parte inferior izquierda existe una forma de ingreso con dos campos: nombre de usuario y contraseña, para que el usuario pueda ingresar a su cuenta ya sea profesor o Alumno. De lado derecho de la forma de registro se comprende una pequeña descripción de lo que realiza la aplicación, así como la opción para visitar la Página principal de la UVAQ para mayor información académica, actividades artísticas, deportivas etc.

Por debajo de la forma de ingreso existen tres hipervínculos a páginas donde existen aplicaciones que desarrollan exámenes, en caso de que el usuario esté interesado en desarrollar otro tipo diferente de exámenes que no comprende esta aplicación. En la parte inferior de la interfaz se encuentra la leyenda de Derechos Reservados y la opción de mandar un correo electrónico al *webmaster*.

Al dar click en el menú Profesores se abre otra página la cual en vez de tener información de la aplicación y la referencia para visitar la página principal de la UVAQ, contendrá una forma de registro con los siguientes campos: Nombre, Dirección, Teléfono, Email, Usuario, Contraseña y Clave (la cual será proporcionada por la UVAQ, la clave predeterminada es “profuvaq”). Todo esto para que el Profesor ingrese su información y pueda registrarse para utilizar la aplicación. Como se muestra en la figura 6.2:

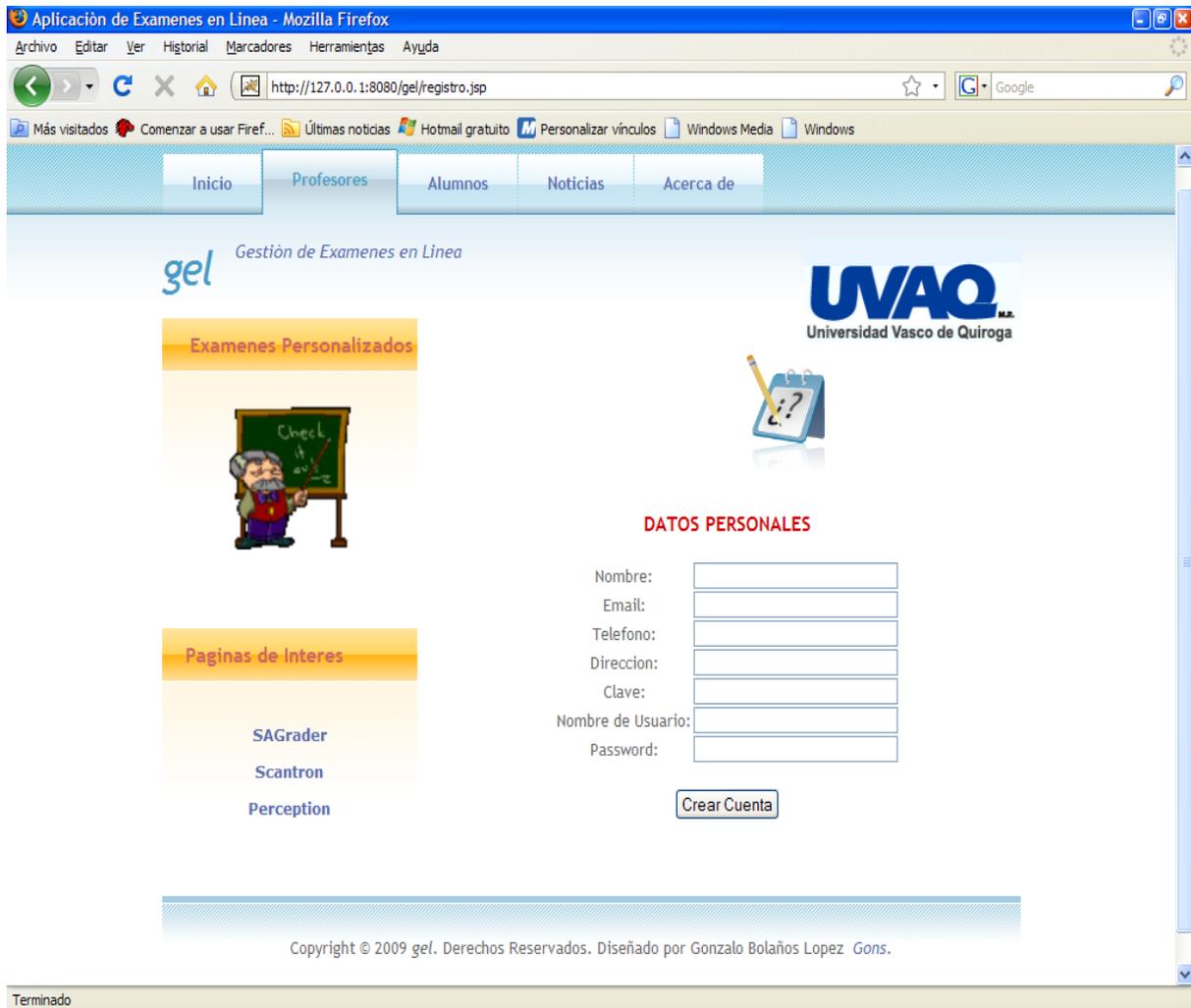


Figura 6.2 Interfaz Menú Profesores

Al dar click en el menú Alumnos se abre otra pagina igual a la mencionada anteriormente solamente que la forma de registro será para que ingrese la información el alumno, se registre y pueda utilizar la aplicación. Esta forma de registro contendrá los siguientes campos: Nombre, Dirección, Teléfono, Email, Carrera, Semestre, Usuario, Contraseña y Clave (la cual será proporcionada por la UVAQ, la clave predeterminada es “alumuvaq”).

La interfaz es como se muestra en la figura 6.3:

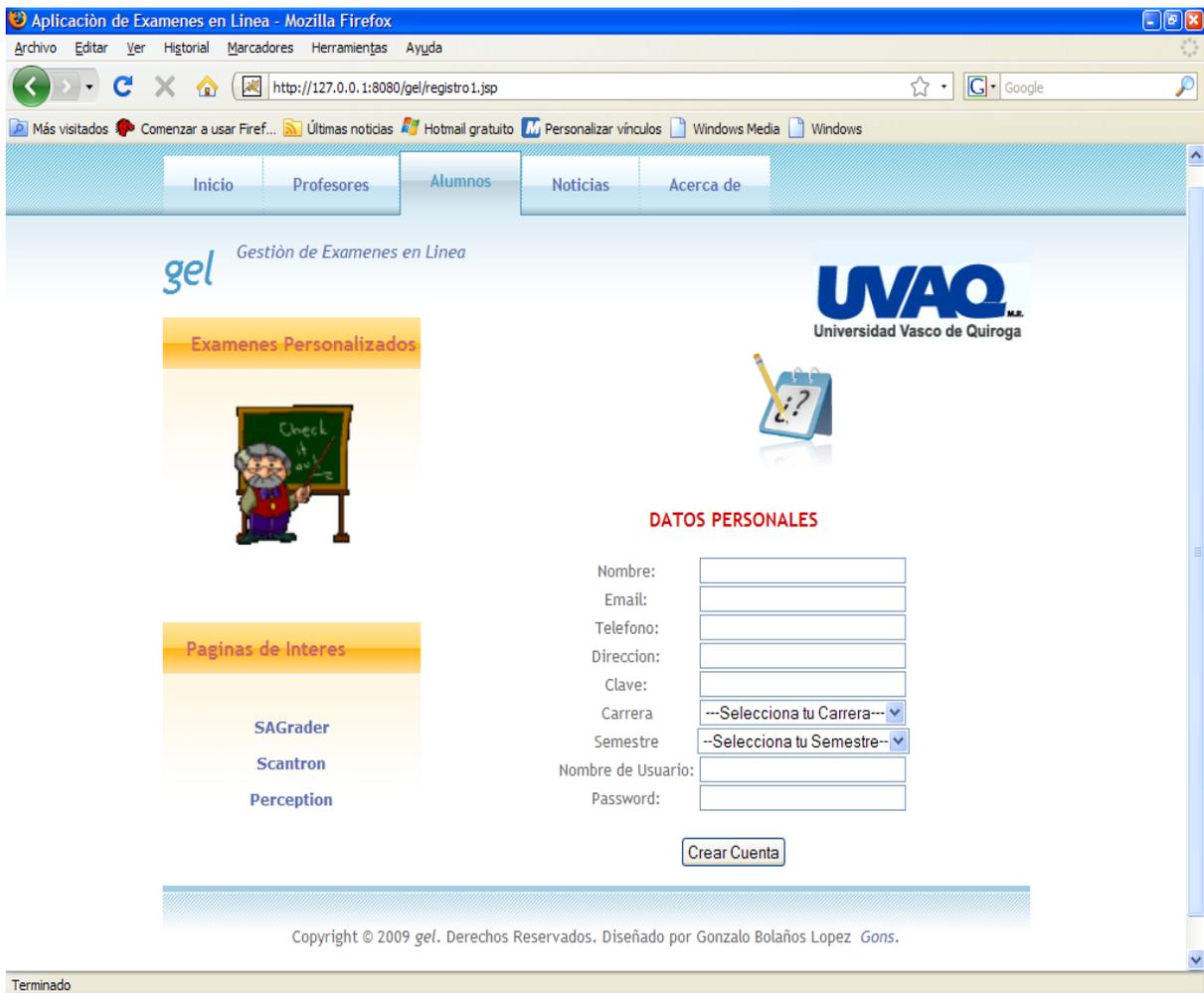


Figura 6.3 Interfaz Menú Alumnos

En el menú Noticias se abre otra pagina igual a la mencionada anteriormente, comprende noticias acerca de exámenes nuevos realizados por profesores para que el alumno revise que exámenes le corresponden contestar.

Como se muestra en la figura 6.4:

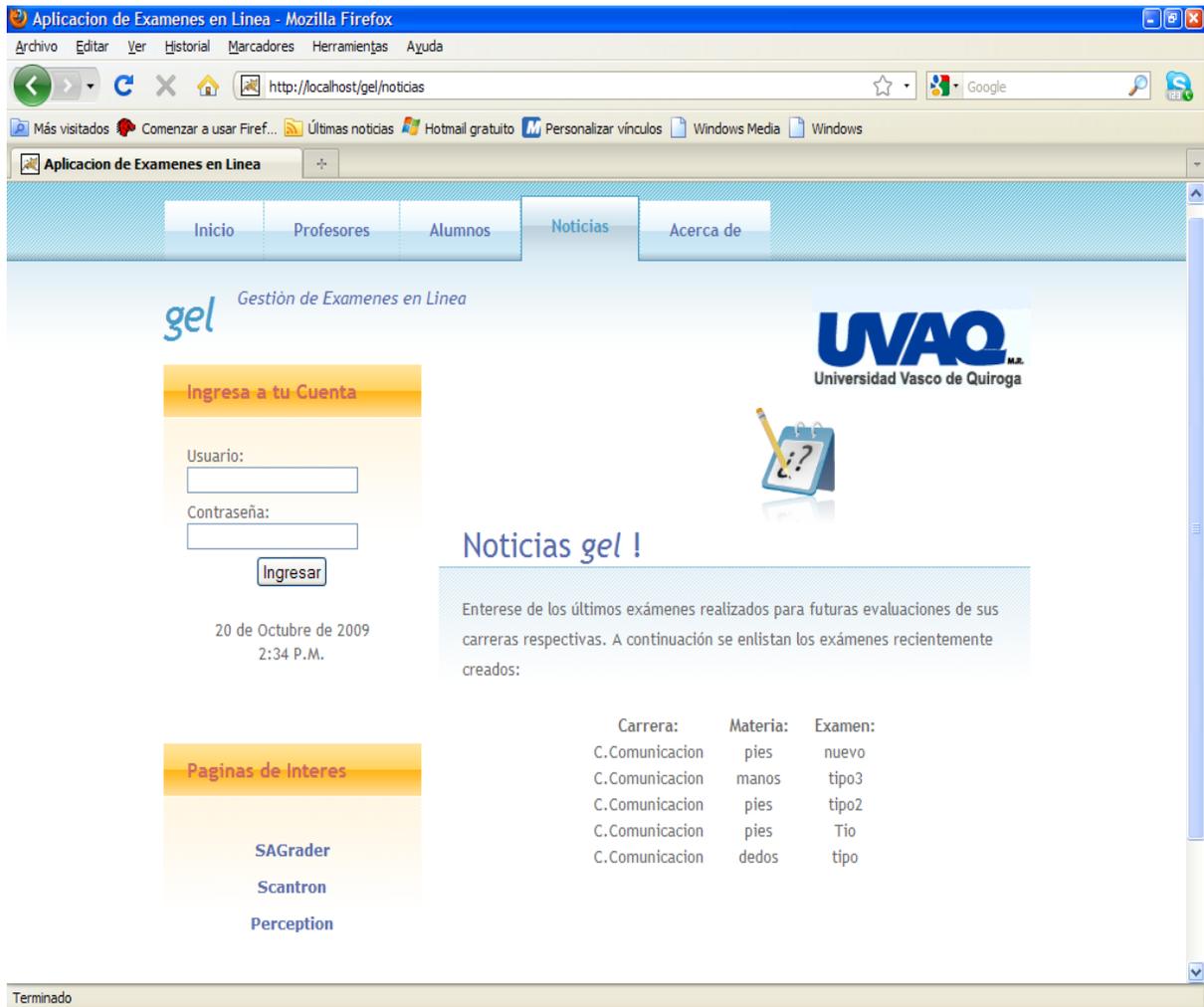


Figura 6.4 Interfaz Menú Noticias

En el menú 'Acerca de' se abre otra página igual a la anterior, varía únicamente en que proporciona información generalizada de las funciones y características que lleva a cabo la aplicación, para que el usuario se informe adecuadamente de lo que puede realizar al registrarse para utilizar la aplicación.

Como se muestra en la figura 6.5:

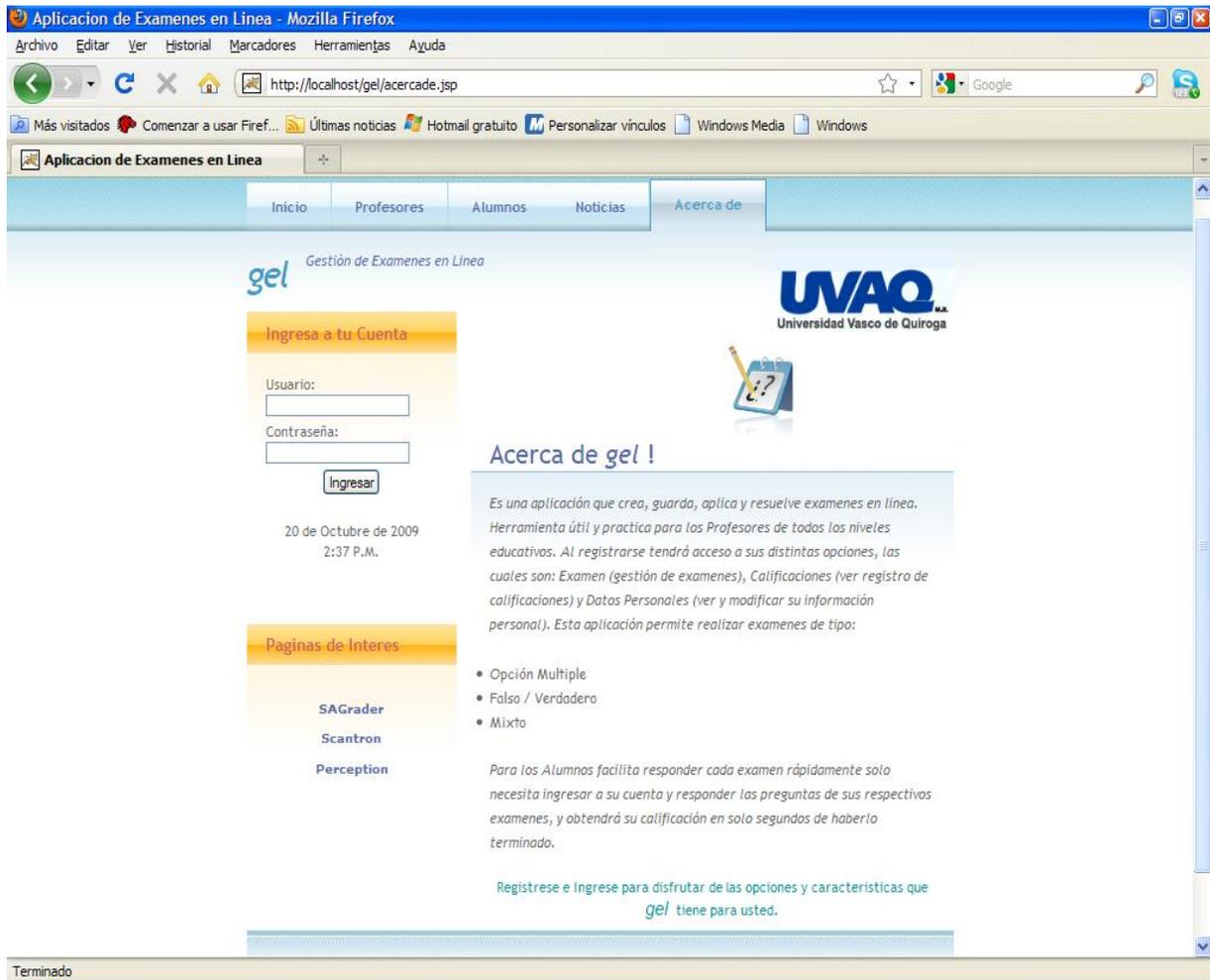


Figura 6.5 Interfaz Menú Acerca de

Los archivos que contienen estas interfaces se enlistan a continuación:

- encabezado.html. Contiene el menú de la Pagina Principal. Resalta la opción de Profesores.
- encabezado1.html. Contiene el menú de la Pagina Principal. Resalta la opción de Inicio.
- encabezado2.html. Contiene el menú de la Pagina Principal. Resalta la opción de Alumnos.
- encabezado3.html. Contiene el menú de la Pagina Principal. Resalta la opción de Noticias.

-
- encabezado4.html. Contiene el menú de la Pagina Principal. Resalta la opción de Acerca de.
 - encabezado5.html. Contiene el menú de la Pagina Principal. Sin ninguna opción resaltada.
 - encabezado6.html, encabezado7.html, encabezado8.html, encabezado9.html. Contiene el menú de la Pagina Principal. Sin ninguna opción resaltada para distintas funcionalidades de la aplicación.
 - pi.html. Contiene la Bienvenida a la página con una pequeña descripción de la aplicación, el acceso directo a la página principal de la UVAQ, la leyenda Derechos Reservados y la opción de mandar un correo al *webmaster*.
 - pie.html. Contiene la leyenda Derechos Reservados y la opción de mandar un correo al *webmaster*.
 - pie1.html. Contiene la Bienvenida a la página con una pequeña descripción de la aplicación, el acceso directo a la página principal de la UVAQ, la leyenda Derechos Reservados y la opción de mandar un correo al *webmaster*.
 - pie2.html. Contiene la Bienvenida a la página con una pequeña descripción de la aplicación, el acceso directo a la página principal de la UVAQ, la leyenda Derechos Reservados y la opción de mandar un correo al *webmaster*.
 - pie3.html. Contiene Noticias referentes a exámenes recientemente creados por profesores.
 - pie4.html. Contiene información generalizada acerca de las opciones y características de la aplicación.
 - pie5.html. Contiene la leyenda Derechos Reservados y la opción de mandar un correo al *webmaster*.
 - index.jsp. Incluye la forma de ingreso la cual redirecciona los datos ingresados al archivo LoginServlet.java. Incluye tanto encabezado1.html como pie1.html.
 - registro.jsp. Incluye la forma de registro la cual redirecciona los datos ingresados al archivo ProfeServlet.java para registrar al profesor. Incluye tanto encabezado.html como pie.html.

-
- registro1.jsp. Incluye la forma de registro la cual redirecciona los datos ingresados al archivo AlumServlet.java para registrar al alumno. Incluye tanto encabezado2.html como pie.html.
 - carrerain.jsp. Muestra el formulario del alumno y la leyenda “Seleccione una carrera” en caso de que el alumno no haya señalado una. Incluye los archivos encabezado8.html y pie5.html.
 - cinvalidada.jsp. Muestra el formulario del profesor con la leyenda “Clave errónea, Ingrése de nuevo” en caso de que la clave no corresponda a la almacenada en el servlet de registro. Incluye los archivos encabezado5.html y pie5.html.
 - cinvalidada1.jsp. Muestra el formulario del alumno con la leyenda “Clave errónea, Ingrése de nuevo” en caso de que la clave no corresponda a la almacenada en el servlet de registro. Incluye los archivos encabezado8.html y pie5.html.
 - invalidado.jsp e invalidado1.jsp. Muestra la leyenda “” No se encontró el usuario, Regístrese” en caso de que no haya un usuario en la base de datos ya sea profesor o alumno. Incluye los archivos encabezado9.html y pie5.html.
 - logout.jsp. Redirecciona a la página principal cuando el usuario sale de su sesión respectiva.
 - noticias.jsp. Incluye noticias relevantes acerca de nuevos exámenes creados por profesores para que el alumno revise que exámenes tiene que realizar. Incluye tanto encabezado3.html como pie3.html.
 - acercade.jsp. Incluye información generalizada acerca de las funciones y características de la aplicación. Incluye tanto encabezado4.html como pie4.html.
 - error.jsp. Muestra la página principal, aquí regresan todas las sesiones que no estén activas. Incluye los archivos encabezado1.html y pi.html.
 - semestrein.jsp. Formulario con los datos de registro del alumno y la leyenda “Ingresa el semestre”. Incluye los archivos encabezado8.html y pie5.html.
 - urepetido.jsp. Formulario con los datos de registro del profesor y la leyenda “Usuario ya registrado, Ingrese otro distinto”. Incluye los archivos encabezado5.html y pie5.html.

- urepetido1.jsp. Formulario con los datos de registro del alumno y la leyenda “Usuario ya registrado, Ingrese otro distinto”. Incluye los archivos encabezado8.html y pie5.html.
- validado.jsp. Muestra la leyenda “Usuario registrado” cuando se ingresaron los datos de forma correcta tanto para el profesor como el alumno. Incluye los archivos encabezado7.html y pie5.html.
- default.css. Contiene las especificaciones de la vista de las Páginas como: Imágenes, tipo de letra, etc.

* Interfaz Profesores.

A esta interfaz es donde los profesores ingresan una vez introducido sus datos de nombre de usuario y contraseña. Se encuentra la página de bienvenida donde se le especifican al profesor los tipos de exámenes que puede crear y se le invita a utilizar las funciones que tiene la aplicación, las cuales se encuentran en la parte superior derecha de la interfaz, es el menú de Opciones. La interfaz es como se muestra en la figura 6.6:



Figura 6.6 Interfaz Profesores

El menú Opciones comprende:

- Examen. Para creación y finalización de exámenes.
- Calificaciones. Resultados de exámenes de alumnos respectivos a cada maestro.
- Datos Personales. Registro y / o modificación de datos personales.
- Salir. Para salir de la aplicación, Regresar a la Pagina principal.

En la parte inferior a éste menú de Opciones se encuentra otro menú de Actualizaciones en el cual se proporciona información de los exámenes creados, donde se puede ver toda la información relevante a cada uno en particular.

En la parte inferior de esta interfaz se encuentra la leyenda de Derechos Reservados y un hipervínculo para contactarse con el *webmaster*. Al abrir el menú de Examen se encuentran dos opciones: crear y abrir un examen. En la opción crear se abre una nueva página en donde se tiene que llenar un formulario con los siguientes datos:

- * Nombre del examen
- * Carrera
- * Nombre de la materia
- * Semestre
- *Tipo de examen

La interfaz es como se muestra en la figura 6.7:

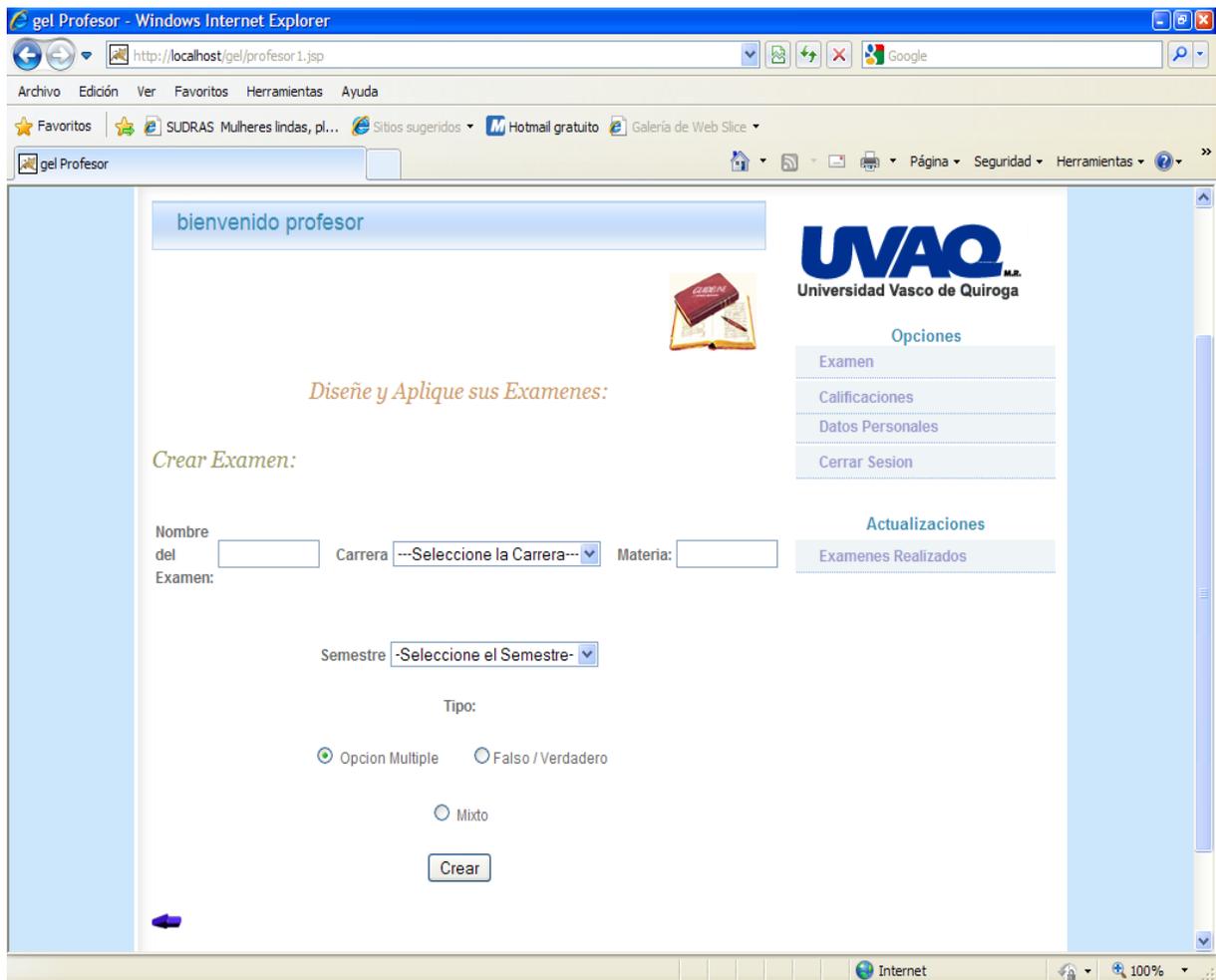


Figura 6.7 Interfaz Crear Examen

En la opción abrir se realiza una búsqueda de exámenes que no hayan sido finalizados por el profesor a causa de falta de tiempo o por otras circunstancias que hayan impedido la finalización de un examen en particular.

Dependiendo si es la primera vez que ha ingresado al sistema mostrará un mensaje de que no existen registros o si ya existen exámenes finalizados, mostrará su información correspondiente para que sea finalizado.

La interfaz es como se muestra en la figura 6.8:

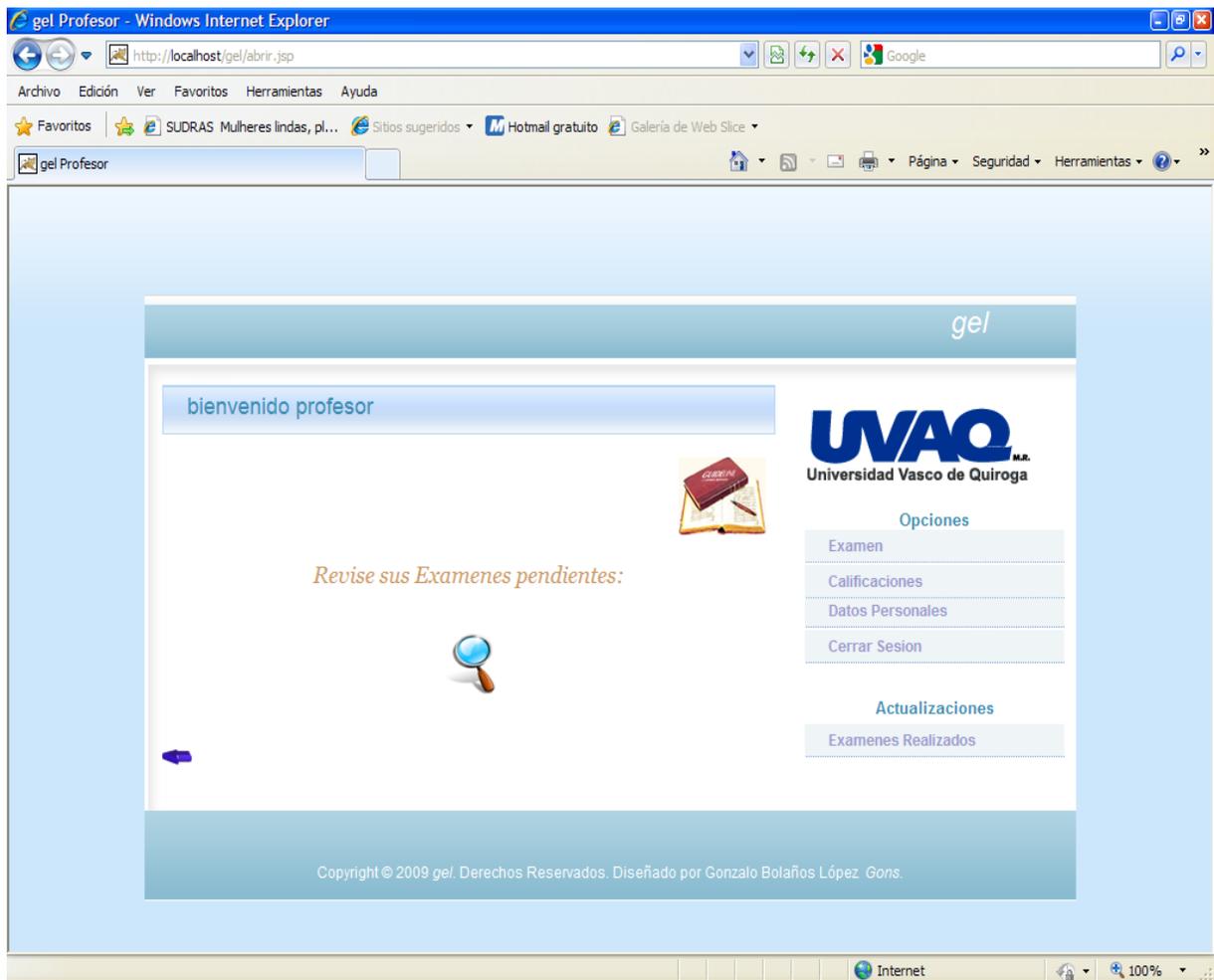


Figura 6.8 Interfaz Abrir Examen

Al seleccionar la opción de examen Falso / Verdadero se abrirá una ventana con un campo de texto para ingresar la pregunta, otro campo para el valor de la pregunta y su respuesta correspondiente.

La interfaz es como se muestra en la figura 6.9:

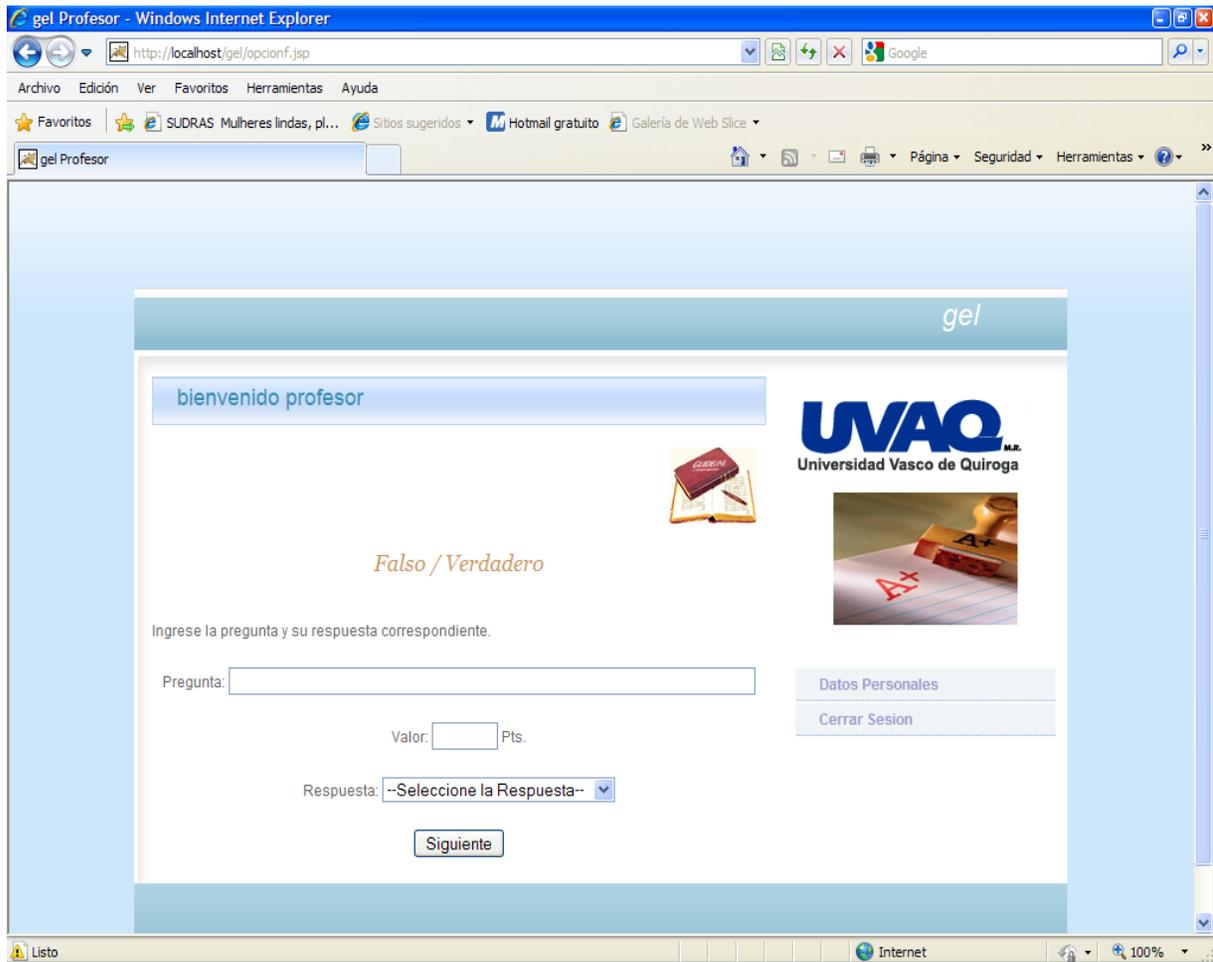


Figura 6.9 Interfaz Examen Falso / Verdadero

Al seleccionar la opción de examen Opción Múltiple se abrirá una ventana con un campo de texto para ingresar la pregunta y otro campo para su valor.

La interfaz es como se muestra en la figura 6.10:

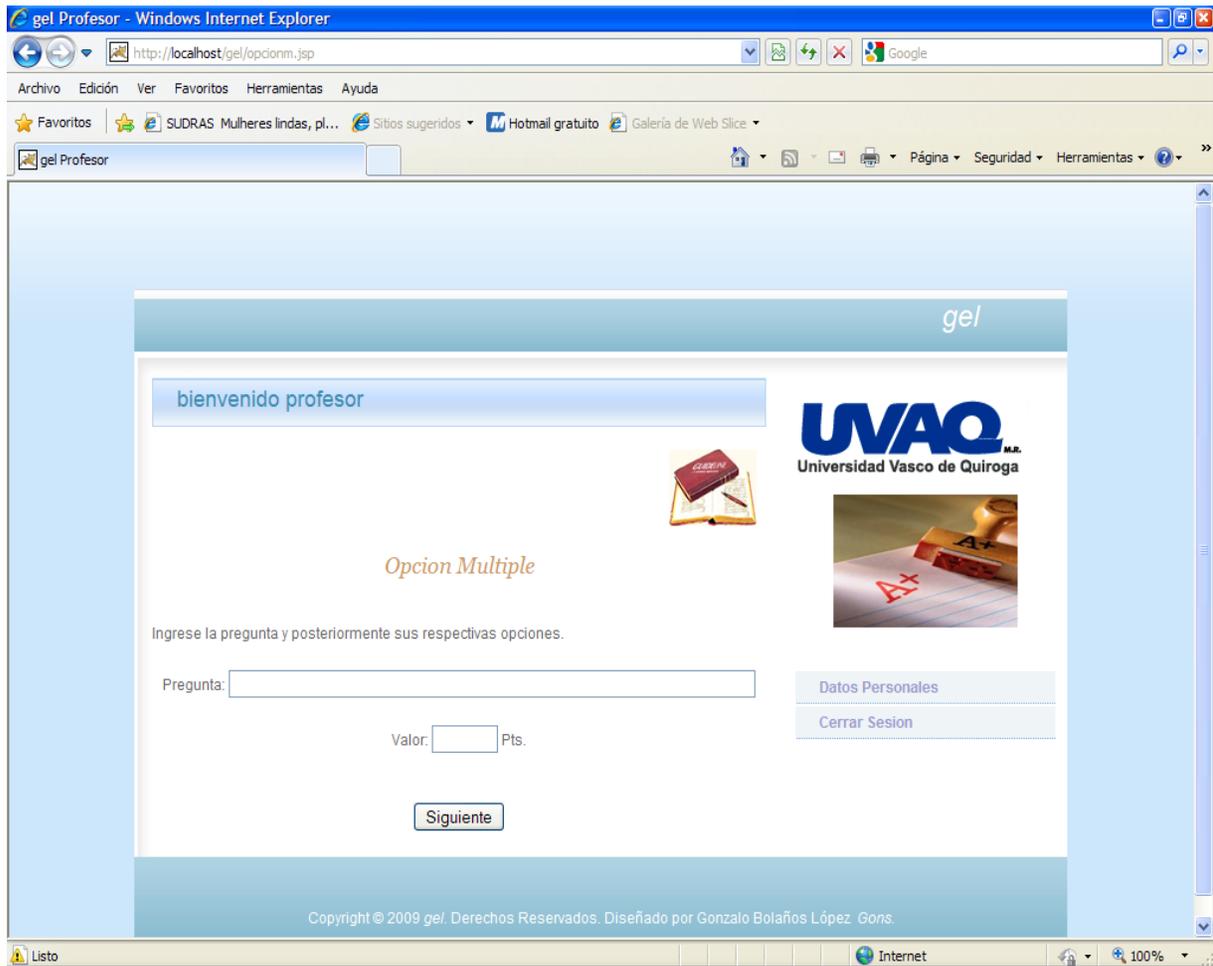


Figura 6.10 Interfaz Examen Opción Múltiple

Posteriormente se ingresaran las opciones que el usuario requiera para su pregunta correspondiente.

La interfaz es como se muestra en la figura 6.11:

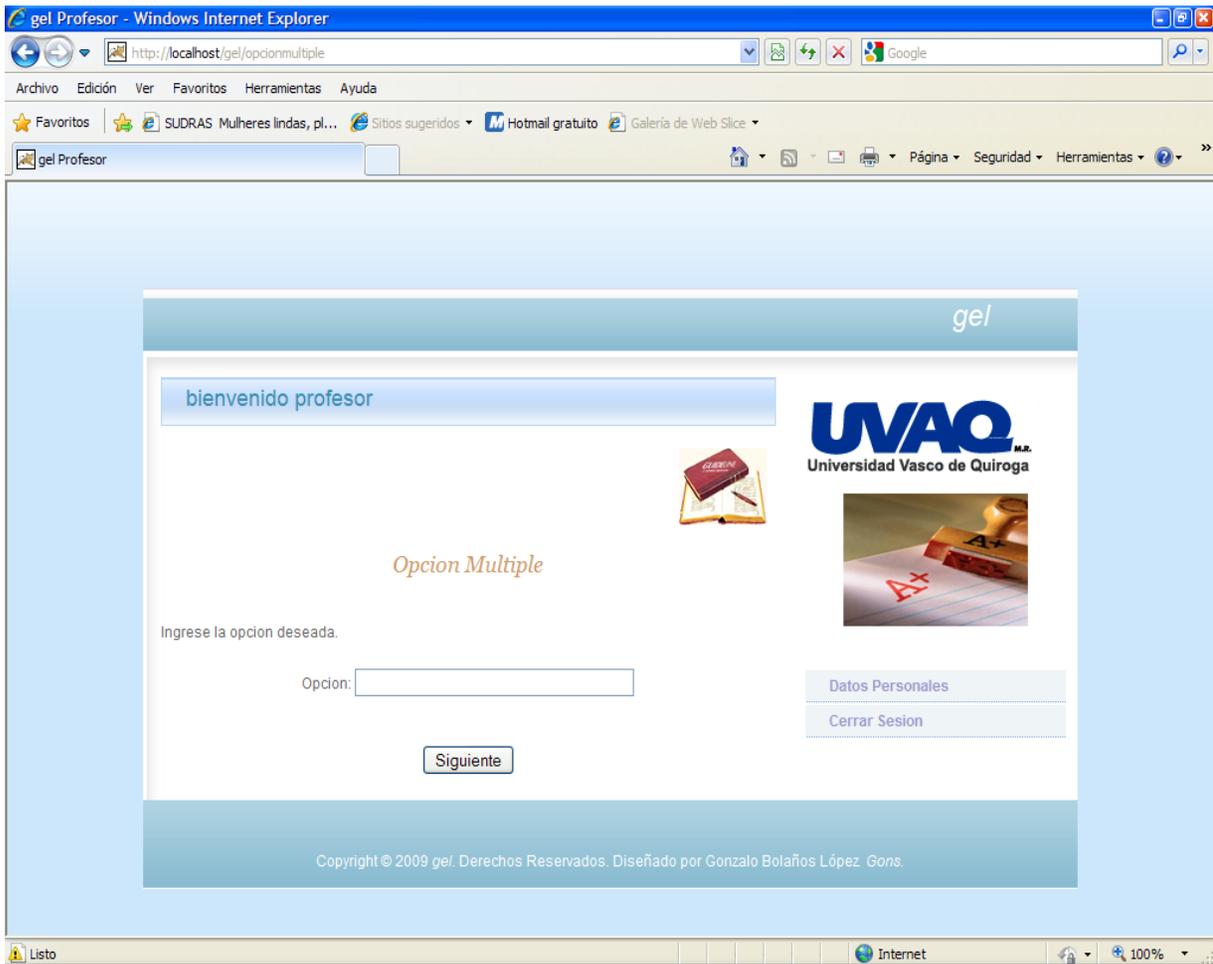


Figura 6.11 Interfaz Opciones

Y por ultimo ingresará la respuesta correspondiente a la pregunta, en base a las opciones anteriormente ingresadas.

La interfaz es como se muestra en la figura 6.12.

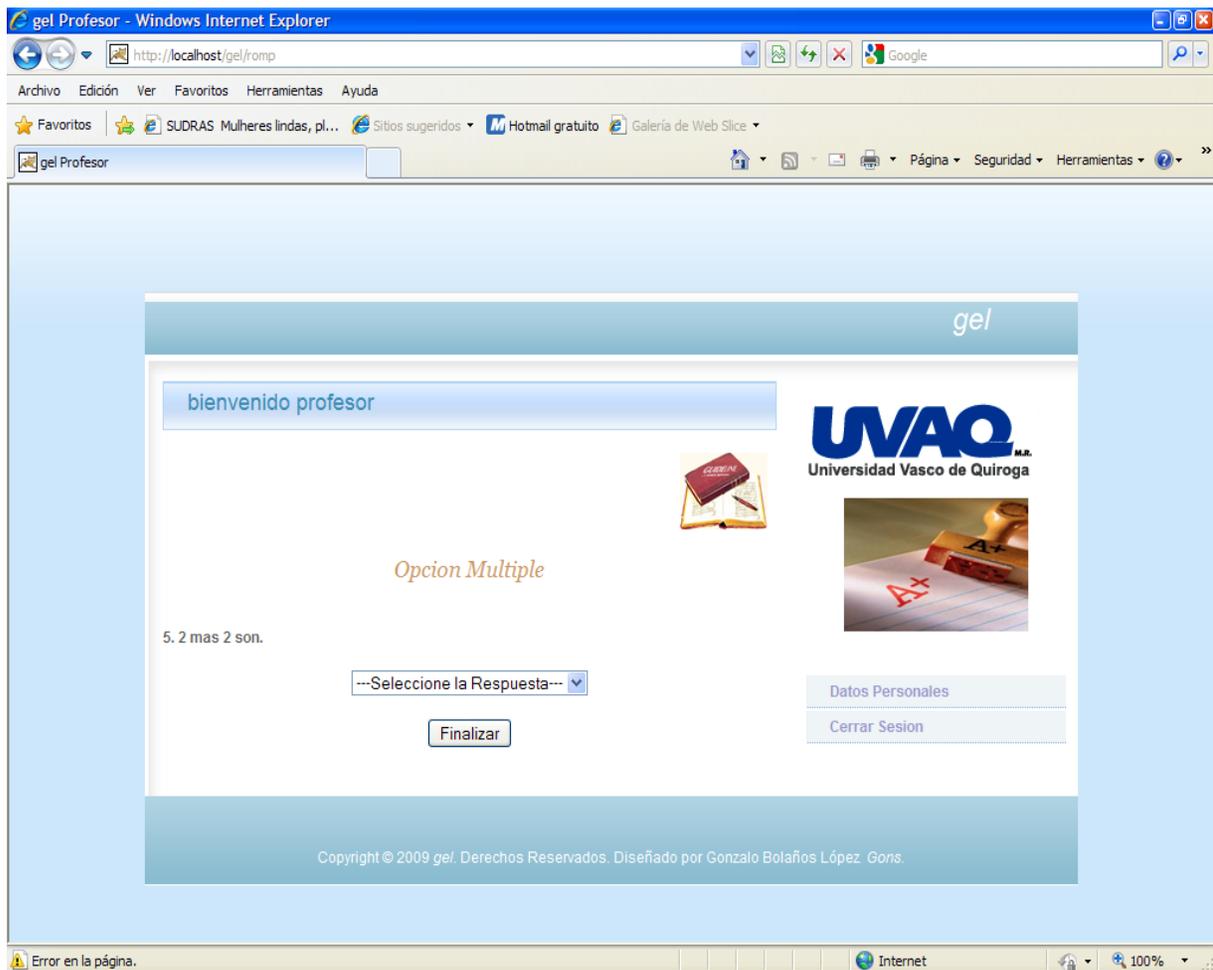


Figura 6.12 Interfaz Respuesta Opción

Al seleccionar la opción de examen Mixto se abrirá una ventana con dos formularios uno para Falso / Verdadero y otro para Opción Múltiple, el usuario elegirá que tipo de pregunta ingresará primero.

La interfaz es como se muestra en la figura 6.13:

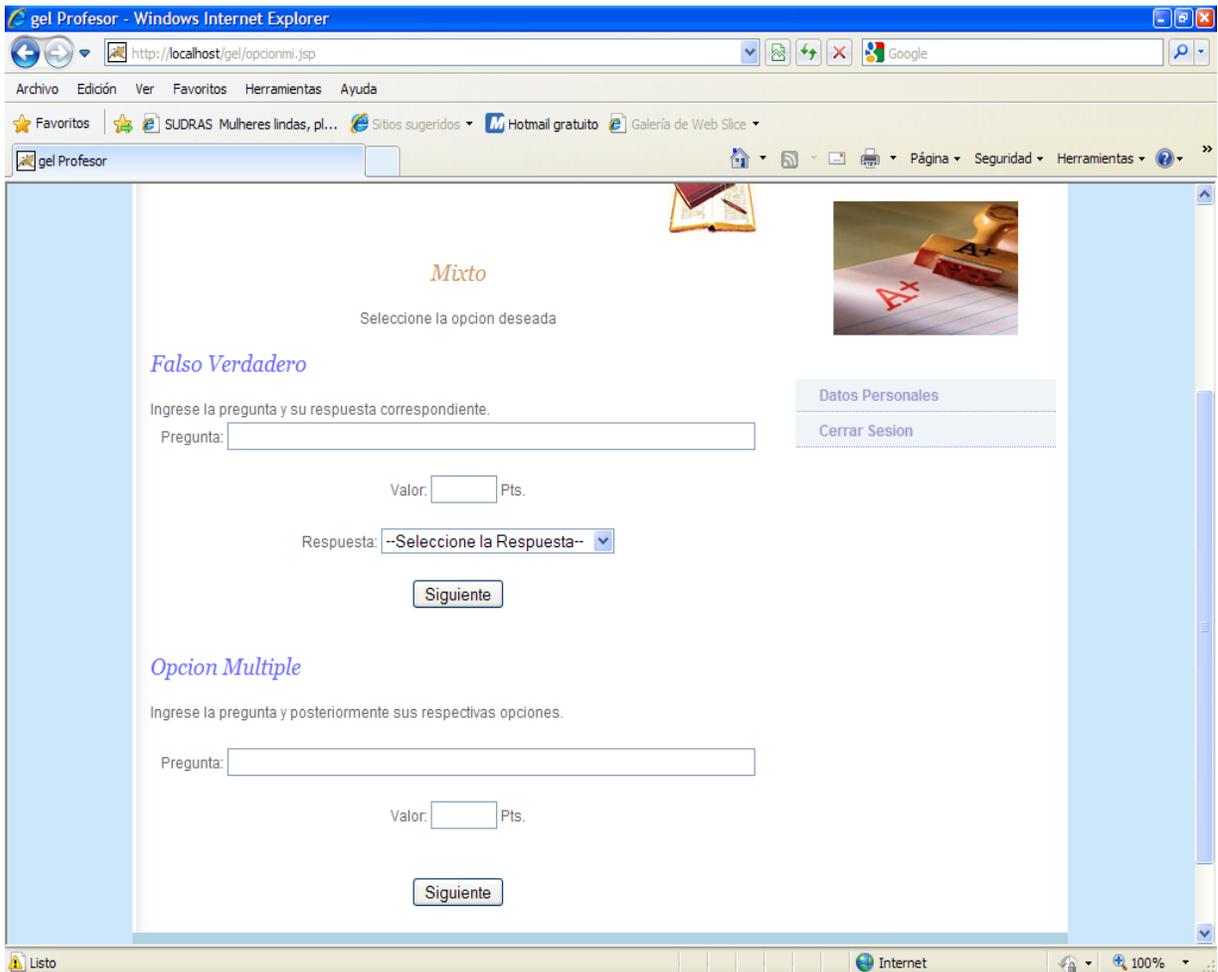


Figura 6.13 Interfaz Examen Mixto

Finalmente se le pedirá al usuario que ingrese la fecha de inicio y terminación del examen así como su tiempo de evaluación.

La interfaz es como se muestra en la figura 6.14:

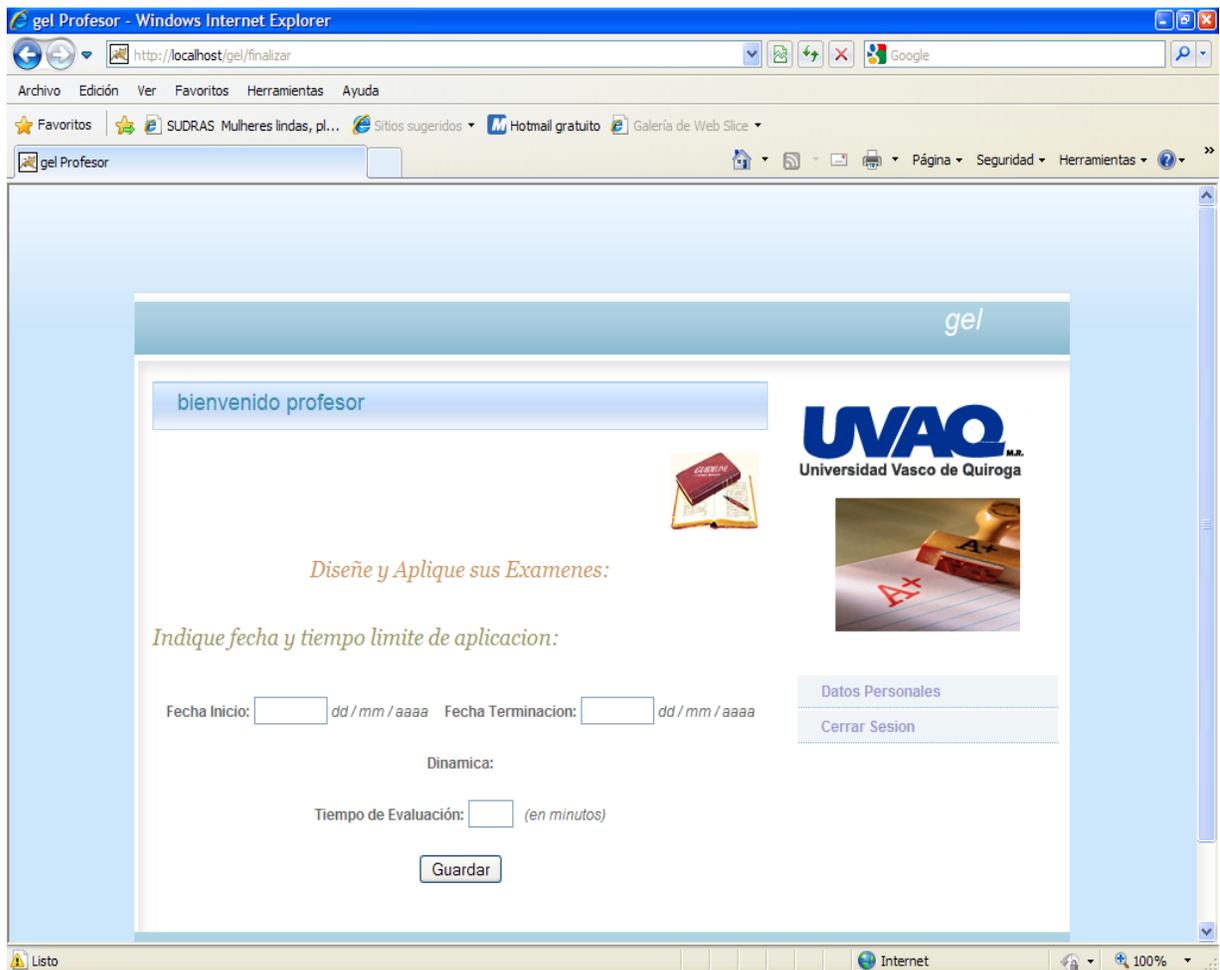


Figura 6.14 Interfaz Finalización

En el menú Calificaciones se tiene la opción de verificar, en la cual se lleva a cabo una búsqueda de todas las evaluaciones de la sesión del profesor que se encuentre activa, existen dos tipos de búsqueda: por examen y por materia.

La interfaz es como se muestra en la figura 6.15:

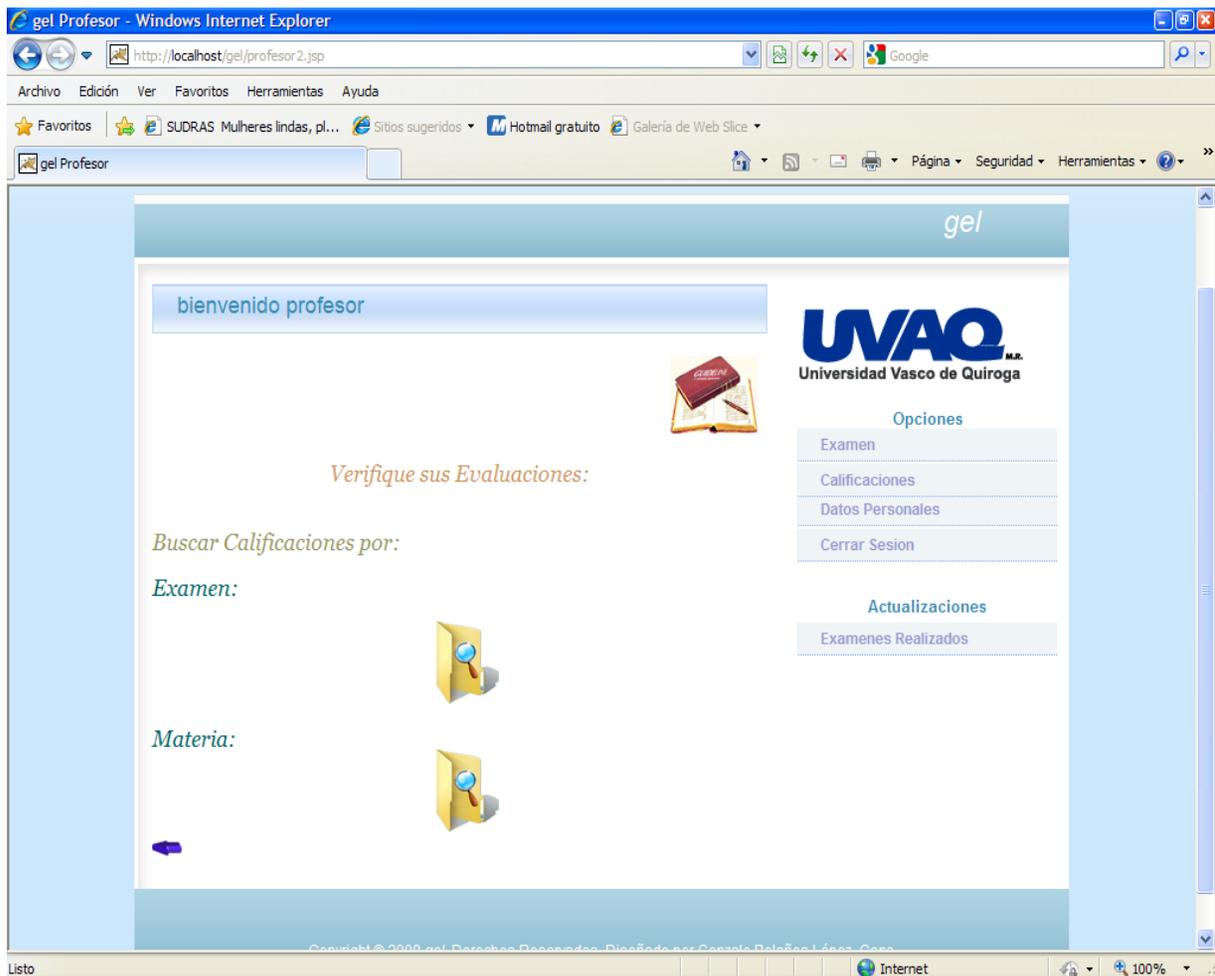


Figura 6.15 Interfaz Buscar

En la opción de Datos Personales se abre una nueva ventana mostrando información del usuario (en este caso el profesor) donde tiene la posibilidad de modificar:

- * Email
- * Teléfono
- * Dirección
- * Usuario
- * Contraseña

La interfaz es como se muestra en la figura 6.16:

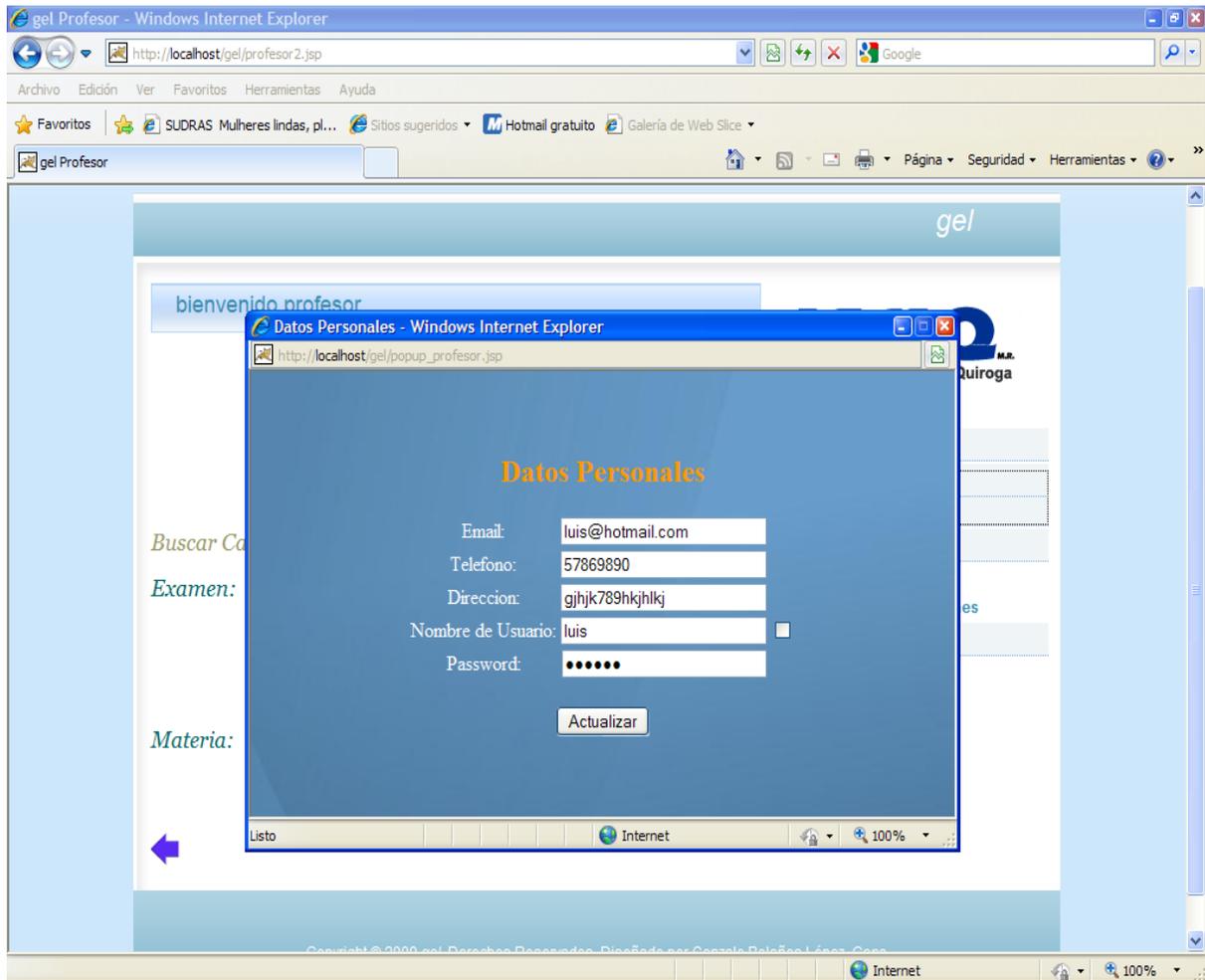


Figura 6.16 Interfaz Popup Profesor

En la opción Cerrar Sesión termina la sesión del usuario y regresa a la página principal. Finalmente en la opción Exámenes Realizados muestra todos los exámenes creados por el profesor mostrando la siguiente información:

- * Nombre del examen
- * Tipo
- * Carrera
- * Semestre
- * Materia
- * Fecha de creación

En caso de que el profesor no haya realizado un examen aún se mostrará un mensaje de que no se encontraron registros. La interfaz es como se muestra en la figura 6.17:

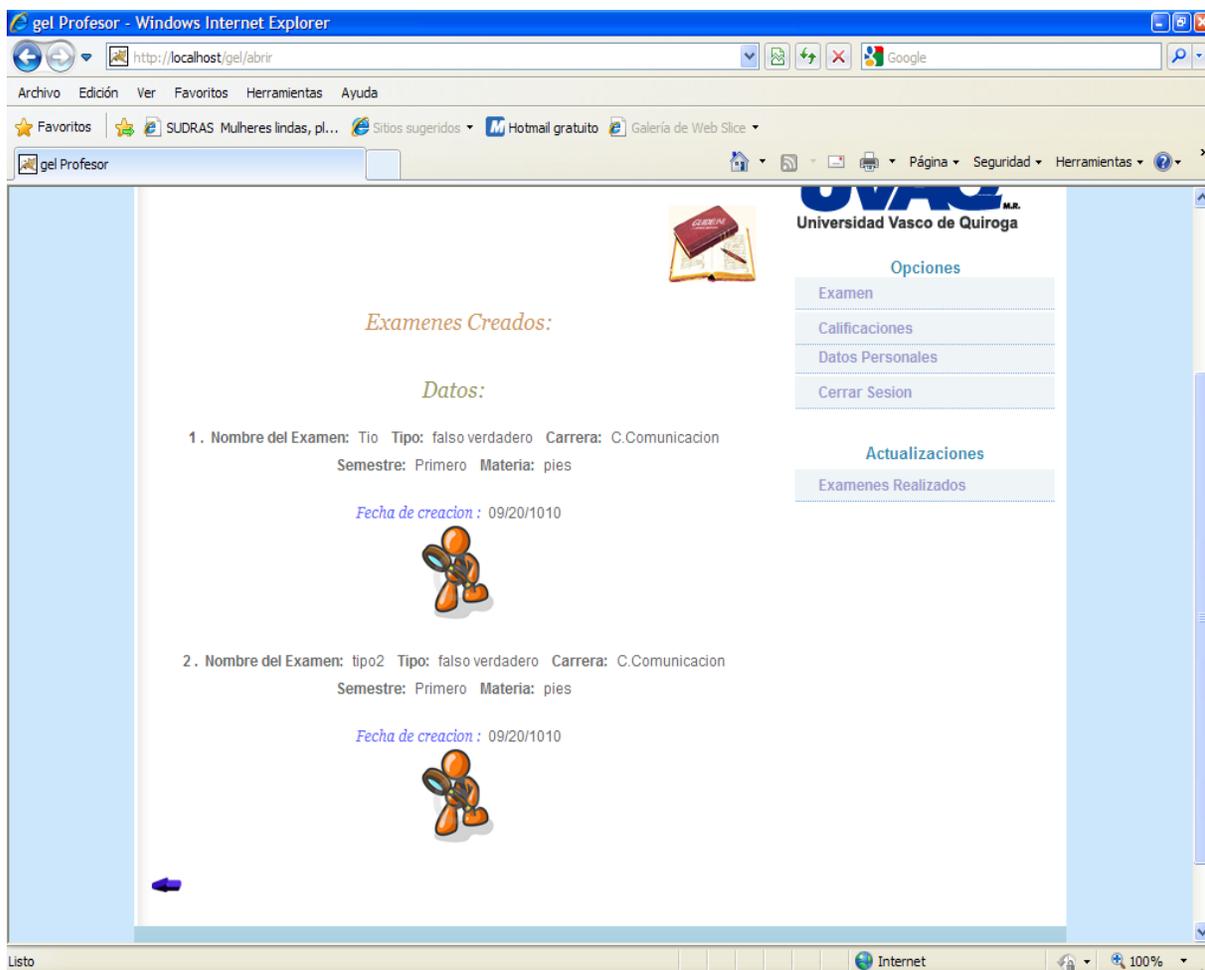


Figura 6.17 Interfaz Exámenes Realizados Profesor

Los archivos para la creación de estas interfaces son:

- profesor.html, profesorc.html. Los cuales contienen la bienvenida a la página.
- profesor1.html, profesor3.html, profesor4.html, profesor5.html, profesor6.html, profesor7.html, profesor8.html, profesor9.html y profesor10.html. Contienen la bienvenida al profesor.
- pieprof.html. El cual contiene la información sobre los tipos de exámenes para realizar, menú Opciones y menú Actualizaciones así como la leyenda Derechos Reservados y el hipervínculo para el *webmaster*.

- pieprofexam.html. Contiene los menús y opciones antes mencionados y la leyenda de derechos Reservados así como un hipervínculo para el *webmaster*.
- pieprofexam1.html. Contiene solo las opciones de Datos Personales y Cerrar Sesión al igual que la leyenda de Derechos Reservados, así como un hipervínculo para el *webmaster*.
- popup_profesor.html. Interfaz de la ventana de datos personales del profesor.
- style.css. Es el archivo donde se encuentran todas las especificaciones de la interfaz como: tipo de letra, imágenes, etc.
- profesor.jsp. Incluye el nombre del usuario que abrió sesión y los archivos profesor.html y pieprof.html.
- profesor1.jsp. Incluye un formulario para a creación de un examen con sus diferentes campos (señalados anteriormente), así como también los archivos profesor3.html y pieprofexam.html.
- profesor2.jsp. Contiene los tipos de búsqueda para las evaluaciones del profesor: Examen y Materia; al igual que los archivos profesor1.html y pieprofexam.html.
- abrir.jsp, abrir.jsp1, abrir2.jsp. Contiene la interfaz de abrir los exámenes pendientes del profesor, muestra que no se encontraron registros en caso de que no existan exámenes y de igual manera si no hay exámenes inconclusos muestra “No se encontraron exámenes sin finalizar”. Incluye los archivos profesor1.html y pieprofexam.html.
- carrerain1.jsp. Muestra el formulario para crear exámenes y la leyenda de seleccionar una carrera en caso de que no haya señalado una el profesor. Incluye los archivos profesor3.html y pieprofexam.html.
- actualizaciones.jsp, actualizaciones1.jsp. Contienen la interfaz tanto de ingreso y visualización de los exámenes que creó el profesor respectivamente. Incluye los archivos profesor9.html y pieprofexam.html.
- calificaciones.jsp, calificaciones1.jsp, calificaciones2.jsp, calificaciones3.jsp, calificaciones5.jsp. Contiene las interfaces para mostrar las materias y exámenes correspondientes al profesor, al momento de seleccionar alguna de estas opciones obtendrá las calificaciones de sus alumnos. Incluye los archivos profesor9.html y pieprofexam.html.

-
- eliminado.jsp. Muestra que el examen ha sido eliminado. Incluye los archivos profesor1.html y pieprofexam.html.
 - erroneo.jsp, erroneo1.jsp. Muestra la leyenda “Seleccione el nombre del examen” y “Seleccione el nombre de la materia” respectivamente en caso de que el profesor no lo haya indicado para su búsqueda de evaluaciones. Incluyen los archivos profesor9.html y pieprofexam.html.
 - finalización.jsp. Muestra un formulario para indicar la fecha de inicio y terminación del examen así como el tiempo de evaluación. Incluye los archivos profesor4.html y pieprofexam1.html.
 - finalizado.jsp. Muestra la leyenda “Se ha creado exitosamente el examen” cuando se finalizo el examen. Incluye los archivos profesor3.html y pieprofexam.html.
 - opcionf.jsp, opcionf1.jsp y opcionf2.jsp. Muestra los campos de inserción para las preguntas de Falso y Verdadero. Incluye lo archivos profesor5.html y pieprofexam1.html.
 - opcionfin.jsp, opcionfin1.jsp y opcionfin2.jsp. Muestra los campos de inserción para las preguntas de Falso y Verdadero al igual que la leyenda “Ingrese la respuesta” en caso de que el usuario no la haya indicado. Incluye lo archivos profesor5.html y pieprofexam1.html.
 - opcionm.jsp, opcionm1.jsp, opcionm2.jsp, opcionm3.jsp, opcionm4.jsp, opcionm5.jsp y opcionm6.jsp. Muestra los campos de inserción para las preguntas de Opción Múltiple, ingresar las opciones y sus respuestas correspondientes. Incluye los archivos profesor6.html, profesor7.html, profesor8.html y pieprofexam1.html.
 - opcionm4in.jsp. Muestra el campo para ingresar la respuesta con la leyenda “Ingrese la respuesta” en caso de que el usuario no la haya indicado. Incluye los archivos profesor8.html y pieprofexam1.html.
 - opcionmi.jsp, opcionmi1.jsp y opcionmi2.jsp. Muestra los formularios para las preguntas Falso / Verdadero y Opción Múltiple respectivamente. Incluye los archivos profesor9.html y pieprofexam1.html.
 - opcionmiin.jsp, opcionmi1in.jsp y opcionmi2in.jsp. Muestra los formularios para las preguntas Falso / Verdadero y Opción Múltiple respectivamente, y la leyenda

“Ingrese la respuesta” en caso de que el usuario no la haya indicado. Incluye los archivos profesor9.html y pieprofexam1.html.

- popup_profesor.jsp. Formulario de datos personales del profesor. Incluye los archivos popup_profesor.html y popup_piealumno.html.

- semestrein1.jsp. Formulario de datos del examen con la leyenda “Ingrese el semestre”. Incluye los archivos profesor3.html y pieprofexam.html.

- urepetido3.jsp. Formulario de datos personales del profesor con la leyenda “Usuario ya registrado, Ingrese otro distinto”. Incluye los archivos popup_profesor.html y popup_piealumno.html.

- visualizar.jsp, visualizar1.jsp, visualizar2.jsp y visualizar6.jsp. Muestran los exámenes que no han sido finalizados por el usuario ya sea : Falso / Verdadero, Opción Múltiple o Mixto. Ofrece la opción de continuar el examen, finalizarlo o borrarlo. Incluye los archivos profesor1.html, profesor6.html, profesor9.html y pieprofexam.html.

- visualizar3.jsp, visualizar4.jsp y visualizar5.jsp. Muestran los datos de exámenes finalizados y da la opción de ingresar para ver las preguntas y sus respuestas respectivas. Incluye los archivos profesor9.html y pieprofexam.html.

* Interfaz Alumnos.

Esta interfaz es donde los alumnos ingresan una vez introducido sus datos de nombre de usuario y contraseña. Se encuentra la página de bienvenida donde se le especifica al alumno como podrá encontrar su examen a resolver y la manera práctica y efectiva que obtendrá sus resultados de las evaluaciones respectivas, a través del menú de Opciones que tiene a su disposición en la parte inferior izquierda de la interfaz. La interfaz es como se muestra en la figura 6.18:

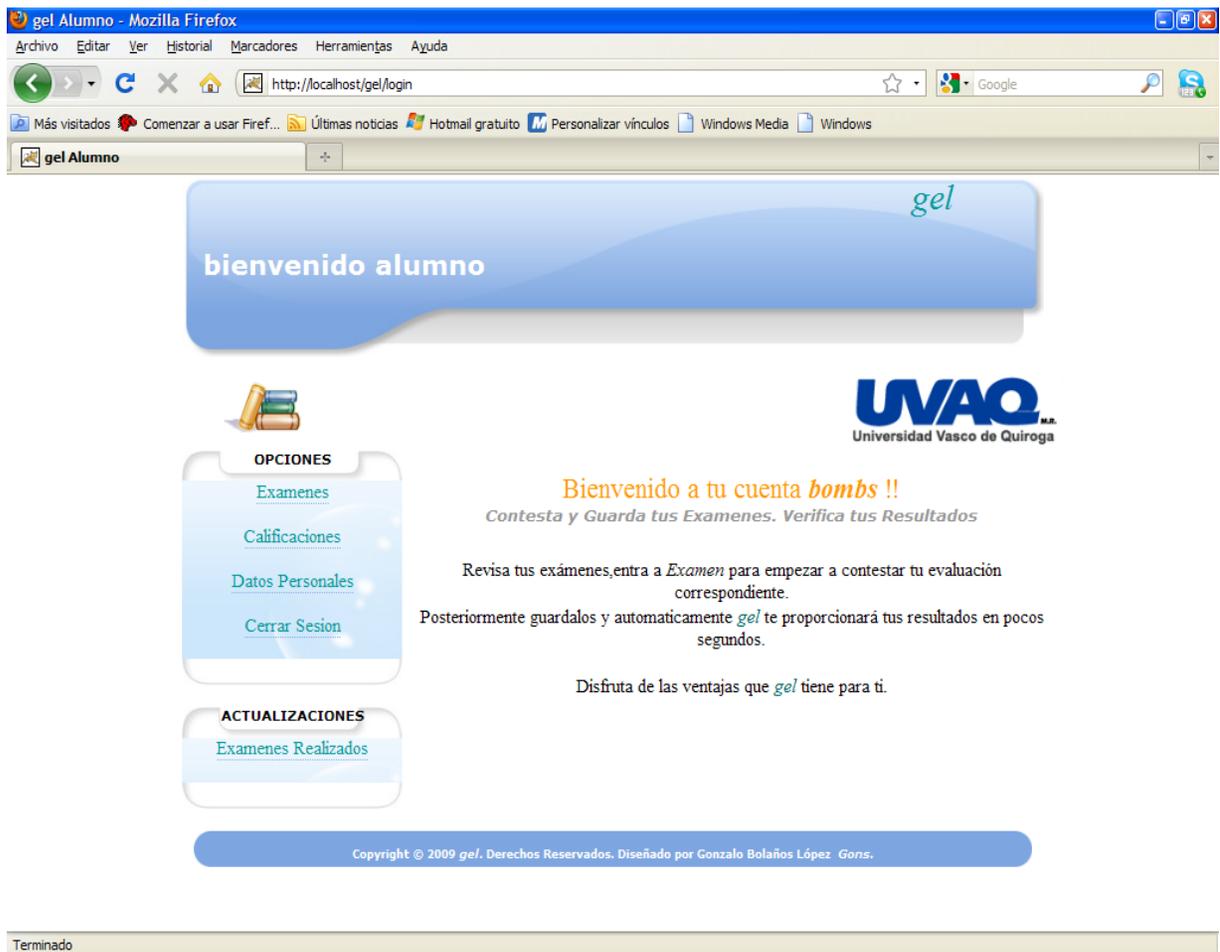


Figura 6.18 Interfaz Alumnos

El menú Opciones comprende:

- Examen. Para abrir, responder y guardar sus exámenes.
- Calificaciones. Registro de calificación de sus exámenes correspondientes cada una de sus materias
- Datos Personales. Registro y/o modificación de datos personales.

- Cerrar Sesión. Para salir de la aplicación, Regresar a la Pagina principal.

En la parte inferior a éste menú de Opciones se encuentra otro menú de Actualizaciones en el cual se proporciona información de los exámenes recientemente realizados y evaluados.

En la parte inferior de esta interfaz se encuentra la leyenda de Derechos Reservados y un hipervínculo para contactarse con el *webmaster*. Al seleccionar la opción Exámenes se despliegan en una tabla los exámenes que el alumno deberá contestar, en caso de que no exista alguno tendrá la leyenda “No se han creado exámenes recientemente”. La interfaz es como se muestra en la figura 6.19:

gel Alumno - Mozilla Firefox

http://localhost/gel/examenes

gel Alumno

bienvenido alumno gel

OPCIONES

- Exámenes
- Calificaciones
- Datos Personales
- Cerrar Sesión

ACTUALIZACIONES

- Exámenes Realizados

UVAQ
Universidad Vasco de Quiroga

Exámenes:

Tienes nuevos exámenes a realizar, elige uno y comienza tu evaluación:

Nombre del Examen	Materia
* New1	ingles

Copyright © 2009 gel. Derechos Reservados. Diseñado por Gonzalo Bolaños López Gons.

Terminado

Figura 6.19 Interfaz Exámenes Alumno

Al seleccionar la opción Calificaciones el alumno podrá buscar sus evaluaciones de dos formas, por nombre de examen o por materia. En caso de que no hubieran evaluaciones tendrá la leyenda “No se han evaluado exámenes aún”. La interfaz es como se muestra en la figura 6.20:



Figura 6.20 Interfaz Calificaciones Alumno

Seleccionando la opción Datos Personales el alumno será capaz de verificar y / o actualizar su información personal.

La interfaz es como se muestra en la figura 6.21:

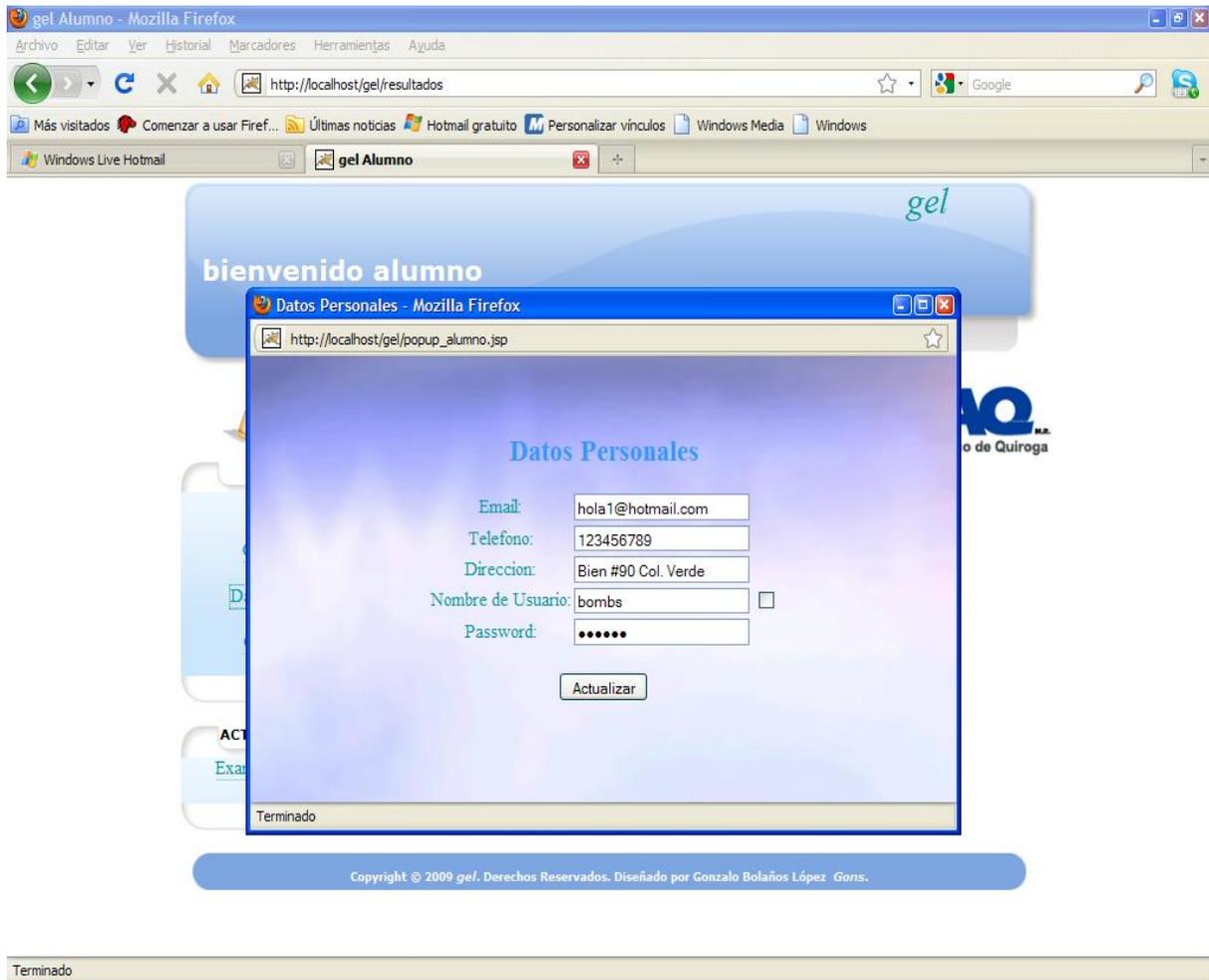


Figura 6.21 Interfaz Popup Alumno

La opción Cerrar Sesión termina la sesión del alumno y lo redirecciona a la página principal de la aplicación. Finalmente la opción Exámenes Realizados permite al alumno ver cada examen que ha realizado y verificar que respuestas fueron acertadas y cuales erróneas.

La interfaz es como se muestra en la figura 6.22:



Figura 6.22 Interfaz Exámenes Realizados Alumno

Los archivos para la creación de esta interfaz son:

- alumno.html, alumno1.html y alumno3.html. Contienen la bienvenida a la página y el menú de opciones: Exámenes, Calificaciones, Datos Personales, Cerrar Sesión y Exámenes Realizados.
- alumno2.html. Contiene el menú de opciones anteriormente mencionado y el título calificaciones señalando al alumno que verifique sus evaluaciones.
- alumno4.html. Contiene solo la opción datos Personales y una imagen de reloj que indica el tiempo de evaluación del examen.
- piealumn.html. El cual contiene la información general sobre como abrir un examen, contestarlo y guardarlo para su futura evaluación, así como la leyenda Derechos Reservados y el hipervínculo para el *webmaster*.

-
- piealumn1.html y piealumn2.html. Contienen una flecha de hipervínculo que los manda a la página principal de la sesión del alumno y la leyenda Derechos Reservados y el hipervínculo para el *webmaster*.
 - piealumn3.html y piealumn4.html. Contienen solo la leyenda Derechos Reservados y el hipervínculo para el *webmaster*.
 - popup_alumno.html. Parte de la interfaz de la ventana de datos personales del alumno.
 - popup_piealumno.html. Parte de la interfaz de la ventana de datos personales del alumno.
 - style.css. Es el archivo donde se encuentran todas las especificaciones de la interfaz como: tipo de letra, imágenes, etc.
 - alumno.jsp. Contiene el nombre del usuario, los menús correspondientes al alumno. Incluye los archivos alumno.html y piealumn.html.
 - alumno1.jsp. Contiene el título de Exámenes y muestra cuando existen evaluaciones para ser contestadas por el alumno, los menús correspondientes al alumno. Incluye los archivos alumno3.html y piealumn1.html.
 - calificaciones4.jsp. Contiene dos tipos de búsqueda de evaluaciones: Examen y Materia, los menús correspondientes al alumno. Incluye los archivos alumno3.html y piealumn1.html.
 - calificacionosal.jsp. Muestra en una tabla el resultado de su evaluación conteniendo los siguientes campos: Examen, Tipo, Calificación y Resultado. Contiene los menús correspondientes al alumno. Incluye los archivos alumno3.html y piealumn1.html.
 - calificacionosal1.jsp. Muestra en una tabla el resultado de búsqueda por materia conteniendo los siguientes campos: Materia, Examen y un hipervínculo para visualizar su evaluación correspondiente. Contiene los menús correspondientes al alumno. Incluye los archivos alumno3.html y piealumn1.html.
 - calificado.jsp. Muestra en una tabla el resultado de su evaluación conteniendo los siguientes campos: Examen, Calificación y Resultado. Contiene los menús correspondientes al alumno. Incluye los archivos alumno3.html y piealumn1.html.
 - contestar.jsp. Muestra en una tabla la información del examen que se va a evaluar conteniendo los siguientes campos: Tipo de Examen, Número de Preguntas, Valor

del Examen, Tiempo de Evaluación y Vigencia. Contiene los menús correspondientes al alumno. Incluye los archivos alumno1.html y piealumn1.html.

- erroneo2.jsp y erroneo3. Contiene dos tipos de búsqueda de evaluaciones: Examen y Materia, al igual que dos leyendas cuando no se ha seleccionado un examen o materia (“Selecciona un examen”, “Selecciona una materia”) respectivamente. Contiene los menús correspondientes al alumno. Incluye los archivos alumno3.html y piealumn1.html.

- popup_alumno.jsp. Contiene los Datos Personales del alumno. Incluye los archivos popup_alumno.html y popup_piealumno.html.

- resuelto.jsp, resuelto1.jsp y resuelto2.jsp. Muestra los exámenes con las respuestas del usuario señalando cuales fueron acertadas y cuales erróneas de los tres tipos de exámenes (Falso / Verdadero, Opción Múltiple y Mixto) respectivamente. Incluye los archivos alumno1.html y piealumn4.html.

- resultados.jsp. Muestra en una tabla la información de los exámenes resultados por el alumno, contiene los siguientes campos: Materia, Tipo, Examen, Calificación, Resultado y un hipervínculo para ingresar al examen respectivo. Incluye los archivos alumno1.html y piealumn1.html.

- urepetido2.jsp. Contiene los Datos Personales del alumno con la leyenda “Usuario registrado, ingrese otro distinto”. Incluye los archivos popup_alumno.html y popup_piealumno.html.

- visualizaral.jsp, visualizaral1.jsp y visualizaral2.jsp. Muestra los exámenes en sus distintas modalidades para ser resueltos por el alumno. Incluye los archivos alumno4.html y piealumn3.html.

* actualizado.jsp. Muestra la leyenda “Cuenta actualizada” y la opción para cerrar la ventana, es tanto para el profesor como para el alumno.

Archivos para la funcionalidad general de la aplicación:

* Tooltip.js. Archivo para mostrar diferentes imágenes al pasar el mouse sobre los menús de la página principal.

- * `funcion.js`, `functions.js`, `functionss.js`, `functions1.js`, `functions2.js`, `functions3.js`, `functions4.js`, `functions5.js`, `functions6.js`, `functions7.js` y `functions8.js`. Archivos para validar los diferentes campos de la aplicación.
- * `mootools.js`, `sexyalertbox.css`, `sexyalertbox.packed.js`, `sexyalertbox.v1.1.js`. Archivos para la interfaz del mensaje de alerta acerca del tiempo de evaluación del examen.
- * `activación.html`. Contiene el logo de la UVAQ, el nombre de la aplicación y la leyenda “Necesita activar JavaScript para visualizar esta página”.
- * `pieac.html`. Contiene el hipervínculo a la página principal con la leyenda “Ingrese a la aplicación una vez activado” (Refiriéndose a la activación de JavaScript).
- * `activación.jsp`. Concluye la sesión del usuario, incluye los archivos `activación.html` y `pieac.html`.

Los archivos de `profesor.html` con sus variantes numéricas así como también los del `alumno.html`, se crearon por la necesidad de implementar diversas funciones para cada página que despliegan; de igual manera la información contenida en cada uno de ellos está dispuesta de manera distinta en los *divs* (espacios o dimensiones) de HTML, esto con la finalidad de mostrar la información adecuada para cada función de la aplicación.

6.2 Base de Datos

En este proyecto se comprende una Base de Datos relacional llamada GEL la cual será administrada por PhpMyAdmin que es un sistema de gestión de base de datos basado en MySQL (mencionado en el tema 4.3 PhpMyAdmin). Esta base de datos contendrá diferentes tablas con sus respectivas relaciones, como se muestra en la figura 6.23:

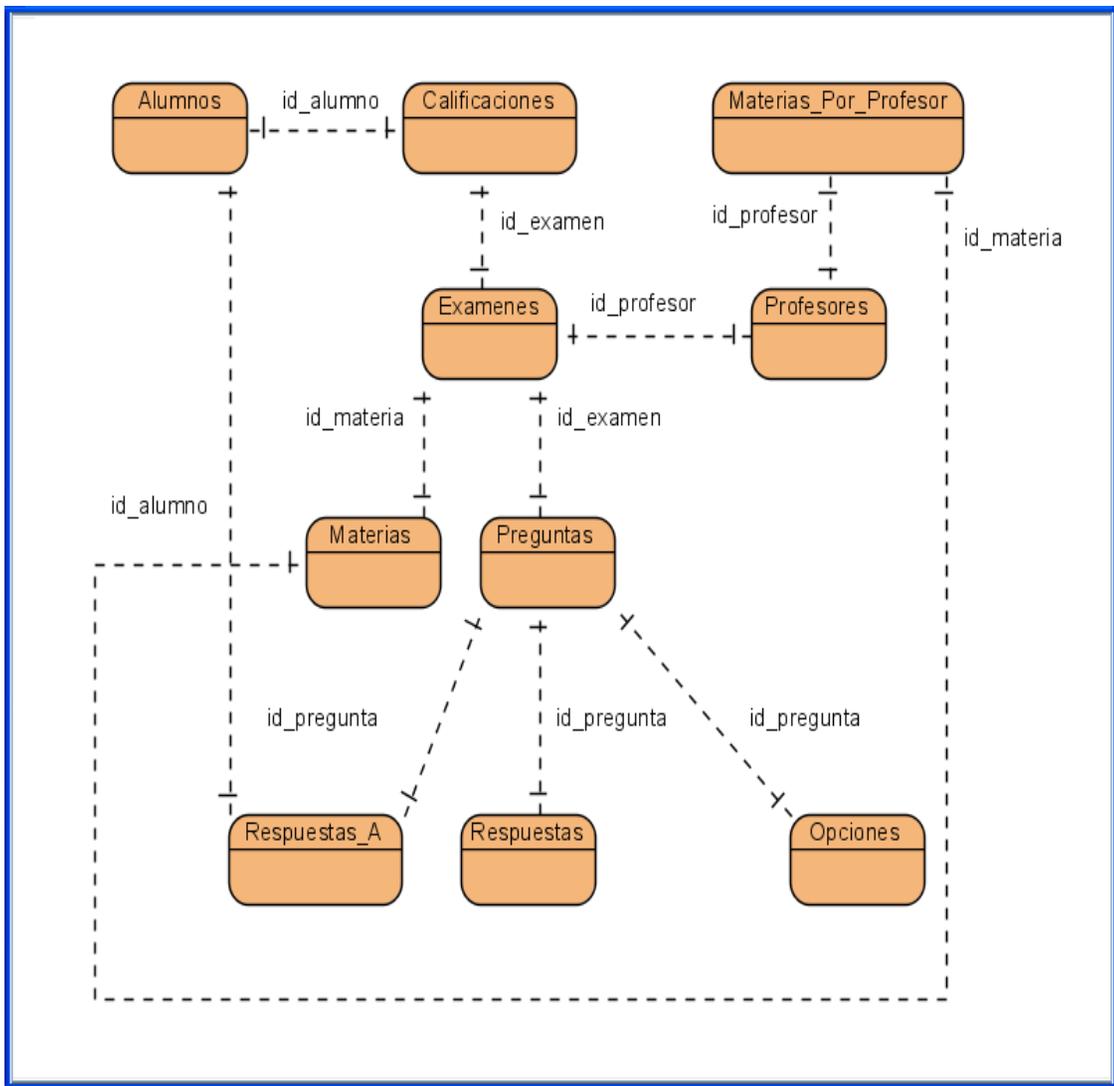


Figura 6.23 Base de Datos GEL

A continuación se señalan y describen cada una de las tablas que conforman la Base de Datos del proyecto:

- Profesores. Contiene información general del profesor, sus campos son:

```
CREATE TABLE IF NOT EXISTS `profesores`
(
  `id_profesor` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(128) CHARACTER SET utf8 NOT NULL,
  `email` varchar(128) CHARACTER SET utf8 NOT NULL,
  `telefono` int(255) NOT NULL,
```

```
`direccion` varchar(128) CHARACTER SET utf8 NOT NULL,  
`clave` varchar(128) CHARACTER SET utf8 NOT NULL,  
`username` varchar(128) CHARACTER SET utf8 NOT NULL,  
`password` varchar(128) CHARACTER SET utf8 NOT NULL,  
PRIMARY KEY (`id_profesor`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

- Alumnos. Contiene información general del alumno, sus campos son:

```
CREATE TABLE IF NOT EXISTS `alumnos`  
(  
  `id_alumno` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `email` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `telefono` int(128) NOT NULL,  
  `direccion` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `clave` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `carrera` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `semestre` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `username` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `password` varchar(128) CHARACTER SET utf8 NOT NULL,  
  PRIMARY KEY (`id_alumno`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

- Exámenes. Contiene información acerca de los exámenes a realizar, sus campos son:

```
CREATE TABLE IF NOT EXISTS `examenes`  
(  
  `id_examen` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `fecha_inicio` varchar(128) DEFAULT 'no señalada',  
  `fecha_terminacion` varchar(128) DEFAULT 'no señalada',
```

```
`tipo` varchar(128) NOT NULL,  
`tiempo` int(3) DEFAULT '0',  
`finalizado` varchar(128) NOT NULL DEFAULT 'No',  
`id_materia` int(11) NOT NULL,  
`id_profesor` int(11) NOT NULL,  
`fecha_creacion` varchar(128) DEFAULT 'no finalizado',  
`fecha_actual` varchar(11) NOT NULL DEFAULT 'indefinido',  
PRIMARY KEY (`id_examen`),  
KEY `id_materia` (`id_materia`),  
KEY `id_profesor` (`id_profesor`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

- Materias. Contiene información acerca de las materias de sus respectivos exámenes, sus campos son:

```
CREATE TABLE IF NOT EXISTS `materias`  
(  
  `id_materia` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `carrera` varchar(128) CHARACTER SET utf8 NOT NULL,  
  `semestre` varchar(128) NOT NULL,  
  PRIMARY KEY (`id_materia`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

- Calificaciones. Contiene información sobre los resultados obtenidos de los exámenes, sus campos son:

```
CREATE TABLE IF NOT EXISTS `calificaciones`  
(  
  `id_calificacion` int(11) NOT NULL AUTO_INCREMENT,  
  `promedio` varchar(128) NOT NULL,  
  `id_alumno` int(11) NOT NULL,  
  `id_examen` int(11) NOT NULL,
```

```
`resultado` varchar(128) DEFAULT 'indefinido',
PRIMARY KEY (`id_calificacion`),
KEY `id_alumno` (`id_alumno`),
KEY `id_examen` (`id_examen`)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
- Materias_Por_Profesor. Contiene información acerca de las materias que imparte
cada profesor, sus campos son:
CREATE TABLE IF NOT EXISTS `materias_por_profesor`
(
  `id_mat_prof` int(11) NOT NULL AUTO_INCREMENT,
  `id_profesor` int(11) NOT NULL,
  `id_materia` int(11) NOT NULL,
  PRIMARY KEY (`id_mat_prof`),
  KEY `id_profesor` (`id_profesor`),
  KEY `id_materia` (`id_materia`)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
- Preguntas. Contiene información sobre la redacción de cada pregunta y su tipo, sus
campos son:
CREATE TABLE IF NOT EXISTS `preguntas`
(
  `id_pregunta` int(11) NOT NULL AUTO_INCREMENT,
  `texto` text CHARACTER SET utf8 NOT NULL,
  `id_examen` int(11) NOT NULL,
  `valor` int(3) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_pregunta`),
  KEY `id_examen` (`id_examen`)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

- Respuestas. Contiene información sobre las respuestas correctas para cada pregunta, sus campos son:

```
CREATE TABLE IF NOT EXISTS `respuestas`  
(  
  `id_respuesta` int(11) NOT NULL AUTO_INCREMENT,  
  `id_pregunta` int(11) NOT NULL,  
  `r_correcta` varchar(128) CHARACTER SET utf8 NOT NULL,  
  PRIMARY KEY (`id_respuesta`),  
  KEY `id_pregunta` (`id_pregunta`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

- Opciones. Contiene información sobre las opciones del Examen de Opción Múltiple, sus campos son:

```
CREATE TABLE IF NOT EXISTS `opciones`  
(  
  `id_opcion` int(11) NOT NULL AUTO_INCREMENT,  
  `id_pregunta` int(11) NOT NULL,  
  `texto` varchar(128) CHARACTER SET utf8 NOT NULL,  
  PRIMARY KEY (`id_opcion`),  
  KEY `id_pregunta` (`id_pregunta`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

- Respuestas_a. Contiene información acerca de las respuestas realizadas por los alumnos en sus exámenes respectivos, sus campos son:

```
CREATE TABLE IF NOT EXISTS `respuestas_a`  
(  
  `id_respuesta` int(11) NOT NULL AUTO_INCREMENT,  
  `id_pregunta` int(11) NOT NULL,  
  `id_alumno` int(11) NOT NULL,  
  `respuesta` varchar(128) NOT NULL,  
  `valor` int(3) DEFAULT '0',
```

```
PRIMARY KEY (`id_respuesta`),  
KEY `id_pregunta` (`id_pregunta`),  
KEY `id_alumno` (`id_alumno`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

La información introducida a través de los formularios por parte de alumnos y profesores, así como de sus calificaciones y exámenes, estará almacenada en esta base de Datos llamada GEL, la cual será el sustento de esta aplicación, conteniendo toda la información relevante para su implementación.

6.3 POJO`s y Servlets

* POJO`s

Un POJO (acrónimo de *Plain Old Java Object*) es una sigla creada por Martin Fowler, Rebecca Parsons y Josh MacKenzie en septiembre de 2000 y utilizada por programadores Java para enfatizar el uso de clases simples y que no dependen de un *framework* en especial. Este acrónimo surge como una reacción en el mundo Java a los *frameworks* cada vez más complejos.

POJO es una nueva palabra para designar algo viejo. No existe en Java una nueva tecnología con ese nombre, sino que el nombre existe en el marco de una revalorización de la programación "simplemente orientada a objetos". Esta revalorización tiene que ver también con el éxito de lenguajes orientados a objetos más puros y sencillos, que empezaron a tomar parte del mercado al que Java apunta, como Ruby y Python. [19]

A continuación se señalan y describen los POJO`s de esta aplicación:

- Alumno.java. Obtiene los datos personales del alumno.
- Profesor.java. Obtiene los datos personales del profesor.
- Validacion.java. Obtiene el *username* y *password* del profesor.
- Validacion1.java. Obtiene el *username* y *password* del alumno.

* **Servlets.**

Los *servlets* son objetos que corren dentro del contexto de un contenedor de *servlets* (en este caso Tomcat) y extienden su funcionalidad. 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*. En cambio, un *servlet* es un programa que se ejecuta en un servidor. 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.

Ciclo de vida

1. El cliente solicita una petición a un servidor vía URL
2. El servidor recibe la petición.
 1. Si es la primera, se utiliza el motor de *Servlets* para cargarlo y se llama al método `init()`.
 2. Si ya está iniciado, cualquier petición se convierte en un nuevo hilo. Un *Servlet* puede manejar múltiples peticiones de clientes.
3. Se llama al método `service()` para procesar la petición devolviendo el resultado al cliente.
4. Cuando se apaga el motor de un *Servlet* se llama al método `destroy()`, que lo destruye y libera los recursos abiertos.

Clases y objetos necesarios

Podemos crear un Servlet haciendo uso del paquete `javax.servlet`.

Clase `HttpServletRequest`

Representa la comunicación desde el cliente hacia el servidor. El flujo de entrada será del tipo *ServletInputStream* y permite acceder a la cabecera HTTP y a los datos del cliente.

Clase HttpServletResponse

Representa la comunicación desde el servidor hacia el cliente, donde el flujo será del tipo *ServletOutputStream*. [20]

A continuación se señalan y describen los *servlets* de esta aplicación:

- ActualizaAlumnServlet.java. Actualiza los datos personales del alumno, los cuales son: email, teléfono, dirección, *username* y *password*.
- ActualizaProfeServlet.java. Actualiza los datos personales del profesor, los cuales son: email, teléfono, dirección, *username* y *password*.
- AlumServlet.java. Inserta en la base de datos GEL los datos personales del alumno que son: nombre, email, teléfono, dirección, clave, carrera, semestre, *username* y *password*.
- BuscarServlet.java. Busca los exámenes correspondientes al profesor.
- CalificacionesAlumnoServlet.java. Busca las calificaciones del alumno por materia, por parte del profesor.
- CalificacionesExamAlumServlet.java. Busca las calificaciones del alumno por nombre de examen.
- CalificacionesMatAlumServlet.java. Busca las calificaciones del alumno por materia, por parte del alumno.
- CalificacionesServlet.java. Obtiene los id y nombres de los exámenes por nombre de examen y materia del profesor.
- CalificacionessServlet.java. Obtiene los datos del examen por nombre de examen y las calificaciones de cada alumno correspondiente.
- CalificacionesssServlet.java. Obtiene los datos del examen por nombre de la materia y las calificaciones de cada alumno correspondiente.
- CalificarMServlet.java. Califica el examen de tipo Mixto.
- Calificar Servlet.java. Califica los exámenes tanto de Falso / Verdadero como de Opción Múltiple.
- ContestarServlet.java. Obtiene los datos de los exámenes para ser visualizados posteriormente por un archivo JSP.

-
- CrearExamenServlet.java. Crea un examen ya sea de tipo: Falso / Verdadero, Opción Múltiple o Mixto. Inserta la información del examen en la base de datos GEL, los cuales son: nombre del examen, carrera, semestre, nombre de la materia y tipo de examen.
 - EliminarServlet.java. Borra un examen no finalizado por el profesor.
 - ExámenesAlumnoServlet.java. Obtiene todos los exámenes correspondientes a la carrera y semestre del alumno.
 - EvaluarServlet.java. Obtiene los datos del examen para ser mostrados posteriormente.
 - ExámenesResueltosAlumServlet.java. Obtiene la información de los exámenes resueltos del alumno, como: nombre de examen, materia, tipo de examen, calificación y resultado.
 - FinalizadoServlet.java. Actualiza la tabla Exámenes de la base de datos GEL para señalar que se ha finalizado un examen por parte del profesor.
 - FinalServlet.java. Inserta en la base de datos GEL información del examen para ser finalizado, como: fecha de inicio y fin del examen y tiempo de evaluación.
 - FOMPServlet.java. Obtiene el texto de la pregunta y la opción del examen de tipo Opción Múltiple para ser mostrado posteriormente por un archivo JSP.
 - FVServlet.java. Inserta en la base de datos Gel la información del examen de tipo Falso / Verdadero, como: texto de la pregunta y su respuesta correspondiente.
 - FVServlet.java. Inserta en la base de datos Gel la información del examen de tipo Falso / Verdadero para posteriormente ser finalizado, como: texto de la pregunta y su respuesta correspondiente.
 - LoginServlet.java. Realiza las validaciones del usuario tanto para profesor como alumno e ingresar posteriormente a sus respectivas cuentas.
 - LogoutServlet.java. Termina la sesión del usuario y redirecciona a la página principal de la aplicación.
 - NoticiasServlet.java. Realiza la búsqueda en la base de datos GEL de los últimos 5 exámenes realizados para posteriormente ser mostrados en un archivo JSP con los datos: nombre de examen, semestre y materia.

-
- OMServlet.java. Inserta en la base de datos GEL el texto de las preguntas del examen de tipo Opción Múltiple.
 - OOMServlet.java. Inserta en la base de datos GEL el texto de las opciones del examen de tipo Opción Múltiple.
 - OOOMServlet.java. . Inserta en la base de datos GEL el texto de la última opción de la pregunta correspondiente para el examen de tipo Opción Múltiple.
 - OOOOMServlet.java. Inserta en la base de datos GEL el texto de la pregunta para posteriormente seleccionar sus respectivas opciones y ser finalizado el examen de tipo Opción Múltiple.
 - ProfeServlet.java. Inserta en la base de datos GEL los datos personales del profesor que son: nombre, email, teléfono, dirección, clave, *username* y *password*.
 - ResultadoExamenServlet.java. Obtiene la información de exámenes evaluados del alumno para posteriormente ser mostrados por un archivo JSP, visualizando a su vez cuales respuestas fueron acertadas y cuales erróneas..
 - ROMPServlet.java. Inserta en la base de datos GEL el texto de la opción para posteriormente seleccionar su respuesta a la pregunta correspondiente y ser finalizado el examen de tipo Opción Múltiple.
 - VisualizarExámenesServlet.java. Busca la información de los exámenes realizados por el profesor para posteriormente ser visualizados por un archivo JSP, los datos son: nombre del examen, materia, carrera, semestre y fecha de creación.
 - VisualizarServlet.java. Obtiene la información de los exámenes realizados por el profesor para mostrarlos en sus respectivas modalidades según sea el caso: Falso / Verdadero, Opción Múltiple y Mixto.

* Los archivos compilados .class de java se encuentran en la carpeta *classes* (mencionado en el tema 5.2 Sistema de archivos de la aplicación en Tomcat).

* Algunos *servlets* realizan funciones muy parecidas unos de otros, pero tienen la particularidad de que dependiendo de la información que reciban redireccionan a páginas distintas dentro de la aplicación.

A continuación se muestran todos los *POJO*'s y *servlets* antes mencionados por medio de Diagramas UML (Diagramas de Clase) con sus atributos y métodos correspondientes. De igual manera se presenta un Diagrama de Casos de Uso en UML donde se señala la funcionalidad general del sistema para los dos tipos de usuario.

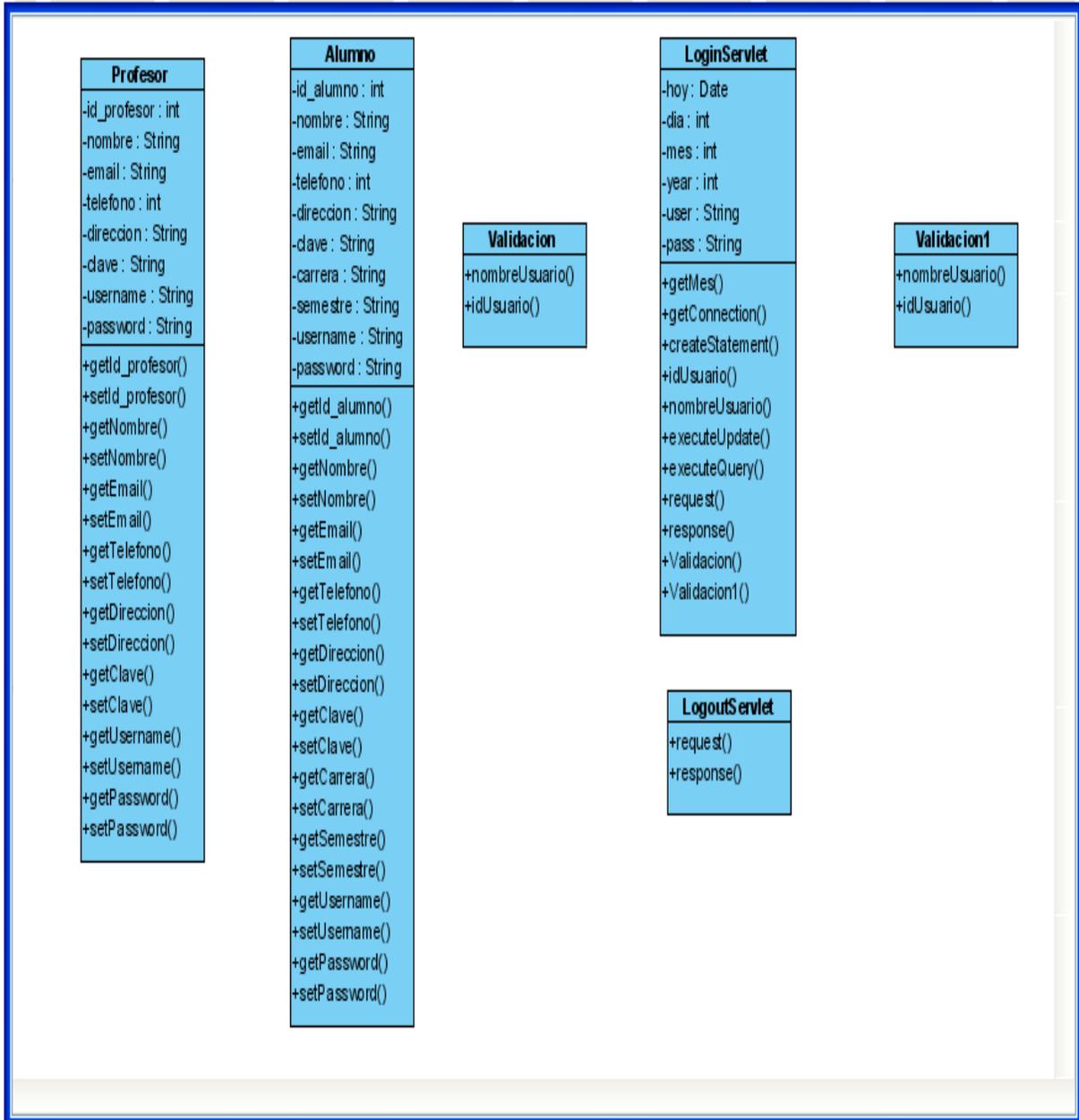


Figura 6.24 Diagrama de Clases Login / Logout



Figura 6.25 Diagrama de Clases Registro y Actualización de Usuarios

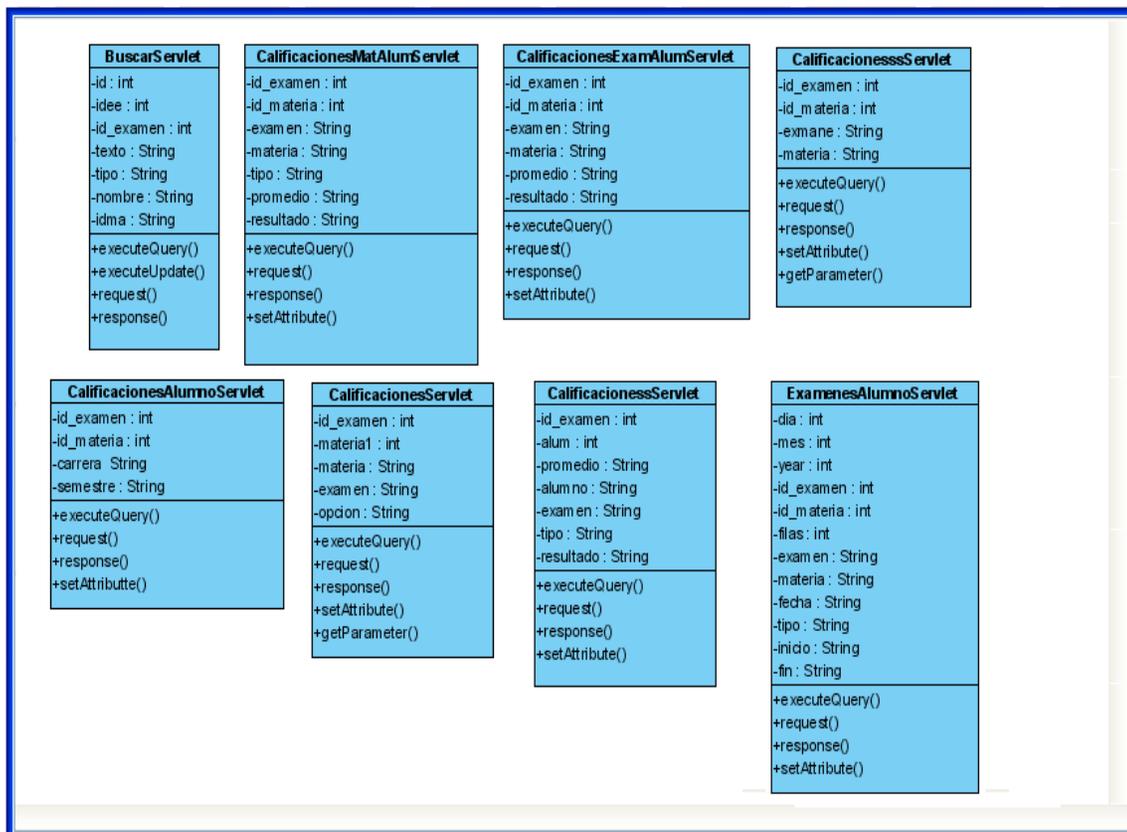


Figura 6.26 Diagrama de Clases de Búsqueda de Evaluaciones

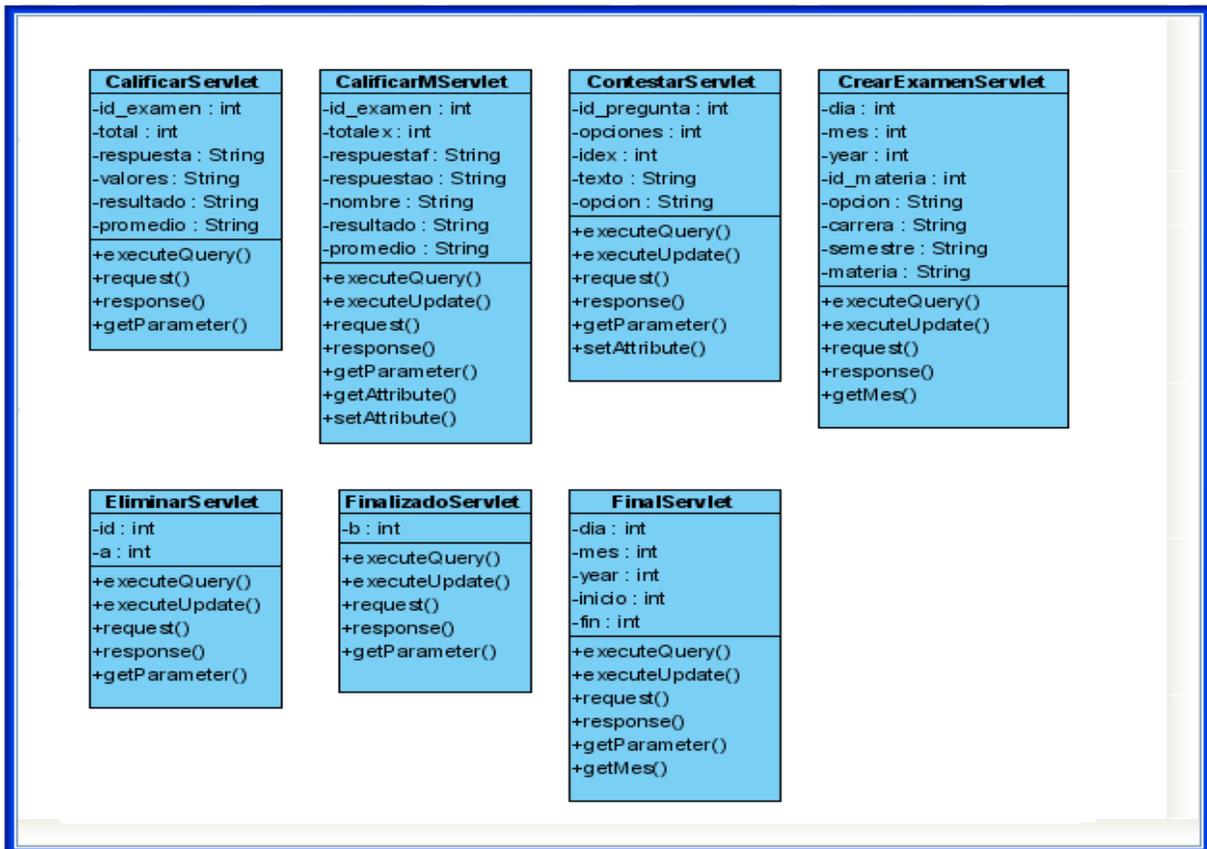


Figura 6.27 Diagrama de Clases para Crear, Eliminar y Evaluar Exámenes

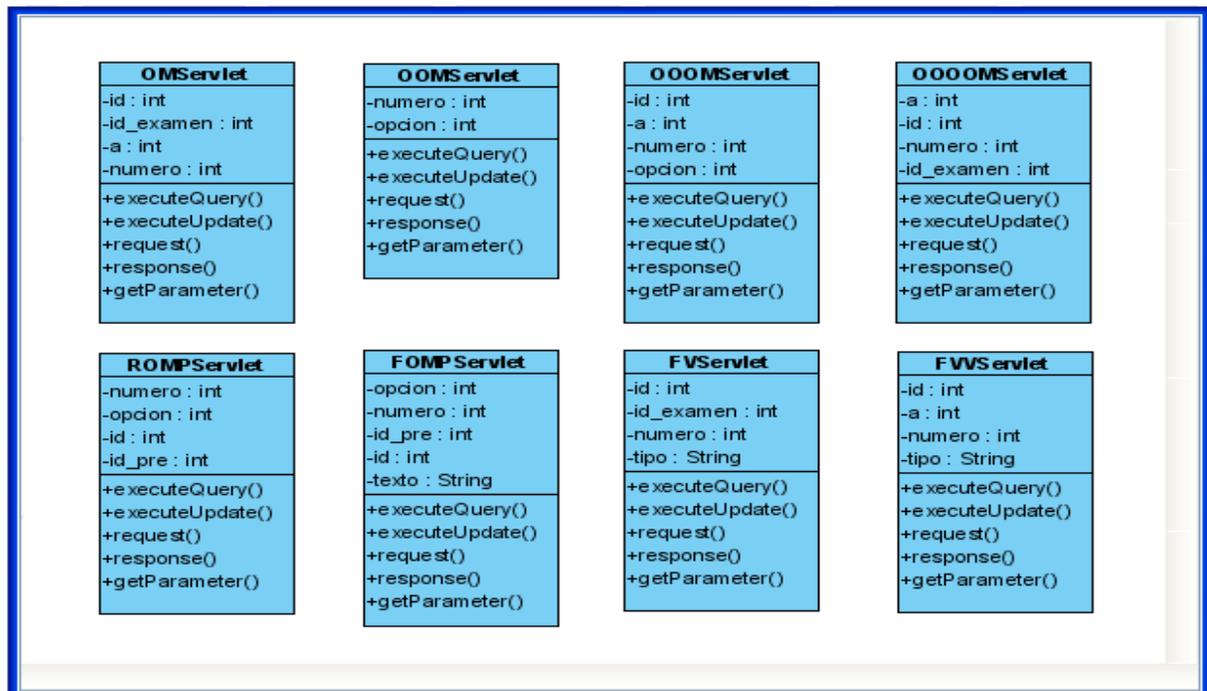


Figura 6.28 Diagrama de Clases para Ingreso de Preguntas

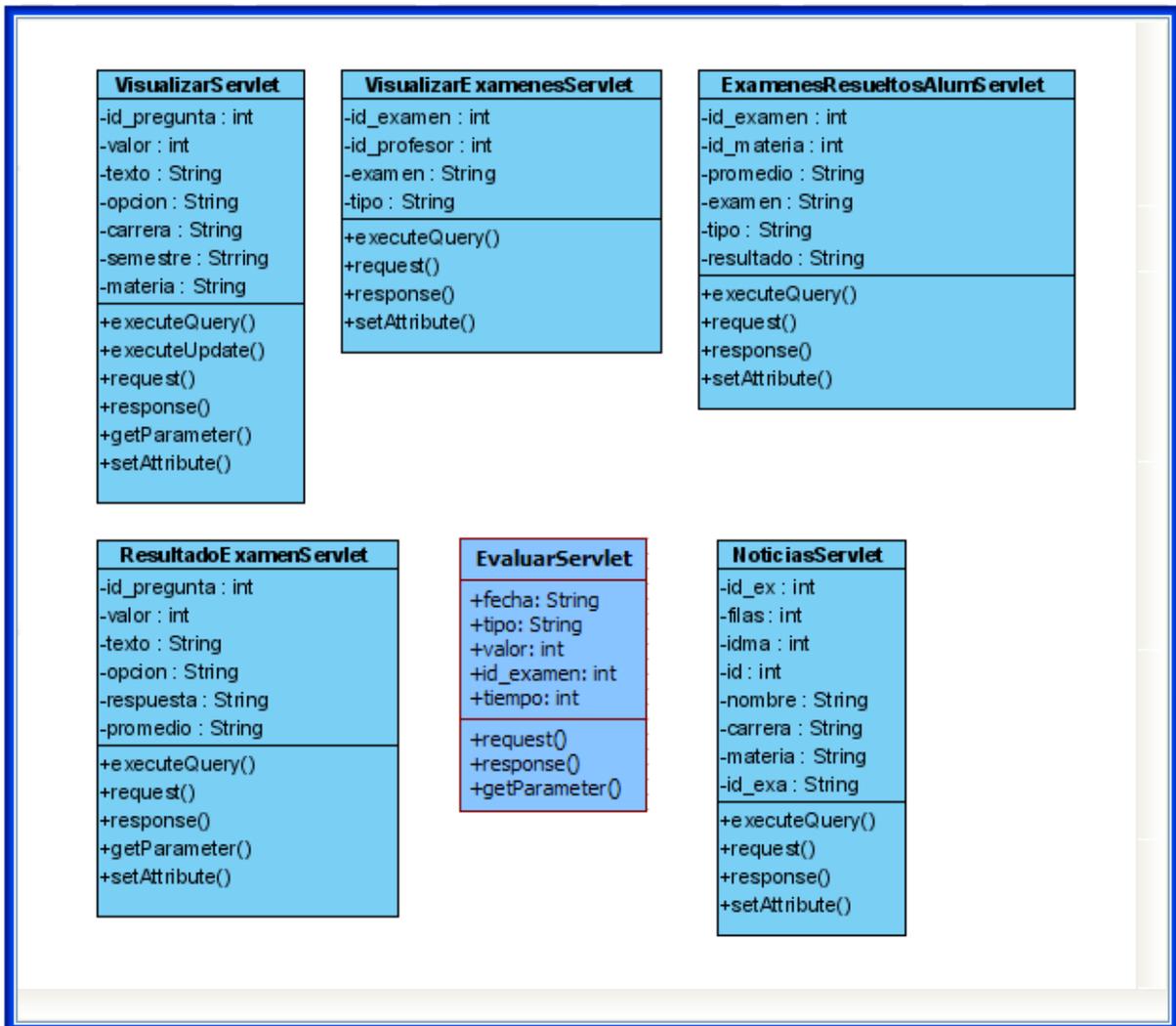


Figura 6.29 Diagrama de Clases para Visualizar Exámenes

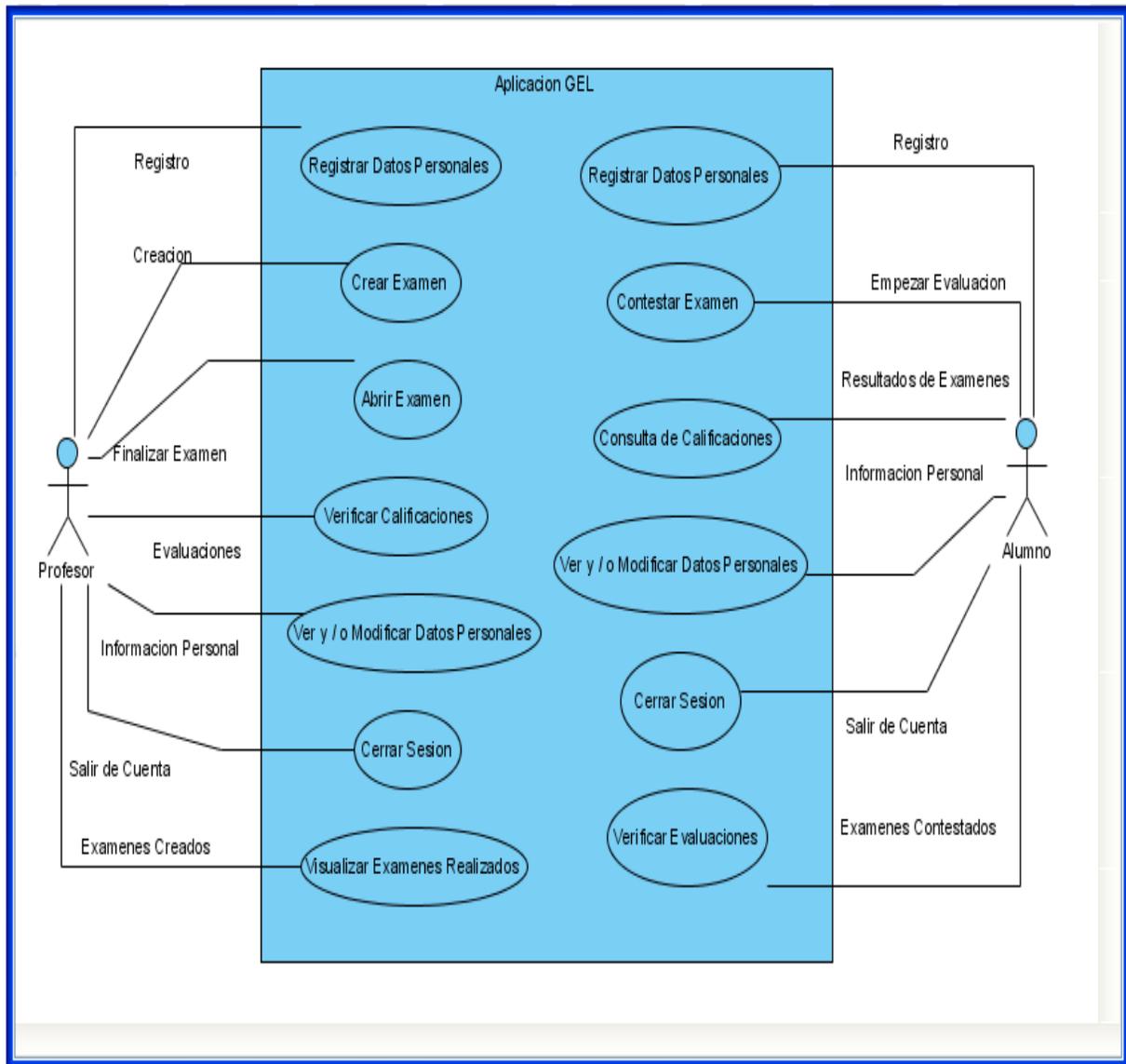


Figura 6.30 Diagrama de Casos de Uso

Funcionalidad del Sistema.

A continuación se puntualiza la funcionalidad total del sistema:

Página Principal.

- Inicio. Bienvenida a la aplicación GEL.
- Profesores. Forma de registro validada por Javascript y un servlet.
- Alumnos. Forma de registro validada por Javascript y un servlet.
- Noticias. Información de exámenes recientemente creados.

- Acerca de. Información general de la aplicación GEL.
- Forma de ingreso de nombre de usuario y contraseña, validada por Javascript y un servlet.

Pagina Profesores.

- Examen:
 - Crear. Forma de registro de un examen, validado por Javascript y un servlet.
 - Abrir. Búsqueda de exámenes no finalizados por el profesor.
- Calificaciones:
 - Verificar. Búsqueda de evaluaciones por el nombre del examen o por materia. Campos validados por un servlet.
- Datos Personales. Muestra los datos personales del profesor para verificarlos y / o modificarlos. Campos validados por Javascript y un servlet.
- Cerrar Sesión. Finaliza la sesión del usuario.
- Exámenes Realizados. Búsqueda de exámenes creados por el profesor.

Pagina Alumnos.

- Exámenes. Búsqueda de exámenes a ser realizados por el alumno.
- Realizará sus exámenes correspondientes en base a su carrera y semestre. Solo los podrá resolver en las fechas previamente establecidas por el profesor, en caso de no contestarlo durante dichas fechas se calificará el examen como "N / P" y no podrá ser nuevamente evaluado.
- Tendrá un tiempo límite de evaluación previamente establecido.
- En caso de no finalizar la evaluación y terminar su sesión, se calificará el examen correspondiente como "N / P" y no podrá ser nuevamente evaluado.
- Calificaciones. Búsqueda de evaluaciones por el nombre del examen o por materia. Campos validados por un servlet.
- Datos Personales. Muestra los datos personales del alumno para verificarlos y / o modificarlos. Campos validados por Javascript y un servlet.

- Cerrar Sesión. Finaliza la sesión del usuario.
- Exámenes Realizados. Búsqueda de exámenes realizados por el alumno, visualiza el examen, sus respuestas e indica cuales fueron acertadas y cuales erróneas.

Capítulo 7 CONCLUSIONES Y TRABAJO FUTURO

Conclusiones

La aplicación GEL (Gestión de Exámenes en línea) es una herramienta práctica y fácil de utilizar para cualquier persona sin conocimiento alguno de computación. Lo único que se requiere es una clave específica ya sea la del profesor o alumno según sea el caso, para tener una cuenta de usuario y así poder utilizar el sistema.

Ofrece varias ventajas:

- Para el profesor.

- Puede crear tres modalidades de exámenes: Falso / Verdadero, Opción Múltiple y Mixto.
- Señala el tiempo exacto que debe tener un examen.
- Señala las fechas en las cuales tendrá vigencia un examen, desde el día de inicio hasta el último día que se tiene para contestar el examen.
- Obtendrá las evaluaciones de sus alumnos en la búsqueda por nombre de examen o materia.
- Los resultados de cada examen indicarán los nombres de los alumnos y tres modalidades de evaluación: Aprobado, Reprobado y N / P (en caso de que haya pasado la fecha de vigencia del examen y no haya sido contestado).

- Para el alumno.

- Contesta sus exámenes respectivos a cada materia según su semestre y carrera.
- Las evaluaciones se dan de manera inmediata y exacta una vez terminado el examen.
- Posteriormente puede ver el examen que ha contestado, y saber cuáles de sus respuestas fueron acertadas y cuales erróneas.

- Tiene un control sobre sus evaluaciones al tener un sistema de búsqueda por nombre de examen o por materia, y así, estar al tanto de sus calificaciones.

Es un sistema en donde el alumno puede contestar sus exámenes a distancia y tener sus resultados inmediatos. Es una aplicación para todos los niveles educativos.

Cabe mencionar que no se ha contratado ningún servicio de *hosting* aún, pues no se han planteado específicamente las características de dicho servicio (costos, disponibilidad de espacio etc.) que la institución (UVAQ) requiere para tener esta aplicación funcional en internet.

Consideraciones

Como trabajo futuro se tiene contemplado adjuntar imágenes a los exámenes, donde puedan contener formulas, tablas, esquemas etc. que aporten un apoyo extra para la resolución del examen.

De igual manera incrementar las modalidades de exámenes como: Orden Predeterminado, Correlacional, Preguntas Abiertas, Completar la Frase, entre otras. Esta aplicación se contempló para ser utilizada principalmente a distancia, pero en caso de aplicar un examen en un laboratorio de cómputo, se realizará un sistema de preguntas al azar, donde el orden de preguntas para cada alumno sea distinto y así el profesor no tenga que preocuparse por estar vigilando a los alumnos.

En caso de que el alumno pase al siguiente semestre, realizar una actualización de dicho semestre al terminar el periodo escolar, sin perder los datos y evaluaciones ya realizadas por el alumno de sus semestres anteriores.

BIBLIOGRAFÍA

- [1] http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java visitado en el mes de marzo de 2009.
- [2] <http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html> visitado en el mes de marzo de 2009.
- [3] <http://www.desarrolloweb.com/articulos/831.php> visitado en el mes de marzo de 2009.
- [4] http://es.wikipedia.org/wiki/Java_Server_Pages visitado en el mes de marzo de 2009.
- [5] <http://websolutionsnic.blogspot.com/2008/09/los-diferentes-lenguajes-de-programacin.html> visitado en el mes de marzo de 2009.
- [6] <http://universitarios.universia.es/exámenes/durante/tipos-exámenes/index.htm> visitado en el mes de abril de 2009.
- [7] <http://www.uam.es/docencia/ace/23-eleccion.htm#emparejamiento> visitado en el mes de abril de 2009.
- [8] <http://espanol.scantron.com/software/examsystem.htm> visitado en el mes de mayo de 2009.
- [9] <http://foro.uptodown.com/showthread.php?t=15480> visitado en el mes de mayo de 2009.
- [10] <http://espanol.scantron.com/software/perception.htm> visitado en el mes de mayo de 2009.
- [11] http://es.wikipedia.org/wiki/Apache_Tomcat visitado en el mes de mayo de 2009.
- [12] <http://es.wikipedia.org/wiki/JDK> visitado en el mes de mayo de 2009.
- [13] http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/A_I.htm visitado en el mes de mayo de 2009.
- [14] <http://www.masadelante.com/faq-servidor-web.htm> visitado en el mes de mayo de 2009.
- [15] <http://www.masadelante.com/faq-hosting.htm> visitado en el mes de mayo de 2009.
- [16] http://72.14.205.132/translate_c?hl=es&sl=en&u=http://en.wikipedia.org/wiki/Virtual_hosting&prev=/search%3Fq%3Dweb%2Bservers%26hl%3Des%26sa%3DG&rurl=translate.google.com.mx&usq=ALkJrhh4WHIB9CDBALuCHih4IKVAM9Jcvg visitado en el mes de mayo de 2009.
- [17] <http://free-web-hosting-gratis.net76.net/pagina-web/definicion-y-tipos-de-web-hosting-o-alojamiento-web/> visitado en el mes de mayo de 2009.
- [18] <http://translate.google.com.mx/translate?hl=es&sl=en&u=http://www.jedit.org/&ei=ATgkSq6CAaiEtAOIrK2JBA&sa=X&oi=translate&resnum=1&ct=result&prev=/search%3Fq%3DJEedit%26hl%3Des%26client%3Dfirefox-a%26channel%3Ds%26rls%3Dorg.mozilla:es-ES:official%26sa%3DG> visitado en el mes de junio de 2009.

-
- [19] http://es.wikipedia.org/wiki/Plain_Old_Java_Object visitado en el mes de junio de 2009.
- [20] http://es.wikipedia.org/wiki/Java_Servlet visitado en el mes de junio de 2009.
- [21] <http://programas.navegalis.com/programa/29/phpmyadmin.php> visitado en el mes de Octubre de 2009.
- [22] Deitel, Harvey M., Deitel Paul J. *Como Programar en Java*. Pearson Educación. 5a. Edición. México 2004.
- [23] Jorge Salvador Regalado Meza, “*Apuntes, Diplomado en Programación de Aplicaciones Web*”, Lugar: UVAQ, Febrero – Mayo del 2009.
- [24] Misael Madrigal Torres, “*Tutorial JSP, Diplomado en Programación de Aplicaciones Web*”, Lugar: UVAQ, Febrero – Mayo del 2009.
- [25] <http://www.alegsa.com.ar/Dic/uml.php> visitado en el mes de noviembre de 2009.

ÍNDICE DE FIGURAS

Figura 2.1 Examen tipo orden predeterminado.	3
Figura 2.2 Examen tipo Correlacional.	5
Figura 2.3 Examen tipo Miscelaneo.	7
Figura 5.1 Estructura de archivos MVC.....	31
Figura 6.1 Interfaz Pagina Principal	36
Figura 6.2 Interfaz Menu Profesores.	38
Figura 6.3 Interfaz Menu Alumnos.	39
Figura 6.4 Interfaz Menu Noticias.....	40
Figura 6.5 Interfaz Menu Acerca de.	41
Figura 6.6 Interfaz Profesores.	44
Figura 6.7 Interfaz Crear Examen	46
Figura 6.8 Interfaz Abrir Examen.....	47
Figura 6.9 Interfaz Examen Falso / Verdadero.....	48
Figura 6.10 Interfaz Examen Opcion Multiple.....	49
Figura 6.11 Interfaz Opciones.	50
Figura 6.12 Interfaz Respuesta Opcion.....	51
Figura 6.13 Interfaz Examen Mixto.....	52
Figura 6.14 Interfaz Finalizacion.	53
Figura 6.15 Interfaz Buscar.	54
Figura 6.16 Interfaz Popup Profesor.	55
Figura 6.17 Interfaz Exámenes Realizados Profesor.	56
Figura 6.18 Interfaz Alumnos.	60
Figura 6.19 Interfaz Exámenes Alumno.	61
Figura 6.20 Interfaz Calificaciones Alumno.	62
Figura 6.21 Interfaz Popup Alumno.....	63
Figura 6.22 Interfaz Exámenes Realizados Alumno.....	64
Figura 6.23 Base de Datos GEL.....	68
Figura 6.24 Diagrama de Clases Login / Logout	78
Figura 6.25 Diagrama de Clases Registro y Actualizacion de Usuarios.....	79
Figura 6.26 Diagrama de Clases Busqueda de Evaluaciones.....	79
Figura 6.27 Diagrama de Clases Crear, Eliminar y Evaluar Exámenes	80
Figura 6.28 Diagrama de Clases Ingreso de Preguntas.....	80
Figura 6.29 Diagrama de Clases Visualizar Exámenes	81
Figura 6.30 Diagrama de Casos de Uso.	82

GLOSARIO DE TÉRMINOS

Script. Es un guión o conjunto de instrucciones. Permiten la automatización de tareas creando pequeñas utilidades. Son ejecutados por un intérprete de línea de órdenes y usualmente son archivos de texto.

Multithreading. Es una manera de programar software que permite compartir el mismo estado de memoria entre varios hilos de ejecución(procesos que representan una secuencia simple de instrucciones ejecutada en paralelo con otras secuencias) Los hilos permiten dividir un programa en dos o más tareas que corren simultáneamente.

Intranet. Es una red de ordenadores privados que utiliza tecnología Internet para compartir de forma segura cualquier información o programa del sistema operativo para evitar que cualquier usuario de Internet pueda ingresar.

Browser. Es un programa que permite visualizar la información que contiene una página *Web* (ya esté alojada en un servidor dentro de la *World Wide Web* o en uno local).

Hardware. Corresponde a todas las partes físicas y tangibles de una computadora: sus componentes eléctricos, electrónicos, electromecánicos y mecánicos; sus cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado.

Software. Equipamiento lógico o soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica. Tales componentes lógicos incluyen, entre otros, aplicaciones informáticas, software de sistema y el resto de las aplicaciones.

Byte code. Es un código intermedio, es tratado como un archivo binario que contiene un programa ejecutable, que es un archivo binario producido por el compilador cuyo contenido es el código objeto o código máquina.

CD –ROM. Por sus siglas del inglés *Compact Disc - Read Only Memory*, "Disco Compacto - Memoria de Sólo Lectura", es un disco compacto utilizado para almacenar información no volátil.

PC. Por sus siglas en inglés *Personal Computer*. Computadora personal que suele estar equipada para cumplir tareas comunes de la informática.

Windows. Es una familia de sistemas operativos desarrollados y comercializados por Microsoft. Existen versiones para hogares, empresas, servidores y dispositivos móviles.

UNIX. Es un sistema operativo portable, multitarea y multiusuario.

Mac. También llamada Macintosh, es el nombre con el que actualmente nos referimos a cualquier computadora personal diseñada, desarrollada, construida y comercializada por *Apple Inc.*

Qualrus. Es una herramienta de análisis de datos cualitativos distingue por su rápida curva de aprendizaje, sin la codificación de la eficiencia y flexible lenguaje de scripting.

LAN. Por sus siglas en inglés *Local Area Network* es la interconexión de varios ordenadores y periféricos para crear una red de área local.

GNU. Significa GNU No es Unix (*GNU is Not Unix*), sistemas desarrollado con el objetivo de crear un sistema operativo completamente libre.

Shell. Es un programa informático que actúa como interfaz de usuario para comunicar al usuario con el sistema operativo mediante una ventana que espera órdenes escritas por el usuario en el teclado, los interpreta y los entrega al sistema operativo para su ejecución.

Framework. Es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.

Webmaster. Término comúnmente usado para referirse a las personas responsables de un sitio *Web* específico.

UML. Por sus siglas en inglés "*Unified Modeling Language*" Lenguaje de Modelado Unificado, es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos.

XML. Por sus siglas en inglés "*Extensible Markup Language*", o lenguaje de anotación extensible. Permite que los diseñadores creen sus propias etiquetas, así como la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones.

DTD. Por sus siglas en inglés "*Document Type Definition*", especifica cuáles son los tipos de elementos y atributos que podemos utilizar en nuestra página web. También define las reglas de cómo podemos utilizar estos elementos juntos, de acuerdo a la especificación del lenguaje de marcado

Logs. Son registros de actividad de un sistema, que generalmente se guardan en un archivo de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema.

Hotlinking. Es cuando se extrae la imagen o cualquier otro contenido de una web para colocarlo en otra, esto sin que lo permita el propietario de la *Web*.

Div. Es una etiqueta de un archivo HTML la cual describe una división o espacio, donde se pueden contener todo tipo de información como: texto, imágenes, video etc.

Struts. Es un marco de trabajo (*framework*) utilizado para desarrollar aplicaciones *Web* mediante la plataforma J2EE (*Java 2 Enterprise Edition*). Implementa el patrón Modelo Vista Controlador (MVC) y se distribuye como software libre por la *Apache Software Foundation*.

