



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

PLATAFORMA WEB DE EVALUACIÓN DIFUSA CON INFORMACIÓN HETEROGÉNEA

Alumno: Antonio Javier Sánchez Santiago

Tutores: Luis Martínez López
Pedro J. Sánchez Sánchez

Dpto: Informática

Septiembre, 2007

Índice

CAPITULO 1: Introducción.....	5
1.1 Introducción al proyecto.....	6
1.2 Propósito.....	8
1.3 Objetivos.....	9
CAPITULO 2: Antecedentes.....	11
2.1 Modelos de Evaluación basados en Procesos de Toma de Decisiones.....	12
2.2 Marco de evaluación: Herramientas y modelos.....	15
2.2.1 Marco de Evaluación.....	15
2.2.2 Proceso de Evaluación.....	17
CAPITULO 3: Proyecto.....	21
3.1 Introducción.....	22
3.2 Especificación de requerimientos	23
3.2.1 Requerimientos funcionales	24
3.2.2 Requerimientos no funcionales	28
3.3 Análisis del sistema	33
3.3.1 Perfil de usuario	33
3.3.2 Casos de uso	39
3.3.3 Escenarios	52
3.4 Diseño del sistema	54
3.4.1 Diagrama de clases	56
3.4.1.1 Diagrama completo de clases	57
3.4.1.2 Diagrama por paquete	60
3.4.1.2.1 Paquete Base de Datos.....	60
3.4.1.2.2 Paquete Modelo	61
3.4.1.2.3 Paquete Controlador	62
3.4.1.2.4 Paquete HeterogeneousInformation	63
3.4.1.2.5 Paquete EncuestaEvaluacion	64
3.4.2 Diseño de los datos	65
3.4.3 Diseño de la interfaz	75
3.4.3.1 Definir estilo	76
3.4.3.2 Metáforas	78
3.4.3.3 Pantallas	79
3.4.3.4 Caminos de navegación	80
3.4.3.4.1 Storyboard Crear Encuesta	82
3.4.3.4.2 Storyboard Nueva Pregunta	83
3.4.3.4.3 Storyboard Modificar Pregunta	84
3.4.3.4.4 Storyboard Alumno Contesta Encuesta	86
3.4.3.4.5 Storyboard Profesor Consulta Resultados	88

3.4.3.5 Mensajes de error	89
3.5 Implementación	91
3.5.1 Tipo de arquitectura de la aplicación	92
3.5.2 Lenguajes de programación utilizados	93
3.5.3 Herramienta de desarrollo	95
3.5.4 Instalación en el servidor y funcionamiento	95
3.6 Implantación y pruebas	96
3.6.1 Pruebas y validación	96
3.6.1.1 Casos de Test	96
3.6.1.2 Resultados obtenidos	108
CAPITULO 4:Conclusiones.....	113
4.1 Conclusiones	114
4.2 Mejoras futuras	115
Bibliografía.....	117
ANEXO I: Descripción del proceso de evaluación	121
I.1 Introducción	122
I.2 Enfoque Lingüístico Difuso	126
I.2.1 Modelado Lingüístico basado en 2-tuplas	127
I.3 Modelo propuesto de decisión con información heterogénea	131
I.3.1 Fase de agregación	132
I.3.2. Fase de explotación	142
I.4 Un problema TDG con información no homogénea	143
I.5 Conclusiones	146
Bibliografía Anexo I	147
ANEXO II: Manual de Instalación.....	151
ANEXO III: Manual de Usuario: Administrador	159
ANEXO IV: Manual de Usuario: Alumno.....	195
ANEXO V: Manual de Usuario: Profesor.....	211

Plataforma web de evaluación difusa
con información heterogénea

Antonio Javier Sánchez Santiago

CAPITULO 1

Introducción

1.1 Introducción al proyecto

Actualmente, la evaluación del profesorado se realiza mediante una encuesta con distintas preguntas que es rellenada por cada uno de los alumnos de un profesor en una asignatura. Las respuestas a todas las preguntas se evalúan en una escala numérica discreta de 1 a 5, a pesar de que muchas de las preguntas están relacionadas con percepciones propias del alumno que son difíciles de cuantificar de forma precisa ya que, implican incertidumbre en el conocimiento de los alumnos. Además, los indicadores evaluados en estas encuestas no tienen siempre la misma naturaleza por lo que, unos pueden ser evaluados fácilmente mediante información precisa (indicadores cuantitativos) mientras que otros se adaptarán mejor a una evaluación cualitativa.

Dada la situación anterior, nos proponemos crear un sistema para la evaluación del profesorado que sea capaz de tener como información entrada, información valorada en distintos dominios. Esto es debido a que normalmente en los tests de evaluación de los docentes aparecen preguntas que tienen distinta naturaleza y que podrían ser valoradas mediante valores numéricos, lingüísticos e incluso intervalares, dependiendo de la misma. Por tanto nuestro sistema será capaz de manejar esta información heterogénea, para ello utilizaremos modelos basados en la lógica difusa y en el análisis de decisiones. De esta forma obtener un valor global que evalúe al docente en cuestión mediante un valor global cualitativo en una escala predefinida a partir de los datos de entrada.

En este proyecto, tal y como veremos en detalle posteriormente, se propone el desarrollo de un sistema de evaluación Cliente/Servidor WEB con tecnología JAVA basándonos en el prototipo del proyecto [Mar05] que ayude a realizar de forma automática y flexible los procesos de evaluación del profesorado universitario que se lleva a cabo en todas y cada una de las Universidades Andaluzas.

Con la utilización de una plataforma WEB se consigue una mayor flexibilidad a la hora de

poder acceder al sistema ya que sólo se requiere que esté instalado en el ordenador servidor y los ordenadores clientes no requieren de ningún software especial. Sólo necesitan tener un navegador WEB en su ordenador para poder acceder al sistema.

Una vez comentados la situación actual y el porqué del proyecto podemos pasar a introducir brevemente en qué consiste el proyecto que se documenta en esta memoria.

En este proyecto se pretende construir un sistema Web con tecnología JAVA que automatice el proceso de evaluación docente. Este sistema puede ser utilizado fácilmente en cualquier momento a otro ámbito de evaluación.

Actualmente se encuentran desarrollados los siguientes elementos:

- La Base de Datos, tanto el diseño de tablas como los ficheros necesarios para la misma. La Base de Datos se encuentra desarrollada completamente con tecnología JAVA.
- Biblioteca para el tratamiento de la Información Heterogénea implementada en JAVA.
- Biblioteca para el modelo de evaluación basado en un proceso de decisión difuso con Información Heterogénea

Por tanto este proyecto esta centrado en la implementación del sistema con una arquitectura Cliente/Servidor y en la creación de las interfaces necesarias para cada uno de los usuarios presentes en el sistema. Dichas interfaces estarán desarrolladas con tecnología JAVA, es decir, utilizando Servlets y JSP para el tratamiento de la información entre el servidor y los distintos clientes del sistema.

Por último se expondrán una serie de conclusiones sobre el desarrollo de este proyecto así como diversos manuales y cualquier descripción que se considere útil para el correcto entendimiento y funcionamiento de los modelos y aplicaciones desarrolladas en este proyecto. Finalmente, para aquellos lectores que estén interesados en el tema, se incluirá una amplia bibliografía.

1.2 Propósito

La principal tarea del proyecto es desarrollar un sistema con arquitectura Cliente/Servidor basada en WEB y con tecnología JAVA que lleve a cabo de forma automática procesos de evaluación en contextos heterogéneos y cálculo de resultados. Para así, de esta forma poder aplicar el sistema de evaluación del profesorado y posteriormente ampliarlo fácilmente a cualquier contexto de evaluación.

1.3 Objetivos

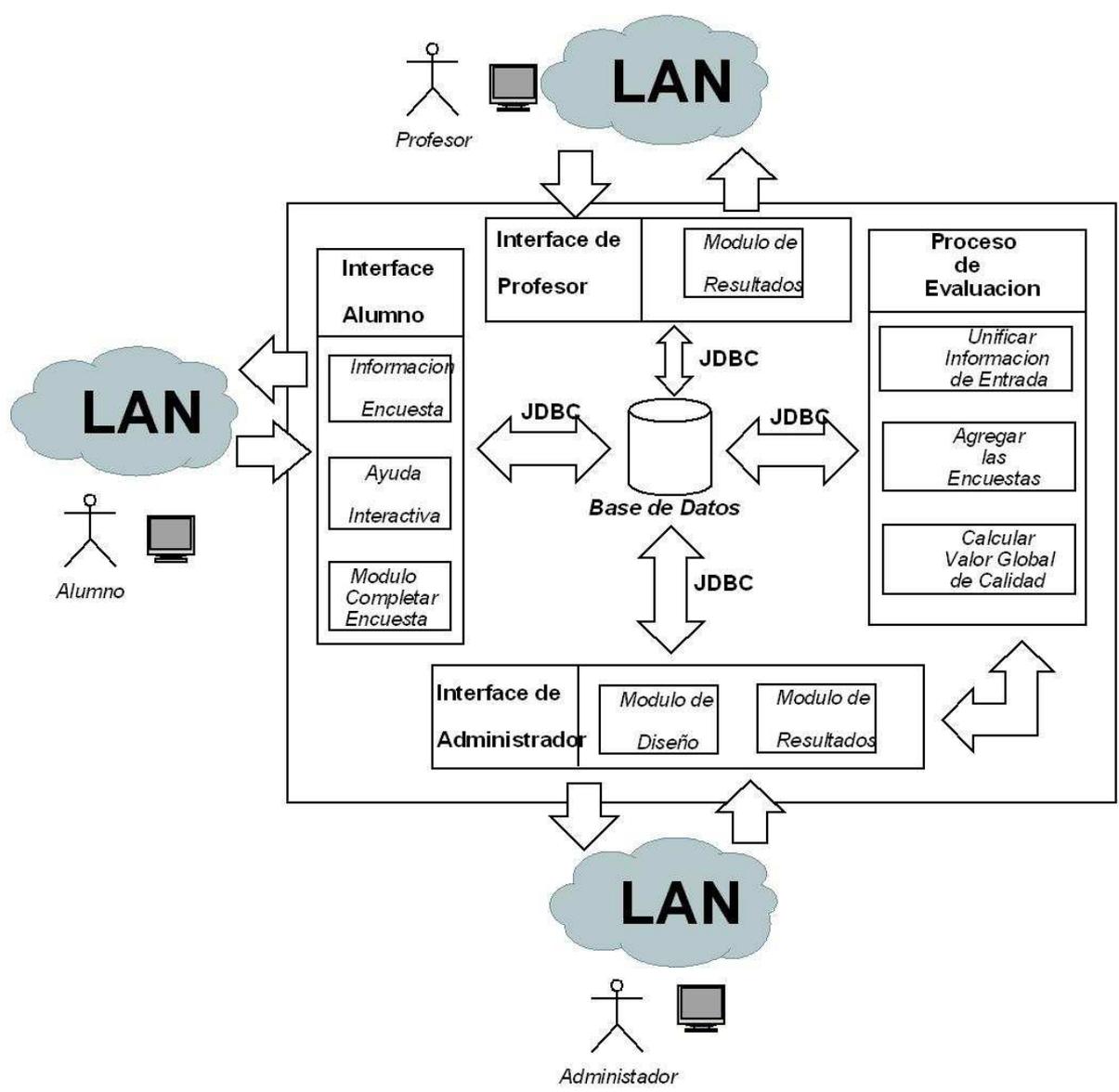


Figura 1.1: Arquitectura Cliente/Servidor de la aplicación

El objetivo principal del proyecto será la construcción de un módulo web que implemente la arquitectura que se muestra en la Figura 1.1. Para alcanzar este objetivo principal habrá que cumplir otros objetivos como son:

1. Búsqueda y revisión bibliográfica.
2. Analizar el sistema existente.
3. Diseño y análisis del sistema con arquitectura Cliente/Servidor.
4. Implementar el acceso de usuario usando un sistema con arquitectura Cliente/Servidor. De forma que los alumnos y profesores puedan acceder a las encuestas mediante Servlets JAVA en un entorno html fácil de entender por ellos.
5. Realizar manuales para el uso correcto de la aplicación.

Para alcanzar los objetivos propuestos, en primer lugar había que desarrollar una metodología matemática que nos proporcionase las herramientas para manejar contextos heterogéneos y pudiésemos obtener los resultados que buscábamos. Estos objetivos ya fueron alcanzados a la conclusión del proyecto [Mar05], los cuales se detallan en el siguiente capítulo.

CAPITULO 2

Antecedentes

2.1 Modelos de Evaluación basados en Procesos de Toma de Decisiones

La evaluación es un área que involucra procesos complejos para la obtención de un resultado global que nos indique la valoración global del elemento, modelo, producto, persona o sistema, que se esté evaluando.

En la literatura científica podemos encontrar como problemas de evaluación complejos pueden ser resueltos mediante técnicas relacionadas con la Toma de Decisiones [Che01,Dev99,Her03,Her03a]. Incluso estas técnicas junto al uso de la Lógica Difusa nos permiten introducir una mayor flexibilidad en el modelado de la información que se manejará en el proceso de evaluación. A continuación haremos una breve introducción sobre la Toma de Decisiones y su relación con los problemas de evaluación.

La Toma de Decisiones (TD) es sin duda una de las actividades fundamentales de los humanos, ya que constantemente nos estamos enfrentando a situaciones en las que existen varias alternativas y, al menos en algunas ocasiones, tenemos que decidir cuál es mejor, o cuál llevar a cabo. La TD se aplica en distintas disciplinas, tales como, las Ciencias Sociales, la Economía, la Ingeniería, etc. Esta amplia gama de campos de aplicación tiene como consecuencia la existencia de diferentes modelos de Toma de Decisión [Bui87], que han dado lugar a la Teoría de Decisión.

En la TD nos podemos encontrar distintos tipos de situaciones de decisión dependiendo del contexto en el que se vaya a desarrollar el problema de decisión:

1. Ambiente de certidumbre.
2. Ambiente de riesgo.
3. Ambiente de incertidumbre.

La Teoría Clásica de Decisión proporciona una gran cantidad de modelos sobre las distintas situaciones de decisión, aunque no es capaz de dar solución a todos los problemas de decisión. Los métodos clásicos se adecuan fácilmente a problemas desarrollados en ambiente de certidumbre y de riesgo, sin embargo no son adecuados en situaciones de incertidumbre, es decir, en problemas que presentan información vaga e imprecisa como los del problema que afrontamos en este proyecto. En estas situaciones hablamos de problemas de decisión en contexto difuso o de Toma de Decisiones Difusa [Car01, Tri00].

En la TD hay dos áreas principales a la hora de resolver los problemas: la *Selección de Alternativas y el Consenso*. En una primera aproximación a los problemas de “evaluación” mediante técnicas de toma de decisiones el área más interesante es el de Selección de alternativas.

Selección de alternativas: en estos procesos se busca el conjunto solución de alternativa/s que mejor se adecue al problema de Decisión. En este proceso distinguimos dos fases [Rou97]: (i) Una fase de Agregación en la que se combinan las preferencias de los expertos para obtener una preferencia colectiva sobre las alternativas del problema y (ii) una fase de Explotación en la que a partir de las preferencias colectivas se aplicará un grado de selección para obtener un conjunto solución de alternativas para el problema.

1. Los *procesos de agregación*, juegan un papel fundamental en la resolución de problemas de Decisión. En la TD podemos encontrarnos problemas en los que distintas fuentes de información pueden tener distinto grado de conocimiento sobre los aspectos del problema lo que puede originar que las preferencias estén modeladas con información de distinta naturaleza o expresadas sobre distintas estructuras [Chi98, Her00c, Her01a], lo cuál dificulta los procesos de agregación. El modelado de preferencias es un elemento muy importante en los procesos de TD ya que se utiliza para expresar conocimiento, gustos, afinidades, etc. En problemas de “evaluación”

la valoración de aspectos cualitativos mediante valores precisos no es muy adecuada, por lo que, el uso del Enfoque Lingüístico [Zad75] basado en la Teoría de Conjuntos Difusos [Zad65, Dub80] se ha mostrado útil a la hora de modelar este tipo de preferencias en TD difusa. El uso del enfoque lingüístico implica la necesidad de realizar procesos de operar con palabras, denominados en inglés como Computing with words (CW) [Her00a]. Estos procesos se han llevado a cabo en la TD difusa utilizando distintos modelos:

- Modelo basado en el Principio de Extensión [Deg88]. En el cuál las operaciones se realizan utilizando la aritmética difusa [Dub80] sobre los números difusos que soportan la semántica de las etiquetas lingüísticas.
- Modelo Simbólico [Del93]. Las operaciones se realizan sobre los índices de las etiquetas lingüísticas que son valores discretos.
- Modelo basado en la 2-tupla lingüística [Her00a, Her01a]. Las operaciones se realizan sobre información lingüística expresada mediante 2-tuplas, que permite trabajar en un dominio de expresión lingüístico pero tratándolo como un universo continuo.

Los dos primeros modelos basados en el uso del Enfoque Lingüístico presentan, una serie de limitaciones a la hora de resolver problemas definidos en contextos complejos como los mostrados en [Del98] y en [Her01b]. El modelo de representación de información lingüística denominado 2-tuplas Lingüísticas [Her00a, Her01a] permite abordar algunos de estos problemas abiertos. Lo que ha supuesto un nuevo enfoque a la hora de diseñar modelos de resolución de problemas de TD difusa porque obtiene resultados más precisos y facilita el diseño de modelos de decisión sobre

problemas definidos en contextos heterogéneos [Chi98, Her00c, Her01a, Her05]. Los problemas de “evaluación” de calidad suelen evaluar tanto criterios cuantitativos como cualitativos por lo que el uso de una metodología que nos proporcione flexibilidad a la hora de trabajar con este tipo de información es muy importante.

2. *Procesos de Explotación*, en la TD normalmente se aplica un grado de selección para ordenar las distintas alternativas. Sin embargo, cuando utilizamos los procesos de TD en problemas de “evaluación” podemos encontrarnos que en ocasiones es necesario ordenar las alternativas que se evalúan [Her03] o bien a veces lo que necesitamos es obtener un valor general del elemento o elementos evaluados para saber cuál es su grado de calidad [Her03a]. Por tanto, en los procesos evaluativos dependiendo del tipo de evaluación abordada diseñaremos distintas fases de explotación para obtener el resultado buscado en cada tipo de problema.

2.2 Marco de evaluación: Herramientas y modelos

Antes de poder implementar el Sistema de Evaluación en un entorno WEB con tecnología JAVA, hay que definir el marco de evaluación en el que se va a llevar a cabo el proceso evaluativo y los modelos y procesos que se utilizarían para obtener los resultados buscados en la evaluación.

El modelo teórico desarrollado es el siguiente:

2.2.1 Marco de Evaluación

El contexto en el que se desenvuelve la evaluación de cualquier servicio puede describirse con la estructura de un problema decisión Multi-Experto ya que, para obtener una evaluación

se utilizan las opiniones de distintos actores (usuarios, expertos, etc.) que participan en la evaluación.

Por tanto el marco de evaluación se describe como:

$$\textit{Expertos (usuarios, expertos)} \quad E = \{e_1, \dots, e_n\} \quad n \geq 2$$

$$\textit{Indicadores} \quad X = \{x_1, \dots, x_m\} \quad m \geq 2$$

$$\textit{Vectores de Utilidad} \quad V_{ij}^k \{S_{i1}^k, \dots, S_{im}^k\}$$

$$\textit{Dominios} \quad D^K$$

Un grupo de expertos formado por los usuarios y opcionalmente otro conjunto de observadores evalúan una serie de indicadores sobre un elemento a evaluar. La información para la evaluación es organizada mediante vectores de utilidad tales que, cada evaluador proporciona su vector de utilidad. En dichos vectores hay un valor para cada indicador, S_{ij}^k siendo i el identificador del experto que proporciona el vector de utilidad, j el indicador evaluado y k el dominio de expresión utilizado:

$$S_{ij}^k \quad (i \in \{1, \dots, m\}, j \in \{1, \dots, n\}, k \in \{N, I, L\})$$

Los valores de los distintos indicadores pueden ser valorados en distintos dominios (Numérico, Intervalar y Lingüístico). El dominio de expresión de cada indicador se decide después de estudio realizado según modelos de la psicología para definir escalas de valoraciones.

2.2.2 Proceso de Evaluación

En este momento describiremos resumidamente el modelo matemático utilizado para realizar la evaluación de los docentes a partir de información valorada en un contexto heterogéneo con información numérica, intervalar y lingüística. Una descripción completa del modelo puede ser consultada en el Anexo I de este documento.

Al haber modelado el problema de evaluación como un problema de decisión multi-experto, utilizaremos un modelo de decisión con las fases mostradas en la Figura 2 para obtener un valor de la calidad de los distintos indicadores de cada profesor:

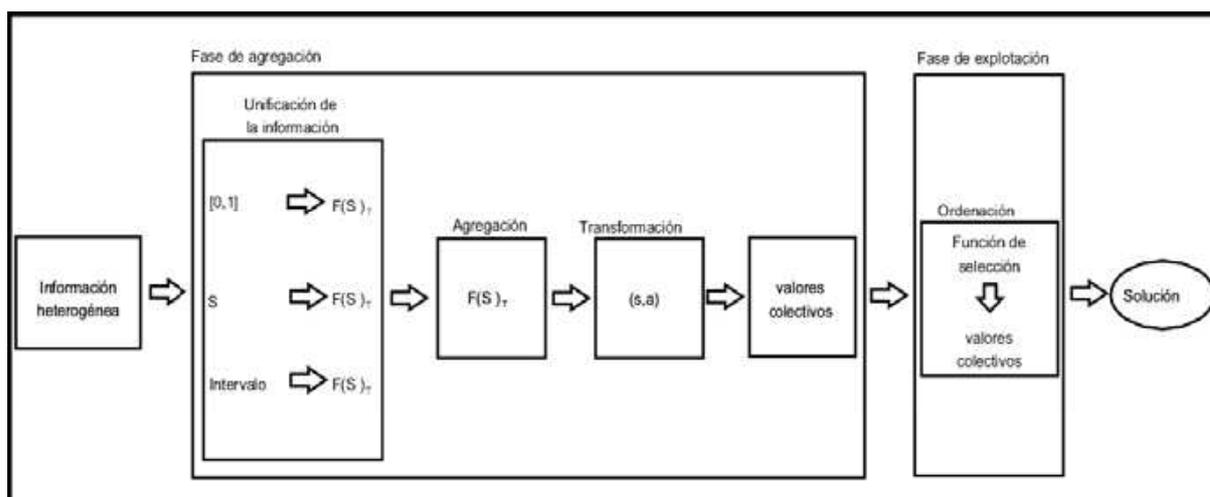


Figura 2.1: Modelo de decisión con Información Heterogénea

A. Recolección de información:

En la fase de recolección se obtiene la información heterogénea que será utilizada en el modelo de decisión para obtener una solución.

B. Fase de Agregación:

En la fase de agregación perseguimos obtener un vector de utilidad que exprese la evaluación global del docente por parte de todos los alumnos. Para ello se realizarán los siguientes pasos:

Unificación de la Información:

Al estar la información expresada en distintos dominios es necesario, para operar con ella, unificarla en un único dominio de expresión. Para llevar a cabo este proceso de unificación se utilizará el modelado lingüístico basado en 2-tuplas [Her00a]. La selección de este modelo de representación se debe a que se ha demostrado como muy útil a la hora de trabajar con información expresada en contextos no homogéneos [Her00,Her01a,Her05,Liu04,Xu04]. Para una descripción en profundidad de este modelo de representación de información consultar [Her00a].

Pero para unificar mediante 2-tuplas la información heterogénea en primer lugar tenemos que unificarla mediante conjuntos difusos valorados en un conjunto básico de términos lingüísticos denominado S_T . Para ello, se definen funciones de transformación de información numérica, intervalar y lingüística en conjuntos difusos sobre un conjunto de etiquetas básico. La información de entrada heterogénea ya está expresada de forma uniforme mediante conjuntos difusos en un conjunto básico de términos lingüísticos S_T . Sin embargo, este modelado de la información es difícil de entender y de manejar matemáticamente, por lo que, transformaremos estos conjuntos difusos en 2-tuplas valoradas en S_T para mejorar el entendimiento de los valores

de evaluación obtenidos y mejorar los procesos matemáticos sobre la información de entrada.

De esta forma la información proporcionada por los alumnos y/o distintos expertos en las encuestas está expresada de forma uniforme en un único dominio de expresión lingüístico mediante 2-tuplas lingüísticas. El objetivo de la agregación es obtener un vector de utilidad colectivo para cada profesor. Esto se consigue en el siguiente paso de esta fase.

– Agregación de la Información:

Una vez la información ha sido expresada de forma uniforme agregamos todos los vectores de utilidad individuales proporcionados por los distintos expertos para obtener un vector de utilidad colectivo mediante operadores de agregación para 2-tuplas [Her00a]. Los valores de este vector de utilidad colectivo serán los que se utilicen para obtener la evaluación en cuestión.

C. Fase de Explotación:

Esta fase tiene como entrada un vector de utilidad colectivo expresado mediante 2-tuplas lingüísticas. A partir de este vector de utilidad colectivo calcularemos un valor de evaluación utilizando un operador de agregación que pondere de forma adecuada los distintos parámetros que estamos evaluando de forma similar a la utilizada en [Her03a].

En esta fase obtenemos un valor global de calidad del docente a partir de los datos iniciales con los que partíamos. Este es el objetivo principal de esta

aplicación, pero a partir de la información que hemos utilizado para obtener este valor global podemos desarrollar nuevos módulos que realicen recomendaciones o realicen informes automáticos sobre lo evaluado.

Una descripción completa y detallada de este proceso de evaluación puede verse en el Anexo I del documento.

CAPITULO 3

Proyecto

3.1 Introducción

Una vez presentado el proyecto a grandes rasgos con su propósito, objetivos y realizada una introducción al modelo teórico que vamos a utilizar para la evaluación llega el momento de detallar el desarrollo del proyecto realizado.

El proyecto aborda el desarrollo de un sistema Cliente/Servidor basado en WEB con tecnología JAVA que realiza de forma automática el proceso de evaluación descrito anteriormente.

En este capítulo se detallan las fases de Ingeniería del Software para el desarrollo de un prototipo software.

No existe una definición única y estandarizada para la Ingeniería del Software pero las dos que se presentan a continuación pueden resultar válidas:

- Ingeniería del Software es la construcción de software de calidad con un presupuesto limitado y un plazo de entrega en contextos de cambio continuo.
- Ingeniería del Software es el establecimiento y uso de principios y métodos firmes de ingeniería para obtener software económico que sea fiable y funcione de manera eficiente en máquinas reales.

Las actividades que conforman la Ingeniería del Software son las siguientes:

- **Especificación de Requerimientos:** se obtienen el propósito del sistema y las propiedades y restricciones del mismo.

- **Análisis del Sistema:** se obtiene un modelo del sistema correcto, completo, consistente, claro y verificable.
- **Diseño del Sistema:** se definen los objetivos del proyecto y las estrategias a seguir para conseguirlos.
- **Implementación:** se traduce el modelo a código fuente.

En los puntos siguientes se profundizará en cada una de estas actividades y en como se han llevado a cabo en el ámbito nuestro proyecto.

3.2 Especificación de requerimientos

El primer paso en la Ingeniería del Software debe ser determinar el propósito último del proyecto, las propiedades que debe satisfacer y las restricciones a las que está sometido. Este es, sin duda, una paso de vital importancia dentro del desarrollo de un proyecto software ya que, sin conocer el propósito del proyecto y todas las limitaciones de diversa índole a las que debe hacer frente, difícilmente se podrá realizar una aplicación software que cumpla dicho propósito.

En un proyecto de ámbito comercial para una empresa real, para determinar el propósito del mismo se recurre a una serie de estudios como pueden ser entrevistas con los clientes, encuestas con posibles usuarios, estudios de la situación actual del sistema o estudios de viabilidad. En nuestro caso no nos encontramos ante un proyecto comercial sino ante uno académico por lo que el propósito es conocido desde el mismo momento de la concepción del mismo:

La principal tarea del proyecto es desarrollar un sistema con arquitectura Cliente/Servidor basada en WEB y con tecnología JAVA que lleve a cabo de forma automática procesos de evaluación en contextos heterogéneos y cálculo de resultados. Para así, de esta forma poder aplicar el sistema de evaluación del profesorado y posteriormente ampliarlo fácilmente a cualquier contexto de evaluación.

Una vez determinado el propósito último del proyecto, el siguiente paso consiste en especificar los requerimientos del mismo. Los requerimientos de un proyecto software son el conjunto de propiedades o restricciones definidas con total precisión, que dicho proyecto software debe satisfacer. Existen dos tipos bien diferenciados de tales requerimientos:

Requerimientos funcionales: aquellos que se refieren específicamente al funcionamiento de la aplicación o sistema.

Requerimientos no funcionales: aquellos no referidos al funcionamiento estricto sino a otros factores externos.

En los dos siguientes subapartados se pasarán a definir cuales son estos requerimientos (tanto funcionales como no funcionales) para el proyecto del que se ocupa esta memoria. Sin embargo, estas definiciones sólo serán previas ya que en la actividad de análisis del sistema, donde se crearán los casos de uso y sus escenarios, se descubrirán nuevas necesidades que no son observables en esta primera actividad y que permitirán refinar completamente estos requerimientos.

3.2.1 Requerimientos funcionales

Los requerimientos funcionales de un sistema software son aquellos que se encargan de describir las funcionalidades que el sistema debe proporcionar a los usuarios del mismo para cumplir sus expectativas.

Normalmente, estos requerimientos se obtendrían de la interacción con el cliente mediante diversas entrevistas y/o encuestas. En nuestro caso, al tratarse de un proyecto académico, nos encontramos ante la situación de la no existencia de cliente alguno por lo que la información sobre las funcionalidades que debe disponer el sistema se han obtenido de las necesidades de profesores y alumnos que actualmente realizan este tipo de encuestas de forma manual.

En base a estas investigaciones realizadas se ha llegado a la conclusión de que las funcionalidades que el usuario potencial espera de un sistema como el nuestro son las siguientes:

- Para el usuario Administrador:
 - Insertar/Modificar alumnos, profesores y asignaturas de la base de datos.
 - Asignar a un alumno las asignaturas que tiene que evaluar.
 - Consultar los datos de los alumnos y de los profesores.
 - Crear y modificar encuestas para ser evaluadas.
 - Asignar encuestas a asignaturas.
 - Abrir y cerrar el plazo para evaluar una asignatura.
 - Consultar resultados de una encuesta.

- Para el alumno:
 - Rellenar una encuesta.
- Para el profesor:
 - Consultar los resultados de una encuesta asociada a la asignatura que imparte docencia.

Una vez definidas cuales son las funcionalidades que los distintos usuarios pueden reclamar a nuestro sistema, se hace necesario caracterizar de una manera más formal y concreta como va a responder a estas funcionalidades nuestro sistema. Además, para distinguir cada uno de los requisitos utilizaremos este identificador: RF-Número del Requisito:

1) RF-01:Insertar/Modificar alumnos, profesores y asignaturas de la base de datos.

El sistema debe proporcionar al administrador la posibilidad de crear nuevos usuarios y nuevos profesores para ser evaluados, así como modificarlos una vez que están creados.

2) RF-02:Asignar a un alumno el profesor que tiene que evaluar.

El sistema debe proporcionar al administrador la posibilidad de asignar la asignatura de un profesor a un alumno para que este la pueda evaluar.

3) RF-03:Consultar los datos de los alumnos y de los profesores.

El sistema debe permitir que el administrador consulte los datos de todos los alumnos y de los profesores.

4) RF-04: Crear y modificar encuestas para ser evaluadas.

El sistema debe proporcionar la posibilidad de crear encuestas de evaluación y permitir la modificación de estas antes de que estén asignadas a una evaluación.

5) RF-05: Asignar encuestas a profesores.

El sistema debe permitir al administrador del mismo asignar una o varias encuestas a uno o varios profesores.

6) RF-06: Abrir y cerrar el plazo para evaluar un profesor.

El sistema debe proporcionar al administrador la posibilidad de abrir el plazo para que los alumnos rellenen la encuesta asignada a una asignatura en la que se encuentran matriculados. También debe proporcionar la posibilidad de cerrar el plazo para cumplimentar dicha encuesta.

7) RF-07: Consultar resultados de una encuesta.

Debe permitir al administrador obtener y consultar los resultados de una evaluación, una vez que esté cerrado el plazo para evaluar al profesor.

Estas son las funcionalidades que debe proporcionar nuestro sistema de evaluación teniendo en cuenta que este proyecto va a desarrollar una versión prototipal del mismo. En una versión final del sistema, este debería satisfacer otras funcionalidades, como podrían ser:

- Automatización de la apertura y cierre del plazo de evaluación de un profesor.
- Mandar un correo electrónico a los alumnos cuando este abierto el plazo para rellenar las encuestas asociadas a sus asignaturas.
- Mandar un correo electrónico a los profesores para indicarle que ya se encuentran

disponibles los resultados asociados a una de sus asignaturas.

- Posibilidad de eliminar alumnos, profesores o encuestas.
- Consultas avanzadas, con parámetros, de alumnos y de profesores.
- Crear nuevos dominios de valoración.
- Crear una base de datos vacía, a partir de la que tenemos actualmente.

3.2.2 Requerimientos no funcionales

Los requerimientos no funcionales son aquellos que restringen los requerimientos funcionales. Son tan importantes como los propios requerimientos funcionales y pueden incluso a llegar a ser críticos para la aceptación del sistema. Estos requerimientos normalmente especifican propiedades del sistema o del producto en si (plataforma, velocidad, rendimiento...) y del diseño de la interfaz gráfica con el usuario además de todas las restricciones impuestas por la organización (políticas de empresa, estándares, legalidad vigente...).

Al no ser este un proyecto comercial para ninguna organización o empresa real, no debemos someternos a restricciones de la organización. Por lo tanto, los requerimientos no funcionales que se deben obtener y analizar son los referentes a las necesidades hardware y software de los equipos informáticos para que estos proporcionen al usuario las funcionalidades requeridas de forma eficiente y los referentes a la interfaz gráfica entre la aplicación y el usuario.

A) Requerimientos del equipo informático

Al hablar de los requerimientos del equipo informático y debido a que el marco del desarrollo de la aplicación es una arquitectura Cliente/Servidor, debemos diferenciar los requerimientos de equipo que necesita el servidor y los que necesita el cliente.

Las necesidades de equipo informático del cliente son muy simples ya que tan solo le hace falta un computador conectado a Internet (preferiblemente de banda ancha) y tener instalado un navegador capacitado para visualizar de forma correcta la aplicación (se recomienda Firefox).

Los requerimientos del equipo informático del servidor, el cual se aconseja que sea un equipo dedicado, son más amplios y se dividen en dos tipos: los requerimientos de hardware y los requerimientos software.

1) Hardware

- ✓ *Velocidad:* el equipo debe ser lo suficientemente rápido como para ejecutar la aplicación en el menor tiempo posible y con la mayor fiabilidad. Cualquier microprocesador actual es capaz de cumplir con esta labor.
- ✓ *Memoria:* el equipo debe disponer de la suficiente memoria RAM libre par realizar las operaciones que se soliciten entre la aplicación y la base de datos.
- ✓ *Almacenamiento:* el equipo que haga la labor de servidor debe tener una capacidad de almacenamiento suficiente para almacenar la base de datos con la que trabaja la aplicación y permitir con holgura las transacciones entre ambas entidades.

- ✓ *Tarjeta gráfica:* las tarjetas gráficas de las que disponen los equipos informáticos actuales son de gran potencia por lo que cualquiera de ellas puede ser utilizable.
- ✓ *Monitor:* el monitor debe soportar una resolución de 800x600 y superiores.
- ✓ *Conexión a Internet:* el servidor debe encargarse de que la aplicación sea accesible a través de Internet para todos sus usuarios por lo que es indispensable que se encuentre conectado a Internet a través de banda ancha las 24 horas del día.

2) Software

- ✓ *Sistema Operativo:* el servidor de la aplicación puede trabajar sobre un sistema operativo Windows XP o Linux.
- ✓ *Navegador:* la aplicación debe poder ser visualizada desde cualquier navegador web actual aunque se recomienda el uso de Firefox en sus últimas versiones.
- ✓ El resto del software necesario (servidor Web Apache, servidor Tomcat, Máquina Virtual Java,...) será proporcionado al administrador de la aplicación, el cual dispone de un manual para su instalación en el Anexo II.

B) Requerimientos de la interfaz

Los requerimientos de la interfaz gráfica entre la aplicación y el usuario están íntimamente ligados a la usabilidad y sus principios. La usabilidad se puede definir de varias formas [Man02]

- Usabilidad se define coloquialmente como facilidad de uso, ya sea de una página web, una aplicación informática o cualquier otro sistema que interactúe con un usuario.
- La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.
- Usabilidad es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico.

A partir de estas tres definiciones se pueden obtener los principios básicos de la usabilidad, los cuales se asociarán a los requerimientos no funcionales que deberá cumplir la interfaz gráfica:

- ***Facilidad de aprendizaje:*** se refiere a la facilidad con la que nuevos usuarios pueden tener una interacción efectiva. Depende de los siguientes factores:
 - ***Predecible:*** una vez conocida la aplicación, se debe saber en cada momento a que estado se pasará en función de la tarea que se realice.
 - ***Sintetizable:*** los cambios de estado tras una acción deben ser fácilmente captados.
 - ***Generalizable:*** las tareas semejantes se resuelven de modo parecido.
 - ***Familiarizable:*** el aspecto de la interfaz tiene que resultar conocido y

familiar para el usuario.

- *Consistente*: siempre se han de seguir una misma serie de pasos para realizar una tarea determinada.
- *Flexible*: relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.
- *Robustez*: es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos o, también, la capacidad del sistema para tolerar fallos. Esta relacionada con los siguiente factores:
 - *Navegable*: el usuario debe poder observar el estado del sistema sin que esta observación repercuta de forma negativa en él.
 - *Recuperación de información*: la aplicación debe poder deshacer alguna operación y permitir volver a un estado anterior.
 - *Valores por defecto*: ayudan al usuario introduciendo posibles valores de uso común permitiendo el recuerdo pasivo.
 - *Tiempo de respuesta*: es el tiempo necesario para que el sistema pueda mostrar los cambios realizados por el usuario.

3.3 Análisis del sistema

Una vez conocido el propósito del proyecto software, las propiedades que debe cumplir y las restricciones a las que debe someterse, llega el momento de analizar el sistema y crear un modelo del mismo que sea correcto, completo, consistente, claro y verificable. Para conseguir esto se estudiarán los perfiles de usuario y se crearán y definirán casos de uso en base a los requerimientos previamente obtenidos. Por último se describirán ciertos escenarios de acción de dichos casos de uso.

3.3.1 Perfil de usuario

En esta fase el primer paso es determinar quienes son los usuarios potenciales de la aplicación, para a partir de esto, obtener las características generales que nos permitan caracterizar los requisitos de usabilidad que posteriormente habrá que tener en cuenta en el diseño de la aplicación y de su interfaz gráfica.

Nuestro sistema, por lo general va a contar con dos tipos de usuarios principales: alumno y profesor. Contendrá también un tercer usuario, que será el administrador del sistema, el cuál no hará uso de la aplicación en si, aunque también analizaremos su perfil, para decidir sobre la interfaz gráfica del sistema.

Pasamos a comentar las características de cada uno de los usuarios de nuestro sistema:

Administrador:

- *Conocimientos del dominio del problema:* debe tener conocimiento del dominio suficiente como para realizar nuevas encuestas.

- *Sobre uso de equipos/programas informáticos:* es importante que tenga conocimientos sobre programas informáticos, ya que será el encargado del mantenimiento de la base de datos. También deberá tener conocimientos de servidores Tomcat, que es el utilizado en la aplicación. En cuanto al uso de equipos informáticos debe tener conocimientos a nivel experto del manejo de un servidor.
- *Entorno de trabajo:* el lugar de trabajo será una oficina, estará situado en un punto concreto, por lo cual no necesitará el uso de ningún software o hardware especial.
- *Nivel cultural:* el administrador debe tener un nivel cultural medio, ya que es necesario para la creación y modificación de encuestas.
- *Habilidades sociales:* puesto que no es una ocupación de cara al público, el administrador no tiene porque tener habilidades específicas de trato hacia el público, pero si será importante que sepa trabajar en grupo y tener habilidades de interrelación personal con sus compañeros de trabajo.

Profesor:

- *Conocimientos del dominio del problema:* no es necesario que tenga conocimientos sobre el dominio del problema, ya que sólo se va a limitar a comprobar los resultados de la evaluación de las asignaturas en las que imparte docencia.
- *Sobre uso de equipos/programas informáticos:* no es necesario que tenga conocimientos sobre otros programas informáticos, sólo sobre nuestra aplicación y el navegador que utilice, ya que no será necesario el uso de ningún otro programa específico. En cuanto al manejo de equipos si deberá tener unos conocimientos básicos a nivel de usuario como son: encendido y apagado del equipo, manejo del

ratón, teclado,...

- *Entorno de trabajo:* el lugar de trabajo estará situado en un punto concreto, por lo cual no necesitará el uso de ningún software o hardware especial.
- *Nivel cultural:* se presupone un nivel cultural medio-alto, al ser profesor. Es necesario, este nivel, para la comprensión de sus resultados asociados.
- *Habilidades sociales:* puesto que no es una ocupación de cara al público, no tiene porque tener habilidades específicas de trato hacia el público.

Alumno:

- *Conocimientos del dominio del problema:* no es necesario que tenga conocimientos sobre el dominio del problema, ya que solo se va a limitar a contestar encuestas.
- *Sobre uso de equipos/programas informáticos:* al igual que con profesor, no es necesario que tenga conocimientos sobre otros programas informáticos, sólo sobre nuestra aplicación y el navegador que utilice, ya que no será necesario el uso de ningún otro programa específico. En cuanto al manejo de equipos si deberá tener unos conocimientos básicos a nivel de usuario como son: encendido y apagado del equipo, manejo del ratón, teclado,...
- *Entorno de trabajo:* el lugar de trabajo del alumno, normalmente será su casa, biblioteca o aula con un ordenador conectado a Internet por lo que estará situado en un punto concreto en el que no necesitará el uso de ningún software o hardware especial.

- *Nivel cultural:* se le supone un nivel cultural medio, al ser alumno universitario, ya que es necesario que comprenda las preguntas de las encuestas, así como las posibles respuestas de cada pregunta.
- *Habilidades sociales:* puesto que no es una ocupación de cara al público, el alumno no tiene porque tener habilidades específicas de trato hacia el público.

Tras definir las características del perfil de usuario podemos tomar una serie de decisiones de diseño sobre la aplicación y su interfaz gráfica, tales como:

- La aplicación debe ser guiada por el usuario. El usuario posee los conocimientos necesarios sobre este tipo de aplicaciones, no a nivel experto, pero sí como usuario habitual.
- La interfaz debe presentar toda su funcionalidad en una pantalla principal. De este modo el usuario tiene a su alcance todas las acciones propias de la aplicación de un modo más accesible, facilitándole su trabajo.
- Aunque anteriormente no se ha planteado la posibilidad de deficiencias visuales en el usuario, hay que tenerlas en cuenta a la hora de determinar el aspecto de la interfaz, ya que una parte importante de la población las presenta y como consecuencia, existe una elevada posibilidad de que algunos usuarios de esta aplicación necesiten de características especiales en la interfaz para facilitarles su trabajo. Un ejemplo de esto puede ser un usuario con visión reducida en este caso se podría ampliar el tamaño de los botones, de los mensajes de diálogo y de toda la información escrita.
- Para realizar el diseño de la interfaz también es importante tener en cuenta aspectos como los colores empleados, ya que estos facilitan al usuario la percepción de las distintas tareas o acciones. La percepción del usuario hacia la interfaz en muchas

ocasiones no solo depende de problemas en la falta de visión, sino también en la forma de percibir los colores, por lo que cabe plantearse la existencia de disfunciones en la percepción del color tales como:



Figura 3.1: Imagen original. Inicio de la aplicación

- Deuteranopía: No responde al color verde. Como podemos ver en la imagen, esta deficiencia visual no influye en la comprensión de la aplicación, ya que sólo cambia el color de la cabecera. El color del menú sigue viéndose igual que sin esta deficiencia.



Figura 3.2: Simulación de deuteranopía

- Protanopía: No responde al color rojo. Como se aprecia en la imagen, no tiene ninguna influencia en los menús de la aplicación, por lo que un usuario con protanopia puede utilizarla correctamente.



Figura 3.3: Simulación de protanopía

- Tritanopía: No responde al azul y al amarillo. Se da muy raramente. En este caso vemos como cambia el color del menú, pasa de azul a verde oscuro, pero observamos que se puede leer los diferentes apartados del menú, por lo que un usuario con esta deficiencia puede usar nuestro sistema sin ningún problema.



Figura 3.4: Simulación de tritanopía

Estas disfunciones están presentes en más de un 10% de la población, por lo que son importantes contemplarlas ya que a la hora de diseñar la interfaz es posible que los iconos o botones empleados no sean percibidos como deberían serlo, por ello, una vez diseñadas las pantallas debemos realizar una serie de simulaciones con la interfaz, para comprobar que no existe ningún problema para percibir los distintos iconos.

Hay que puntualizar que todos los aspectos comentados anteriormente sobre deficiencias visuales y disfunciones en la percepción del color son cuestiones no específicas a esta aplicación, sino que según los Criterios de Usabilidad, deben ser tomadas en cuenta para el desarrollo de la interfaz gráfica de cualquier programa informático.

3.3.2 Casos de uso

Un caso de uso representa una clase de funcionalidad dada por el sistema como un flujo de eventos. También se puede definir como la representación de una situación o tarea de interacción de un usuario con la aplicación. Los casos de uso describen como se realiza una tarea de manera exacta y constan de los siguientes elementos:

- Nombre único e unívoco
- Actores participantes
- Condiciones de entrada
- Flujo de eventos
- Condiciones de salida
- Requerimientos especiales

Por lo tanto, es necesario determinar cuales son los actores participantes en cada uno de los casos de uso. Un actor modela una entidad externa que se comunica con el sistema, es decir, es un tipo de usuario del sistema. Un actor, al igual que un caso de uso, debe tener un

nombre único y puede tener una descripción asociada.

En nuestro sistema contamos con los actores descritos en el punto anterior, por lo que es el momento de crear los distintos casos de uso. A la hora de realizar esta acción es importante que cada uno de los requerimientos funcionales ya definidos aparezca en al menos uno de los casos de uso aunque, por otra parte, puede haber casos de uso nuevos, en los que no aparezca ninguno de los requerimientos, ya que estamos en una fase de refinamiento del sistema donde queremos construir un modelo detallado del mismo.

Un paso previo a la creación y descripción de los distintos casos de uso es la obtención de los diversos diagramas de casos de uso de nuestro sistema, al que vamos a llamar Encuesta Evaluación.

El primero es un diagrama frontera, es decir, un diagrama que describe completamente la funcionalidad de un sistema:

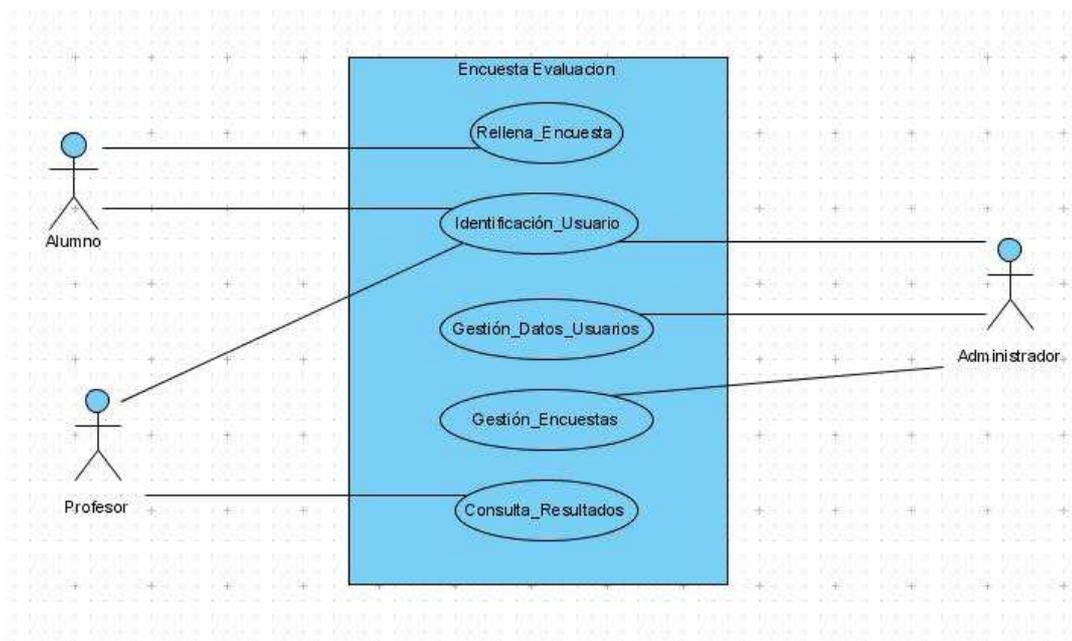


Figura 3.5: Diagrama frontera de Encuesta Evaluación

Los casos de uso mostrados en un diagrama frontera pueden ser lo suficientemente exactos o, por el contrario, pueden ser concretados con un mayor detalle. A la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones:

- <<extend>>: es una relación cuya dirección es hacia el caso de uso a detallar que representa comportamientos excepcionales del caso de uso.
- <<include>>: es una relación cuya dirección es contraria a la de la relación <<extend>> que representa un comportamiento común del caso de uso

En nuestro caso nos encontramos con que los casos de uso **Gestión_Datos_Usuarios** y **Gestión_Encuestas** requieren ser detallados con mayor profundidad.

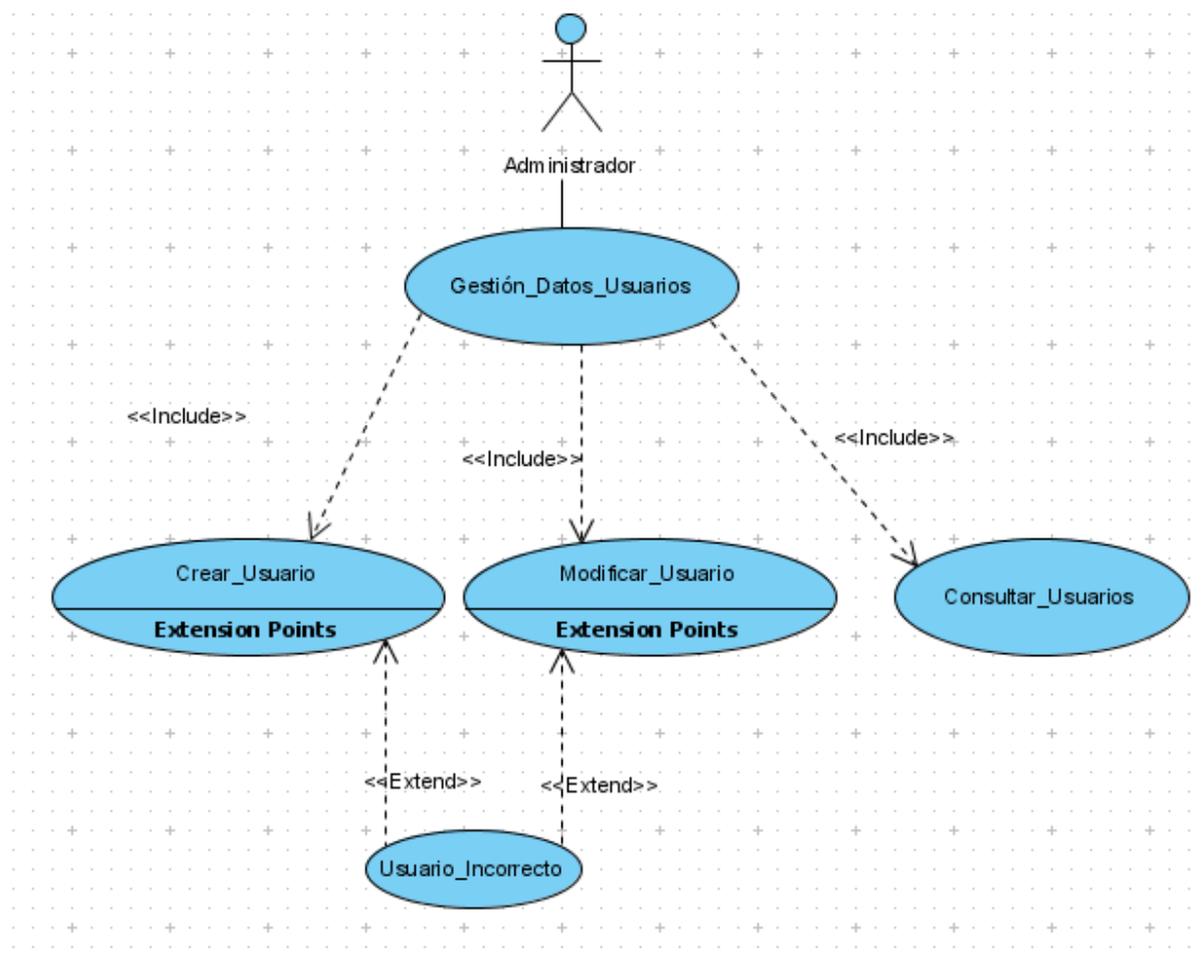


Figura 3.6: Diagrama del caso de uso Gestión_Datos_Usuarios

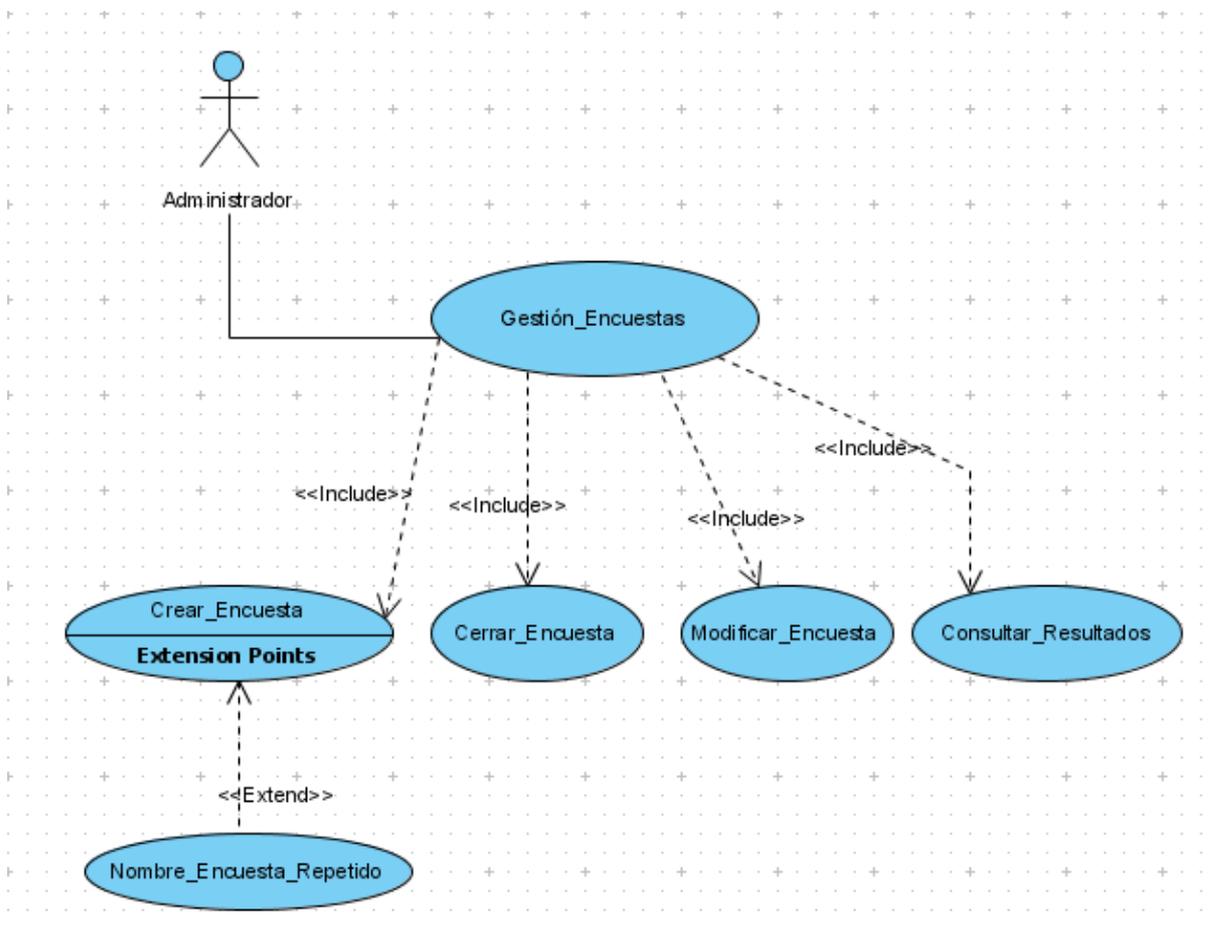


Figura 3.7: Diagrama del caso de uso Gestión_Encuestas

A continuación, se describen detalladamente cada uno de los casos de uso mostrados en las figuras anteriores:

➤ **Caso de Uso 1: Identificación Usuario**

Actores participantes: Alumno, Profesor y Administrador

Condiciones de entrada: Que exista una cuenta de usuario creada.

Flujo de Eventos:

1. El usuario (Alumno, Profesor o Administrador) inicia la aplicación.
2. El sistema muestra un formulario de entrada.
3. El usuario introduce su identificador.
4. El sistema comprueba que el identificador es válido (E-1).
5. Según el identificador sea de:
 - 5.1 Alumno, el usuario entra al sistema y este le muestra una lista desplegable con las asignaturas en las que está matriculado.
 - 5.2 Profesor, el usuario entra al sistema y este le muestra una lista desplegable con las asignaturas en las que imparte docencia.
 - 5.3 Administrador, entonces el usuario entra al sistema y este le muestra el menú de administrador.

Condiciones de salida: La contraseña ha sido comprobada.

Excepciones:

E-1: El identificador introducido por el usuario no es válido. El sistema informa al usuario de dicha situación. El usuario puede intentar introducir un identificador válido de Alumno, Profesor o Administrador, o salir del caso de uso.

➤ **Caso de Uso 2: Gestión_Datos_Usuario**

Actores participantes: Administrador

Condiciones de entrada: Que el Administrador esté identificado en el sistema.

Flujo de Eventos:

1. El sistema muestra el menú principal.
2. Administrador elige la opción Datos Usuario del menú principal.
3. El sistema muestra un menú con tres opciones y le pide a Administrador que elija:
 - Si Administrador elige Crear Usuario, se realiza S-1.
 - Si Administrador elige Modificar Usuario, se realiza S-2.
 - Si Administrador elige Consultar Usuarios, se realiza S-3.
 - Si Administrador elige Salir, se termina el caso de uso.

Subflujos de eventos:

S-1: Crear_Usuario

- 1.1. El sistema muestra al Administrador un formulario de entrada de datos.
- 1.2. Administrador introduce DNI, nombre, apellidos del usuario.
- 1.3. El sistema actualiza (E-1) la base de datos con la información introducida.
- 1.4. El caso de uso se inicia de nuevo.

S-2: Modificar_Usuario

- 2.1. El sistema pide el DNI del usuario.
- 2.2. El sistema muestra los datos del usuario, con la posibilidad de ser modificados.

2.3. El administrador modifica los datos del usuario y los guarda (E-1) en la BBDD.

2.4. El caso de uso se inicia de nuevo.

S-3: Consultar_Usuarios

3.1 El sistema muestra al Administrador el DNI, nombre y apellidos de todos los usuarios existentes en el sistema.

3.2 El caso de uso se inicia de nuevo.

Condiciones de salida: Se actualiza la BBDD con la información introducida.

Excepciones:

E-1:El usuario introducido por el Administrador no es válido. El sistema informa al usuario de dicha situación. El Administrador debe introducir de forma correcta todos los datos del usuario.

➤ **Caso de Uso 3: Gestión_Encuestas**

Actores participantes: Administrador

Condiciones de entrada: Que el Administrador esté identificado en el sistema.

Flujo de Eventos:

1. El sistema muestra el menú principal.
2. Administrador elige la opción Encuestas del menú principal.
3. El sistema muestra un menú con tres opciones y le pide a Administrador que elija:
 - Si Administrador elige Crear Encuesta, se realiza S-1.
 - Si Administrador elige Cerrar Encuesta, se realiza S-2.
 - Si Administrador elige Modificar Encuesta, se realiza S-3.
 - Si Administrador elige Consultar Resultados, se realiza S-4.
 - Si Administrador elige Salir, se termina el caso de uso.

Subflujos de eventos:

S-1: Crear_Encuesta

- 1.1. El sistema muestra al Administrador un formulario de entrada para el nombre de la encuesta.
- 1.2. Administrador introduce nombre de la encuesta.
- 1.3. El sistema actualiza la base de datos con la información introducida (E-1).
- 1.4. El sistema muestra al administrador el menú para insertar las preguntas de la encuesta.
- 1.5. El administrador termina la realización de la encuesta y la guarda.
- 1.6. El caso de uso se inicia de nuevo.

S-2: Cerrar_Encuesta ¹⁾

- 2.1. El sistema pide al Administrador que indique la encuesta que asignatura cerrar.
- 2.2. El sistema muestra pantalla de confirmación de cierre de encuesta.
- 2.3. El administrador elige si quiere cerrar la encuesta o no.
- 2.4. El sistema realiza la operación indicada por el administrador.
- 2.5. El caso de uso se inicia de nuevo.

S-3: Modificar_Encuesta

- 3.1. El sistema pide al Administrador que indique la encuesta que quiere modificar.
- 3.2. El administrador elige la encuesta a modificar.
- 3.3. El sistema muestra la encuesta.
- 3.4. El administrador modifica la encuesta y guarda los cambios.
- 3.5. El caso de uso se inicia de nuevo.

S-4: Consultar_Resultados

- 4.1. El sistema pide al Administrador que indique la encuesta de que asignatura quiere consultar sus resultados.
- 4.2. El sistema muestra pantalla los resultados de encuesta.
- 4.3. El caso de uso se inicia de nuevo.

¹⁾ En otro tipo de evaluación, cerrar una encuesta, dependerá de que se hayan recibido todas las respuestas esperadas.

Condiciones de salida: Se solicita la salida del caso de uso.

Excepciones:

E-1: El nombre de encuesta introducido por el Administrador no es válido. El sistema informa al Administrador de dicha situación. El Administrador debe introducir un nombre de encuesta correcto.

➤ **Caso de Uso 4: Rellena_Encuesta**

Actores participantes: Alumno

Condiciones de entrada: Que exista una encuesta creada para la/s asignatura/s en las que el alumno está matriculado.

Flujo de Eventos:

1. El Alumno elige la asignatura y la encuesta a rellenar.
2. El sistema va mostrando cada una de las preguntas de la encuesta.
3. El Alumno va contestando todas las preguntas de la encuesta.
4. El sistema pide al alumno que envíe (E-1) o rechace la encuesta.
5. El sistema realiza la petición del alumno.
6. El caso de uso finaliza.

Condiciones de salida: La encuesta es almacenada en el sistema.

Excepciones:

E-1: Se produce un error al enviar la encuesta. El sistema le indica al Alumno que la encuesta no se ha almacenado correctamente en la BBDD.

➤ **Caso de Uso 5: Consulta_Resultados**

Actores participantes: Profesor

Condiciones de entrada: Que exista una encuesta cerrada para las asignaturas en las que imparte docencia.

Flujo de Eventos:

1. El Profesor elige la encuesta a consultar.
2. El sistema muestra los resultados de la encuesta y ofrece dos opciones al profesor:
 - Finalizar el caso de uso.
 - Consultar los resultados de otra encuesta.

Condiciones de salida: Profesor visualiza correctamente los resultados asociados a su asignatura.

3.3.3 Escenarios

Un caso de uso es una representación abstracta, una abstracción, de una funcionalidad del sistema a realizar. La representación concreta de un caso de uso se realiza mediante la creación de uno o más escenarios que muestren todas las interacciones posibles entre el sistema y sus usuarios.

Un escenario esta formado por los siguientes elementos:

- Un nombre único y unívoco
- Una descripción
- Los actores participantes
- El flujo de eventos

Como se ha indicado, para cada caso de uso puede haber varios escenarios. Para nuestro proyecto se han creado y descrito una cantidad importante de casos de uso por lo que no vamos a definir todos los escenarios de cada uno de ellos sino que vamos a definir unos pocos que puedan servir como ejemplo de las principales funcionalidades que el sistema va a desarrollar: Rellena encuesta y Consulta resultados.

Nombre: RellenarEncuestaCurso06-07

Descripción: El usuario 12345678 desea rellenar y enviar la encuesta Curso 06-07, de la asignatura S. Informáticos en la que se encuentra matriculado.

Actor: Alumno

Flujo de eventos:

1. El usuario entra al sistema.
2. El sistema muestra el formulario de entrada.
3. El usuario introduce 12345678.
4. El sistema valida al usuario y entra a la aplicación como el usuario 12345678
5. El sistema muestra una lista con las asignaturas en las que esta matriculado.
6. El usuario elige la asignatura S. Informáticos.
7. El sistema muestra una lista con las encuestas no contestadas relacionadas con la asignatura S. Informáticos.
8. El usuario elige la encuesta Curso 06-07.
9. El sistema se comunica con la BBDD y obtiene la encuesta Curso 06-07.
10. El sistema muestra la primera pregunta de la encuesta Curso 06-07.
11. El usuario contesta a esta primera pregunta.
12. El sistema va mostrando el resto de preguntas de la encuesta Curso 06-07.
13. El usuario contesta el resto de preguntas hasta finalizar la encuesta.
14. El sistema pregunta al usuario si quiere enviar la encuesta.
15. El usuario envía la encuesta.
16. El sistema actualiza la BBDD, con las respuestas del usuario.
17. El sistema comunica al usuario que la encuesta se ha guardado correctamente.

Nombre: ConsultarResultadosCurso06-07

Descripción: El usuario 87654321 desea consultar los resultados obtenidos en la encuesta Curso 06-07, de la asignatura S. Informáticos de la que es profesor.

Actor: Profesor

Flujo de eventos:

1. El usuario entra al sistema.
2. El sistema muestra el formulario de entrada.
3. El usuario introduce 87654321.
4. El sistema valida al usuario y entra a la aplicación como el usuario 87654321.
5. El sistema muestra una lista con las encuestas finalizadas de su asignatura.
6. El usuario elige la encuesta Curso 06-07.
7. El sistema se comunica con la BBDD y obtiene los resultados de la encuesta Curso 06-07.
8. El sistema muestra los resultados de la encuesta Curso 06-07.

3.4 Diseño del sistema

Sin duda, realizar de manera adecuada cada una de las actividades que conlleva la Ingeniería del Software es indispensable para la realización de un proyecto software de calidad. Por lo tanto, no se puede decir que ninguna de estas actividades sea más importante que otra. Sin embargo, si podemos decir que la actividad de diseño es la más delicada y la más laboriosa de llevar a cabo.

Es delicada porque si no se lleva a cabo correctamente se hace imposible el codificar, de manera correcta, en la actividad de implementación el modelo obtenido en el análisis del

sistema, lo que puede repercutir en el desperdicio de todo el esfuerzo realizado durante las primeras actividades de la Ingeniería del Software.

Y es laboriosa porque las estrategias a seguir para conseguir que esta traducción entre modelo y código se lleve a cabo correctamente son muy diversas y bastante complejas. Se puede decir, por tanto, que el diseño del sistema es la actividad de la Ingeniería del Software en la que se identifican los objetivos finales del sistema y se plantean las diversas estrategias para alcanzarlos en la actividad de implementación.

Sin embargo, el sistema no se suele diseñar de una sola vez sino que hay que diferenciar entre el diseño y el diagrama de clases que se van a manejar y el diseño de la interfaz entre la aplicación y el usuario. Estas dos fases del diseño no se realizan de forma consecutiva una detrás de la otra sino que lo normal es realizarlas de manera concurrente y finalizarlas a la vez.

Las distintas fases de diseño que veremos a continuación son:

- Diagrama de clases: muestran la estructura del sistema.
- Diseño de los datos: estructura de los datos.
- Diseño de la interfaz: se define la apariencia visual de la aplicación.
- Implementación: el modelo obtenido se transforma en código.

3.4.1 Diagrama de clases

Los diagramas de clases se utilizan para mostrar la estructura estática del sistema modelado. Pueden estar contenidos por clases, interfaces, paquetes, relaciones e incluso instancias, como objetos o enlaces.

Son una potente herramienta de diseño, ayudando a los desarrolladores a planificar y establecer la estructura del sistema y subsistema antes de escribir ningún código. Esto permite asegurar que el sistema está bien diseñado desde el principio.

Son utilizados en la fase de diseño prácticamente en la totalidad de los sistemas que utilizan UML para su modelado.

Los diagramas de clases tienen los siguientes componentes:

- Clases: son los componentes fundamentales de los diagramas de clase. Su notación general es un rectángulo dividido en tres secciones, mostrando la primera el nombre de la clase, la siguiente los atributos y la última las operaciones.
- Relaciones: es una conexión semántica entre elementos. Existen cuatro tipos principales de relaciones:
 - o Generalización: es una relación de especialización.
 - o Asociación: es una relación estructural. Existen dos subtipos, la agregación y la composición.
 - o Realización: es una relación contractual, en la cual una clase especifica un contrato que otra clase garantiza que cumplirá (por ejemplo, una interface).

- Dependencia: Es una relación de uso.

Una vez descritos brevemente los componentes de un diagrama de clases, vamos a proceder a analizar los propios del sistema.

3.4.1.1 Diagrama completo de clases

Antes de mostrar el diagrama de clases, vemos apropiado explicar como va a ser el funcionamiento o la lógica de negocio del sistema para ver más claramente el paso de información o comunicación entre las distintas capas diseñadas.

Tendremos una capa modelo de la información que representará todos los datos del sistema. Por lo tanto esta capa es la encargada de obtener los datos necesarios de la BBDD.

Esta información será gestionada por una serie de servlets que harán la función de controlar el sistema y realizar cualquier operación necesaria.

Finalmente, cualquier representación gráfica de datos en la pantalla será lanzada por los Servlets utilizando para ello llamadas a páginas JSP haciendo éstas últimas la labor de la Vista de la aplicación.



Figura 3.8: Diagrama Esquema de la Aplicación

El diagrama completo de clases es el siguiente:

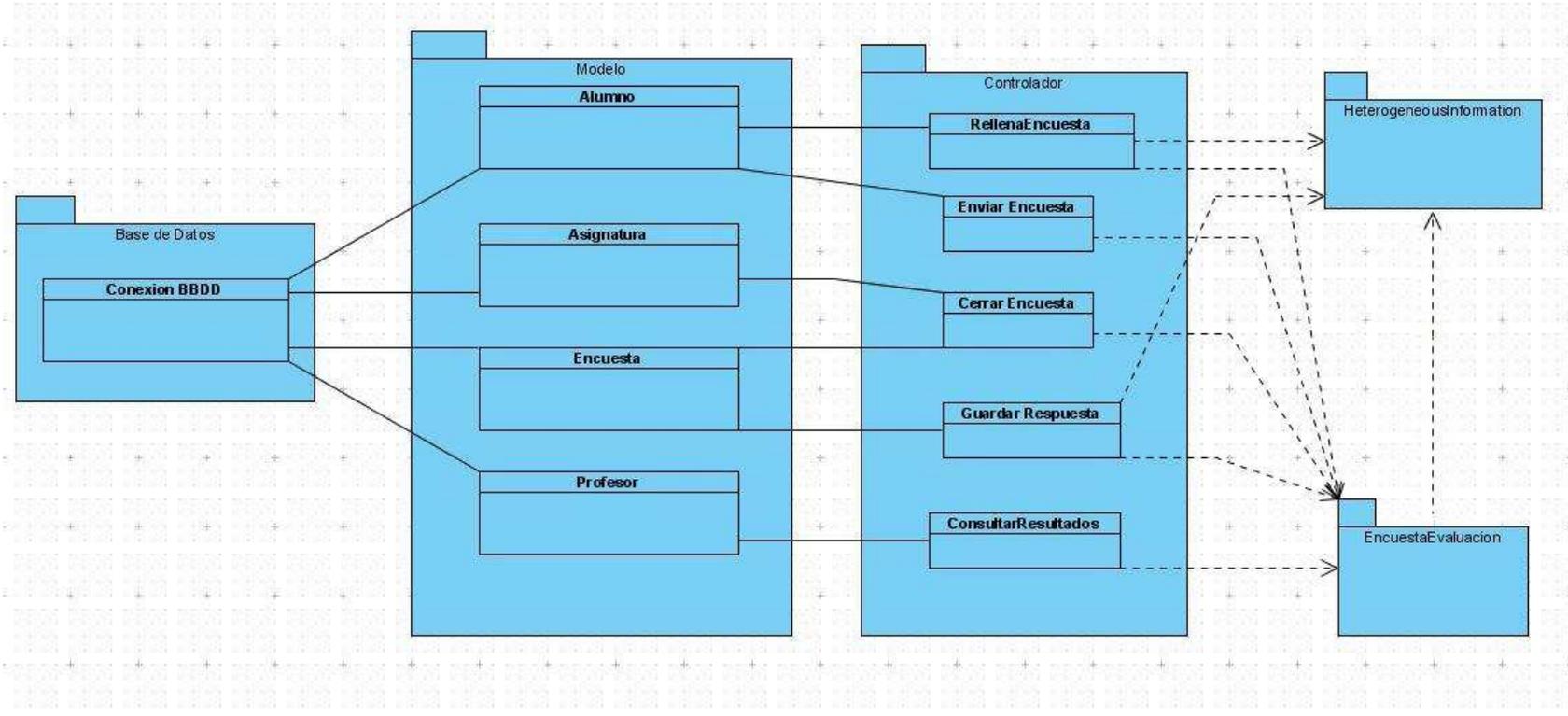


Figura 3.9: Diagrama de clases

Analizamos cada uno de los paquetes de nuestro diagrama de clases:

- Paquete Base de Datos: es el paquete al que hacen referencia todas las clases del modelo, para obtener la conexión a la BBDD.
- Paquete Modelo: Representa la capa de modelo de la información de la aplicación y, como tal, debe ser gestionado por otros paquetes controladores para administrar de forma adecuada toda la información. Todas las clases de este paquete son DataModule. Cada una de las clases contiene un QueryDataSet para acceder a las tablas de la BBDD.
- Paquete Controlador: la capa controladora, que en nuestro caso está formada por servlets, llevará a cabo todo el control del sistema y permitirá al usuario realizar cualquier operación sobre éste. Como capa de negocio, utilizará constantemente clases del paquete Modelo para obtener y guardar información, y será la encargada de operar con la información para mostrarla en la Vista, que en nuestro caso serán JSP`s.
- Los paquetes HeterogeneousInformation y EncuestaEvaluacion, son utilizados en este proyecto. El primer paquete contiene todo lo necesario para el manejo de la información heterogénea, mientras que el segundo nos define varias clases, entre las que destaca EvaluationalQuality, que será utilizada para evaluar las encuestas y Survey que es la clase encuesta, que contiene las preguntas de cada encuesta y las valoraciones de los alumnos.

3.4.1.2 Diagrama por paquete

3.4.1.2.1 Paquete Base de Datos

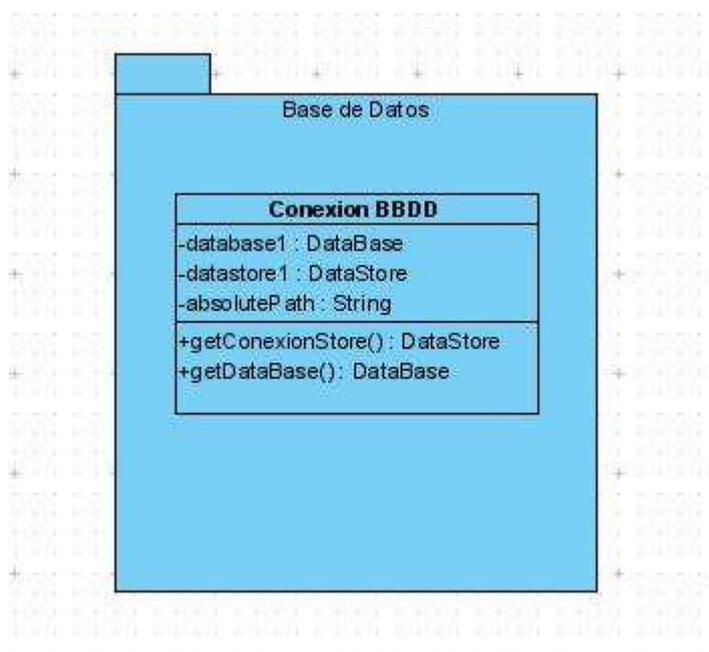


Figura 3.10: Diagrama de clases. Paquete Base de Datos

3.4.1.2.2 Paquete Modelo

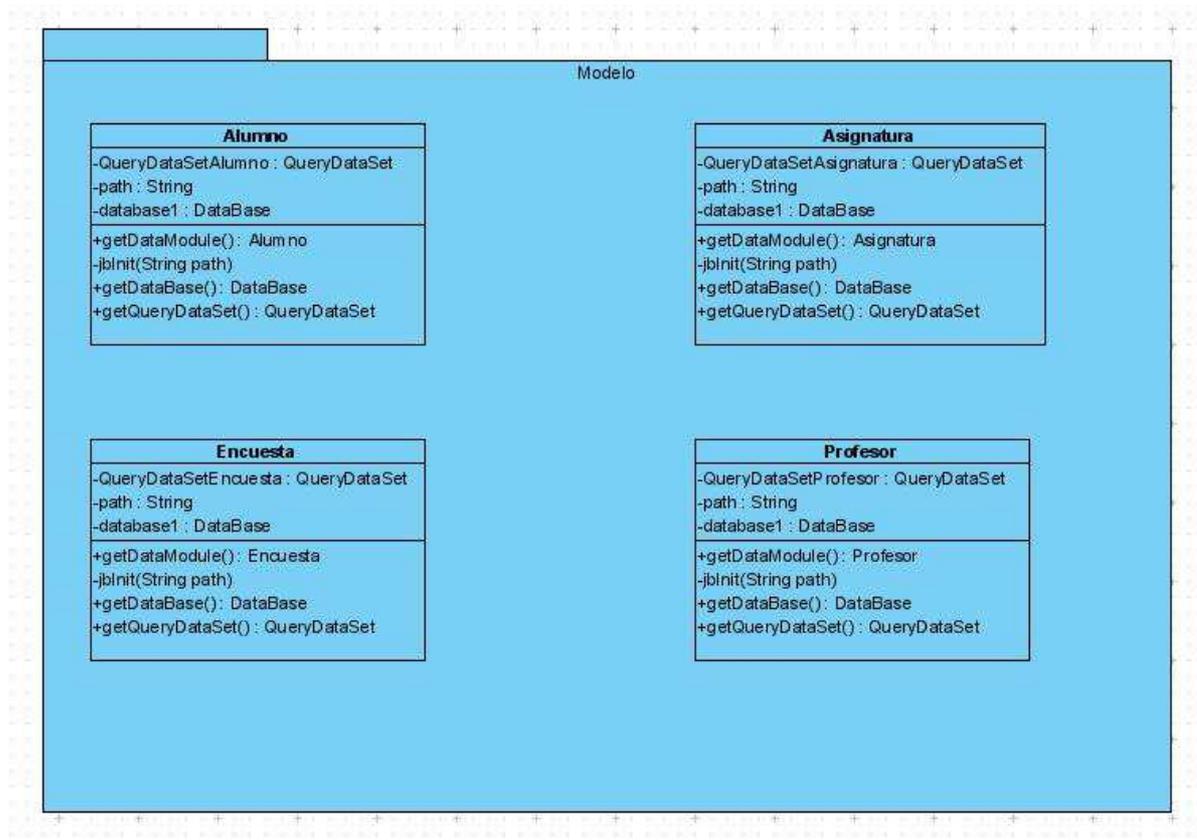


Figura 3.11: Diagrama de clases. Paquete Modelo

3.4.1.2.3 Paquete Controlador

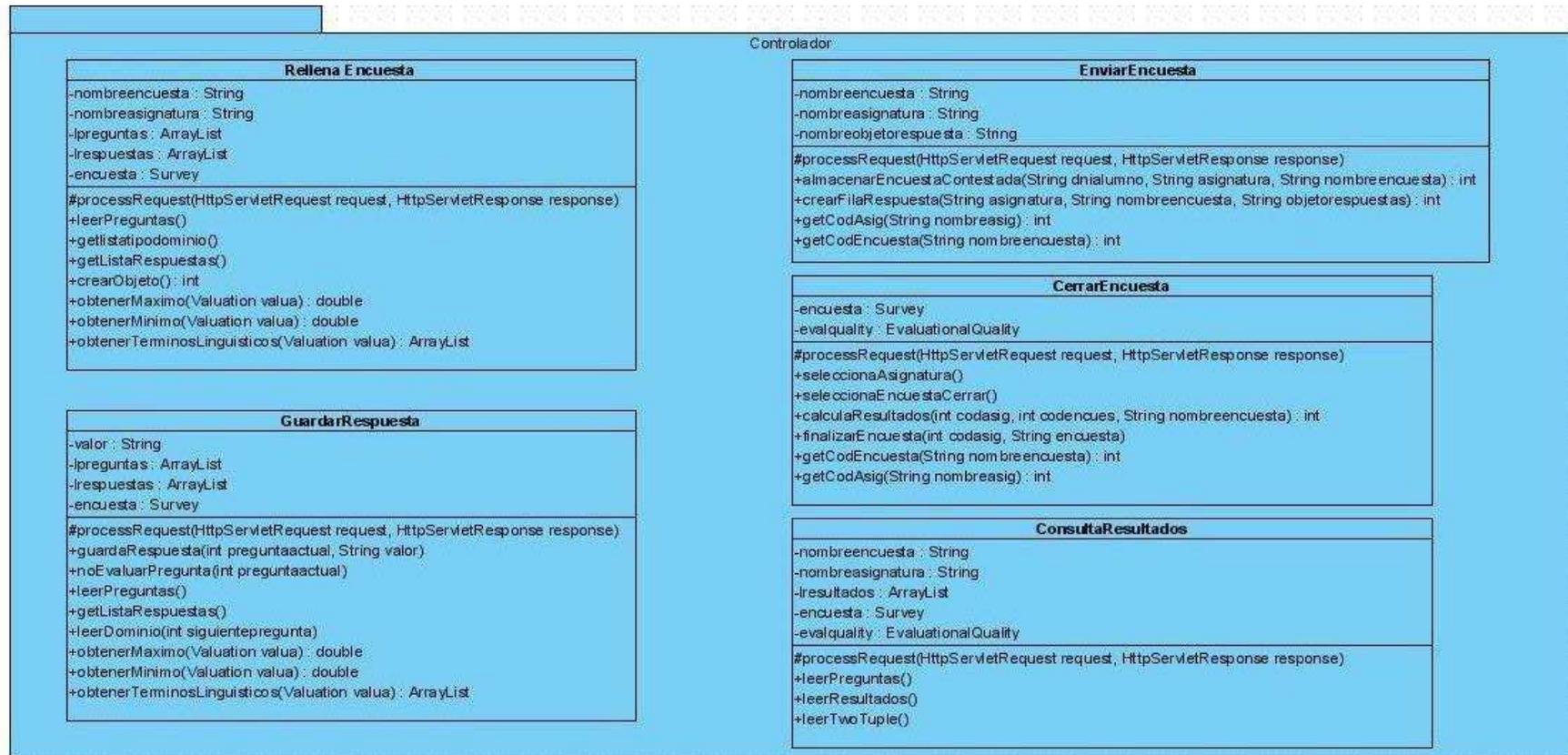


Figura 3.12: Diagrama de clases. Paquete Controlador

3.4.1.2.4 Paquete HeterogeneousInformation

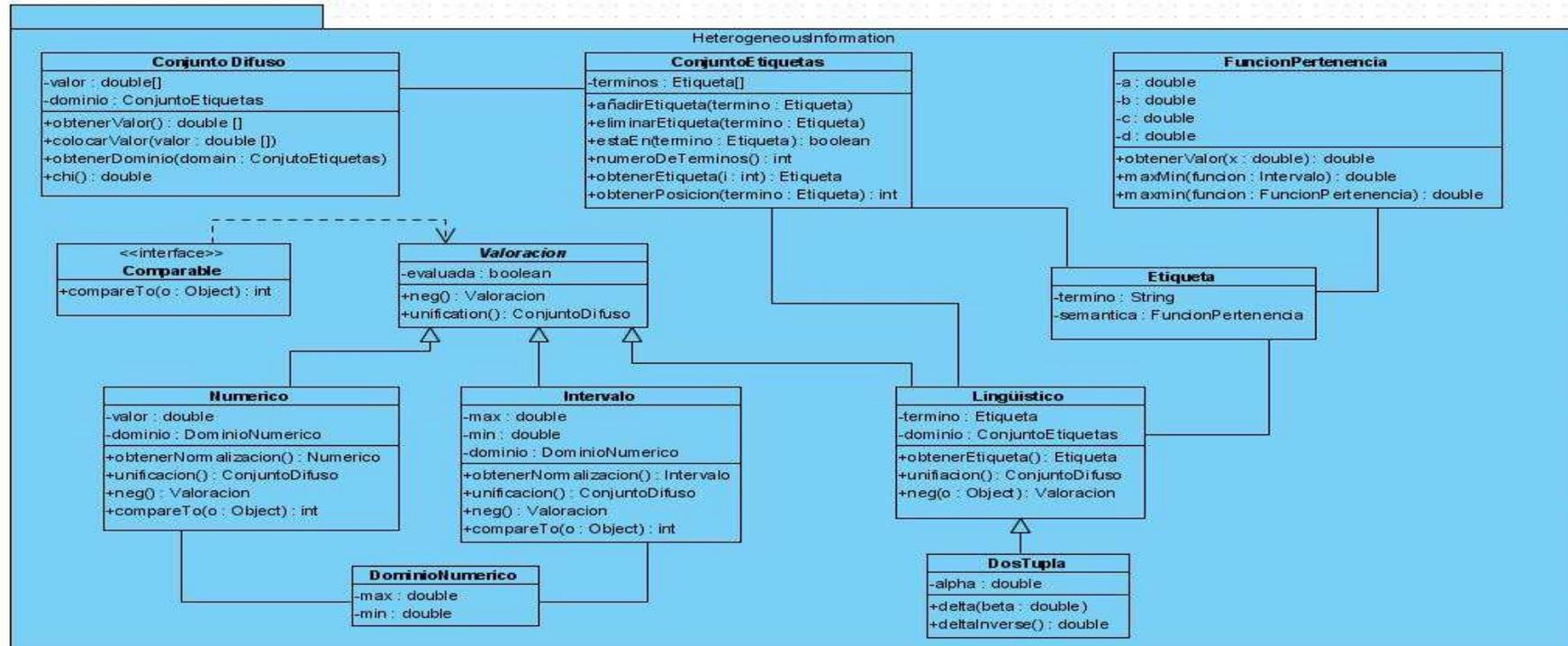


Figura 3.13: Diagrama de clases. Paquete HeterogeneousInformation

3.4.1.2.5 Paquete EncuestaEvaluacion

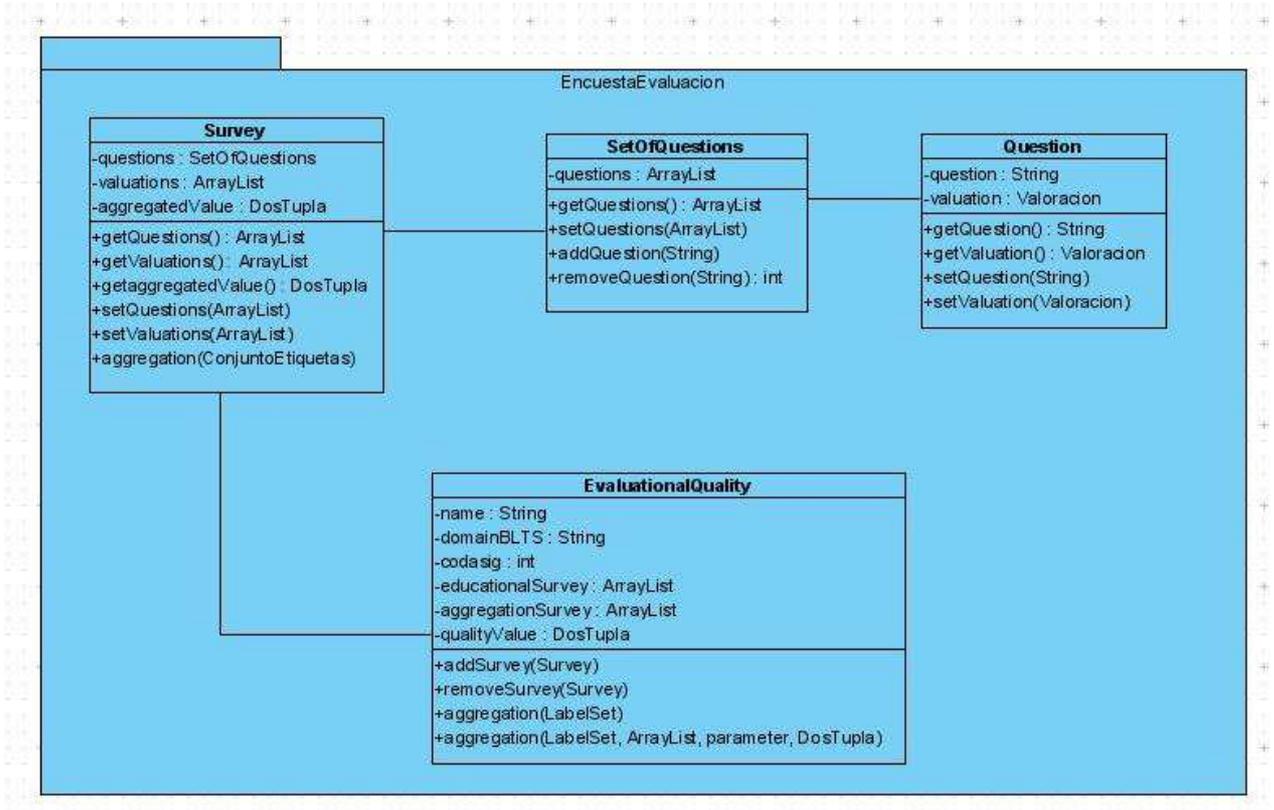


Figura 3.14: Diagrama de clases. Paquete EncuestaEvaluacion

3.4.2 Diseño de los datos

La intención de esta fase del diseño software es determinar la estructura que poseen cada uno de los elementos de información del sistema, es decir, la estructura de los datos sobre los que va a trabajar. Estos elementos son:

- Los *alumnos*, de los que conocemos su DNI, nombre y apellidos.
- Los *profesores*, de los que conocemos su DNI, nombre y apellidos.
- Las *asignaturas*, de las que conocemos su código y su nombre.
- Las *encuestas*, de las que conocemos su código y su nombre.
- Las *respuestas* de un alumno deben almacenarse de forma anónima. Por lo tanto debemos conocer el código de la encuesta, código de la asignatura y el nombre del objeto que almacenará las respuestas. Al ser respuestas anónimas no se debe almacenar el código del alumno al que corresponde la respuesta.
- Los *resultados* asociados a una encuesta también deben almacenarse. De los resultados debemos conocer el código de la encuesta, código de la asignatura a la que pertenecen los resultados y el nombre del objeto que contiene estos resultados.

Una vez determinados cuales son los elementos de información del sistema, se deben obtener sus representaciones en forma de tablas de una base de datos. Para ello, se debe realizar primeramente un diseño conceptual de la base de datos para, posteriormente, obtener las tablas requeridas. Para realizar este diseño conceptual se utilizará el modelo Entidad-Relación.

Modelo Entidad-Relación

El modelo Entidad-Relación (también conocido por sus iniciales: E-R) es una técnica de modelado de datos que utiliza diagramas Entidad-Relación. No es la única técnica de modelado pero si es la más extendida y utilizada.

Un diagrama entidad-relación esta compuesto por tres tipos de elementos principales:

- *Entidades*: objetos (cosas, conceptos o personas) sobre los que se tiene información. Se representan mediante rectángulos etiquetados en su interior con un nombre. Una instancia es cualquier ejemplar concreto de una entidad.
- *Relaciones*: interdependencias entre una o más entidades. Se representan mediante rombos etiquetados en su interior con un verbo. Si la relación es entre una entidad consigo mismo se denomina reflexiva, si es entre dos entidades se denomina binaria, ternaria si es entre tres y múltiple si es entre más.
- *Atributos*: características propias de una entidad o relación. Se representan mediante elipses etiquetados en su interior con un nombre.

En los diagramas Entidad-Relación también hay que tener en cuenta otros aspectos como pueden ser:

- *Entidades débiles*: son aquellas que no se pueden identificar unívocamente solo con sus atributos, es decir, necesitan de estar relacionadas con otras entidades para existir. Se representan con dos rectángulos concéntricos de distinto tamaño con un nombre en el interior del más pequeño.
- *Cardinalidad de las relaciones*: existen tres tipos de cardinalidades de una

relación según el número de instancias de cada entidad que involucren:

- *Uno a uno*: una instancia de la entidad A se relaciona solamente con una instancia de la entidad B. (1:1)
 - *Uno a muchos*: cada instancia de la entidad A se relaciona con varias de la entidad B. (1:*)
 - *Muchos a muchos*: cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. (*:*)
- *Claves*: cada entidad de un diagrama entidad-relación debe tener una clave, debe estar formada por uno o más de sus atributos.

Una vez conocidos los elementos que forman parte de un diagrama Entidad-Relación podemos empezar a desarrollar el modelo Entidad-Relación. Los pasos a seguir son los siguientes:

1. Convertir el enunciado del problema (o, como es nuestro caso, los elementos del sistema software) en un Esquema Conceptual del mismo.
2. Convertir este Esquema Conceptual (o EC) en uno más refinado conocido como Esquema Conceptual Modificado (ECM).
3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado y normalizarlas.

Normalización en el modelo Entidad-Relación

La normalización es un proceso consistente en imponer a las tablas ciertas restricciones mediante una serie de transformaciones consecutivas. Con ello se asegura que las tablas contengan los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando aquellos que pueden contener información cuya relevancia permite la creación de otra nueva tabla.

Para asegurar la normalización Codd estableció tres formas normales, las cuales hacen que una base de datos (si las cumple) esté normalizada.

Estas formas normales son:

- Primera forma Normal (FN1).

Una tabla está en FN1 si todos los atributos no clave, dependen funcionalmente de la clave, o lo que es lo mismo, no existen grupos repetitivos para un valor de clave.

- Segunda forma Normal (FN2)

Una tabla está en FN2 si está en FN1 y además todos los atributos que no pertenecen a la clave dependen funcionalmente de forma completa de ella. De esta definición se desprende que una tabla en FN1 y cuya clave está compuesta por un único atributo está en FN2.

- Tercera forma Normal (FN3)

Una tabla está en FN3 si está en FN2 y además no existen atributos no clave que dependan transitivamente de la clave.

1. Esquema Conceptual

Necesitamos convertir nuestros elementos del sistema en entidades o relaciones. Es obvio que alumnos, profesores, asignaturas y encuestas se convertirán en Entidades en nuestro Esquema Conceptual.

En cuanto a las relaciones:

- Los elementos respuesta y resultado se convertirán en relaciones entre profesor y encuesta. Cada una de estas relaciones tendrán un atributo denominado nombreobjeto, que almacenará el nombre del objeto que en la base de datos contiene la respuesta de un alumno y los resultados de una encuesta, respectivamente, y un código que hará de clave de cada respuesta o resultado.
- Un alumno estará matriculado en muchas asignaturas y en una asignatura podrá haber diversos alumnos matriculados, esta relación se llamara matriculado y será de muchos a muchos.
- Un profesor puede dar muchas asignaturas y estas pueden tener a varios profesores como docentes, por lo que tendremos una relación de muchos a muchos entre las entidades profesor y asignatura denominada *imparte*.
- Un alumno sólo puede responder una vez a una encuesta por lo que tendremos una relación entre las entidades alumno y encuesta. Esta relación será de muchos a muchos, ya que una encuesta es contestada por varios alumnos y un alumno contestará varias encuestas.

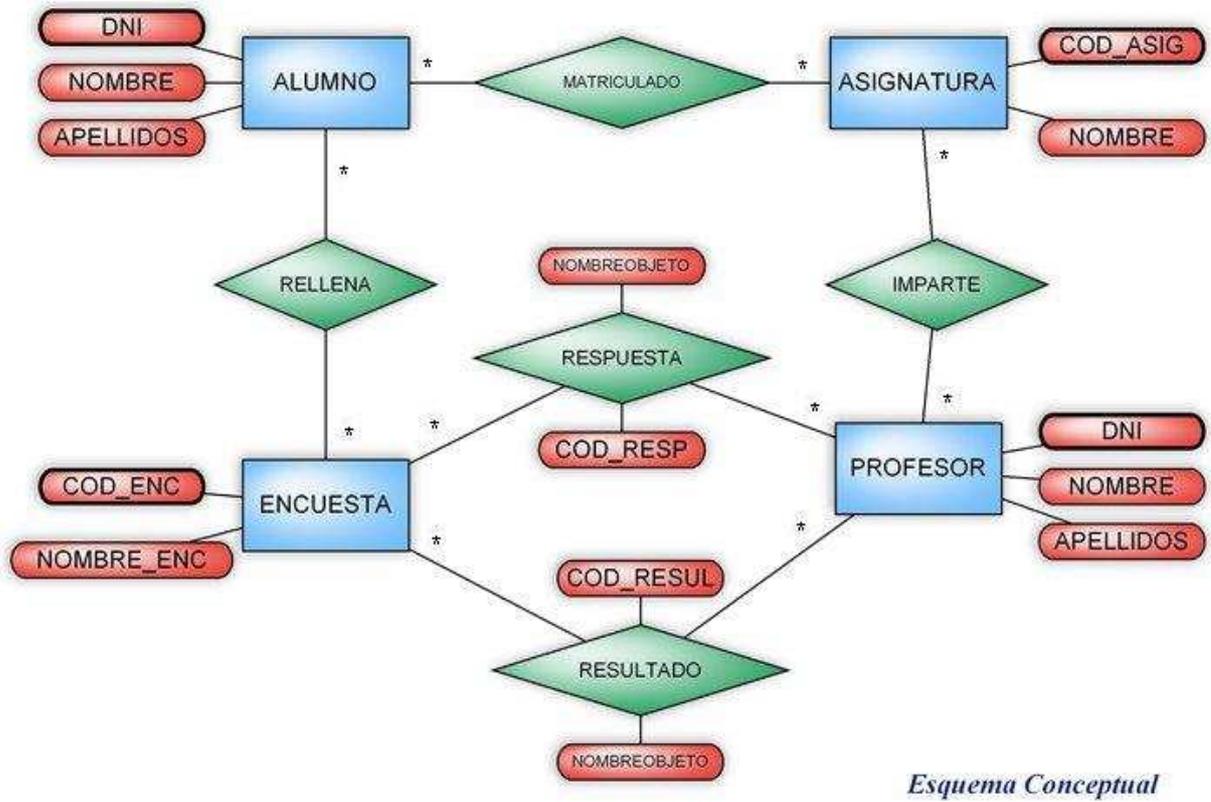
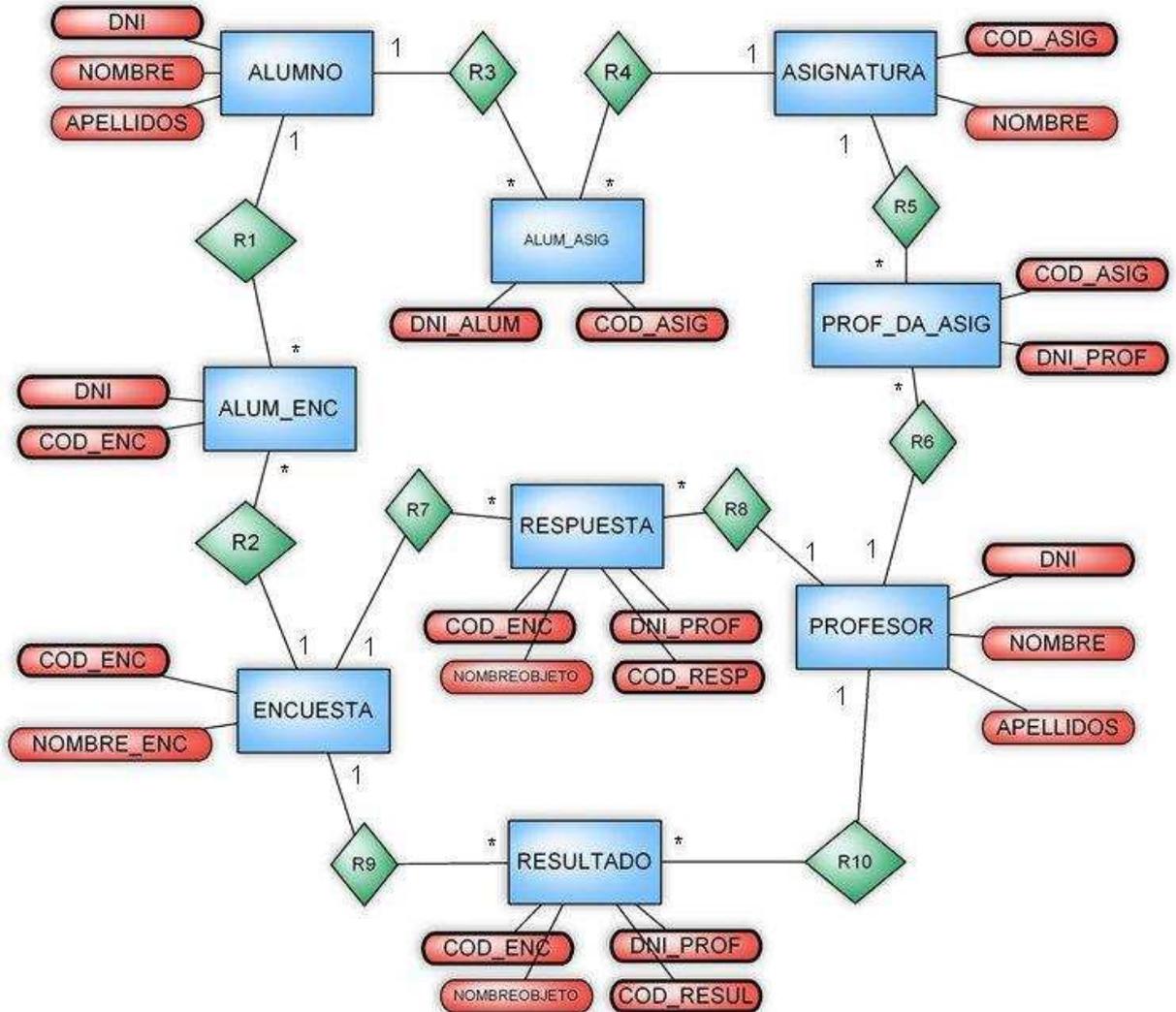


Figura 3.15: Esquema Conceptual

2. Esquema Conceptual Modificado

Para obtener el Esquema Conceptual Modificado debemos eliminar todas las entidades débiles, relaciones muchos a muchos y relaciones con atributos que haya en nuestro Esquema Conceptual. Por lo tanto, nuestro ECM queda como sigue:



Esquema Conceptual Modificado

R1: Rellena R2: Es rellenada
 R3:Matriculado_en R4:matriculados
 R5:Es_impartida R6: Imparte
 R7:Enc_respuesta R8: Prof_respuesta
 R9:Enc_resultado R10:Prof_resultado

Figura 3.16:Esquema Conceptual Modificado

Nota: El atributo NOMBRE_ENC de la tabla encuesta hace referencia a un objeto tipo Survey almacenado en la BBDD (se puede ver en el diagrama de clase de la figura 3.14), que contiene las preguntas de esa encuesta y sus valoraciones. De igual forma, el atributo

NOMBREOBJETO, de la tabla Respuesta, hace referencia a un objeto tipo Survey que contiene las respuestas de un alumno sobre una encuesta. El atributo NOMBREOBJETO de la tabla Resultado hace referencia a un objeto tipo EducationalQuality (se puede ver en el diagrama de clase de la figura 3.14), que contiene los resultados asociados a una encuesta.

3. Tablas

A partir del ECM obtenido previamente podemos determinar las tablas de la base de datos, teniendo en cuenta que:

- Cada entidad del ECM se transforma en una tabla.
- Los atributos de una entidad se convierten en los campos de las tablas respectivas.

Por lo tanto, obtendremos las siguientes tablas: ALUMNO, ALUM_ENC, ENCUESTA, ALUM_ASIG, ASIGNATURA, PROF_DA_ASIG, PROFESOR, RESPUESTA y RESULTADO. A continuación se detallan cada una de estas tablas.

ALUMNO

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
DNI	STRING	DNI del alumno	*	SI
NOMBRE	STRING	Nombre del alumno		SI
APELLIDOS	STRING	Apellidos del alumno		SI

Tabla 1: Campos de la tabla ALUMNO

ALUM_ENC

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
DNI	STRING	DNI del alumno	*	SI
COD_ENC	INT	Código de encuesta	*	SI

*Tabla 2: Campos de la tabla ALUM_ENC***ENCUESTA**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
COD_ENC	INT	Código de encuesta	*	SI
NOMBRE_ENC	STRING	Nombre de encuesta y nombre del objeto tipo Survey		SI

*Tabla 3: Campos de la tabla ENCUESTA***ALUM_ASIG**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
DNI_ALUM	STRING	DNI del alumno	*	SI
COD_ASIG	INT	Código de asignatura	*	SI

*Tabla 4: Campos de la tabla ALUM_ASIG***ASIGNATURA**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
COD_ASIG	INT	Código de asignatura	*	SI
NOMBRE	STRING	Nombre asignatura		SI

Tabla 5: Campos de la tabla ASIGNATURA

PROF_DA_ASIG

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
COD_ASIG	INT	Código de asignatura	*	SI
DNI_PROF	STRING	DNI profesor	*	SI

Tabla 6: Campos de la tabla PROFESOR_DA_ASIG

PROFESOR

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
DNI	STRING	DNI profesor	*	SI
NOMBRE	STRING	Nombre profesor		SI
APELLIDOS	STRING	Apellidos profesor		SI

Tabla 7: Campos de la tabla PROFESOR

RESPUESTA

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
DNI_PROF	STRING	DNI profesor	*	SI
COD_ENC	INT	Código de encuesta	*	SI
COD_RESP	INT	Código de respuesta	*	SI
NOMBREOBJETO	STRING	Nombre del objeto que almacena las valoraciones de un alumno		SI

Tabla 8: Campos de la tabla RESPUESTA

RESULTADO

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
DNI_PROF	STRING	DNI profesor	*	SI
COD_ENC	INT	Código de encuesta	*	SI
COD_RESUL	INT	Código del resultado	*	SI
NOMBREOBJETO	STRING	Nombre del objeto que almacena el resultado de una encuesta		SI

*Tabla 9: Campos de la tabla RESULTADO***3.4.3 Diseño de la interfaz**

En esta fase del diseño del sistema software se define cual va a ser la apariencia visual de la aplicación, es decir, se define la interfaz visual entre el usuario y la aplicación. Sin duda, realizar un buen diseño de la interfaz resulta primordial ya que está debe presentarse atractiva al usuario de la aplicación pero a la vez le debe de resultar fácil de entender y trabajar sobre ella.

Esta importancia es mayor en nuestro caso ya que la interfaz de nuestro proyecto es una interfaz web. Para las aplicaciones con interfaces web no existe una guía de estilo estándar como existe, por ejemplo, para desarrollar interfaces para aplicaciones de escritorio de Windows XP y que resulten, a la vez, atractivas y familiares. Cada programador, desarrollador o diseñador web debe definir su propia guía de estilo y procurar que, en base a ella, la interfaz resultante consiga unas cotas dignas de atractivo visual, familiaridad y facilidad de uso.

La ingeniería de usabilidad define en esta fase los siguientes aspectos:

- Definir estilo.
- Metáforas.
- Pantallas.
- Caminos de navegación.
- Secuencias de diálogo.
- Mensajes de error.

Cada uno de estos aspectos lo veremos a continuación de forma detallada.

3.4.3.1 Definir estilo

Antes de ponerse a diseñar una interfaz de usuario, se debe definir el estilo de la misma. Esto es de vital importancia cuando el diseño va a ser compartido entre varios diseñadores, ya que ayuda a mantener la coherencia interna de la interfaz.

Sin embargo, en contra de lo que pueda parecer en un principio, también es de mucha utilidad definir una guía de estilo cuando sólo hay un diseñador encargado de la interfaz. Esto se debe a varias razones:

- A veces es posible que mantener la coherencia y consistencia de una interfaz, si ésta es muy grande o muy ambiciosa, sea complicado incluso si sólo hay un diseñador.
- El diseñador primitivo puede, por las más diversas razones, abandonar el diseño y es de utilidad para sus sustitutos contar con una guía de estilo predefinida para

no tener que empezar de cero otra vez. Lo mismo puede aplicarse si no es el diseñador original el que se encarga del mantenimiento o la actualización de la interfaz.

Quedando demostrada la utilidad del uso de guías de estilo podemos pasar a definir las reglas, normas y recomendaciones que contendrá la guía de estilo de nuestra interfaz:

- Fuentes: para escribir cualquier texto en nuestra interfaz utilizaremos una tabla cuyas propiedades son las siguientes:

Tipo de letra: "Times New Roman"

Tamaño: 16px;

Color de fondo: #FFFFFF;

Alineación del texto: left;

Color borde de la tabla: solid #000080;

- Enlaces: sin subrayado. Fondo azul oscuro. Los enlaces de la aplicación se abren en la misma ventana o pestaña.
- Cabeceras: las cabeceras de cada .JSP tendrán las siguientes propiedades:

Tipo de letra: "Arial"

Tamaño: 20px

Color de fondo: #000080

Alineación del texto: center;

- Logotipo: arriba a la izquierda. Esta presente en todas las paginas .JSP.

3.4.3.2 Metáforas

Una metáfora es el empleo de un objeto con un significado o dentro de un contexto diferente al habitual. Al diseñar una interfaz gráfica, la utilización de metáforas resulta muy útil ya que permiten al usuario, por comparación con otro objeto o concepto, comprender de una manera más intuitiva las diversas tareas que la interfaz permite desarrollar.

Al igual que pasa en el ámbito de la literatura, para que una metáfora cumpla con su cometido, el desarrollador de la aplicación y el usuario final de está deben tener una base cultural similar. Es muy posible que el uso de un icono de manera metafórica sea entendido de una manera por el usuario occidental y de otra bien distinta por un usuario oriental. Hay que intentar, por lo tanto, que las metáforas empleadas sean lo más universales posibles para que así sean comprendidas a la perfección por la mayor parte del público potencial.

Las aplicaciones de escritorio de Windows suelen seguir la Guía de Estilo XP y utilizan una serie de metáforas con las que el usuario esta plenamente familiarizado (por ejemplo, una lupa con un signo '+' en su interior establece que la función del icono es, inequívocamente, la de realizar un aumento de zoom). En el mundo de las aplicaciones web también existen una cantidad de metáforas de amplia difusión como puede ser, por ejemplo, el célebre carrito de la compra que emplean casi todos los comercios online.

Pero las metáforas no solo dependen del tipo de aplicación (escritorio o web) sino también del ámbito de la misma. Por ejemplo, el carrito de la compra es una metáfora conocida por todos pero si nuestra aplicación no va a vender nada al usuario no resulta conveniente utilizarla ya que puede confundir.

En nuestra aplicación, debido a que es un prototipo, no hemos creído conveniente la realización de ninguna metáfora.

3.4.3.3 Pantallas

Como hemos visto en apartados anteriores, para diseñar una buena interfaz, es necesario un concienzudo trabajo de análisis y diseño.

El diseño gráfico de la interfaz de usuario juega un papel fundamental, ya que una buena distribución, ordenación, y codificación de colores de los elementos de la interfaz hace que ésta sea más clara y por tanto más fácil de usar para el usuario. Por el contrario una colocación incorrecta de los elementos puede hacer muy compleja la utilización de una aplicación por parte de un usuario.

Un principio general es que, los elementos que posee la interfaz tienen que estar colocados de tal modo que, a la hora de realizar una tarea, el usuario deba recorrer la interfaz en la misma dirección que lee un texto, es decir, de izquierda a derecha y de arriba abajo. En países orientales no se realizaría de la misma forma.

Una interfaz de usuario visualmente atractiva hace que sea percibida por el usuario como una mejor interfaz, mientras que una interfaz compleja con sus elementos desordenados, provoca en el usuario insatisfacción y desorientación. No hay que olvidar que una buena interfaz es aquella que encuentra el equilibrio entre la funcionalidad y la estética.

En nuestra aplicación se situaran los botones relacionados en el mismo grupo. Los botones relacionados con una pregunta (Ir a Pregunta, No contestar pregunta, Siguiente Pregunta, Guardar Respuesta) estarán en un mismo grupo, mientras que los relacionados con la encuesta (Salir y Terminar Encuesta) se encontrarán en otro grupo.

El orden de los botones no es aleatorio. Los botones más importantes, los que permiten aceptar una acción (No contestar pregunta, Siguiente Pregunta, Guardar Respuesta, Terminar Encuesta) deben estar situados más a la derecha del grupo de botones al que

pertenecen. A la izquierda se encuentran los botones menos importantes que normalmente permiten salir del sistema o de la pantalla en la que nos encontremos.

3.4.3.4 Caminos de navegación

Hasta este momento tenemos un diseño visual de la interfaz estática, es decir, cada pantalla diseñada individualmente, pero no tenemos una idea de si en el conjunto de la interacción, la acción va a transcurrir de forma comprensible para el usuario. Para ello vamos a diseñar la interfaz en movimiento y comprobar que es usable.

Para estudiar los caminos de interacción se empleara una herramienta llamada *storyboard*, que consiste en mostrar, a modo secuencia, las distintas pantallas por las que se va pasando al realizar el usuario una determinada acción sobre la aplicación. Mediante flechas se ayuda a entender que es lo que a desencadenado el paso de una pantalla a otra. Los *storyboards* también están muy ligados a los escenarios vistos anteriormente.

El *storyboard* sirve de prototipo para ser evaluado por el usuario y poder introducir correcciones en fases iniciales, ya que cuanto más tiempo se tarde en validar una interfaz, más coste de tiempo y trabajo nos ocasionará.

No se han desarrollado *storyboards* para todas las acciones de nuestro sistema por lo que los que se muestran a continuación son los que se han considerado más importantes:

- Storyboard Crear Encuesta
- Storyboard Nueva Pregunta
- Storyboard Modificar Pregunta
- Storyboard Alumno Contesta Encuesta
- Storyboard Profesor Consulta Resultados

Una vez diseñados deben ser validados por los posibles usuarios para comprobar que realmente la aplicación es usable. Se ha pedido ayuda a dos amigos, estudiantes universitarios, para que nos ayuden a validar los *storyboards* desarrollados.

La opinión de ambos es similar, validando y aceptando los *storyboards* desarrollados, por lo que no realizaremos ningún cambio en nuestra interfaz.

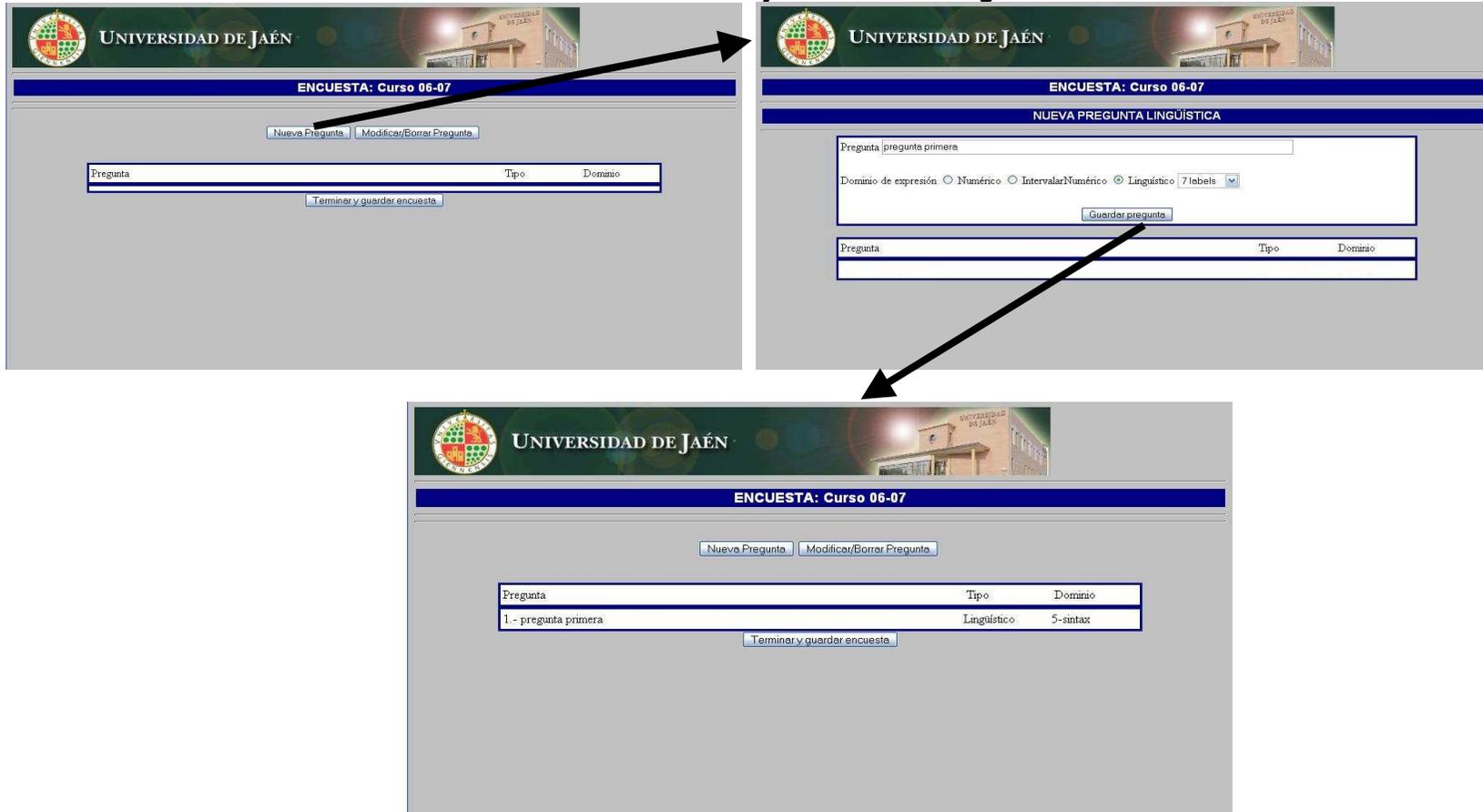
3.4.3.4.1 Storyboard Crear Encuesta



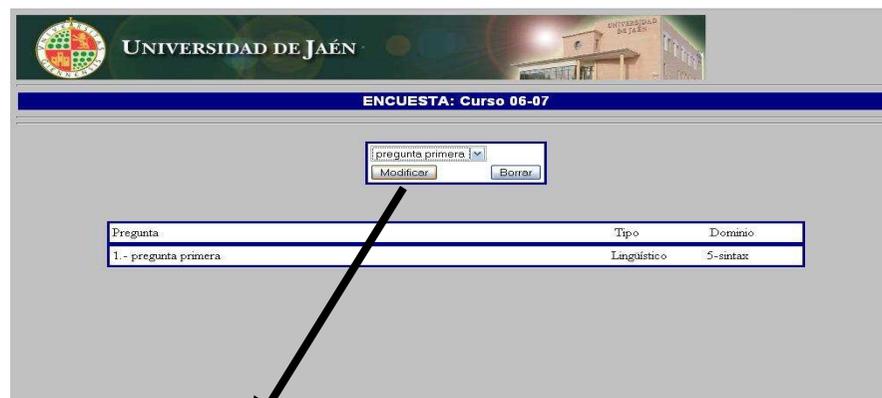
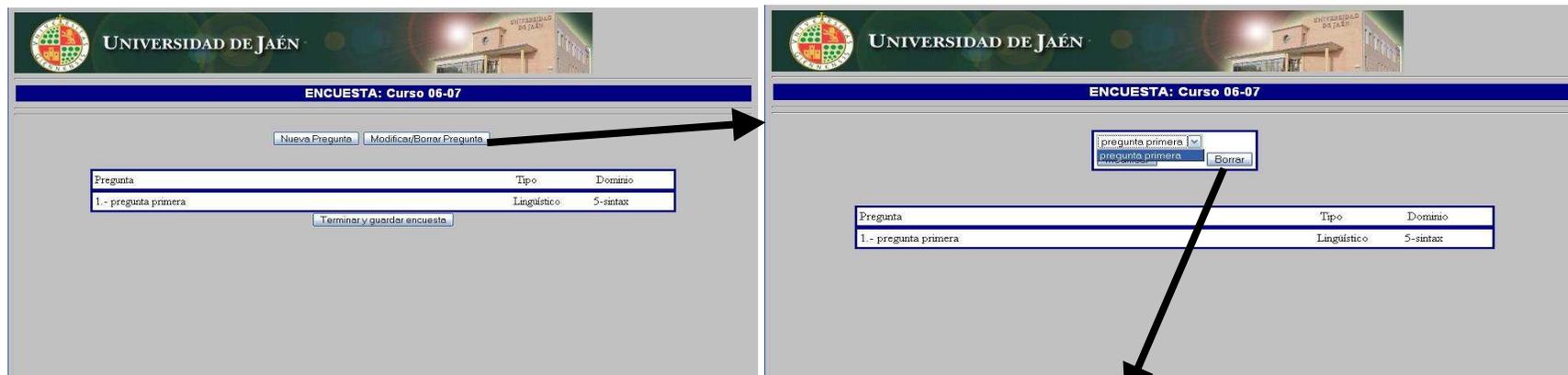
Antonio Javier Sánchez Santiago

Plataforma web de evaluación difusa con información heterogénea

3.4.3.4.2 Storyboard Nueva Pregunta



3.4.3.4.3 Storyboard Modificar Pregunta



Antonio Javier Sánchez Santiago

Plataforma web de evaluación difusa con información heterogénea

ENCUESTA: Curso 06-07

NUEVA PREGUNTA LINGÜÍSTICA

Pregunta: pregunta primera

Dominio de expresión: Numérico IntervalarNumerico Linguistico

5-sintax

Guardar pregunta

Pregunta	Tipo	Dominio
pregunta primera	Lingüístico	5-sintax

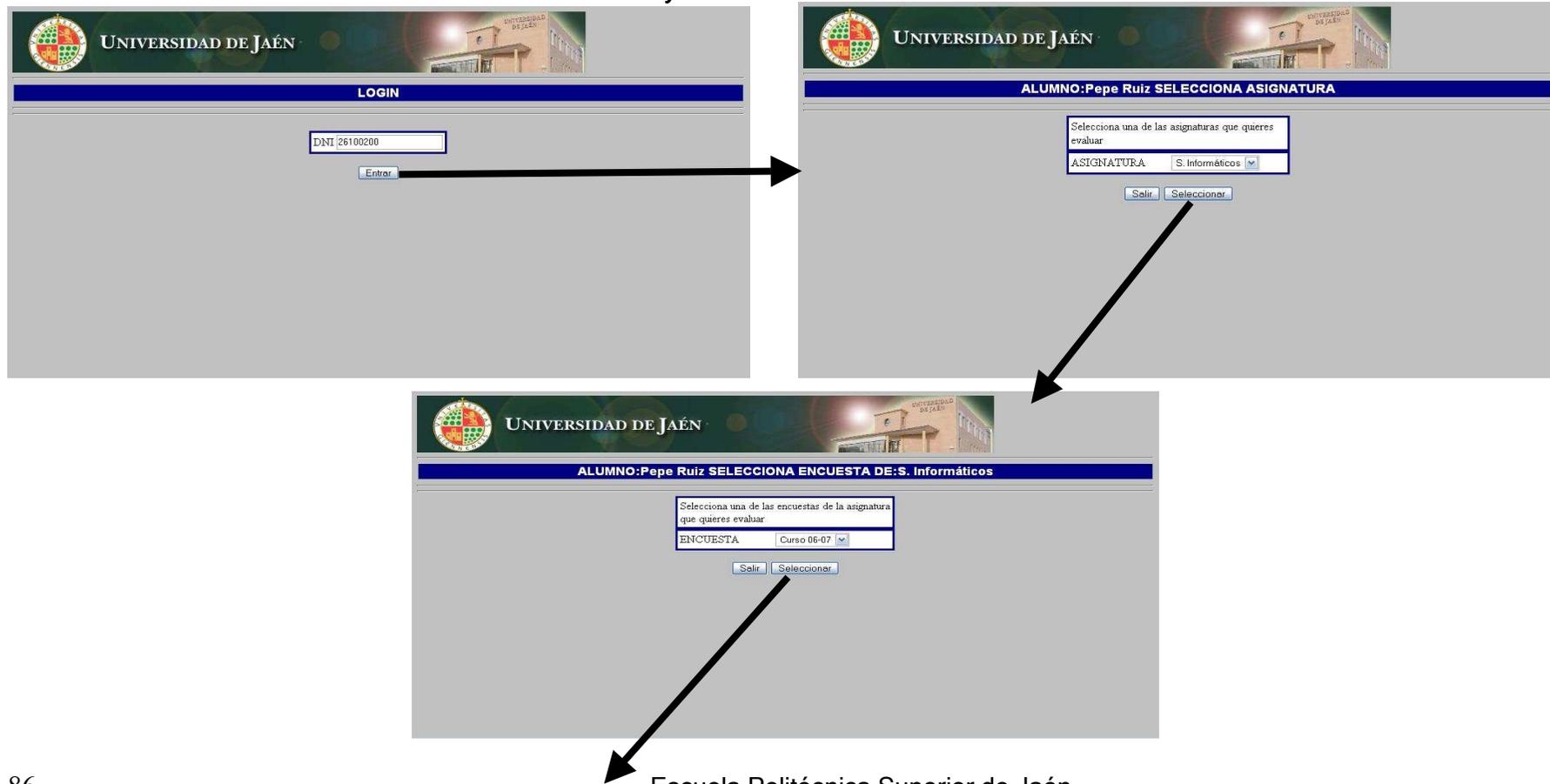
ENCUESTA: Curso 06-07

Nueva Pregunta Modificar/Borrar Pregunta

Pregunta	Tipo	Dominio
1. PRIMERA PREGUNTA	Lingüístico	7-sintax

Terminar y guardar encuesta

3.4.3.4.4 Storyboard Alumno Contesta Encuesta



Antonio Javier Sánchez Santiago

Plataforma web de evaluación difusa con información heterogénea

The image displays three sequential screenshots of a web application interface for a survey. The interface is titled "ENCUESTA: Curso 06-07" and features the logo of the "UNIVERSIDAD DE JAÉN".

Left Screenshot: Shows the question input stage. A text field contains "PRIMERA PREGUNTA". Below it, a dropdown menu is set to "Med". Navigation buttons include "Ir a pregunta", "No Contestar esta pregunta", "Siguiente Pregunta", "Guardar Respuesta", "Salir", and "Terminar Encuesta". A table at the bottom shows the current question status:

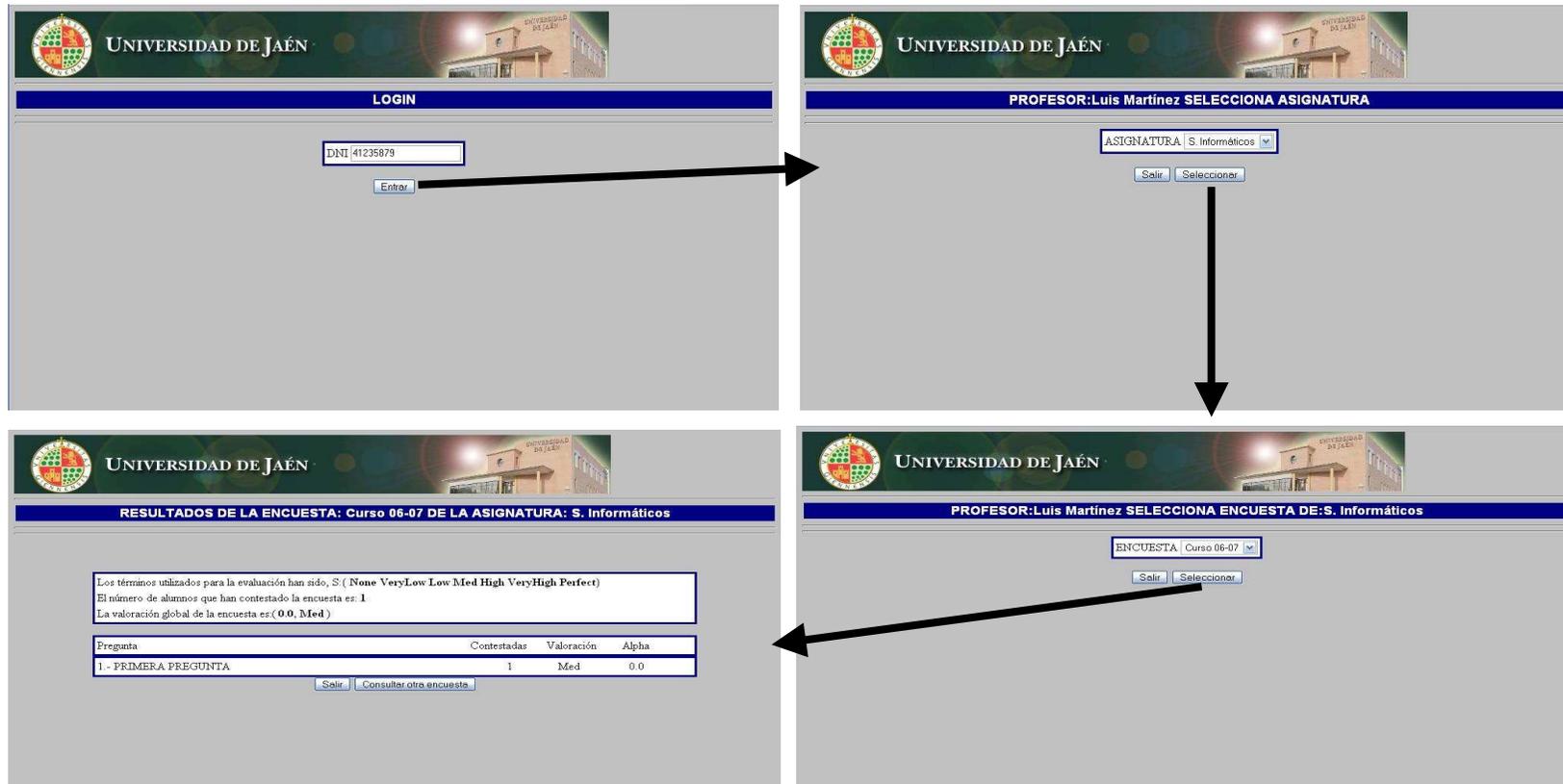
Pregunta	Tipo	Respuesta
1 - PRIMERA PREGUNTA	Lingüístico	No Evaluado

Middle Screenshot: Shows the review stage. A message states: "Estas son tus respuestas. Si deseas modificar alguna respuesta pulsa en Ir a Pregunta. Si has finalizado la encuesta pulsa Enviar Encuesta para almacenarla en el sistema." Below this is a table with the question details and navigation buttons:

Pregunta	Tipo	Respuesta
1 - PRIMERA PREGUNTA	Lingüístico	Med

Right Screenshot: Shows the final confirmation stage. The title is "NOTA" and the message reads: "Encuesta guardada en el sistema". A button "Responder otra encuesta" and a link "Salir" are visible.

3.4.3.4.5 Storyboard Profesor Consulta Resultados



3.4.3.5 Mensajes de error

Una vez diseñadas las pantallas, los caminos de navegación y las secuencias de diálogo, se tiene una idea muy clara de cómo se va a desarrollar físicamente la interacción entre usuario y ordenador, y por tanto de las posibles situaciones de error que pueden darse en esa interacción. Por ello ahora es el momento de diseñar los mensajes de error.

Los mensajes de error son el medio por el que el sistema comunica al usuario que se ha producido un error en la interacción.

Los errores se producen por falta de conocimiento sobre la interfaz, porque se no se ha entendido correctamente el estado del sistema o bien inadvertidamente (por ejemplo se pulsa un botón cuando se intentaba pulsar otro). Ello lleva al usuario a sentirse confuso y aumenta su ansiedad, sobre todo en usuarios noveles en los que su falta de conocimiento y confianza en el uso de la interfaz lleva a amplificar el stress lo que puede llevarles a una experiencia de uso de la interfaz frustrante.

Los mensajes de error son muy importantes de cara a la usabilidad, ya que bien diseñados, permiten aumentar la confianza del usuario en el uso de la interfaz y que pueda seguir con su tarea. Por ello hay que diseñarlos con cuidado. A la hora de diseñar mensajes de error se deberían seguir las siguientes reglas:

- Ser breves: el usuario es una persona ocupada tratando de llevar a cabo una tarea. Si se le presenta un mensaje de error muy largo, lo normal es que no lo lea ya que no tendrá tiempo para hacerlo. Por ello hay que diseñar mensajes de error claros, pero breves.
- Ser específico: los mensajes demasiado generales no son adecuados ya que dicen al usuario que algo ha ido mal, pero no le indican claramente qué. Por lo

tanto el usuario no sabrá qué hacer para impedir que se produzca el error y su sensación de frustración aumentará

- Usar un tono positivo y guía constructiva: nunca recriminar al usuario lo que ha hecho mal, en su lugar siempre que sea posible indicarle que debe hacer para eliminar el error.
- Usar un formato físico apropiado: la mayoría de usuarios encuentran más fácil leer un mensaje donde se mezclan letras mayúsculas y minúsculas de la forma habitual, por tanto este formato es siempre preferible. Los mensajes escritos únicamente en mayúsculas deberían reservarse para avisos breves y graves. Si el mensaje de error debe contener un código numérico, este debería figurar al final del mensaje y entre paréntesis.

Una vez expuestas las reglas a seguir se exponen algunos de los mensajes de error utilizados en nuestro sistema. Todos los mensajes de error que ofrece nuestra aplicación se encuentran en el archivo error.jsp.

- Pantalla Inicial: no existe ninguna situación anómala, por lo que no aparece ningún tipo de mensaje.
- Pantalla de identificación: existe la posibilidad de que un usuario se identifique de forma incorrecta. Si esto ocurre el mensaje de error que se muestra es el siguiente: "Error: DNI/Login inválido"
- Pantalla de crear una encuesta: si el nombre introducido para la nueva encuesta se encuentra repetido el error mostrado es: "Error: Este nombre de encuesta ya existe."

- Pantalla insertar un alumno: si introducimos un DNI erróneo, el sistema nos muestra este mensaje de error: "Error: Introduzca el DNI sin letra, o introduzca los ocho números del DNI"
- Pantalla insertar un profesor: si se deja en blanco alguno de los campos la aplicación nos muestra un error dependiendo del campo que se encuentre en blanco. Por ejemplo si dejamos en blanco los apellidos de un profesor, nos muestra este error: "Error: Por favor, introduzca los apellidos del Profesor."
- Pantalla contestar una encuesta: si en una pregunta de tipo intervalar o numérico introducimos un valor que no está dentro del dominio elegido, nos muestra este mensaje de error: "Error: El valor introducido está fuera del rango. Vuelva a introducir un valor"

3.5 Implementación

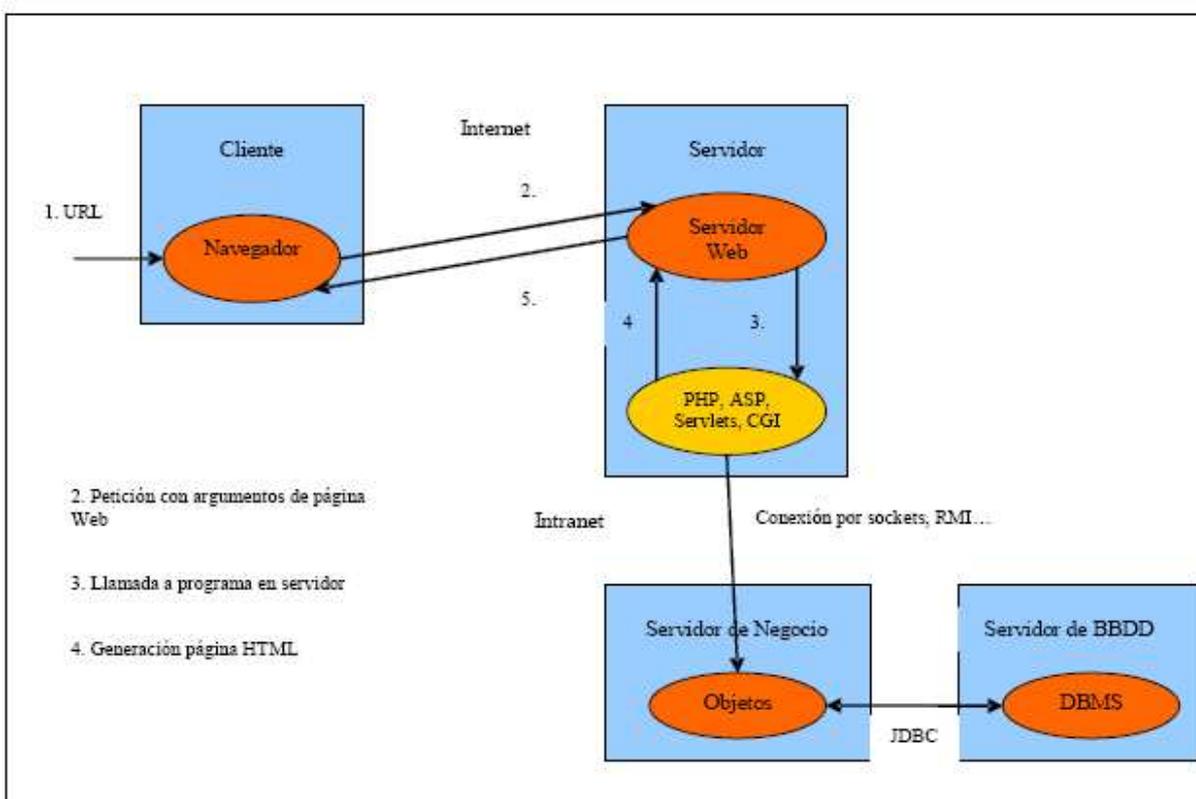
La implementación es la actividad final de la Ingeniería del Software, aquella en la que el modelo obtenido en las actividades anteriores se debe transformar en código fuente.

Para ello se debe ser cuidadoso en la elección del lenguaje de programación empleado para la codificación y de la herramienta utilizada para generarla.

En nuestro caso la elección del lenguaje de programación así como el de la herramienta utilizada para su desarrollo viene dado desde la definición del proyecto.

3.5.1. Tipo de arquitectura de la aplicación

En nuestro caso, vamos a desarrollar una sistema con una arquitectura cliente/servidor y una interfaz web de comunicación con los usuarios. El funcionamiento de las arquitecturas de este tipo es sencilla: la aplicación se encuentra en un servidor central al que los usuarios acceden a través de un software cliente, en nuestro caso un navegador web. Una vez que ha accedido a la aplicación, el usuario realiza peticiones que el servidor tiene que atender para generar una respuesta comprensible para el cliente.



Una arquitectura cliente/servidor Web libera, por lo tanto, al usuario final de la aplicación de tener que instalarla en su máquina y consigue que cada usuario solo pueda acceder a la información que le corresponde. Además, este tipo de arquitectura, gracias a su diseño

modular, es fácilmente escalable y ampliable tanto en nuevos clientes como en servidores añadidos.

3.5.2 Lenguajes de programación utilizados

Resulta obvio ante la arquitectura y el funcionamiento previsto de nuestra aplicación que el uso de HTML simple y llano no es adecuado sino que se necesita otro lenguaje capaz de generar contenido dinámico desde el servidor de manera transparente al usuario final. Para este fin hemos utilizado JSP.

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

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

JSP no se puede considerar un script al 100%, ya que antes de ejecutarse el Servidor de Aplicaciones compila el contenido del documento JSP (script y etiquetas) y genera una clase Servlet. Por lo tanto, se puede decir que aunque este proceso sea transparente para el programador no deja de ser una tecnología compilada.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos (así como las que requieren más seguridad) y dejando la parte encargada de formatear el documento HTML en el archivo JSP. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información.

Para realizar una interfaz web adecuada se hace necesario el uso de otros dos lenguajes de programación: CSS y Javascript.

CSS, acrónimo de Cascade Style Sheets, es un lenguaje formal que ayuda a separar la estructura interna de un documento de su presentación externa. Las etiquetas de estilo CSS pueden presentarse tanto dentro de un documento HTML (encerradas dentro de las etiquetas `<style type="text/css"></style>` en la cabecera) como en un documento aparte (con extensión `.css`) al que el documento HTML se encarga de llamar cuando es necesario. De esta última manera no solo se consigue separar la estructura de la presentación sino que se consigue la centralización del estilo ya que una sola hoja de estilos CSS puede ser invocada por distintas páginas de la aplicación web lo que ayuda de manera muy importante al mantenimiento de la coherencia y consistencia del diseño de la aplicación.

Por su parte, Javascript, lenguaje interpretado de sintaxis similar a lenguajes como Java o C que se ejecuta del lado del cliente, ayuda a comprobar si los datos que introduce el cliente son erróneos informando de su error mediante mensajes de error o alerta.

Al igual que ocurre con CSS, el código Javascript puede ir incrustado dentro del documento HTML (entre las etiquetas `<script type="text/javascript"></script>` en el cuerpo o la cabecera) o estar almacenado en ficheros aparte (con extensión `.js`) y ser invocados por el documento. Para nuestra aplicación, para los estilos CSS nos hemos decantado por la segunda opción, mientras que para el código Javascript, al ser utilizado muy pocas veces lo hemos incrustado dentro del documento HTML.

Como ya se vio al comienzo de esta memoria, para la lógica de negocio del sistema se va a utilizar tecnología JAVA, es decir, Servlets.

3.5.3 Herramienta de desarrollo

La herramienta escogida para desarrollar todo el código ha sido JBuilder de Borland, y más concretamente la edición Enterprise que es la destinada para aplicaciones J2EE, Web Services y Struts.

3.5.4 Instalación en el servidor y funcionamiento

La instalación de la aplicación así como la puesta en marcha del servidor viene documentada paso a paso en el Anexo II.

Por su parte, en el Anexo III se encuentra disponible el manual de usuario para el Administrador. Los manuales de uso para el alumno y el profesor se encuentran en los Anexos IV y V, respectivamente.

3.6 Implantación y pruebas

Esta fase de implantación junto con la del mantenimiento del software son partes muy importantes de la Ingeniería del software.

Consisten en desplegar el software realizado y tratar de mejorar u optimizar cualquier problema encontrado.

El mantenimiento de software involucra cualquier tipo de pruebas del software realizadas, las cuales son un elemento crítico para la garantía de calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación.

3.6.1 Pruebas y validación

El objetivo de esta fase es realizar un conjunto de pruebas sobre el sistema. Con esto intentaremos conseguir llegar a un sistema sin errores garantizando, como hemos dicho, la calidad del software. Para comprobar esto realizaremos unas pruebas de sistema.

3.6.1.1 Casos de Test

Los test diseñados son los siguientes:

Test 1: Nuevo alumno.

Requisitos testeados	RF01
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Alumnos
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Alta.
Checkpoint 2	El sistema debe mostrar una pantalla con la petición de datos del alumno.
Acción	Rellena todos los datos y pulsa Guardar.
Checkpoint 3	El sistema informa de que se ha insertado un nuevo alumno en la base de datos.

Test 2: Nuevo profesor.

Requisitos testeados	RF01
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Profesores
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Alta.
Checkpoint 2	El sistema debe mostrar una pantalla con la petición de datos del profesor.

Acción	Rellena todos los datos y pulsa Guardar.
Checkpoint 3	El sistema informa de que se ha insertado un nuevo profesor en la base de datos.

Test 3: Nuevo alumno con DNI repetido.

Requisitos testeados	RF01
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Alumnos
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Alta.
Checkpoint 2	El sistema debe mostrar una pantalla con la petición de datos del alumno.
Acción	Rellena todos los datos y pulsa Guardar.
Checkpoint 3	El sistema informa de que ya existe un alumno con este DNI. Pide al administrador que introduzca un DNI válido.

Test 4: Nuevo profesor con apellidos en blanco

Requisitos testeados	RF01
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Profesores
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Alta.
Checkpoint 2	El sistema debe mostrar una pantalla con la petición de datos del profesor.
Acción	Rellena todos los datos y pulsa Guardar.
Checkpoint 3	El sistema informa de que no puede haber campos en blanco. Pide al administrador que introduzca los apellidos del profesor.

Test 5: Modificar los datos de un profesor

Requisitos testeados	RF01
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Profesores
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Modificación.
Checkpoint 2	El sistema debe mostrar una pantalla con una lista desplegable, que contenga todos los DNI de los profesores del sistema

Acción	Elige el DNI del profesor que quiere modificar. Pulsa el botón Modificar.
Checkpoint 3	El sistema muestra los datos de ese profesor.
Acción	Modifica los datos del profesor y pulsa el botón Guardar.
Checkpoint 4	El sistema informa de que se han modificado y guardado los datos de ese profesor.

Test 6: Asignar a un alumno una asignatura

Requisitos testeados	RF02
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Alum/Asignaturas
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Alta.
Checkpoint 2	El sistema debe mostrar una pantalla con dos listas desplegadas. Una con los DNI de los alumnos y otra con los nombres de las asignaturas.
Acción	Elige el DNI del alumno y la asignatura a asignar. Pulsa el botón Guardar
Checkpoint 3	El sistema informa de que la asignación se ha realizado de forma correcta

Test 7: Asignar a un alumno una asignatura a la que ya esta asignado.

Requisitos testeados	RF02
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Alum/Asignaturas
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Alta.
Checkpoint 2	El sistema debe mostrar una pantalla con dos listas desplegables. Una con los DNI de los alumnos y otra con los nombres de las asignaturas.
Acción	Elige el DNI del alumno y la asignatura en la que está matriculado
Checkpoint 3	El sistema informa de que la asignación se ha realizado de forma correcta

Test 8: Consultar los alumnos del sistema

Requisitos testeados	RF03
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Alumnos
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Consultar.
Checkpoint 2	El sistema debe mostrar la información de todos los alumnos que existen en el sistema.

Test 9: Consultar los profesores del sistema

Requisitos testeados	RF03
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración de Profesores
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Alta, Modificación y Consulta.
Acción	Pulsa la opción de Consultar.
Checkpoint 2	El sistema debe mostrar la información de todos los profesores que existen en el sistema.

Test 10: Crear una encuesta

Requisitos testeados	RF04
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración encuestas
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Nueva Encuesta, Modificar/Consultar, Cerrar Encuesta
Acción	Pulsa la opción de Nueva Encuesta
Checkpoint 2	El sistema debe mostrar una pantalla en la que pide el nombre de la encuesta.
Acción	Introduce un nombre de encuesta correcto.
Checkpoint 3	El sistema debe mostrar una pantalla con los botones: Nueva Pregunta, Modificar/Borrar Pregunta, Terminar y Guardar

	Encuesta
Acción	Pulsa el botón Nueva Pregunta
Checkpoint 4	El sistema debe mostrar el formulario para introducir una pregunta
Acción	Rellena todos los campos y pulsa el botón Guardar Pregunta.
Checkpoint 5	El sistema debe mostrar una pantalla con los botones: Nueva Pregunta, Modificar/Borrar Pregunta, Terminar y Guardar Encuesta y una tabla con la pregunta que acaba de ser introducida.
Acción	Pulsa el botón Terminar y Guardar Encuesta.
Checkpoint 6	El sistema debe informar de que la encuesta se ha grabado correctamente.

Test 11: Crear una encuesta con un nombre ya utilizado.

Requisitos testeados	RF04
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración encuestas
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Nueva Encuesta, Modificar/Consultar, Cerrar Encuesta
Acción	Pulsa la opción de Nueva Encuesta
Checkpoint 2	El sistema debe mostrar una pantalla en la que pide el nombre de la encuesta.
Acción	Introduce un nombre de encuesta ya utilizado
Checkpoint 3	El sistema debe informar de que ese nombre de encuesta ya ha sido utilizado. Pide al usuario que introduzca un nombre correcto.

Test 12: Modificar una encuesta.

Requisitos testeados	RF04
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración encuestas
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Nueva Encuesta, Modificar/Consultar, Cerrar Encuesta
Acción	Pulsa la opción de Modificar/Consultar
Checkpoint 2	El sistema debe mostrar una pantalla con una lista desplegable que contiene los nombres de todas las encuestas.
Acción	Selecciona el nombre de la encuesta que quiere modificar y pulsa el botón Modificar.
Checkpoint 3	El sistema debe mostrar una pantalla con los botones: Nueva Pregunta, Modificar/Borrar Pregunta, Terminar y Guardar Encuesta. Debe mostrar también una tabla con las preguntas ya existentes de esa encuesta.
Acción	Pulsa el botón Modificar/Borrar Pregunta
Checkpoint 4	El sistema debe mostrar una lista desplegable con las preguntas de la encuesta.
Acción	Selecciona la pregunta que quiere modificar y pulsa el botón Modificar.
Checkpoint 5	El sistema debe mostrar el formulario para introducir o modificar una pregunta
Acción	Rellena todos los campos y pulsa el botón Guardar Pregunta.
Checkpoint 6	El sistema debe mostrar una pantalla con los botones: Nueva Pregunta, Modificar/Borrar Pregunta, Terminar y Guardar

	Encuesta y una tabla con las preguntas de la encuesta
Acción	Pulsa el botón Terminar y Guardar Encuesta.
Checkpoint 7	El sistema debe informar de que la encuesta se ha grabado correctamente.

Test 13: Modificar una encuesta ya asignada.

Requisitos testeados	RF04
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración encuestas
Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Nueva Encuesta, Modificar/Consultar, Cerrar Encuesta
Acción	Pulsa la opción de Modificar/Consultar
Checkpoint 2	El sistema debe mostrar una pantalla con una lista desplegable que contiene los nombres de todas las encuestas.
Acción	Selecciona el nombre de la encuesta que quiere modificar y pulsa el botón Modificar.
Checkpoint 3	El sistema debe mostrar una pantalla una tabla con las preguntas de esa encuesta y otra tabla indicando al Administrador que la encuesta ya ha sido asignada y que por lo tanto no puede ser modificada.
Acción	Pulsa el botón Volver
Checkpoint 4	El sistema muestra el menú principal.

Test 14: Asignar una encuesta a un profesor.

Requisitos testeados	RF05
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración Encuesta/Asig
Checkpoint 1	El sistema debe mostrar las dos opciones posibles: Alta y Consulta
Acción	Pulsa la opción de Alta
Checkpoint 2	El sistema debe mostrar una pantalla con dos listas desplegables una que contiene los nombres de todas las encuestas y otra con los nombres de las asignaturas.
Acción	Selecciona el nombre de la encuesta y de la asignatura, en la que el profesor imparte docencia, para ser evaluado.
Checkpoint 3	El sistema debe mostrar una pantalla en la que informa de la acción a realizar y pide confirmación por parte del administrador
Acción	Pulsa el botón Si.
Checkpoint 4	El sistema informa de que la asignación se ha realizado de forma correcta.

Test 15: Cerrar en plazo para evaluar un profesor

Requisitos testeados	RF06
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Administración encuestas

Checkpoint 1	El sistema debe mostrar las tres opciones posibles: Nueva Encuesta, Modificar/Consultar, Cerrar Encuesta
Acción	Pulsa la opción de Cerrar Encuesta
Checkpoint 2	El sistema debe mostrar una pantalla con una lista desplegable que contiene los nombres de todas las asignaturas.
Acción	Selecciona el nombre de la asignatura en la que el profesor imparte docencia.
Checkpoint 3	El sistema debe mostrar una pantalla con una lista desplegable que contiene los nombres de todas las encuestas relacionadas con la asignatura, que actualmente están sin cerrar.
Acción	Pulsa el botón Cerrar
Checkpoint 4	El sistema debe mostrar una pantalla en la que informa de la acción a realizar y pide confirmación por parte del administrador
Acción	Pulsa el botón Si.
Checkpoint 5	El sistema informa de que el cierre se ha realizado de forma correcta.

Test 16: Consultar los resultados de una encuesta

Requisitos testeados	RF07
Precondiciones	Usuario Administrador identificado, se encuentra en el menú del administrador.
Contenido	
Acción	Accede a: Consulta de Resultados
Checkpoint 1	El sistema debe mostrar una pantalla con una lista desplegable que contiene los nombres de todas las asignaturas.
Acción	Selecciona el nombre de la asignatura de la que consultar resultados

Checkpoint 2	El sistema debe mostrar una pantalla con una lista desplegable que contiene los nombres de todas las encuestas cerradas, de la asignatura seleccionada.
Acción	Selecciona el nombre de la asignatura en la que el profesor imparte docencia.
Checkpoint 3	El sistema debe mostrar una pantalla con los resultados de la encuesta.
Acción	Pulsa el botón Salir
Checkpoint 4	El sistema vuelve al menú principal.

3.6.1.2 Resultados obtenidos

Mostramos la hoja de resultados de los test diseñados una vez ya realizados sobre el sistema:

TEST	RESULTADO
Test 1: Nuevo alumno	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 2: Nuevo profesor.	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 3: Nuevo alumno con DNI repetido.	
Checkpoint 1	OK
Checkpoint 2	OK

Checkpoint 3	OK
Test 4: Nuevo profesor con apellidos en blanco	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 5: Modificar los datos de un profesor	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Test 6: Asignar a un alumno una asignatura	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 7: Asignar a un alumno una asignatura a la que ya esta asignado	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 8: Consultar los alumnos del sistema	
Checkpoint 1	OK
Checkpoint 2	OK
Test 9: Consultar los profesores del sistema	
Checkpoint 1	OK
Checkpoint 2	OK
Test 10: Crear una encuesta	

Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Checkpoint 5	OK
Checkpoint 6	OK
Test 11: Crear una encuesta con un nombre ya utilizado	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 12: Modificar una encuesta	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Checkpoint 5	OK
Checkpoint 6	OK
Checkpoint 7	OK
Test 13: Modificar una encuesta ya asignada	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Test 14: Asignar una encuesta a un profesor	
Checkpoint 1	OK
Checkpoint 2	OK

Checkpoint 3	OK
Checkpoint 4	OK
Test 15: Cerrar en plazo para evaluar un profesor	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Checkpoint 5	OK
Test 16: Consultar los resultados de una encuesta	
Checkpoint 1	OK

CAPITULO 4

Conclusiones

4.1 Conclusiones

Una vez concluido el proyecto y haberlo realizado con éxito podemos destacar diversos puntos en este apartado de conclusiones.

El objetivo principal de este era la realización de un modulo web que permitiera a los usuarios de un servicio, en este caso alumnos, la realización de encuestas de evaluación a través de una web, sencilla y clara, y que posibilitara realizar la evaluación en contextos heterogéneos y, la consulta y cálculo de resultados.

Tras la realización del Proyecto podemos decir que este objetivo principal se ha alcanzado con total éxito.

Por otra parte, el uso de la tecnología J2EE, es un factor del proyecto que ha causado en mí un gran interés. En este conjunto de servicios basados en el lenguaje de programación Java he encontrado multitud de características tales como su potencial, elegancia al desarrollar, claridad y orden, y una facilidad de diseño que marcaran, tal vez, mi el deseo de inclinarme sobre esta tecnología como la predilecta o elegida con la que desarrollar mi carrera profesional.

Pero no debemos olvidar otra parte fundamental de este Proyecto Fin de Carrera, y es diseñar y desarrollar este siguiendo detenidamente cada una de las fases de la Ingeniería del Software.

Se han comprendido los objetivos de cada etapa y se ha puesto en práctica cada una de ellas, observando que todo lo estudiado en la Ingeniería del Software ha dado su fruto y se ha sido capaz sin apenas problemas de realizar todas las fases de la Ingeniería del Software.

Finalmente, comentar que, la realización de este proyecto ha servido para poner en

práctica muchas de las metodologías y habilidades adquiridas durante los años de formación académica y otras muchas que he tenido que adquirir durante su desarrollo.

4.2 Mejoras futuras

Como es lógico, hay que comentar que este sistema se trata de un Proyecto Fin de Carrera y que por lo tanto, no es más que una versión demo o un prototipo el cual tan sólo posee las funcionalidades básicas.

En este apartado explicaremos las posibles ampliaciones, las cuales fueron comentadas en el capítulo anterior, que se podrían realizar para versiones futuras. Veamos algunas de ellas:

- *Automatización de apertura y cierre del plazo de evaluación de un profesor.* La finalidad de esta posible ampliación es la de permitir al Administrador, cuando realiza una encuesta, introducir la fecha en la que puede empezar a ser contestada y la fecha en la que finalizará el plazo para ser contestada, realizando a continuación el cálculo de resultados.
- *Mandar un correo electrónico a los alumnos cuando este abierto el plazo para rellenar las encuestas asociadas a sus asignaturas.* Con esta ampliación se facilitaría a los alumnos la posibilidad de contestar una encuesta, ya que no tendrían que estar entrando constantemente en el sistema para comprobar si tienen una encuesta que contestar.
- *Mandar un correo electrónico a los profesores para indicarle que ya se encuentran disponibles los resultados asociados a una de sus asignaturas.* Esta ampliación es similar a la anterior, solo que aplicada a los profesores, a los cuales les permitiría conocer de una forma más rápida los resultados de la evaluación de sus

asignaturas.

- *Consultas avanzadas con parámetros.* Esta ampliación estaría dedicada al administrador del sistema, permitiéndole realizar cualquier tipo de consulta y no solo las desarrolladas en este proyecto.
- *Crear nuevos dominios.* Con esta ampliación se permitiría al administrador la creación de nuevos dominios de valoración.
- *Crear una base de datos nueva.* Con esta ampliación se pretende crear una base de datos vacía, sin ningún registro, a partir de la ya existente, pudiendo servir, por ejemplo, para crear encuestas en un nuevo curso académico.

BIBLIOGRAFÍA

Bibliografía específica para modelos de evaluación

[Car01] Carlsson C., Fuller R., Fuzzy Reasoning in Decision Making and Optimization, Studies in Fuzziness and Soft Computing Series, Vol 82, Springer-Verlag, Berlin-Heidelberg, 2001.

[Che01] Chen C.T., *Applying linguistic decision-making method to deal with service quality evaluation problems*. International. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (Suppl.) pp. 103- 114. 2001

[Chi98] Chiclana F., Herrera F., Herrera-Viedma E., Integrating Three Representation Models in Fuzzy Multipurpose Decision Making Based on Fuzzy Preference Relations. Fuzzy Sets and Systems 97 (1998), pp. 33-48

[Deg88] Degani R., Bortolan G., The problem of Linguistic Approximation in Clinical Decision Making, Int. Journal of Approximate Reasoning 2 (1988) 143-162

[Del93] Delgado M., Verdegay J.L., Vila M.A., On Aggregation Operations of Linguistic Labels, Int. J. of Intelligent Systems 8 (1993)351-370.

[Del98] Delgado M., Herrera F., Herrera-Viedma E., Martínez L., Combining Numerical and Linguistic Information in Group Decision Making. Information Sciences 107 (1998) 177-194

[Dev99] Devedzic G.B. , Pap E., *Multicriteria-multistages linguistic evaluation and ranking of machine tools*. Fuzzy Sets and Systems 102 pp. 451-461. 1999.

[Dub80] Dubois D., Prade H., Fuzzy Sets and Systems: Theory and Applications, (Academic, New York, 1980).

[Her00] Herrera F., Herrera-Viedma E., Martínez L., *A Fusion Approach for Managing Multi-Granularity Linguistic Term Sets in Decision Making*. Fuzzy Sets and Systems 114 pp. 43-58. 2000.

[Her00a] Herrera F., Martínez L., A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words. IEEE Transactions on Fuzzy Systems 8:6 (2000) 746-752.

[Her00c] Herrera F., Martínez L., Sánchez P.J., Integration of Heterogeneous Information in Decision-Making Problems. IPMU 2000, Madrid (Spain), Jul 2000.

[Her01a] Herrera F., Martínez L., A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in Multiexpert Decision-Making. IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics, 2001, Vol 31 Num 2 pp. 227.234

[Her01b] **Herrera F., Martínez L.**, The 2-tuple Linguistic Computational Model. Advantages of its linguistic description, accuracy and consistency. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9 (2001) 33-49.

[Her03] **F. Herrera, E. Herrera-Viedma, L. Martínez, F. Mata, P.J. Sanchez.** A Multi-Granular Linguistic Decision Model for Evaluating the Quality of Network Services, in: *Intelligent Sensory Evaluation*, Ruan Da (Ed.) Springer, 2003

[Her03a] **F. Herrera, E. Herrera-Viedma, L. Martínez, P.J. Sanchez.** A Linguistic Decision Process for Evaluating the Installation of an ERP System. 9th International Conference on Fuzzy Theory and Technology, Cary (North Carolina) USA, 2003

[Her05] **Herrera F., Martínez L., Sánchez P.J.**, Managing *nonhomogeneous information in group decision making*. *European Journal of Operational Research* 166:1(2005) pp. 115-132.

[Liu04] **Liu J., Yang J. B., Wang J., Sii H. S., Wang Y. M.**, *Fuzzy rulebased evidential reasoning approach for safety analysis*. *International Journal of General Systems* 33 (2-3) pp. 183-204. 2004.

[Mar05] **Martínez L., Sánchez P.J., García C., Montes B., Mata F., Pérez L.G.**, *Proyecto: UJA 23 "Un Sistema de Evaluación basado en técnicas de decisión difusas"*. Convocatoria UCUA: Grupos de Estudio y Análisis específicos de la Unidad para la Calidad de las Universidades Andaluzas. UCUA 2005.

[Rou97] **Roubens M.**, Fuzzy Sets and Decision Analysis, *Fuzzy Sets and Systems* 90 (1997) 199-206.

[Tri00] **Triantaphyllou E.**, *Multi-Criteria Decision Making Methods: A comparative Study*, Kluwer Academic Publishers, Dordrecht, 2000.

[Xu04] **Xu Z.S.**, *A method based on linguistic aggregation operators for group decision making with linguistic preference relations*. *Information Sciences* 166 pp. 19-30. 2004.

[Zad65] **Zadeh L.A.**, Fuzzy Sets, *Information and Control* 8 (1965) 338-353.

[Zad75] **Zadeh L.A.**, The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. Part I, *Information Sciences* 8 (1975) 199-249, Part II, *Information Sciences* 8 (1975) 301-357, Part III, *Information Sciences* 9 (1975) 43-80.

Bibliografía general para la realización del proyecto

[Bas06] Basham B., Sierra K., Bates B., *Head First Servlets y JSP*. 2006

[Cal 03] Calvert C., Calvert M., *Aprenda JBUILDER con Charlie Calvert*. 2003

[Fre06] Freeman E., Freeman E., *Head First HTML with CSS y XHTML*. 2006

[Man02] Manchón, E., *¿Qué es usabilidad?* 2002

[Tome] <http://tomcat.apache.org/tomcat-4.1-doc/index.html> Documentación Apache Tomcat 4.1

ANEXO I

Descripción del proceso de evaluación

I.1 Introducción

En el área del análisis de la decisión, los problemas de toma de decisión con múltiples fuentes de información han recibido una especial atención por parte de investigadores pertenecientes a un amplio grupo de disciplinas. En esta contribución nosotros nos centramos en los problemas de Toma de Decisión en Grupo (TDG) en los cuales hay un conjunto finito de alternativas $X = \{x_1, \dots, x_n\}$ con $n \geq 2$, así como un conjunto finito de expertos $E = \{e_1, \dots, e_m\}$ con $m \geq 2$ que tienen como objetivo alcanzar una solución al problema teniendo en cuenta todas las opiniones. Los expertos suministrarán sus preferencias mediante relaciones de preferencia [13, 15].

En problemas de decisión con varios individuos, cada uno de ellos tiene su propio conocimiento sobre las alternativas del problema y cada uno de ellos según su área, su conocimiento sobre el problema o a la naturaleza de los aspectos valorados puede expresar sus preferencias en un dominio de expresión diferente. Esto implica que la información que maneja el problema podría ser no homogénea. Sin embargo, en la literatura encontramos que en los problemas de TDG los expertos generalmente expresan sus preferencias en un único dominio: numérico [15, 23] o lingüístico [6, 24]. Si los aspectos a valorar tienen una naturaleza cuantitativa expresar las preferencias mediante valoraciones numéricas suele ser adecuado, sin embargo, si trabajamos con aspectos cualitativos este modo de expresar las preferencias no es el más idóneo, habiendo sido el uso del enfoque lingüístico difuso [26] el que ha dado buenos resultados en el modelado de este tipo de aspectos [3, 4, 17, 18, 24, 25].

En este capítulo nos planteamos construir un modelo de decisión que ofrezca una mayor flexibilidad a los expertos a la hora de proveer sus preferencias sobre el problema, tal que, atendiendo al área del mismo y/o de la naturaleza de las alternativas; dicho experto pueda expresar su conocimiento en el dominio de expresión que le resulte más cómodo para que sus valoraciones se ajusten lo máximo posible a su conocimiento real sobre las mismas.

En [8, 13, 14] se han presentado distintos modelos que nos permiten trabajar con información heterogénea (numérica, intervalar y lingüística) en problemas de Toma de Decisión.

Por tanto, el esquema de definición de un problema de TDG es el siguiente:

Dado de un conjunto finito de alternativas,

$$X = \{x_1, \dots, x_n\} \quad n \geq 2,$$

y un conjunto finito de expertos ,

$$E = \{e_1, \dots, e_m\} \quad m \geq 2.$$

Las preferencias de los expertos son expresadas mediante relaciones de preferencia:

$$P_{e_k} = \begin{pmatrix} p_{11}^k & \dots & p_{1n}^k \\ \vdots & \dots & \vdots \\ p_{n1}^k & \dots & p_{nn}^k \end{pmatrix}$$

en donde p_{ij}^k es el grado de preferencia de la alternativa i sobre la alternativa j dado por el experto k .

Como hemos indicado anteriormente, dependiendo de la naturaleza de las alternativas o del conocimiento que tengan los expertos sobre las alternativas, esta información de preferencia puede venir expresada en distintos dominios: numérico, $[0,1]$, intervalar, $([0, 1])$, o el lingüístico, S .

$$p_{ij}^k \in [0, 1], \text{ o } p_{ij}^k \in \wp([0, 1]), \text{ o } p_{ij}^k \in S$$

Un proceso de decisión constará de las siguientes fases [21]:

- Fase de agregación: esta fase tiene como objeto obtener un valor colectivo para cada una de las alternativas valoradas por los distintos expertos.
- Fase de explotación: busca obtener la alternativa o conjunto de alternativas solución al problema.

Sin embargo, en el tipo de problemas que consideramos en esta contribución, TDG en contextos heterogéneos, la principal dificultad para aplicar este proceso es cómo agregar información expresada en distintos marcos de expresión. Para ello, deberemos unificar en un marco común de expresión dicha información para posteriormente combinarla.

Nuestro objetivo en este capítulo es definir un proceso de decisión para problemas de TDG definidos en contextos heterogéneos. Nuestra propuesta sigue el siguiente esquema (ver figura I.1):

1. Fase de agregación:

- a) *Unificación de información:* elegiremos un marco común de expresión. En nuestra propuesta seleccionaremos un conjunto de etiquetas que denominaremos conjunto básico de términos lingüísticos (CBTL). Y transformaremos la información heterogénea de entrada a conjuntos difusos en este CBTL, mediante distintas funciones de transformación.
- b) *Proceso de agregación:* Una vez unificada la información mediante conjuntos difusos obtendremos para cada alternativa un valor colectivo, por medio de un operador de agregación.

- c) *Transformación a 2-tupla*: El valor colectivo obtenido para cada una de las alternativas expresado por medio de un conjunto difuso valorado en el CBTL se transformará en una 2-tupla lingüística valorada en el CBTL para poder facilitar el proceso de explotación.

2. *Fase de Explotación*: Buscamos la alternativa(s) solución al problema a partir de los valores anteriores.

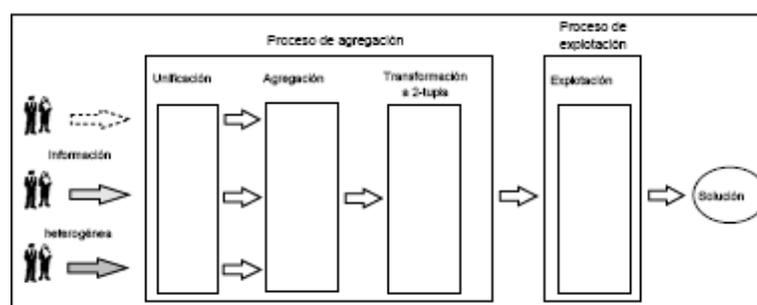


Figura 1.1: Proceso de decisión

Este capítulo está estructurado de la siguiente forma: En la Sección 2 haremos una breve introducción al enfoque lingüístico difuso y revisaremos el modelo lingüístico basado en 2-tuplas. En la Sección 3 presentaremos en detalle el proceso de TDG con información heterogénea que proponemos en este capítulo junto con las herramientas necesarias para llevarlo a cabo. En la Sección 4 presentamos un ejemplo simple de aplicación del modelo propuesto. Finalmente apuntaremos algunas conclusiones sobre el trabajo desarrollado.

I.2 Enfoque Lingüístico Difuso

Como ya hemos comentado el enfoque lingüístico difuso representa información lingüística utilizando valores lingüísticos mediante variables lingüísticas [26].

Cuando trabajamos con información lingüística hay que seleccionar los descriptores lingüísticos adecuados y su semántica. Para llevar a cabo esta tarea, un aspecto importante a considerar es la "granularidad de la incertidumbre", es decir, la capacidad de discriminación entre distintos valores de información. Valores típicos de cardinalidad en los modelos lingüísticos son valores impares, tales como 7 ó 9, donde el término medio representa una valoración de aproximadamente $0,5\tau$ el resto de términos están distribuidos simétricamente a su alrededor. Una posibilidad es generar directamente el conjunto de términos considerando todos los términos distribuidos en una escala sobre la cuál se define un orden total. Un ejemplo de un conjunto de 7 términos es:

$$S = \{s_0 : N, s_1 : V L, s_2 : L, s_3 : M, s_4 : H, s_5 : V H, s_6 : P\}$$

La semántica de los términos lingüísticos viene dada por números difusos definidos en el intervalo $[0,1]$. Estos son valoraciones aproximadas, por lo que diferentes autores consideran que una función de pertenencia trapezoidal es lo suficientemente buena para capturar la vaguedad de los términos lingüísticos. Una representación trapezoidal se consigue con una 4-tupla (a, b, d, c) , donde b y d indican el intervalo donde la función de pertenencia vale 1, con a y c indicando los límites izquierdo y derecho de la función [1]. Un caso particular de este tipo de representación son las valoraciones lingüísticas cuya función sea triangular, es decir, $b = d$, entonces esta función de pertenencia triangular se representa como (a, b, c) . Un ejemplo de semántica para el conjunto de etiquetas anterior podría ser el mostrado en la Figura A.2:

Otros autores usan representaciones no trapezoidales, por ejemplo, funciones gaussianas [2].

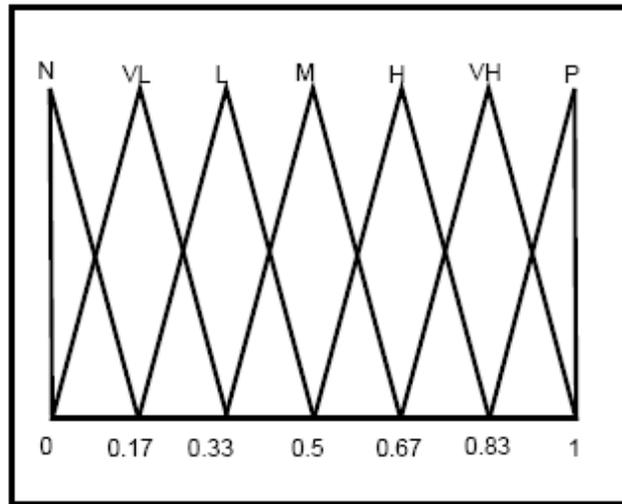


Figura 1.2: Semántica de un conjunto de 7 etiquetas

1.2.1 Modelado Lingüístico basado en 2-tuplas

El modelo de representación lingüístico basado en 2-tuplas fue presentado en [9] para mejorar los problemas de pérdida de información en los procesos de computación con palabras de otros modelos: (i) Modelo basado en el Principio de Extensión [4], (ii) Modelo simbólico [5]. Además se ha demostrado como un modelo útil en el tratamiento de contextos de información no homogéneos [12, 13, 14].

Este modelo se basa en el concepto de traslación simbólica. Sea $S = \{s_0, \dots, s_g\}$ un conjunto de términos lingüísticos, y $\beta \in [0, g]$ un valor en el intervalo de granularidad de S .

Definición 1. La *Traslación Simbólica* de un término lingüísticos s_i es un número valorado en el intervalo $[-5, 5]$ que expresa la “diferencia de información” entre una cantidad de

información expresada por el valor $\beta \in [0, g]$ obtenido en una operación simbólica y el valor entero más próximo, $i \in \{0, \dots, g\}$, que indica el índice de la etiqueta lingüística (s_i) más cercana en S .

A partir de este concepto desarrollaremos un nuevo modelo de representación para la información lingüística, el cuál usa como base de representación un par de valores o 2-tupla, (r_i, α_i) , donde $r_i \in S$ y $\alpha_i \in [-5, 5)$. Este modelo de representación define un conjunto de funciones que facilitan las operaciones sobre 2-tuplas.

Definición 2. Sea $S = \{s_0, \dots, s_g\}$ un conjunto de términos lingüísticos y $\beta \in [0, g]$ un valor que representa el resultado de una operación simbólica, entonces la 2-tupla lingüística que expresa la información equivalente a β se obtiene usando la siguiente función:

$$\Delta : [0, g] \longrightarrow S \times [-5, 5)$$
$$\Delta(\beta) = (s_i, \alpha), \text{ con } \begin{cases} s_i, & i = \text{round}(\beta) \\ \alpha = \beta - i, & \alpha \in [-5, 5), \end{cases}$$

donde round es el operador usual de redondeo, s_i es la etiqueta con índice más cercano a β y α es el valor de la traslación simbólica.

Proposición 1. Sea $S = \{s_0, \dots, s_g\}$ un conjunto de términos lingüísticos y (s_i, α) una 2-tupla lingüística. Existe la función Δ^{-1} , tal que, dada una 2-tupla (s_i, α) esta función devuelve su valor numérico equivalente $\beta \in [0, g]$.

Demostración.

Es trivial si consideramos la siguiente función.

$$\Delta^{-1} : S \times [-5, 5) \longrightarrow [0, g]$$
$$\Delta^{-1}(s_i, \alpha) = i + \alpha = \beta.$$

Comentario: *A partir de las definiciones 1 y 2 y de la proposición 1, la conversión de un término lingüístico en una 2-tupla consiste en añadir el valor cero como traslación simbólica:*

$$s_i \in S \longrightarrow (s_i, 0)$$

Este modelo de representación tiene un modelo computacional asociado presentado en [9]:

1. **Agregación de 2-tuplas:** La agregación de 2-tuplas lingüísticas consiste en la obtención de un valor que resuma un conjunto de valores, por lo tanto, el resultado de la agregación del conjunto de 2-tuplas debe ser una 2-tupla lingüística. En [9] podemos encontrar varios operadores de agregación basados en los operadores de agregación clásicos.

2. **Comparación de 2-tuplas:** La información de comparación representada por las 2-tuplas la realizamos de acuerdo a un orden lexicográfico. Sea (s_k, α_1) y (s_l, α_2) dos 2-tuplas que representan dos valoraciones:

- Si $k < l$ entonces (s_k, α_1) es más pequeño que (s_l, α_2)
- Si $k = l$ entonces:
 - a) Si $\alpha_1 = \alpha_2$ entonces (s_k, α_1) y (s_l, α_2) representan el mismo valor.
 - b) Si $\alpha_1 < \alpha_2$ entonces (s_k, α_1) es más pequeño que (s_l, α_2) .
 - c) Si $\alpha_1 > \alpha_2$ entonces (s_k, α_1) es mayor que (s_l, α_2) .

3. Operador de negación de una 2-tupla: El operador de negación de una 2-tupla se define como:

$$Neg(s_i, \alpha) = \Delta(g - \Delta^{-1}(s_i, \alpha))$$

donde $g + 1$ es la cardinalidad de S , $s_i \in S = \{s_0, \dots, s_g\}$.

I.3 Modelo propuesto de decisión con información heterogénea

El objetivo del modelo que hemos propuesto es resolver un problema de Toma de Decisiones en grupo definido en un contexto heterogéneo en donde las alternativas pueden estar valoradas en distintos dominios (numérico, intervalar, y lingüístico). Ahora vamos a recordar las fases de nuestro proceso de decisión (ver figura I.1):

1. *Agregación de los valores de las preferencias individuales:* (ver figura I.3). En la fase de agregación queremos obtener el valor colectivo de cada una de las alternativas valoradas por los distintos expertos debemos de realizar los siguientes pasos:

- a) Unificación de información.
- b) Proceso de agregación.
- c) Transformación a 2-tupla.

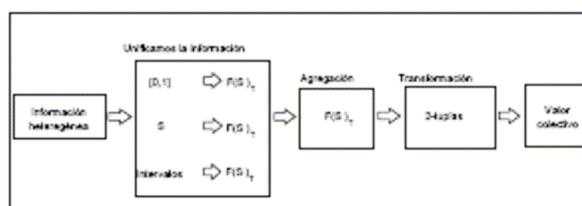


Figura I.3 Proceso de agregación para información heterogénea.

2. *Fase de explotación:* En la fase de explotación queremos ordenar el conjunto de preferencias en función de su valor colectivo y a partir de esta ordenación proporcionar la solución al problema.

A continuación vamos a explicar de forma detallada cada una de las fases.

I.3.1 Fase de agregación

Esta fase tiene como objeto obtener un valor colectivo para cada una de las alternativas valoradas por los distintos expertos en los diferentes dominios (numérico, intervalar y lingüístico). Para ello se efectuarán los pasos de: (i) Unificación de la información (ii) Agregación de los valores de las preferencias individuales (iii) Transformación a 2-tupla, y que explicaremos a continuación.

Unificación de la información

Antes de agregar la información heterogénea, necesitamos expresarla en un marco común para poder operar sobre ella. Este proceso, consiste en unificar la información de entrada (heterogénea) en un único dominio de expresión. Entre los distintos dominios que podemos seleccionar en el contexto que proponemos, hemos decidido unificar sobre el dominio lingüístico, seleccionando un conjunto de etiquetas que denominaremos Conjunto Básico de Términos Lingüísticos (CBTL) y simbolizado por S_T . Cada valor de preferencia numérico, intervalar y lingüístico suministrado por un experto debe ser expresado en el CBTL, para realizar ese proceso se propone convertir cada valor de entrada en un conjunto difuso sobre el CBTL, $F(S_T)$. Para llevar a cabo este proceso de unificación se realizarán los siguientes pasos:

- Seleccionar el CBTL
- Transformar los valores numéricos del intervalo $[0, 1]$ a $F(S_T)$.
- Transformar los intervalos a $F(S_T)$.
- Transformar los términos lingüísticos a $F(S_T)$.

A continuación mostraremos en detalle cómo realizar estos pasos para unificar la información.

Selección del CBTL

Aquí presentamos las reglas que seguiremos para seleccionar el CBTL que utilizaremos como dominio de expresión unificado. En este dominio representaremos toda información de entrada suministrada por los expertos transformándola en información homogénea mediante conjunto difusos en el CBTL.

La información inicial puede estar expresada en cualquier de los siguientes dominios:

1. Numérico.
2. Intervalar.
3. Lingüístico.

Como hemos indicado el CBTL será un dominio lingüístico, para escogerlo, estudiaremos el conjunto de términos lingüísticos S que pertenece al contexto de definición de un problema de TDG. Y si:

1. S es una partición difusa [22], y
2. Las funciones de pertenencia de estos términos son triangulares, $s_i = (a, b, c)$.

Entonces, seleccionaremos S como el CBTL, debido al hecho de que, estas condiciones son necesarias y suficientes para que la transformación entre valores en $[0, 1]$ y 2-tuplas, pudiendo realizarse sin pérdida de información[10].

Si no se cumplen esas condiciones o si el contexto de definición del problema no contienen ningún conjunto de etiquetas lingüísticas, seleccionaremos como CBTL un conjunto de términos con un número de términos más grande de lo que una persona es capaz de discriminar (normalmente 11 o 13, ver [19]) y que satisfaga las condiciones

anteriores. Podemos escoger un CBTL con 15 términos simétricamente distribuidos , con lo que mantendremos la máxima cantidad de información ([7, 10]), cuya semántica es (gráficamente, figura I.4):

s_0	(0, 0, 0,07)	s_1	(0, 0,07, 0,14)
s_2	(0,07, 0,14, 0,21)	s_3	(0,14, 0,21, 0,28)
s_4	(0,21, 0,28, 0,35)	s_5	(0,28, 0,35, 0,42)
s_6	(0,35, 0,42, 0,5)	s_7	(0,42, 0,5, 0,58)
s_8	(0,5, 0,58, 0,65)	s_9	(0,58, 0,65, 0,72)
s_{10}	(0,65, 0,72, 0,79)	s_{11}	(0,72, 0,79, 0,86)
s_{12}	(0,79, 0,86, 0,93)	s_{13}	(0,86, 0,93, 1)
s_{14}	(0,93, 1, 1)		

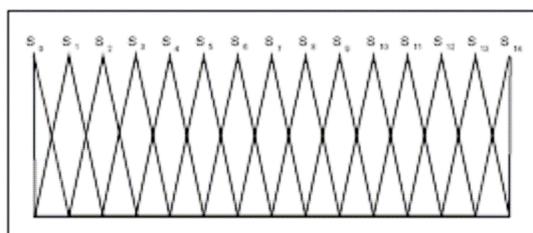


Figura I.4: Un CBTL con 15 términos simétricamente distribuidos.

Una vez que hemos seleccionado el CBTL, como marco común de expresión, hay que unificar la información de entrada en conjuntos difusos sobre el CBTL, para ello definiremos las siguientes funciones de transformación.

Transformar valores numéricos del intervalo [0, 1] en $F(S_T)$

Sea $F(S_T)$ el conjunto de conjuntos difusos en $S_T = \{s_0, \dots, s_g\}$, transformaremos un valor numérico $v \in [0, 1]$ en un conjunto difuso en (S_T) calculando el valor de pertenencia de v en las funciones de pertenencia asociadas con los términos lingüísticos de S_T .

Definición 3. La función τ transforma un valor numérico en un conjunto difuso en S_T :

$$\tau : [0, 1] \rightarrow F(S_T)$$

$$\tau(\vartheta) = \{(s_0, \gamma_0), \dots, (s_g, \gamma_g)\}, s_i \in S_T \text{ y } \gamma_i \in [0, 1]$$

$$\gamma_i = \mu_{s_i}(\vartheta) = \begin{cases} 0, & \text{si } \vartheta \notin \text{Soporte}(\mu_{s_i}(x)) \\ \frac{\vartheta - a_i}{b_i - a_i}, & \text{si } a_i \leq \vartheta \leq b_i \\ 1, & \text{si } b_i \leq \vartheta \leq d_i \\ \frac{c_i - \vartheta}{c_i - d_i}, & \text{si } d_i \leq \vartheta \leq c_i \end{cases}$$

Comentario: Consideramos que las funciones de pertenencia, $\mu_{s_i}(\cdot)$, para etiquetas lingüísticas, definida por una función paramétrica (a_i, b_i, d_i, c_i) . Un caso particular son las valoraciones lingüísticas cuyas funciones de pertenencia son triangulares ($b_i = d_i$).

Ejemplo.

Sea $v = 0,78$ un valor numérico que queremos que sea transformado en un conjunto difuso en $S = \{s_0, \dots, s_4\}$. La semántica de este conjunto de términos es (ver figura 1.5):

$$s_0 = (0, 0, 0,25) \quad s_1 = (0, 0,25, 0,5) \quad s_2 = (0,25, 0,5, 0,75) \quad s_3 = (0,5, 0,75, 1) \quad s_4 = (0,75, 1, 1)$$

Por lo tanto, el conjunto difuso obtenido de transformar 0.78 sobre $F(S_T)$ será: (figura 1.5)

$$\tau(0,78) = \{(s_0, 0), (s_1, 0), (s_3, 0,88), (s_4, 0,12)\}$$

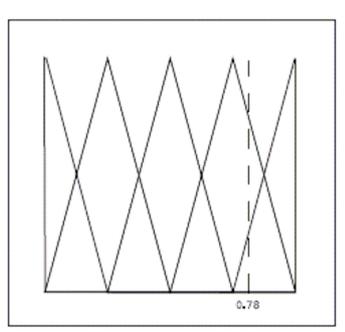


Figura 1.5: Transformación de un valor numérico en un conjunto difuso en S

Transformar intervalos en $F(S_T)$

Nuestro objetivo es el mismo que en el apartado anterior, transformar un intervalo en un $F(S_T)$. En este caso, dado un intervalo $I = [i, \bar{i}]$ valorado en $[0, 1]$, queremos realizar una transformación en la que asumiremos que el intervalo tiene una representación inspirada en una función de pertenencia de conjuntos difusos como la siguiente ([16]):

$$\mu_I(\vartheta) = \begin{cases} 0, & \text{si } \vartheta < i \\ 1, & \text{si } i \leq \vartheta \leq \bar{i} \\ 0, & \text{si } \bar{i} < \vartheta \end{cases}$$

Donde ϑ es un valor en $[0, 1]$. En la figura A.6 podemos observar la representación gráfica de un intervalo. Para realizar la transformación utilizaremos la siguiente función:

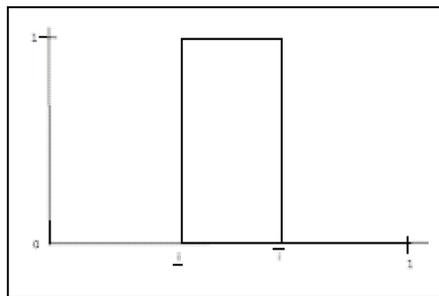


Figura 1.6: Función de pertenencia de $I = [i, \bar{i}]$

Definición 4. Sea $S_T = \{s_0, \dots, s_g\}$ un CBTL. Entonces, la función τ_{IS_T} transforma un intervalo I en $[0, 1]$ en un conjunto difuso en S_T .

$$\begin{aligned} \tau_{IS_T} : I &\rightarrow F(S_T) \\ \tau_{IS_T}(I) &= \left\{ (c_k, \gamma_k^i) / k \in \{0, \dots, g\} \right\} \\ \gamma_k^i &= \max_y \min \{ \mu_I(y), \mu_{c_k}(y) \} \end{aligned}$$

donde $F(S_T)$ es el conjunto de conjuntos difusos definidos en S_T , y $\mu_I(\cdot)$ y $\mu_{c_k}(\cdot)$ son las funciones de pertenencia asociadas con el intervalo I y los términos c_k , respectivamente.

Ejemplo.

Sea $I = [0,6, 0,78]$ un intervalo que debe ser transformado en un conjunto difuso en S_T con cinco términos simétricamente distribuidos. El conjunto difuso obtenido después de aplicar τ_{IST} es (figura 1.7):

$$\tau_{IST}([0,6, 0,78]) = \{(s_0, 0), (s_1, 0), (s_2, 0,6), (s_3, 1), (s_4, 0,2)\}$$

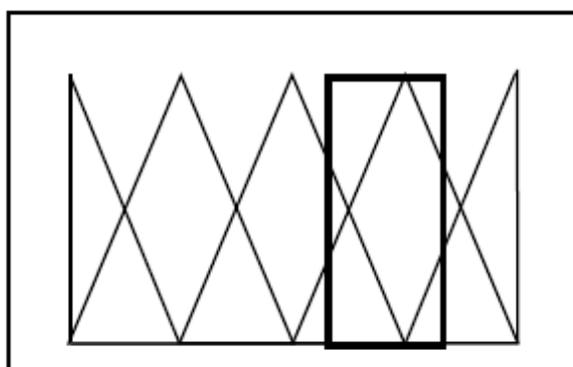


Figura 1.7: Ejemplo de transformación de un intervalo

Cuadro A.1: Relación de preferencia unificada

$$P_{e_k} = \begin{pmatrix} p_{11}^k = \{(s_0, \gamma_0)_k^{11}, \dots, (s_g, \gamma_g)_k^{11}\} & \dots & p_{1n}^k = \{(s_0, \gamma_0)_k^{1n}, \dots, (s_g, \gamma_g)_k^{1n}\} \\ \vdots & \dots & \vdots \\ p_{n1}^k = \{(s_0, \gamma_0)_k^{n1}, \dots, (s_g, \gamma_g)_k^{n1}\} & \dots & p_{nn}^k = \{(s_0, \gamma_0)_k^{nn}, \dots, (s_g, \gamma_g)_k^{nn}\} \end{pmatrix}$$

Transformar términos lingüísticos en $F(S_T)$

Cuando trabajamos con información lingüística en un contexto heterogéneo ocurre que aunque el dominio común de expresión para la información es el lingüístico, la representación unificada es mediante conjuntos difusos, por lo que tenemos que transformar las etiquetas lingüísticas de los conjuntos de etiquetas usados por los expertos en conjuntos difusos en el CBTL.

Para transformar un conjunto de valoraciones lingüísticas definidas en S en $F(S_T)$ debemos realizar los siguientes pasos:

1. En el caso de que el conjunto de términos que ahora estemos transformando, S , hubiese sido escogido como CBTL, entonces el conjunto difuso que representará un término lingüístico será en todos cero excepto en el valor correspondiente al ordinal de la etiqueta lingüística que será uno.
2. En caso contrario, S y S_T son distintos, y debemos transformar estas valoraciones en el CBTL utilizaremos la siguiente función de transformación:

Definición 5. Sea $S = \{l_0, \dots, l_p\}$ y $S_T = \{s_0, \dots, s_g\}$ dos conjuntos de términos lingüísticos, tales que, $g \geq p$. Entonces, una función de transformación lingüísticas, τ_{SS_T} , se define como:

$$\tau_{SS_T} : S \rightarrow F(S_T)$$

$$\tau_{SS_T}(l_i) = \left\{ (c_k, \gamma_k^i) / k \in \{0, \dots, g\} \right\}, \forall l_i \in S$$

$$\gamma_k^i = \max_y \min \{ \mu_{l_i}(y), \mu_{c_k}(y) \}$$

donde $F(S_T)$ es el conjunto de conjuntos difusos definidos en S_T , y $\mu_{l_i}(\cdot)$ y $\mu_{c_k}(\cdot)$ son funciones de pertenencia de los conjuntos difusos asociados con los términos l_i y c_k respectivamente.

Por lo tanto, el resultado de τ_{SS_T} para cualquier valor lingüístico de S es un conjunto difuso definido en el CBTL, $F(S_T)$.

Ejemplo.

Sea $S = \{l_0, l_1, \dots, l_4\}$ y $S_T = \{s_0, s_1, \dots, s_6\}$ dos conjuntos de términos, con 5 y 7 etiquetas, respectivamente, y con la siguiente semántica asociada:

$$\begin{array}{ll}
 l_0 = (0, 0, 0, 25) & s_0 = (0, 0, 0, 16) \\
 l_1 = (0, 0, 25, 0, 5) & s_1 = (0, 0, 16, 0, 34) \\
 l_2 = (0, 25, 0, 5, 0, 75) & s_2 = (0, 16, 0, 34, 0, 5) \\
 l_3 = (0, 5, 0, 75, 1) & s_3 = (0, 34, 0, 5, 0, 66) \\
 l_4 = (0, 75, 1, 1) & s_4 = (0, 5, 0, 66, 0, 84) \\
 & s_5 = (0, 66, 0, 84, 1) \\
 & s_6 = (0, 84, 1, 1)
 \end{array}$$

El conjunto difuso obtenido después de aplicar τ_{SST} para l_1 es (ver figura 1.8):

$$\tau_{SST} = \{(s_0, 0,39), (s_1, 0,85), (s_2, 0,85), (s_3, 0,39), (s_4, 0), (s_5, 0), (s_6, 0)\}$$

Agregación de los valores de las preferencias individuales

Utilizando las funciones de transformación anteriores logramos que la información de entrada este expresada por medio de conjuntos difusos en el CBTL, $S_T = \{s_0, \dots, s_g\}$. El siguiente paso en nuestro proceso es agregar estos valores individuales de preferencia para obtener un valoración colectiva de cada alternativa. Para ello, usaremos una función de agregación para combinar los conjuntos difusos en el CBTL para obtener una valoración colectiva para cada alternativa que sea un conjunto difuso en el CBTL.

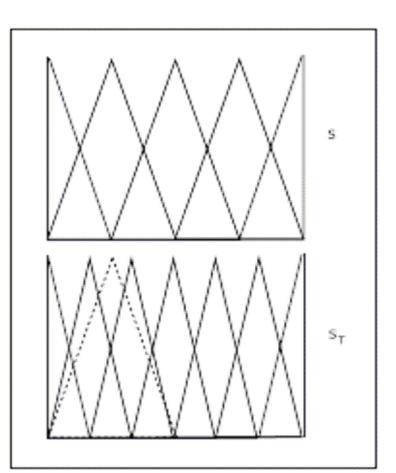


Figura 1.8: Transformación $l_i \in S$ en un conjunto difuso en S_T

En los problemas de TDG la información heterogénea unificada se expresa con relaciones de preferencia cuyos $\in F(S_T)$ (ver Cuadro A.1), donde $p_{ij}^k = (s_\nu, Y_\nu)_{ij}^k$ con $\nu = 0, \dots, g$ es el grado de preferencia de la alternativa x_i sobre x_j proporcionada por el experto e_k y Y_ν representa el grado de pertenencia sobre la etiqueta s_ν del CBTL.

Representaremos cada conjunto difuso, p_{ij}^k , como $r_{ij}^k = (Y_0, \dots, Y_g)_{ij}^k$ siendo los valores de r_{ij}^k sus respectivos grados de pertenencia. Entonces, la valoración colectiva de la relación de preferencia de acuerdo a todas las relaciones de preferencia proporcionados por los expertos $\{r_{ij}^k, \forall e_k\}$ se obtiene agregando estos conjuntos difusos. Esta valoración colectiva global, denotada por r_{ij} , es un nueva relación de preferencia colectiva, R , de conjuntos difusos definida en S_T

$$R = \begin{pmatrix} r_{11} & \dots & r_{1j} & \dots & r_{1n} \\ \vdots & & \vdots & & \vdots \\ r_{i1} & \dots & r_{ij} & \dots & r_{in} \\ \vdots & & \vdots & & \vdots \\ r_{n1} & \dots & r_{nj} & \dots & r_{nn} \end{pmatrix}$$

con $r_{ij} = (Y_0, \dots, Y_\nu, \dots, Y_g)_{ij}$ y donde Y_ν de r_{ij} se calcula con la siguiente función de pertenencia:

$$Y_\nu = f(Y_{\nu 1}, \dots, Y_{\nu k}, \dots, Y_{\nu m})$$

donde f es un operador de agregación, m es el número de expertos, y $Y_{\nu k}$ representa el grado de pertenencia sobre la etiqueta s_ν cuando se está valorando la preferencia de x_i sobre x_j por parte del experto e_k .

Transformando conjuntos difusos en el CBTL a 2-tupla lingüísticas.

Ahora transformaremos los conjuntos difusos que expresan valores colectivos en 2-tuplas lingüísticas sobre el CBTL con el objetivo de facilitar el proceso de ordenación de la fase de explotación del proceso de decisión.

En esta sección presentamos las funciones de transformación que nos permiten transformar los conjuntos difusos del CBTL en 2-tuplas lingüísticas sobre el CBTL. En [11] se presento una función χ que transformaba un conjunto difuso en un valor numérico y mediante Δ transformabamos este valor numérico en una 2-tupla. En este capítulo hemos redefinido χ de forma que ahora transformamos directamente un conjunto difuso definido en $F(S_T)$ en una 2-tupla:

$$\begin{aligned}\chi : F(S_T) &\rightarrow S \times [-0,5, 0,5) \\ \chi(F(S_T)) = \chi(\{(s_j, \gamma_j), j = 0, \dots, g\}) &= \Delta \left(\frac{\sum_{j=0}^g j \gamma_j}{\sum_{j=0}^g \gamma_j} \right) = \\ &= \Delta(\beta) = (s, \alpha)\end{aligned}$$

Ejemplo.

Partimos del conjunto difuso con grados de pertenencia, $(0, 0, 0, ,41, 1, ,19, 0)$, sobre un CBTL similar al mostrado en la figura A.2 y queremos expresarlo mediante una 2-tupla, para ello aplicaremos las función χ :

$$\begin{aligned}\chi((0, 0, 0, ,41, 1, ,19, 0)) &= \Delta \left(\frac{\sum_{j=0}^6 j \gamma_j}{\sum_{j=0}^6 \gamma_j} \right) = \\ &= \Delta(4,33) = (H, ,33)\end{aligned}$$

Por lo tanto, nuestra 2-tupla solución es: $(H, ,33)$.

I.3.2. Fase de explotación

Llegados a esta fase en nuestro modelo de decisión, tenemos para cada alternativa una valoración global expresado por medio de 2-tuplas lingüísticas en el CBTL. Para obtener la mejor alternativa o el mejor conjunto de alternativas podemos utilizar alguna de las funciones de selección presentadas en [20, 21]. Sabemos que el modelo de representación de las 2-tuplas lingüísticas tiene definido un orden total sobre si mismo. Si obtenemos el valor global de cada alternativa podremos ordenar los valores globales [8] de las alternativas utilizando este orden y obtener la solución de nuestro problema.

A continuación veremos un ejemplo de nuestro modelo aplicado a un caso práctico.

I.4 Un problema TDG con información no homogénea

Vamos a suponer que una compañía quiere renovar sus coches. Existen cuatro modelos de coches disponibles, {CAR1,CAR2,CAR3,CAR4} y tres expertos nos proporcionan sus relaciones de preferencia sobre los cuatro coches. El primer experto expresa sus relaciones de preferencia utilizando valores numéricos en $[0, 1]$, P_1^n . El segundo expresa sus preferencias por medio de valores lingüísticos en un conjunto de términos S (ver figura A.2), P_2^S . Y el tercer experto los expresa utilizando valores de preferencia intervalares en $[0, 1]$, P_3^I . Los tres expertos quieren obtener una solución colectiva.

$$P_1^n = \begin{pmatrix} - & ,5 & ,8 & ,4 \\ ,3 & - & ,9 & ,3 \\ ,3 & ,2 & - & ,4 \\ ,9 & ,8 & ,5 & - \end{pmatrix} \quad P_2^S = \begin{pmatrix} - & H & VH & M \\ L & - & H & VH \\ VL & N & - & VH \\ L & VL & N & - \end{pmatrix}$$

$$P_3^I = \begin{pmatrix} - & [,7, ,8] & [,65, ,7] & [,8, ,9] \\ [,3, ,35] & - & [,6, ,7] & [,8, ,85] \\ [,3, ,35] & [,3, ,4] & - & [,7, ,9] \\ [,1, ,2] & [,2, ,4] & [,1, ,3] & - \end{pmatrix}$$

El CBTL elegido en nuestro ejemplo es S (ver figura A.2) dado que cumple con las condiciones necesarias.

Utilizando las funciones de transformación logramos que la información de entrada este expresada mediante conjuntos difusos en el dominio de expresión unificado del CBTL $S_T = \{s_0, \dots, s_9\}$.

El proceso que realizaremos seguirá el siguiente orden:

1. Transformamos los valores numéricos del intervalo $[0, 1]$ al $F(S_T)$.

Cuadro A.2: Información de preferencia expresada en $F(S_T)$.

$$\begin{aligned}
 P_1^T &= \begin{pmatrix} - & (0, 0, 0, 1, 0, 0, 0) & (0, 0, 0, 0, 19, 81, 0) & (0, 0, 59, 41, 0, 0, 0) \\ (0, 19, 81, 0, 0, 0, 0) & - & (0, 0, 0, 0, 59, 41) & (0, 19, 81, 0, 0, 0, 0) \\ (0, 19, 81, 0, 0, 0, 0) & (0, 81, 19, 0, 0, 0, 0) & - & (0, 0, 59, 41, 0, 0, 0) \\ (0, 0, 0, 0, 0, 59, 41) & (0, 0, 0, 0, 19, 81, 0) & (0, 0, 0, 1, 0, 0, 0) & - \end{pmatrix} \\
 P_2^S &= \begin{pmatrix} - & (0, 0, 0, 0, 1, 0, 0) & (0, 0, 0, 0, 0, 1, 0) & (0, 0, 0, 1, 0, 0, 0) \\ (0, 0, 1, 0, 0, 0, 0) & - & (0, 0, 0, 0, 1, 0, 0) & (0, 0, 0, 0, 1, 0) \\ (0, 1, 0, 0, 0, 0, 0) & (1, 0, 0, 0, 0, 0, 0) & - & (0, 0, 0, 0, 0, 1, 0) \\ (0, 0, 1, 0, 0, 0, 0) & (0, 1, 0, 0, 0, 0, 0) & (1, 0, 0, 0, 0, 0, 0) & - \end{pmatrix} \\
 P_3^I &= \begin{pmatrix} - & (0, 0, 0, 0, 81, 81, 0) & (0, 0, 0, 12, 1, 19, 0) & (0, 0, 0, 0, 19, 1, 41) \\ (0, 19, 1, 12, 0, 0, 0) & - & (0, 0, 0, 41, 1, 19, 0) & (0, 0, 0, 0, 19, 1, 12) \\ (0, 19, 1, 12, 0, 0, 0) & (0, 19, 1, 41, 0, 0, 0) & - & (0, 0, 0, 0, 81, 1, 41) \\ (41, 1, 19, 0, 0, 0, 0) & (0, 81, 1, 41, 0, 0, 0) & (41, 1, 81, 0, 0, 0, 0) & - \end{pmatrix}
 \end{aligned}$$

Cuadro A.3: Información de preferencia colectiva en $F(S_T)$.

$$P = \begin{pmatrix} - & (0, 0, 0, 33, 6, 27, 0) & (0, 0, 0, 04, 4, 19, 0) & (0, 0, 2, 47, 06, 33, 04) \\ (0, 13, 94, 04, 0, 0, 0) & - & (0, 0, 0, 14, 67, 26, 14) & (0, 06, 27, 0, 06, 67, 04) \\ (0, 46, 6, 04, 0, 0, 0) & (33, 33, 4, 14, 0, 0, 0) & - & (0, 0, 2, 14, 27, 67, 14) \\ (14, 33, 4, 0, 0, 20, 14) & (0, 6, 33, 14, 06, 27, 0) & (47, 33, 27, 33, 0, 0, 0) & - \end{pmatrix}$$

2. Transformamos los valores lingüísticos al $F(S_T)$.

3. Transformamos los valores intervalares al $F(S_T)$.

Después de aplicar las funciones de transformación obtendremos las preferencias de los expertos valoradas en el CBTL (ver Cuadro A.2).

Una vez expresada toda la información en el CBTL, el siguiente paso es realizar la agregación de los valores de preferencias individuales. Utilizaremos un operador de agregación para combinarlo. En este ejemplo hemos usado el operador de agregación de la media aritmética obteniendo la siguiente relación de preferencia colectiva(ver Cuadro A.3).

Transformaremos las valoraciones globales representadas mediante conjuntos difusos en el CBTL a 2-tupla mediante la función de transformación vista anteriormente (utilizando la función χ).

Así, por ejemplo:

$$\begin{aligned} p_{01} &= \chi(0, 0, 0, ,33, ,6, ,27, 0) = \Delta(3,96) = (H, -,04) \\ p_{02} &= \chi(0, 0, 0, ,04, ,4, ,19, 0) = \Delta(4,24) = (H, ,24) \\ p_{03} &= \chi(0, 0, ,2, ,47, ,06, ,33, 0) = \Delta(3,58) = (H, -,42) \\ p_{10} &= \chi(0, ,13, ,94, ,04, 0, 0, 0) = \Delta(1,92) = (L, -0,08) \end{aligned}$$

...

El resultado de esta transformación es:

$$P = \begin{pmatrix} - & (H, -,04) & (H, ,24) & (H, -,42) \\ (L, -,08) & - & (H, ,33) & (H, ,03) \\ (L, -,38) & (VL, ,29) & - & (H, ,29) \\ (L, ,45) & (L, ,34) & (VL, ,33) & - \end{pmatrix}$$

Por lo tanto, nosotros, en nuestro ejemplo, para obtener el conjunto de alternativas solución hemos escogido una función de selección que calcule el grado de dominancia de cada alternativa, x_i , sobre el resto de alternativas:

$$\Lambda(x_i) = \frac{1}{n-1} \sum_{j=0|j \neq i}^n \beta_{ij}$$

donde n es el número de alternativas y $\beta_{ij} = \Delta^{-1}(p_{ij})$ siendo p_{ij} una 2- tupla lingüística que representa el valor colectivo de la preferencia de la alternativa x_i sobre x_j de acuerdo al grupo de expertos.

Así:

$$\Lambda(CAR1) = \frac{1}{3} \sum_{j=0|j \neq 0}^3 \beta_{ij} = 3,92$$

$$\Lambda(CAR2) = \frac{1}{3} \sum_{j=0|j \neq 1}^3 \beta_{ij} = 3,43$$

$$\Lambda(CAR3) = \frac{1}{3} \sum_{j=0|j \neq 2}^3 \beta_{ij} = 2,4$$

$$\Lambda(CAR4) = \frac{1}{3} \sum_{j=0|j \neq 3}^3 \beta_{ij} = 2,04$$

La solución al problema será aquella que tenga el grado de dominancia más alto. Expresaremos las soluciones obtenidas con 2-tuplas mediante el operador Δ .

Cuadro A.4: Valores globales de las alternativas

CAR1	CAR2	CAR3	CAR4
(H,-.08)	(M,.43)	(L,.4)	(L.04)

Y devolveremos como solución CAR1.

I.5 Conclusiones

En los problemas de decisión con múltiples expertos, no es una situación extraña que distintos expertos tengan diferente conocimiento sobre las alternativas o que estas sean de distinta naturaleza. Por lo que la necesidad de utilizar distintos dominios de expresión puede aparecer. Sin embargo, la mayoría de los modelos de decisión fuerzan a todos los expertos a expresarse en un único dominio de expresión. En este capítulo hemos propuesto un modelo de decisión que ofrece una mayor flexibilidad a los expertos para expresar sus opiniones en distintos dominios (numérico, intervalar o lingüístico). Para desarrollar este modelo hemos presentado una serie de funciones que nos permiten manejar este tipo de información heterogénea.

Bibliografía Anexo I

- [1] P.P. Bonissone and K.S. Decker. *Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-Off Precision and Complexity*. In L.H. Kanal and J.F. Lemmer, Editors., *Uncertainty in Artificial Intelligence*. North-Holland, 1986.
- [2] G. Bordogna and G. Pasi. A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation. *Journal of the American Society for Information Science*, 44:70-82, 1993.
- [3] P. Chang and Y. Chen. A fuzzy multicriteria decision making method for technology transfer strategy selection in biotechnology. *Fuzzy Sets and Systems*, 63:131-139, 1994.
- [4] R. Degani and G. Bortolan. The problem of linguistic approximation in clinical decision making. *International Journal of Approximate Reasoning*, 2:143-162, 1988.
- [5] M. Delgado, J.L. Verdegay, and M.A. Vila. Linguistic decision making models. *International Journal of Intelligent Systems*, 7:479-492, 1993.
- [6] F. Herrera and E. Herrera-Viedma. Linguistic decision analysis: Steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, 115:67-82, 2000.
- [7] F. Herrera, E. Herrera-Viedma, and L. Martínez. A fusion approach for managing multi-granularity linguistic terms sets in decision making. *Fuzzy Sets and Systems*, 114(1):43-58, 2000.
- [8] F. Herrera, E. Herrera-Viedma, L. Martínez, and P.J. Sánchez. A linguistic decision process for evaluating the installation of an ERP system. In *9th International Conference on Fuzzy Theory and Technology*, Cary (North Carolina) USA, 2003.
- [9] F. Herrera and L. Martínez. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on Fuzzy Systems*, 8(6):746-752, 2000.
- [10] F. Herrera and L. Martínez. An approach for combining linguistic and numerical information based on 2-tuple fuzzy representation model in decision-making. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8(5):539-562, 2000.

- [11] F. Herrera and L. Martínez. An approach for combining linguistic and numerical information based on 2-tuple fuzzy representation model in decision-making. *International Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems*, 8(5):539-562, 2000.
- [12] F. Herrera and L. Martínez. A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 31(2):227-234, 2001.
- [13] F. Herrera, L. Martínez, and P.J. Sánchez. Managing non-homogeneous information in group decision making. *European Journal of Operational Research*, page To appear, 2004.
- [14] F. Herrera, L. Martínez, and P.J. Sánchez. *A Multi-Granular Linguistic Decision Model for Evaluating the Quality of Network Services.*, pages 71-92. In: Intelligent Sensory Evaluation: Methodologies and Applications. Springer, 2004.
- [15] J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18:105-118, 1986.
- [16] D. Kuchta. Fuzzy capital budgeting. *Fuzzy Sets and Systems*, 111:367- 385, 2000.
- [17] C.K. Law. Using fuzzy numbers in educational grading systems. *Fuzzy Sets and Systems*, 83:311-323, 1996.
- [18] H.M. Lee. Applying fuzzy sets theory for evaluating the rate of aggregative risk in software development. *Fuzzy Sets and Systems*, 79:323-336, 1996.
- [19] G.A. Miller. The magical number seven plus or minus two: Some limits on our capacity of processing information. *Psychological Review*, 63:81-97, 1956.
- [20] S.A. Orlovsky. Decision-making with a fuzzy preference relation. *Fuzzy Sets Systems*, 1:155-167, 1978.
- [21] M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90:199-206, 1997.
- [22] E.H. Ruspini. A new approach to clustering. *Inform. Control*, 15:22- 32, 1969.
- [23] J.F. Le Téno and B. Mareschal. An interval version of PROMETHEE for the comparison of building products' design with ill-defined data on environmental quality. *European Journal of Operational Research*, 109:522-529, 1998.

[24] R.R. Yager. An approach to ordinal decision making. *International Journal of Approximate Reasoning*, 12:237-261, 1995.

[25] R.R. Yager, L.S. Goldstein, and E. Mendels. Fuzmar: An approach to aggregating market research data based on fuzzy reasoning. *Fuzzy Sets and Systems*, 68:1-11, 1994.

[26] L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. *Information Sciences, Part I, II, III*, 8,8,9:199- 249,301-357,43-80, 1975.

ANEXO II

Manual de Instalación

Este anexo esta dedicado ha realizar las configuraciones necesarias para poner en marcha nuestra aplicación.

Las únicas consideraciones previas que han de tenerse en cuenta son que durante todo este manual se ha supuesto que la unidad principal de disco duro es C:, que la unidad principal de disco flexible es D:.

Hemos de decir que nuestro sistema trabajará con un servidor basado en la plataforma Microsoft Windows, por lo que todas estas instalaciones y configuraciones hay que realizarlas teniendo en cuenta dicho Sistema Operativo.

Todo el material necesario para instalar y dejar operativo el servidor se encuentra disponible en el CD que acompaña a esta memoria. Vaya a dispositivo de CD-ROM D:\EncuestaEvaluacion y compruebe que se encuentran los siguientes archivos:

- jdk-6-windows-i586.exe
- apache-tomcat-4.1.36-LE-jdk14
- EncuestaEvaluacion.war

Si es así podemos proceder a la instalación de nuestro servidor inmediatamente. Si falta algún archivo o alguno de ellos se encuentra dañado póngase en contacto con el responsable de la aplicación para subsanar el percance.

Si ya dispone de un servidor Apache Tomcat 4.1 o superior sólo tendrá que copiar el archivo EncuestaEvaluacion.war en el directorio webapp, que se encuentra donde tenga instalado el servidor. Una vez hecho esto podrá acceder a la aplicación tecleando en su navegador <http://www.ip-del-servidor/EncuestaEvaluacion>. En caso contrario siga los siguientes pasos.

Paso 1: Instalar Máquina Virtual Java

Si no dispone de Java, deberá instalar el entorno de ejecución Java (JRE) y la máquina virtual Java. Para ello vaya al CD-ROM y ejecute el archivo de instalación `jdk-6-windows-i586.exe` y siga los sencillos pasos de instalación.

Paso 2: Instalar Apache-Tomcat

Al ser una aplicación Web, es lógico necesitar un servidor Web para que proporcione los servicios o responda a las peticiones realizadas por cualquier cliente que se quiera conectar al sistema. Para ello utilizaremos el servidor Web Apache Tomcat.

En esta sección explicaremos los pasos a seguir para realizar la instalación y configuración del Servidor Apache Tomcat.

1. En primer lugar necesitaremos disponer del programa de instalación. Este se encuentra en el CD-ROM entregado. El nombre del archivo de instalación es: `apache-tomcat-4.1.36-LE-jdk14`.
2. La primera pantalla, figura II.1, que aparece nos indica que tenemos instalado un JDK superior a la versión 1.4, por lo que podemos seguir con la instalación.



Figura II.1 JDK instalado

3. Una vez que pulsamos aceptar nos aparece una ventana con los términos de licencia. Pulsamos el botón I Agree para continuar con la instalación.

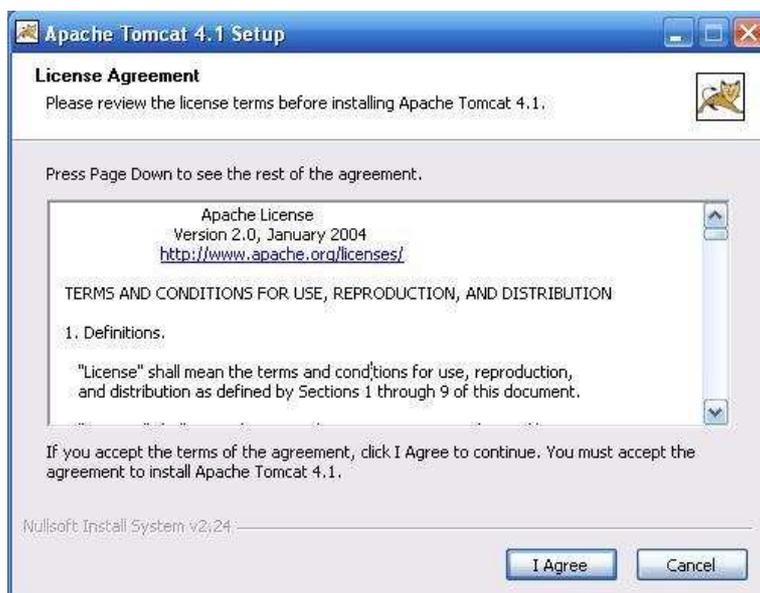


Figura II.2 Términos de la licencia de uso

4. Seleccionamos la instalación Normal y pulsamos Next.

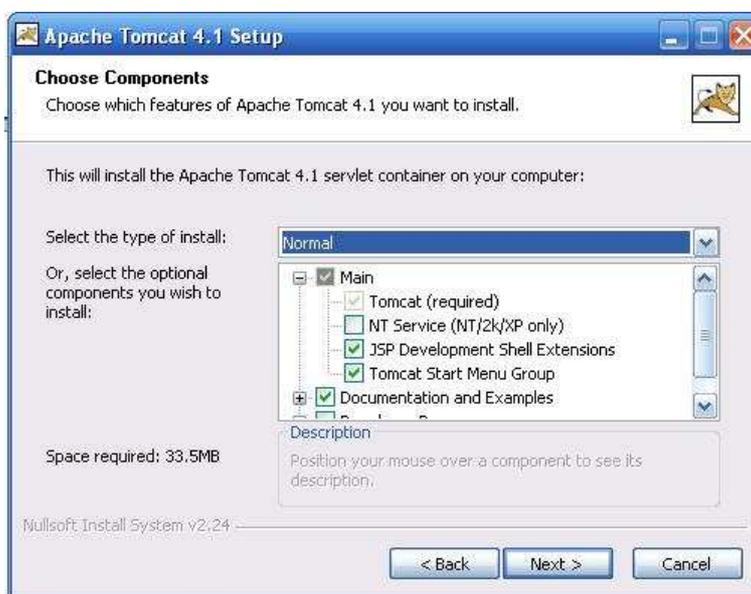


Figura II.3 Selección del tipo de instalación

5. Elegimos el lugar donde se va a realizar la instalación y pulsamos Install.

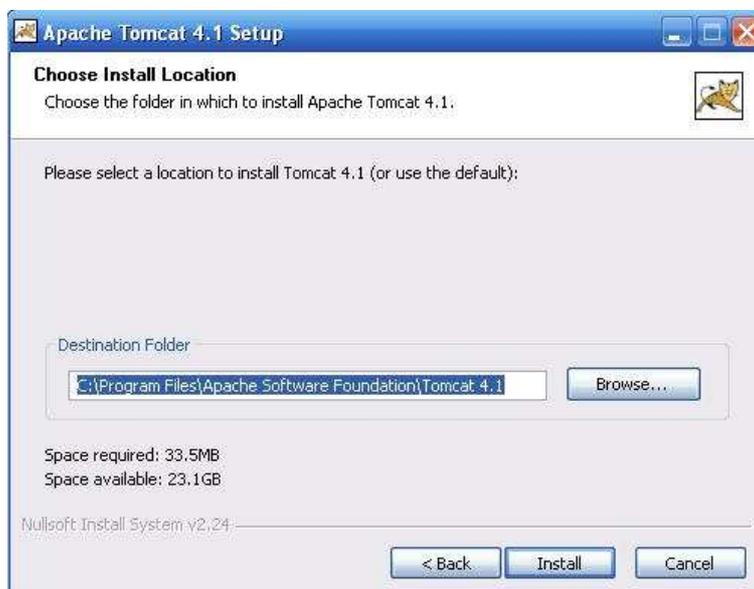


Figura II.4 Selección del directorio de instalación

6. Comienza el proceso de instalación.

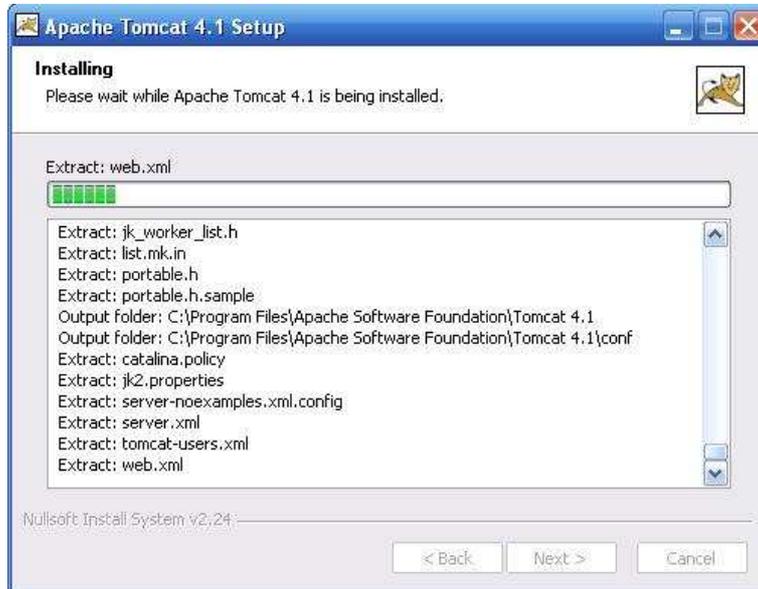


Figura II.5 Durante la instalación

7. Una vez finalizada la instalación pulsamos Next y nos aparece una pantalla como la figura II.6, en la que debemos introducir un usuario y una contraseña para el administrador y el puerto por el cual queremos que se lleve a cabo la comunicación, por defecto el 8080.

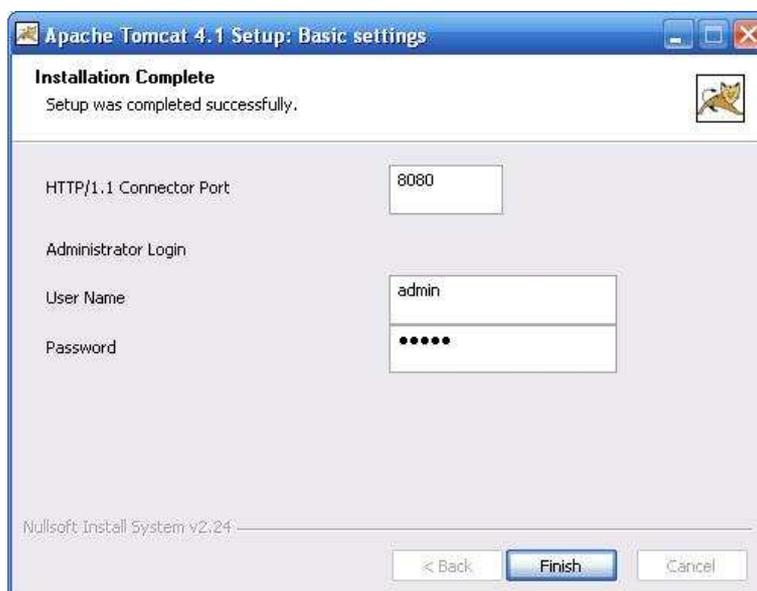


Figura II.6 Selección de usuario y puerto

8. Pulsando finalizar termina la instalación de Apache Tomcat.

Una vez finalizada la instalación debemos poner en marcha el servidor Apache Tomcat para ello, existen dos opciones:

- Mediante variable de entorno:

Para esto, tenemos que establecer la variable de entorno CATALINA_HOME al directorio donde tenga instalado Tomcat 4.1. Ejecutando el siguiente comando:
`%CATALINA_HOME%\bin\startup`

- Ejecutando el siguiente comando:

```
cd %CATALINA_HOME%\bin  
startup
```

Después de iniciar, las aplicaciones por defecto incluidas con Tomcat 4.1 deben ser visualizadas si en: <http://localhost:8080/>

Para detener el servidor dispone de dos opciones:

- Mediante variable de entorno:

Para esto, tenemos que establecer la variable de entorno CATALINA_HOME al directorio donde tenga instalado Tomcat 4.1. Ejecutando el siguiente comando:
%CATALINA_HOME%\bin\shutdown

- Ejecutando el siguiente comando:

```
cd %CATALINA_HOME%\bin  
shutdown
```

Puede encontrar más información sobre como configurar y ejecutar Tomcat 4.1 en la siguiente dirección: <http://tomcat.apache.org/>

Paso 3: Puesta en marcha de la aplicación

Tiene que copiar el archivo EncuestaEvaluacion.war en el directorio webapp, que se encuentra donde ha instalado el servidor. Una vez hecho esto podrá acceder a la aplicación tecleando en su navegador <http://www.ip-del-servidor/EncuestaEvaluacion>.

ANEXO III

Manual de Usuario: Administrador

Este manual de usuario está organizado como una visita guiada por la aplicación. Antes de embarcarse en ella es conveniente que el usuario tenga claros algunos aspectos:

- Ésta es una aplicación web optimizada para su visualización en un navegador Firefox (a ser posible su versión más reciente, la cual se puede descargar en <http://www.mozilla-europe.org/es/>) y con una resolución no inferior a 800x600 pixels. Si se utiliza otro navegador o una resolución inferior a la recomendada se pueden producir fallos de visualización aunque la funcionalidad de la aplicación esta completamente asegurada.
- En este punto de su desarrollo, la aplicación es sólo un prototipo que no permite la eliminación de alumnos, profesores o asignaturas. Cuando la aplicación esté completamente desarrollada, éstas funcionalidades y otras muchas estarán disponibles para los usuarios.

Una vez aclarados los puntos anteriores puede empezar la visita guiada por la aplicación.

El primer paso para utilizar la aplicación es abrir el Navegador (Firefox) y teclear la dirección en la que tenemos alojada esta, que en nuestro caso será: <http://localhost:8083/EncuestaEvaluacion/index.jsp>

Al realizar lo anteriormente comentado, nos encontramos con la página de inicio, tal y como se ve en la figura III.1



Figura III.1: Inicio de la Aplicación

Como somos Administrador, deberemos hacer clic en el botón de Conexión como Administrador, accediendo a la pantalla de identificación, figura III.2

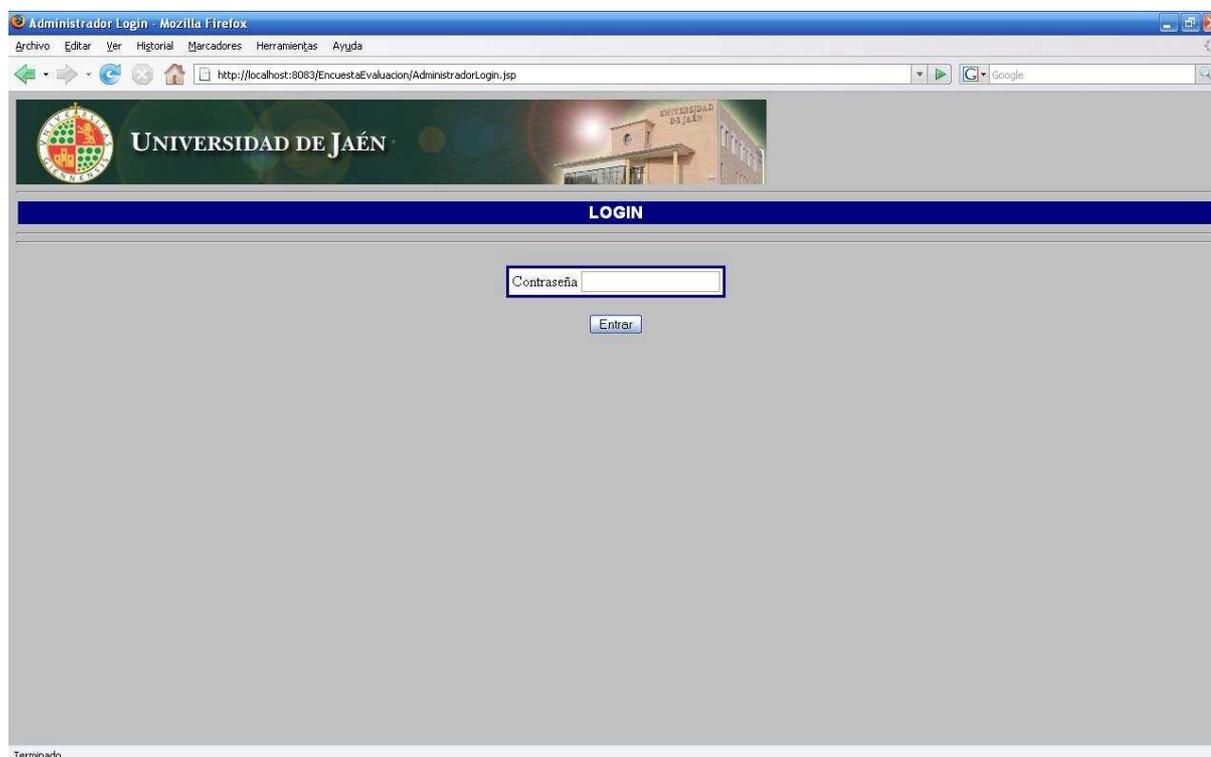


Figura III.2 Identificación de Administrador

Debe introducir la contraseña de Administrador y pulsar Entrar. Si la contraseña introducida no es correcta el sistema nos muestra la pantalla de la figura III.3, indicándonos que la contraseña introducida no es correcta y nos invita a una nueva identificación.

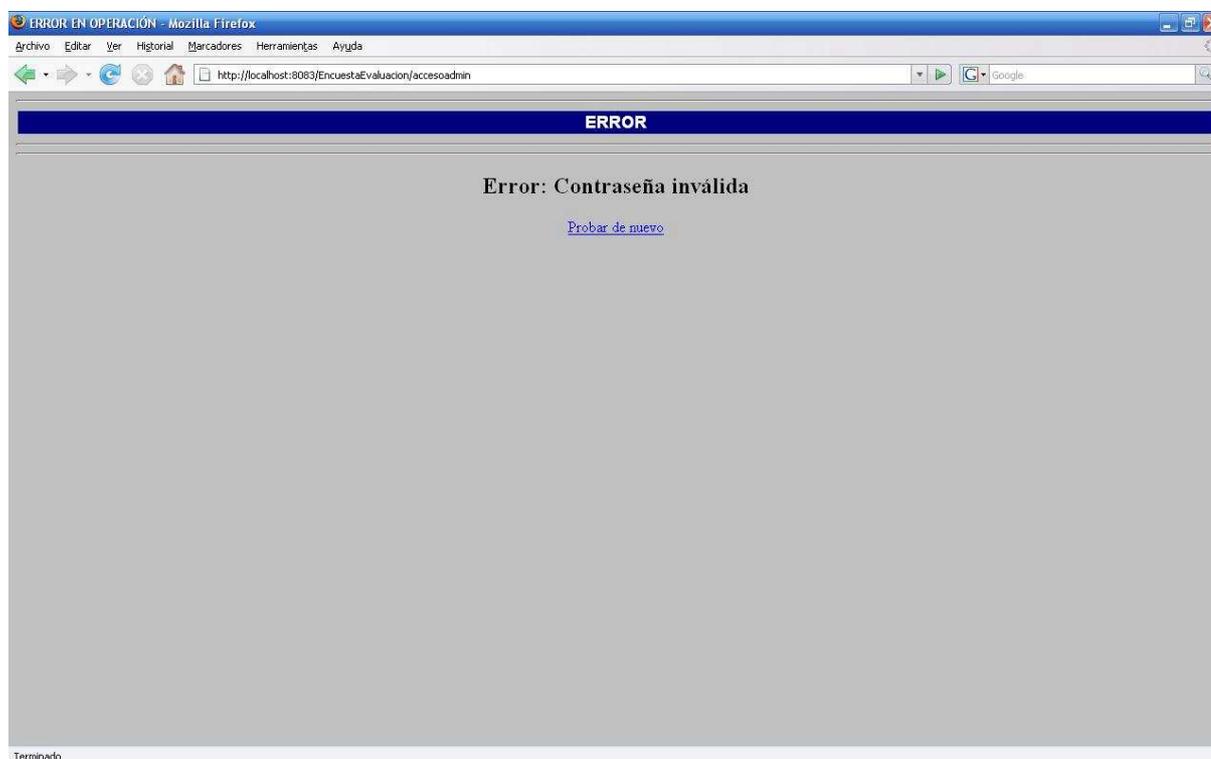


Figura III.3 Identificación de Administrador incorrecta

En el caso de una identificación correcta nos aparece el menú principal del administrador, como podemos ver en la figura III.4.

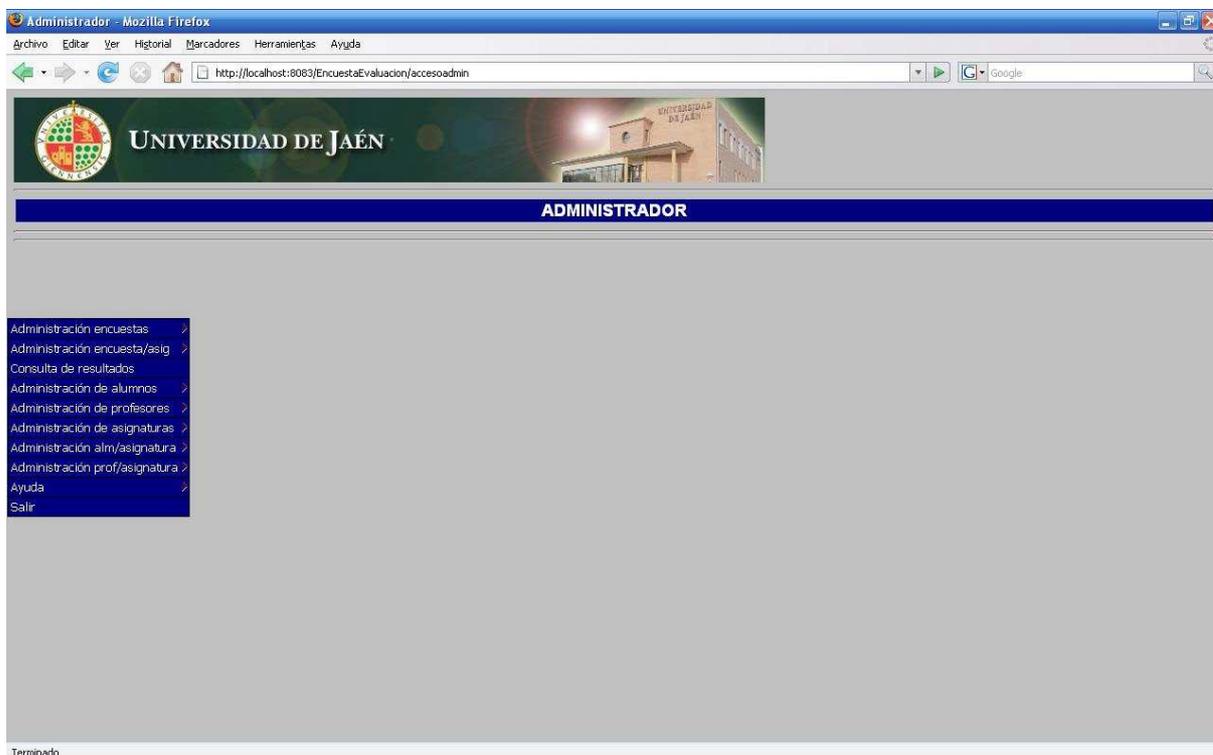


Figura III.4 Menú principal de Administrador

En este menú principal podemos encontrar todas las operaciones disponibles para el Administrador del sistema. Se diferencian tres partes principales:

- Una primera parte relacionada con el **manejo de las encuestas**, que comprende los tres primeros apartados del menú: (Administración Encuestas, Administración Encuesta/Asig y Consulta de Resultados)
- Una segunda parte relacionada con el **manejo de la base de datos**: (Administración de Alumnos, Profesores, Asignaturas, Alum/Asig y Prof/Asig).
- Una tercera y última parte que contiene **información acerca de la aplicación**.

A continuación veremos de forma detallada cada una de estas partes de la aplicación:

1. Manejo de Encuestas

En este apartado nos encontramos las siguientes funcionalidades:

1.1 Administración de encuestas.

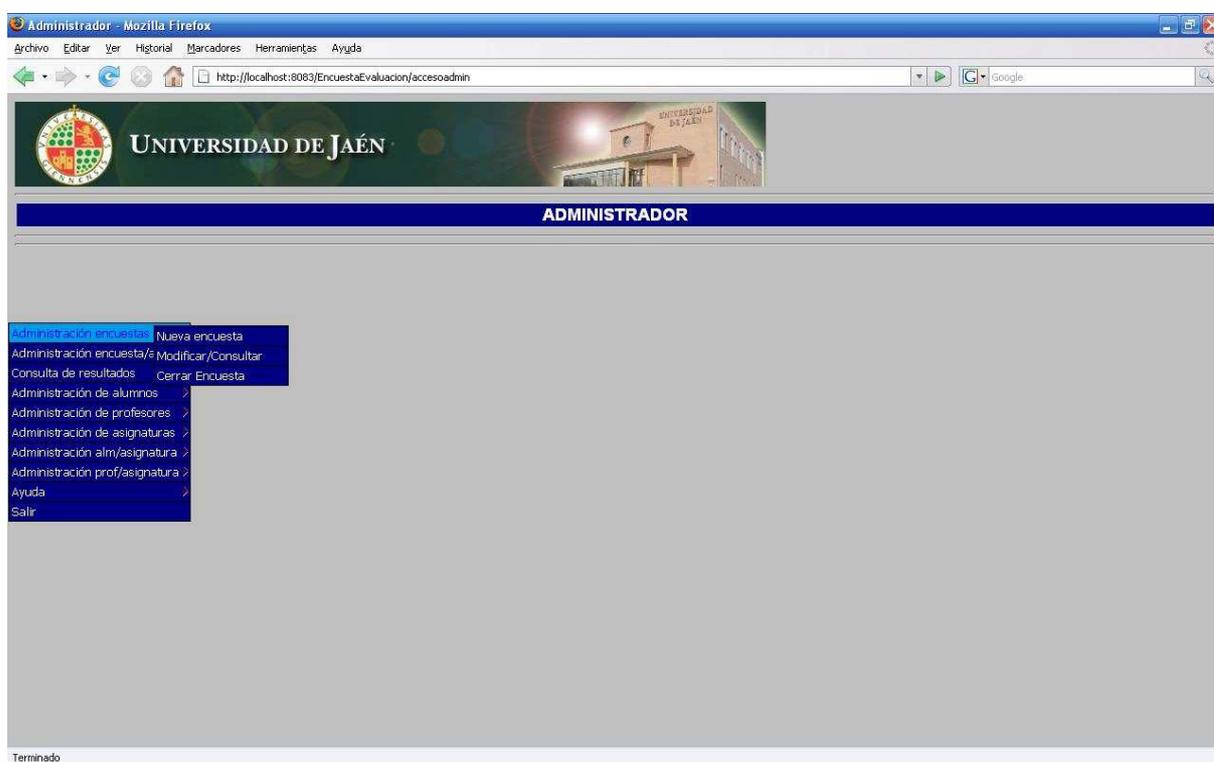


Figura III.5 Menú Administración de Encuestas

1.1.1 Nueva Encuesta

Este apartado nos permite crear una nueva encuesta. El primer paso (figura III.6) para crearla es introducir un nombre de encuesta no existente y pulsar el botón Crear.

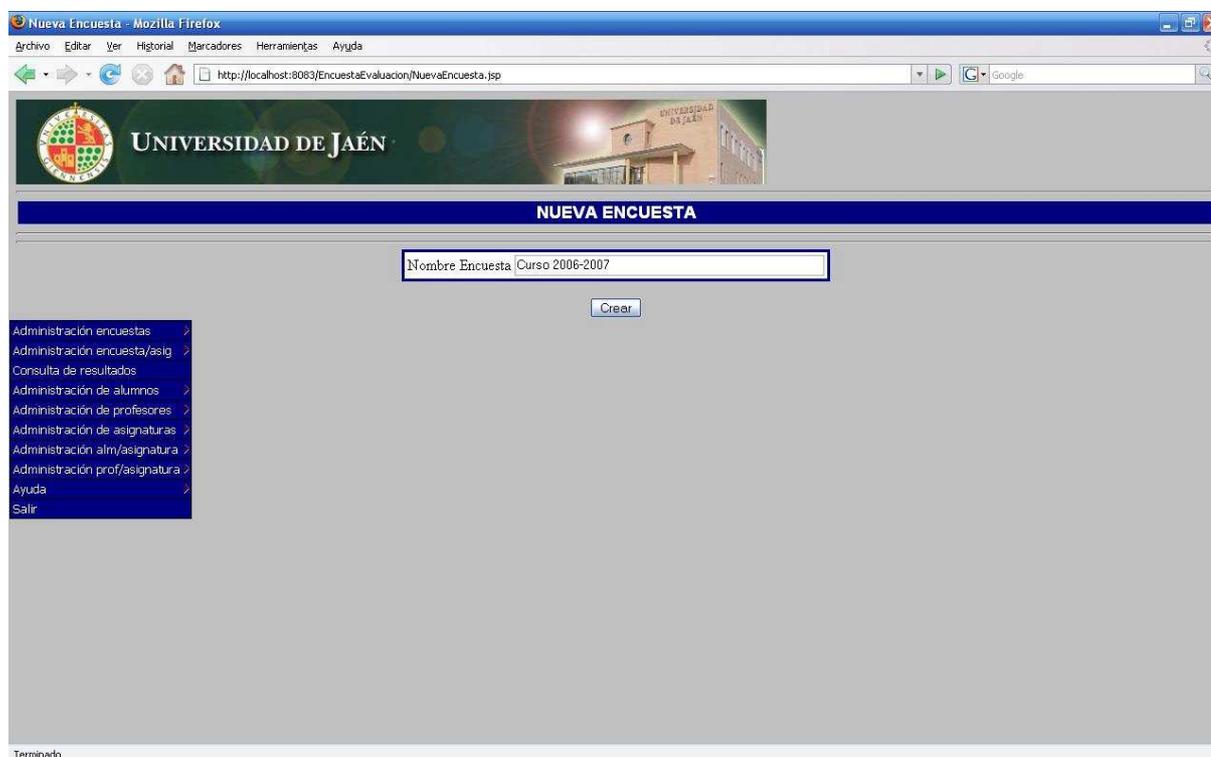


Figura III.6 Crear encuesta

Si ya existe una encuesta con el nombre introducido, la aplicación nos indicará un mensaje de error como el mostrado en la figura III.7, y nos invitará a realizar de nuevo la misma operación.

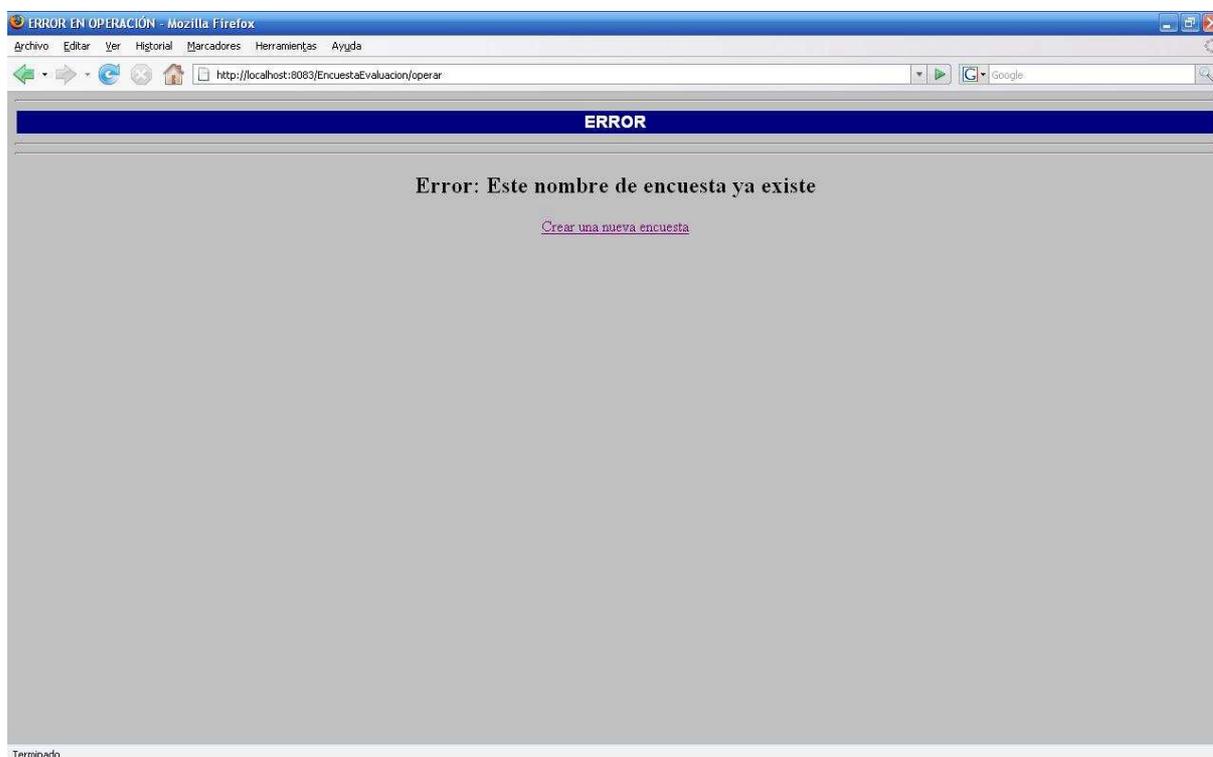


Figura III.7 Error al Crear Encuesta. Nombre Repetido

Si todo es correcto, una vez que hemos pulsado el botón de crear (figura III.6), nos aparecerá una pantalla similar a la figura III.8, en la que podemos observar una tabla en la que aparecerán las preguntas, su tipo de dominio y su dominio, una vez introducidas. También aparecen tres botones que nos permitirán insertar, modificar y borrar preguntas y un botón debajo de la tabla de preguntas que nos permite guardar los cambios realizados en la encuesta.



Figura III.8 Muestra Encuesta

Como hemos comentado anteriormente tenemos la posibilidad de realizar tres acciones:

- Insertar una nueva pregunta, pulsando el botón Nueva Pregunta.
- Modificar o Borrar una pregunta, pulsando el botón Modificar/Borrar Pregunta.
- Terminar la creación de la encuesta, pulsando el botón Terminar y Guardar Encuesta

Insertar una nueva pregunta.

Si pulsamos el botón Nueva Pregunta, nos aparece una pantalla similar a la figura III.9. En esta pantalla encontramos varias cosas.

En primer lugar nos aparece un cuadro de texto en el que debemos introducir la pregunta que deseamos.

A continuación nos aparecen tres botones de radio, para elegir uno, que será el tipo de dominio de la pregunta introducida en el cuadro anterior. Como es lógico cada tipo de dominio tiene un conjunto de dominios, es por esto que si cambiamos el tipo, la lista desplegable que aparece a la derecha también cambia, apareciendo los dominios existentes, según el tipo marcado.

Terminado

Figura III.9 Nueva Pregunta

Debajo aparece una tabla con las preguntas que llevamos introducidas hasta el momento en la encuesta.

Una vez que tengamos introducidos la pregunta, su tipo de dominio y su dominio, sólo nos queda guardar la pregunta, utilizando para ello el botón Guardar Pregunta.

Si dejamos en blanco el texto de la pregunta nos aparece un error como el mostrado en la figura III.10, ya que no puede existir una pregunta en blanco.

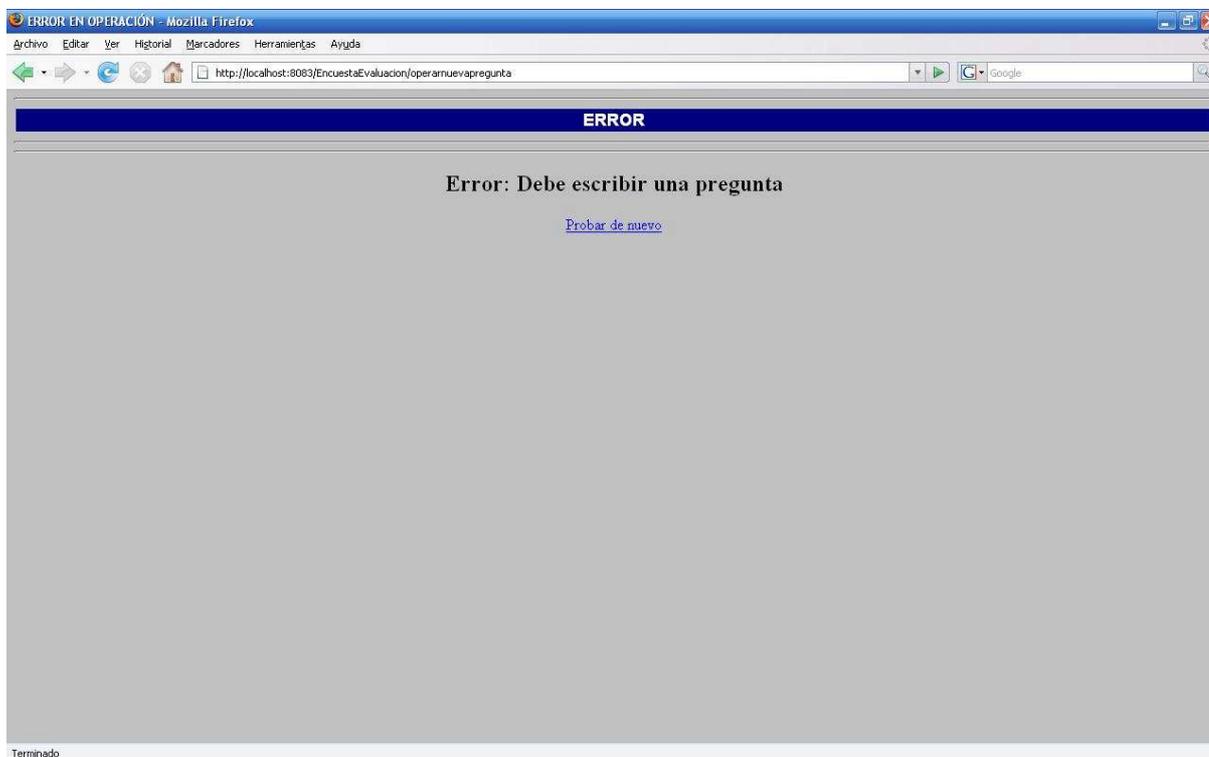


Figura III.10 Error al no introducir texto en una pregunta

Una vez guardada la pregunta el sistema nos vuelve al menú de manejo de encuestas, en el que podemos comprobar que la nueva pregunta insertada aparece ya en la tabla de

preguntas, figura III.11.

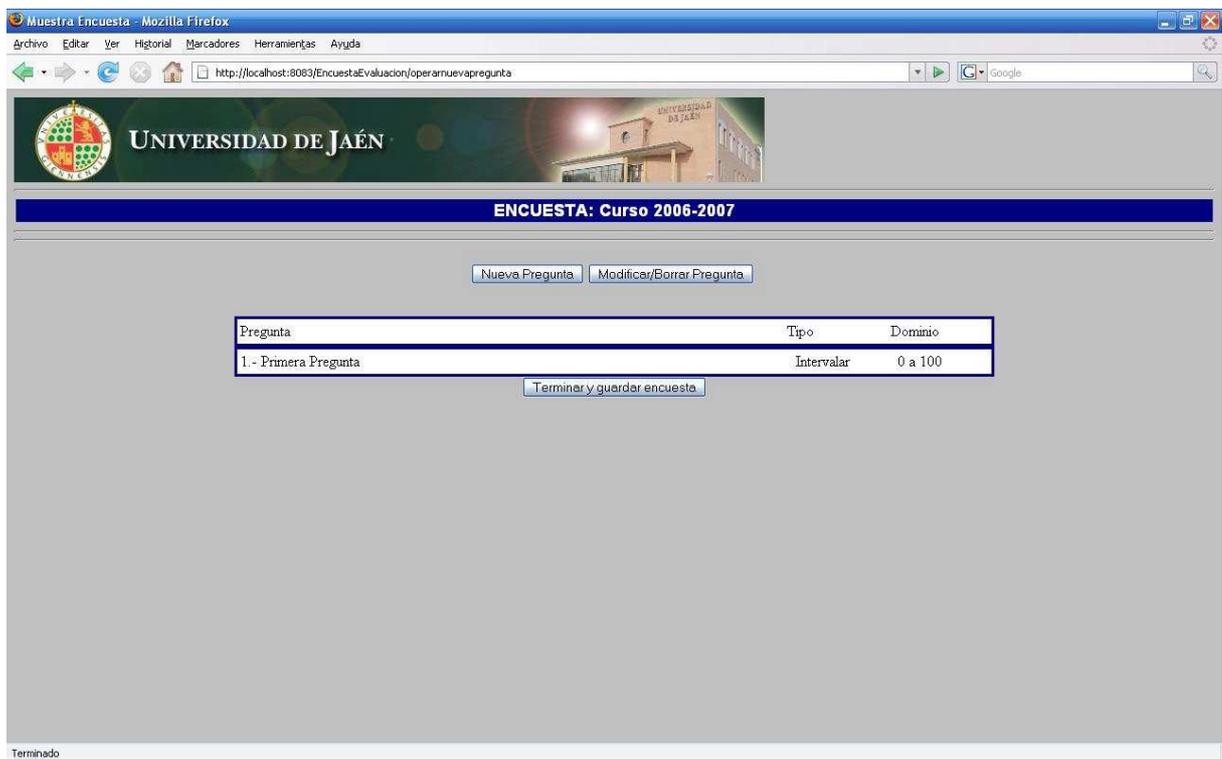


Figura III.11 Pregunta insertada

Repetiendo los pasos anteriores volvemos a insertar una pregunta, tantas como nuestra encuesta requiera.

Modificar una pregunta.

Si pulsamos el botón Modificar/Borrar Pregunta, nos aparece una pantalla similar a la figura III.12. En esta pantalla encontramos una lista desplegable, dos botones y la tabla con las preguntas insertadas hasta ahora.

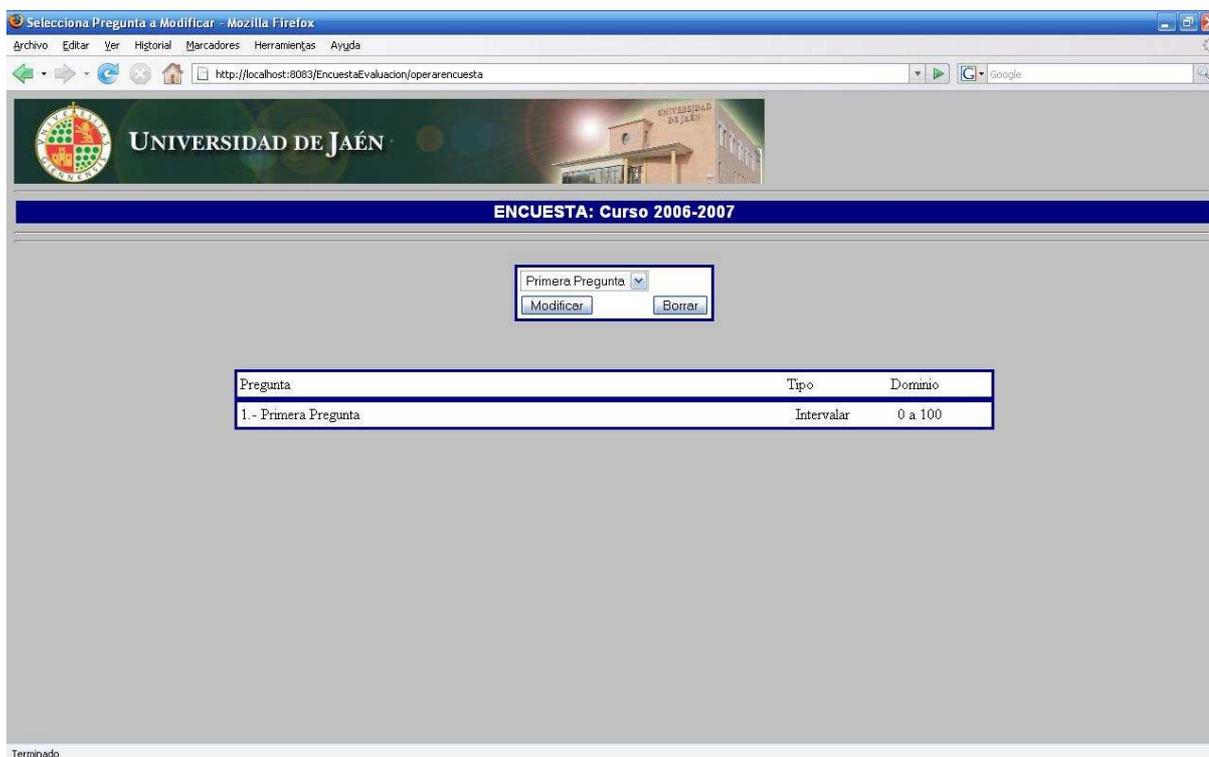


Figura III.12 Modificar pregunta

En este caso queremos modificar una pregunta, para ello debemos elegir, en la lista desplegable, la pregunta que queremos modificar y pulsar el botón modificar.

Una vez hecho esto, nos aparece una pantalla similar a la de Nueva Pregunta (figura III.9), pero en el cuadro de texto nos aparece el texto actual de la pregunta y el tipo de dominio elegido.

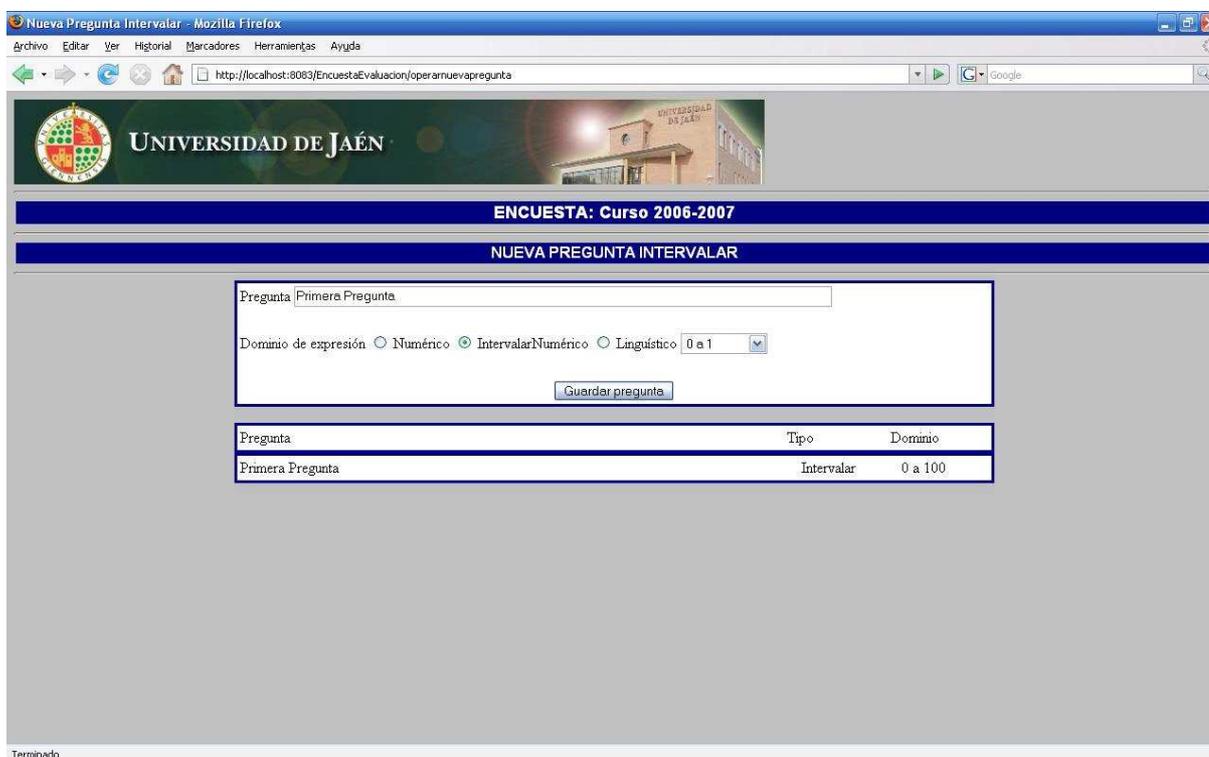


Figura III. 13 Modificar Pregunta

Para modificar la pregunta actuaremos de la misma forma que a la hora de introducir una nueva pregunta, eligiendo el tipo de dominio y el dominio para la pregunta, así como el texto de ésta y pulsando el botón guardar para almacenar los cambios.

Al pulsar Guardar Pregunta, volvemos al menú de manejo de encuesta, comprobando en la tabla de preguntas que ha sido modificada.

Borrar una pregunta.

Si pulsamos el botón Modificar/Borrar Pregunta, nos aparece una pantalla similar a la figura III.14. En esta pantalla encontramos una lista desplegable, dos botones y la tabla con las preguntas insertadas hasta ahora.

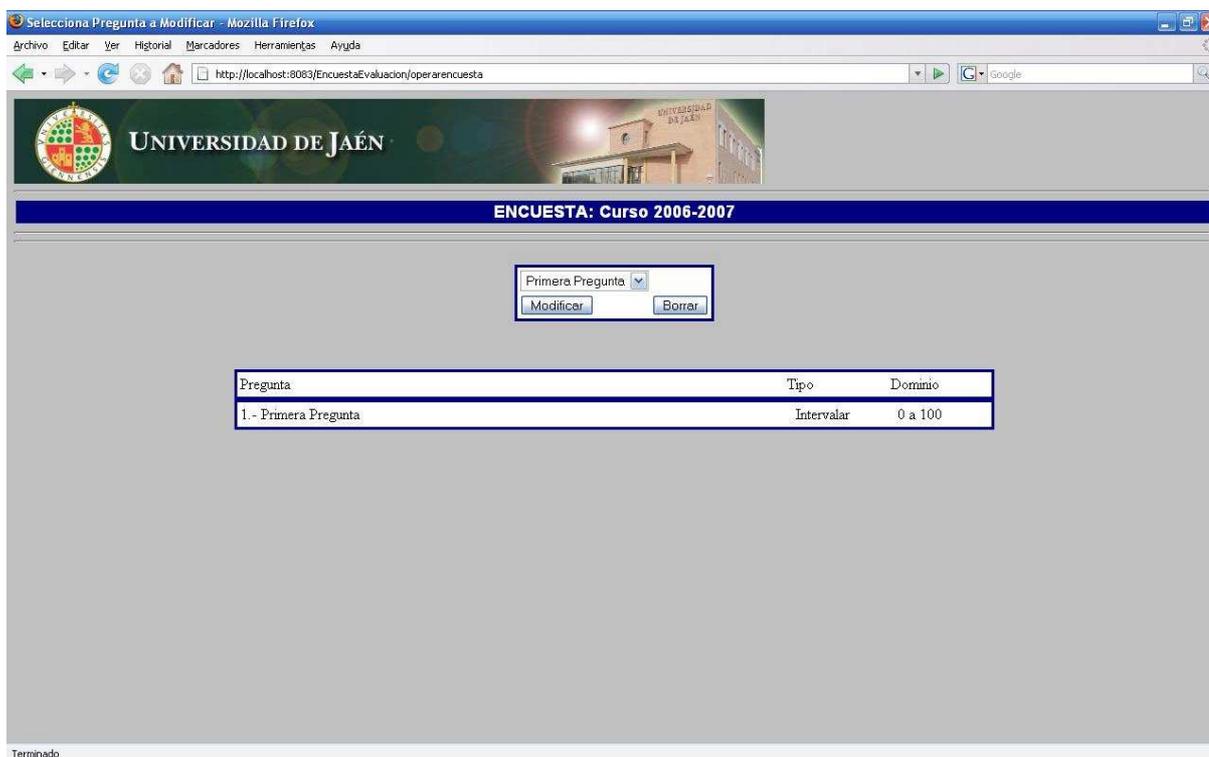


Figura III.14 Borrar Pregunta

En este caso queremos borrar una pregunta, para ello debemos elegir, en la lista desplegable, la pregunta que queremos borrar y pulsar el botón Borrar.

Al pulsar el botón Borrar, la pregunta es eliminada de la lista de preguntas volviendo al menú de manejo de encuestas. Podemos comprobar en la tabla de preguntas que ésta ha sido eliminada.

1.1.2 Modificar/consultar

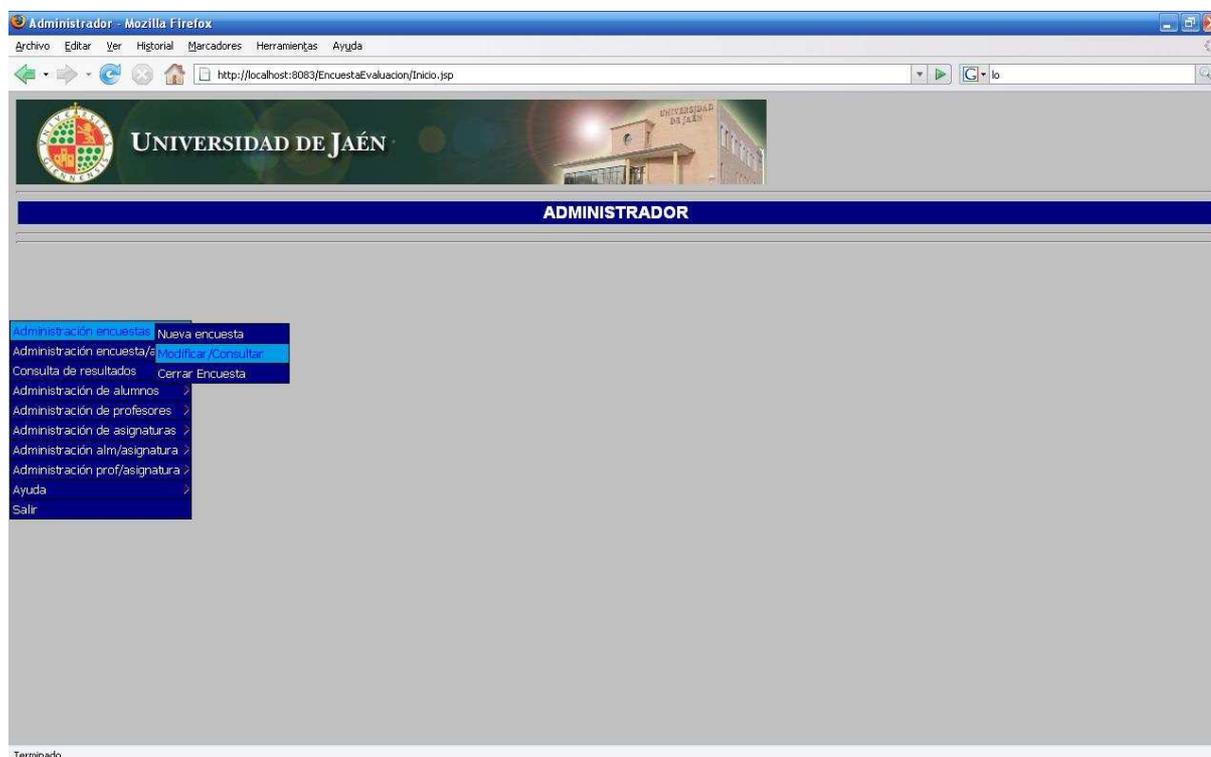


Figura III. 15 Modificar/Consultar Encuesta

En este apartado podemos modificar una encuesta siempre y cuando no este asignada a una asignatura. En el caso de que este asignada sólo podremos consultar las preguntas de esta encuesta, pero en ningún caso modificarla.

Modificar Encuesta:

El primer paso para modificar una encuesta es seleccionar de la lista desplegable la encuesta a modificar. Una vez seleccionada debemos pulsar el botón Modificar/Consultar para acceder a la encuesta elegida.

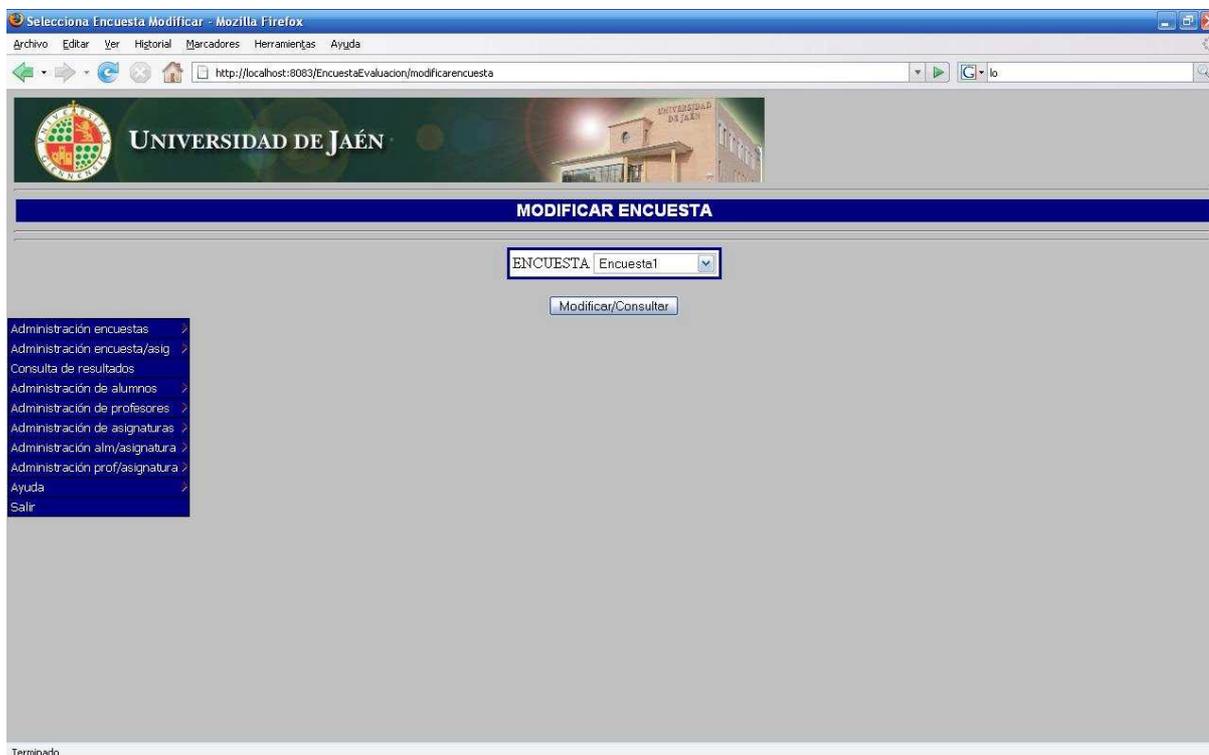


Figura III.16 Selección de encuesta a Modificar/Consultar

Una vez abierta la encuesta el manejo de las preguntas es igual al explicado en el punto anterior de este manual.

Consultar Encuesta:

Si la encuesta esta asignada a una asignatura, nos aparece una pantalla similar a la figura III.17, en la que nos muestra las preguntas de la encuesta y nos indica que no podemos modificarla. Pulsando el botón Volver que aparece al final, volvemos al menú inicial.

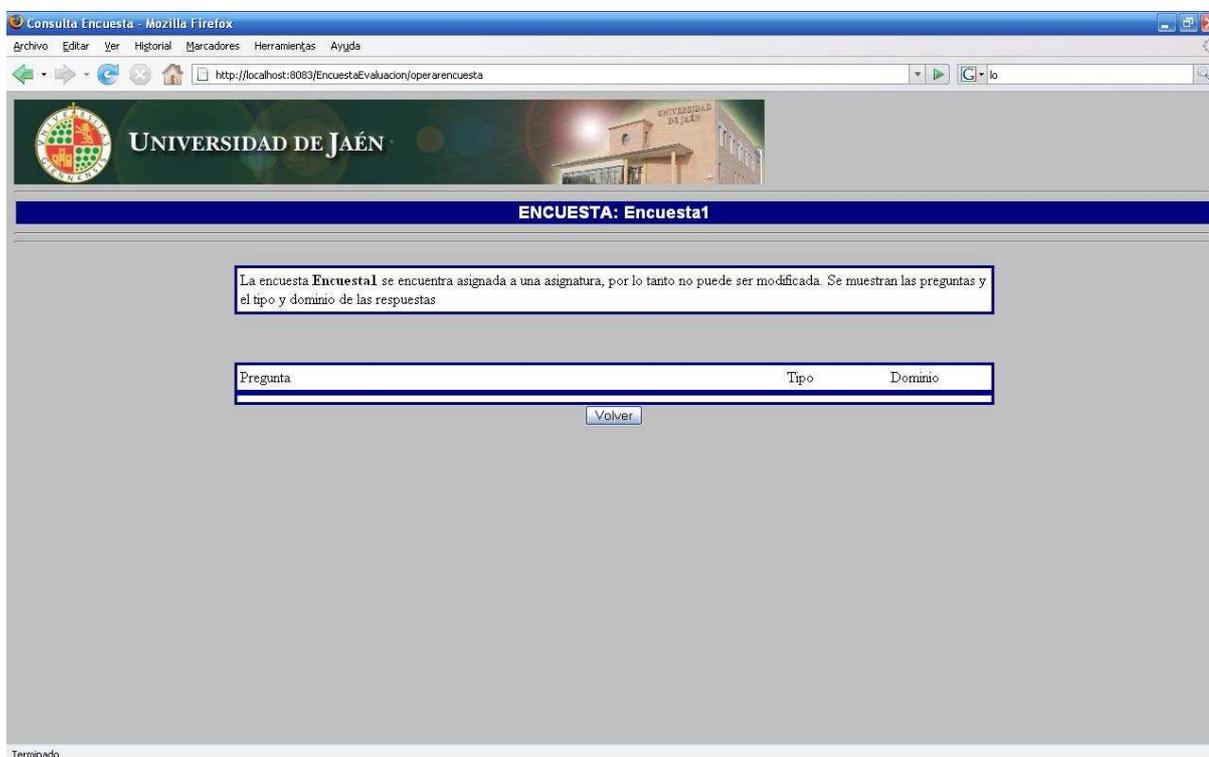


Figura III.17 Consulta de una encuesta que se encuentra ya asignada

1.1.3 Cerrar Encuesta

Este apartado está destinado a cerrar el plazo para contestar una encuesta que ha sido asignada con anterioridad y a la que los alumnos ya han contestado.

Al cerrar una encuesta se realiza el cálculo de los resultados de la misma.

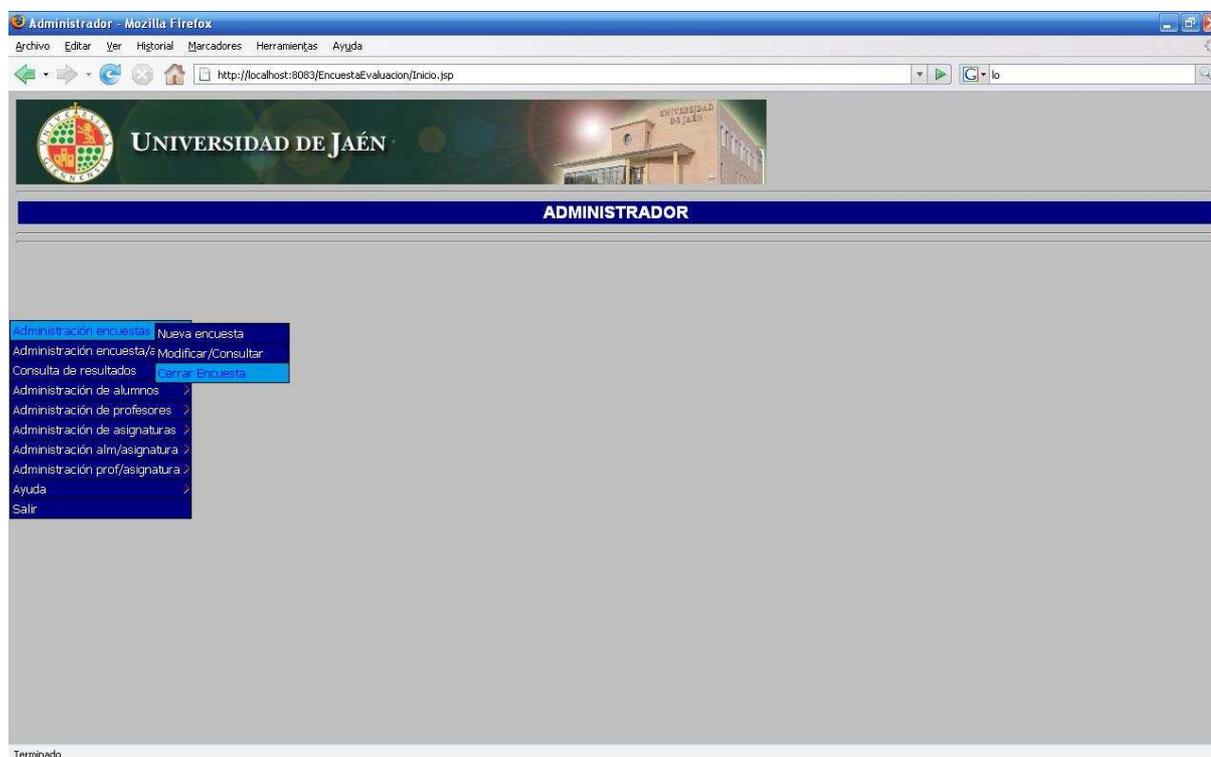


Figura III.18 Cerrar una encuesta

El proceso de cerrar una encuesta es bastante sencillo. En primer lugar hay que elegir, de una lista desplegable (Figura III.19), la asignatura a la cual pertenece la encuesta que queremos cerrar. Una vez hecho esto nos aparece una nueva lista desplegable (Figura III.20) con los nombres de las encuestas que pertenecen a la asignatura seleccionada anteriormente y que todavía están abiertas.



Figura III.19 Selección de asignatura para cerrar una encuesta

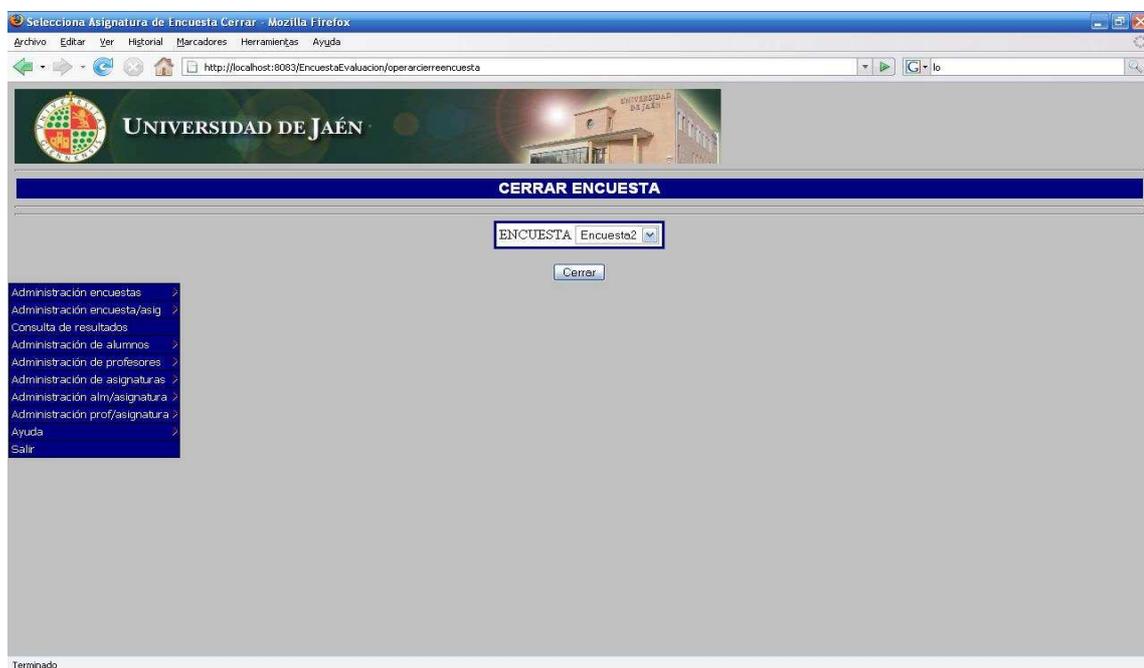


Figura III.20 Selección de encuesta para cerrar

Una vez pulsado el botón Cerrar, nos aparece una ventana indicándonos que es lo que vamos a hacer y que supone cerrar una encuesta, pidiéndonos la confirmación de la acción.

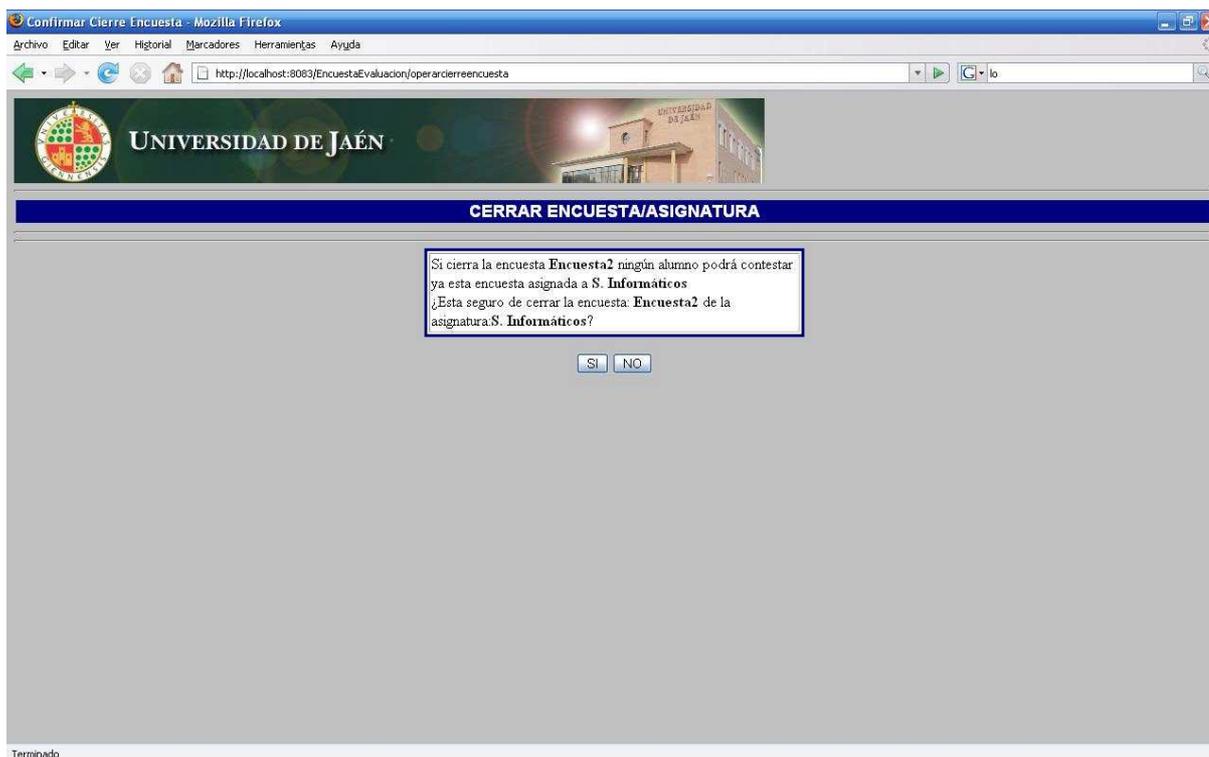


Figura III.21 Petición de confirmación de cierre de encuesta

Si pulsamos en botón Si, aparece una pantalla de confirmación indicando que la encuesta se ha cerrado. Si pulsamos el botón No, no se realizara ninguna acción y volveremos al menú principal.

1.2 Administración Encuesta/Asig

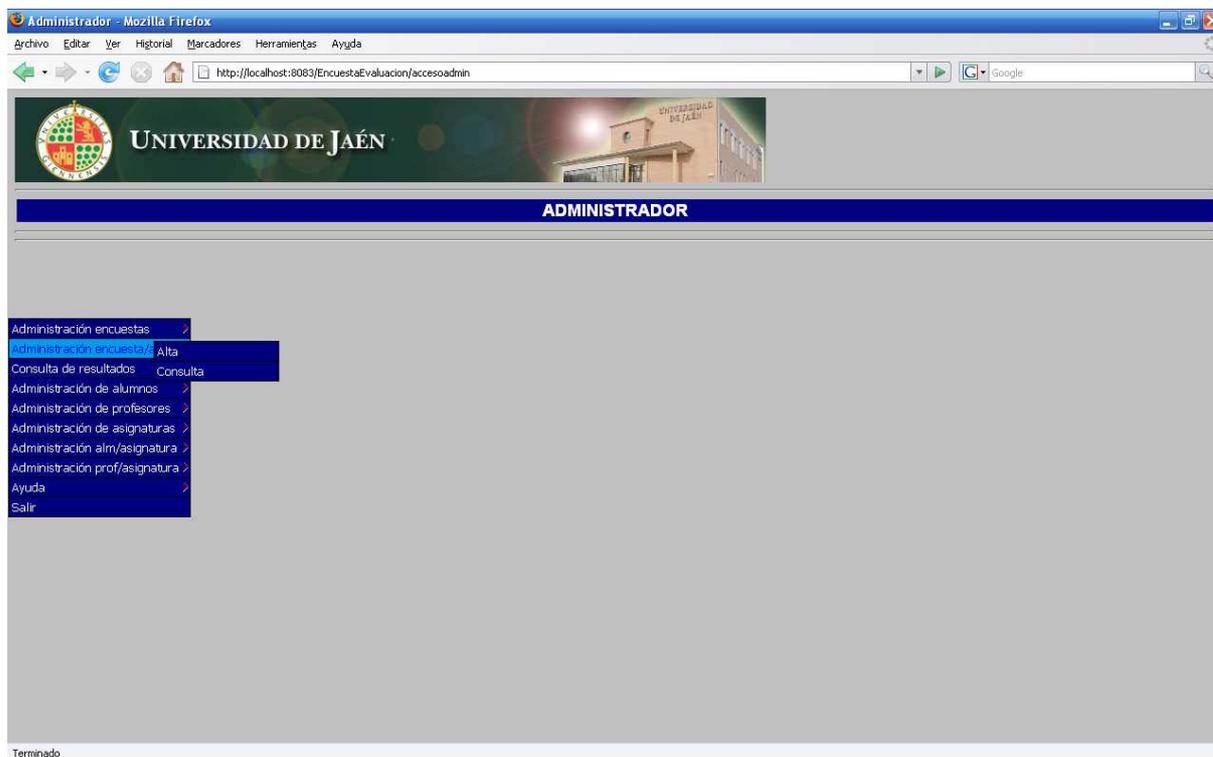


Figura III.22 Menú Administración encuesta/asig

Alta

Si pulsamos en Alta dentro de Administración de encuesta/asig, nos aparece una pantalla con dos listas desplegadas en las que elegir la asignatura que queremos evaluar y la encuesta que queremos asignarle.

Una vez elegidos pulsamos el botón Guardar, informándonos el sistema de que la operación se ha realizado de forma correcta. Si esta asignación ya hubiera sido realizada con anterioridad en sistema nos informará de este hecho.

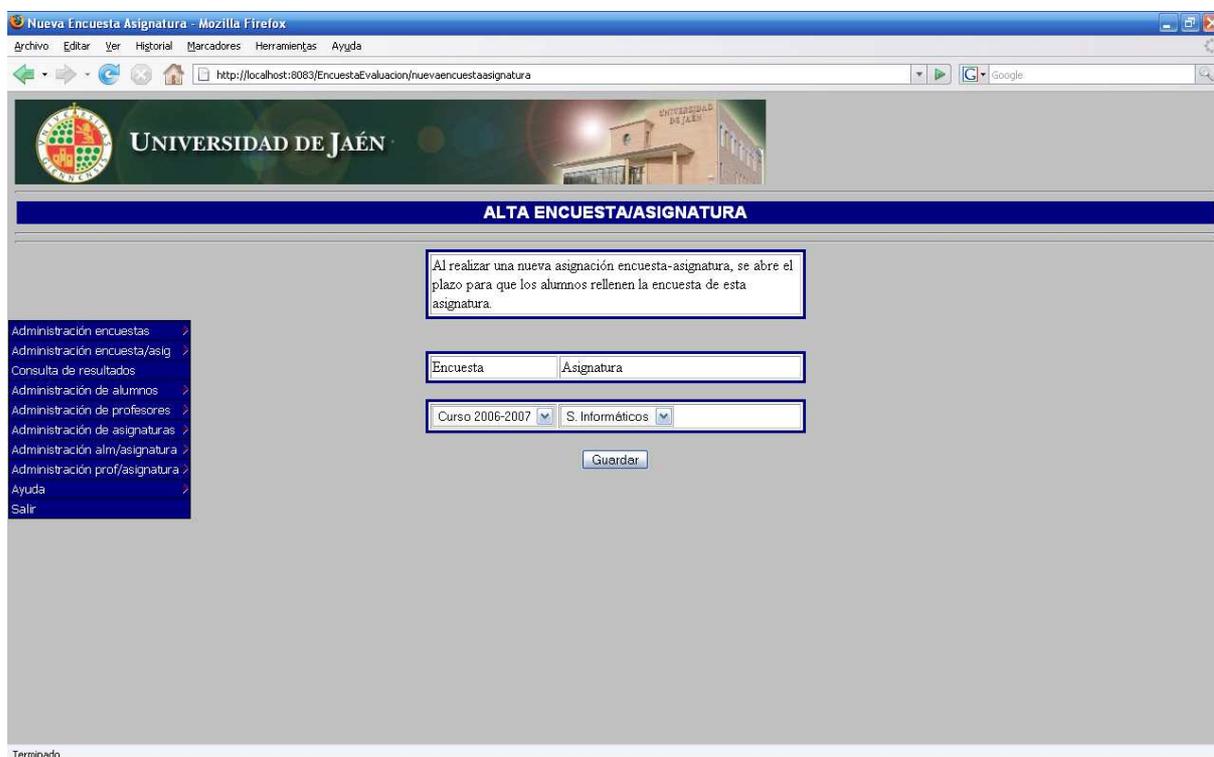


Figura III.23 Alta encuesta/asig

Consulta

En este apartado podemos ver todas las asignaciones encuesta/asignatura que se encuentran almacenadas en la BBDD.

1.3 Consulta de resultados

En esta opción podemos ver los resultados obtenidos de una encuesta, una vez que ésta se encuentra cerrada.

Para ello en primer lugar debemos elegir, de una lista desplegable, la asignatura a la que pertenece la encuesta y en segundo lugar la encuesta que queremos consultar.

Una vez realizado esto, obtendremos una pantalla similar a la figura III.24, en la que observamos en una primera tabla los resultados globales de la encuesta. En una segunda tabla obtenemos los resultados por pregunta.

Muestra Resultados - Mozilla Firefox

http://localhost:8083/EncuestaEvaluacion/operarconsultaresultados

UNIVERSIDAD DE JAÉN

RESULTADOS DE LA ENCUESTA: Curso 2006-2007 DE LA ASIGNATURA: S. Informáticos

Los términos utilizados para la evaluación han sido: S: { None Low Medium High Perfect }

El número de alumnos que han contestado la encuesta es: 3

La valoración global de la encuesta es: (High, -0.27)

Pregunta	Contestadas	Valoración
1.- Pregunta Primera	2	(High , -0.39)
2.- Pregunta Segunda	2	(Medium , 0.33)
3.- Pregunta Tercera	3	(Medium , 0.0)
4.- Pregunta cuarta	3	(Perfect , -0.38)
5.- Ulhma Pregunta	3	(Perfect , -0.39)

Salir Consultar otra encuesta

Terminado

Figura III.24 Consulta de resultados

2. Manejo de la Base de Datos

Como vimos anteriormente, este segundo grupo es el encargado de manejar toda la información de la base de datos referente a alumnos, profesores y asignaturas.

Este grupo esta formado por:

- Administración de alumnos
- Administración de profesores
- Administración de asignaturas
- Administración de Alm/Asignaturas
- Administración de Prof/Asignaturas

Para cada uno de los apartados anteriores existen las mismas opciones: Alta, Modificación y Consulta.

2.1 Administración de alumnos

Alta

Si pulsamos en Alta dentro de administración de alumnos, nos aparecerá una pantalla con un formulario en el que introducir los datos de un alumno. Es necesario introducir todos los datos, ya que no están permitidos datos en blanco. El DNI debe ser introducido sin letra.

Pulsando el botón Guardar se creará un nuevo alumno con los datos introducidos.

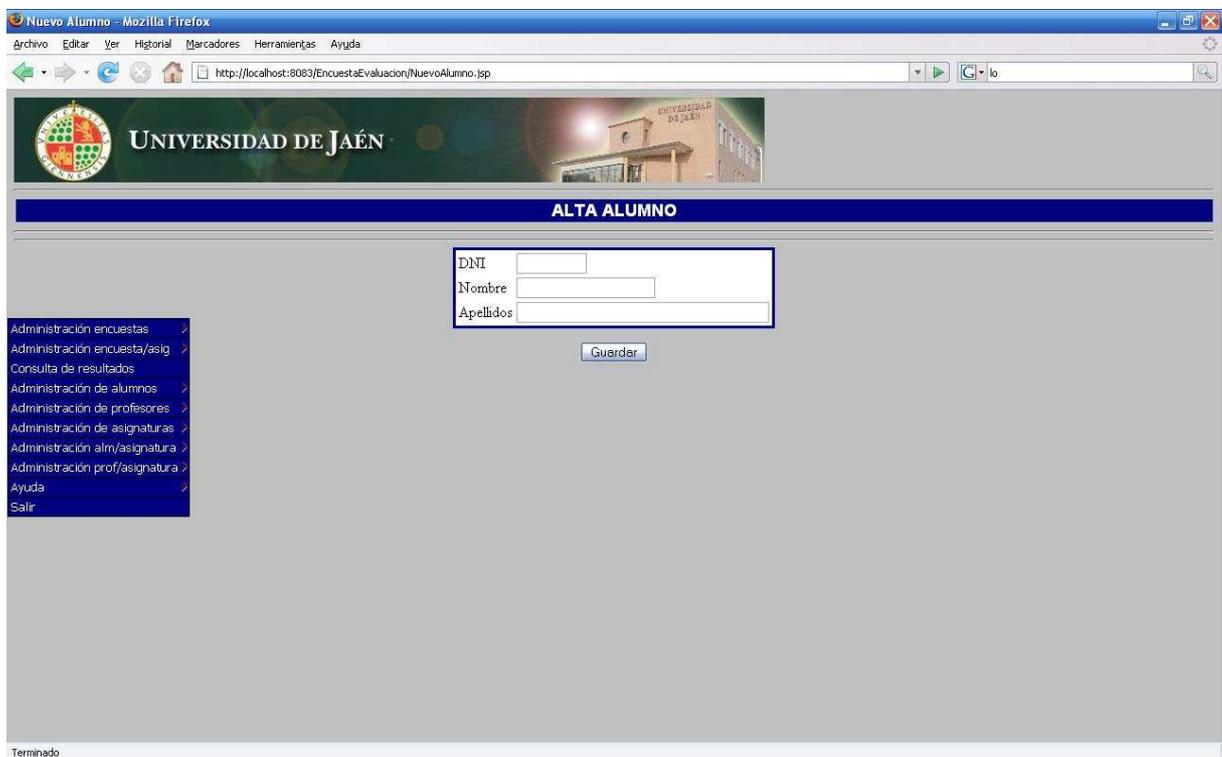


Figura III.25 Formulario de Alta Alumno

Modificación

Si pulsamos en Modificación dentro de administración de alumnos, nos aparecerá una pantalla con una lista que contiene los DNI de todos los alumnos. Seleccionamos el DNI del alumno que queremos modificar y pulsamos modificar, figura III.26

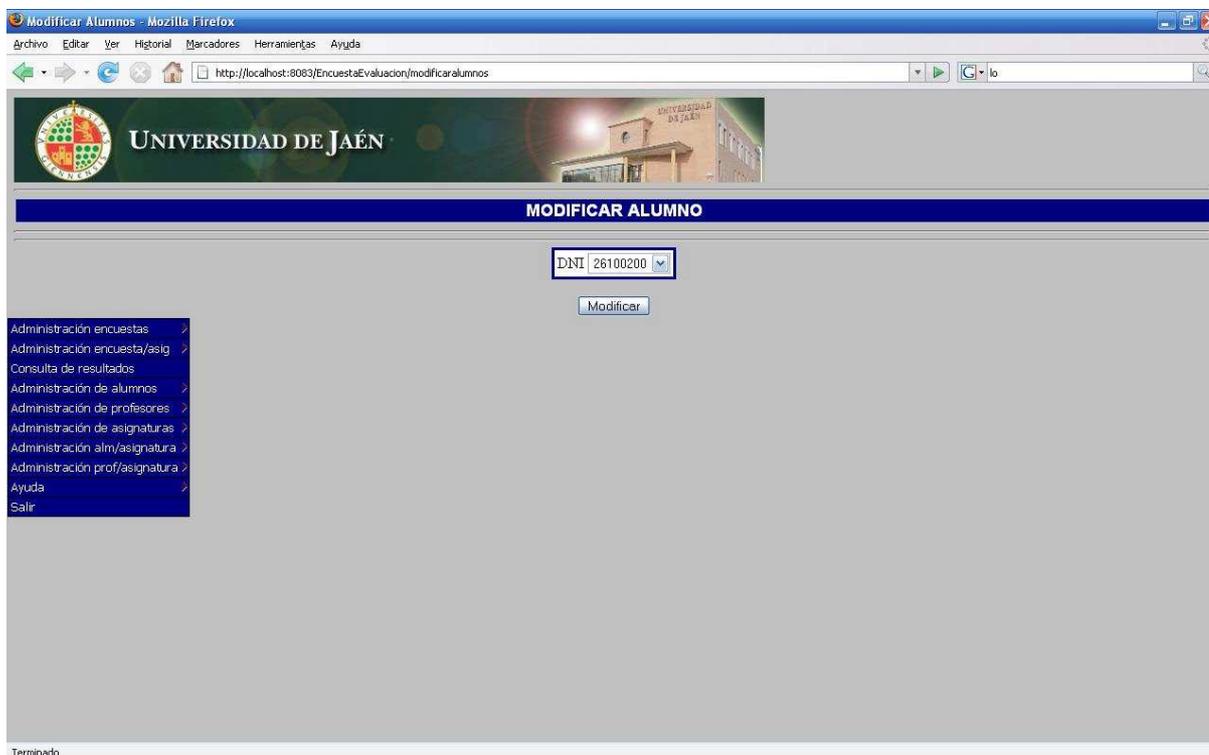


Figura III.26 Modificación de Alumno

Una vez elegido el alumno a modificar, nos aparece un formulario con los datos actuales del alumno, figura III.27. Modificamos los datos incorrectos y pulsamos el botón Guardar Cambios. Si los datos introducidos son correctos nos aparece una pantalla confirmando la modificación. En caso de dejar algún campo en blanco o introducir un DNI erróneo la aplicación nos indicará el error, debiendo introducir los datos de forma correcta.

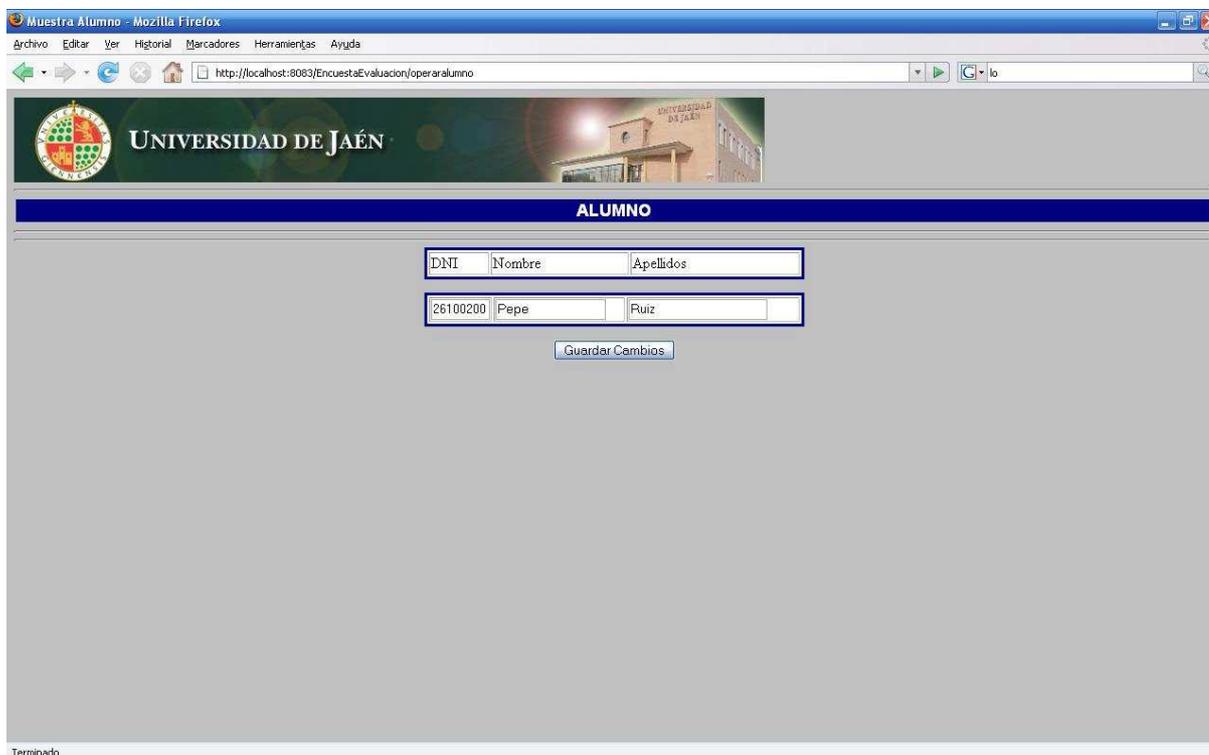


Figura III.27 Modificación de los datos de un Alumno

Consulta

En este apartado podemos ver los datos de todos los alumnos que se encuentran dados de alta en el sistema.

2.2 Administración de profesores

Alta

Si pulsamos en Alta dentro de administración de profesores, nos aparecerá una pantalla con un formulario en el que introducir los datos de un profesor. Es necesario introducir todos los datos, ya que no están permitidos datos en blanco. El DNI debe ser introducido sin letra.

Pulsando el botón Guardar se creará un nuevo profesor con los datos introducidos.

Nuevo Profesor - Mozilla Firefox
http://localhost:8083/EncuestaEvaluacion/NuevoProfesor.jsp

UNIVERSIDAD DE JAÉN

ALTA PROFESOR

DNI
Nombre
Apellidos

Guardar

- Administración encuestas >
- Administración encuesta/asig >
- Consulta de resultados >
- Administración de alumnos >
- Administración de profesores >
- Administración de asignaturas >
- Administración alm/asignatura >
- Administración prof/asignatura >
- Ayuda >
- Salir >

Terminado

Figura III.28 Formulario de Alta de Profesor

Modificación

Si pulsamos en Modificación dentro de administración de profesores, nos aparecerá una pantalla con una lista que contiene los DNI de todos los profesores. Seleccionamos el DNI del profesor que queremos modificar y pulsamos modificar.

Una vez elegido el profesor a modificar, nos aparece un formulario con los datos actuales del profesor, figura III.29. Modificamos los datos incorrectos y pulsamos el botón Guardar Cambios. Si los datos introducidos son correctos nos aparece una pantalla confirmando la modificación. En caso de dejar algún campo en blanco o introducir un DNI erróneo la aplicación nos indicará el error, debiendo introducir los datos de forma correcta.

DNI	Nombre	Apellidos
34123328	Pedro	Sánchez

Figura III.29 Modificación de los datos de un Profesor

Consulta

En este apartado podemos ver los datos de todos los profesores que se encuentran dados de alta en el sistema.

2.3 Administración de Asignaturas

Alta

Si pulsamos en Alta dentro de administración de asignaturas, nos aparecerá una pantalla con un formulario en el que introducir el nombre de una nueva asignatura. No está permitido dejar este dato en blanco.

Modificación

Si pulsamos en Modificación dentro de administración de asignaturas, nos aparecerá una pantalla con una lista que contiene los nombres de todas las asignaturas. Seleccionamos la que queremos modificar y pulsamos el botón Modificar.

A continuación nos aparece un formulario en el que modificar el nombre de la asignatura. Pulsando el botón Guardar Cambios, se modifica el nombre de la asignatura.

Consulta

En este apartado podemos ver las asignaturas que existen en el sistema.

2.4 Administración de Alm/Asignaturas

Alta

Si pulsamos en Alta dentro de administración de Alm/Asignaturas, nos aparecerá una pantalla con dos listas desplegables en las que elegir el DNI del alumno y la asignatura a la que queremos asignarle.

Una vez elegidos pulsamos el botón guardar, informándonos el sistema de que la operación se ha realizado de forma correcta. Si el alumno elegido estuviera ya asignado a esta asignatura, la aplicación no dejaría realizar la operación, indicándolo mediante un error.

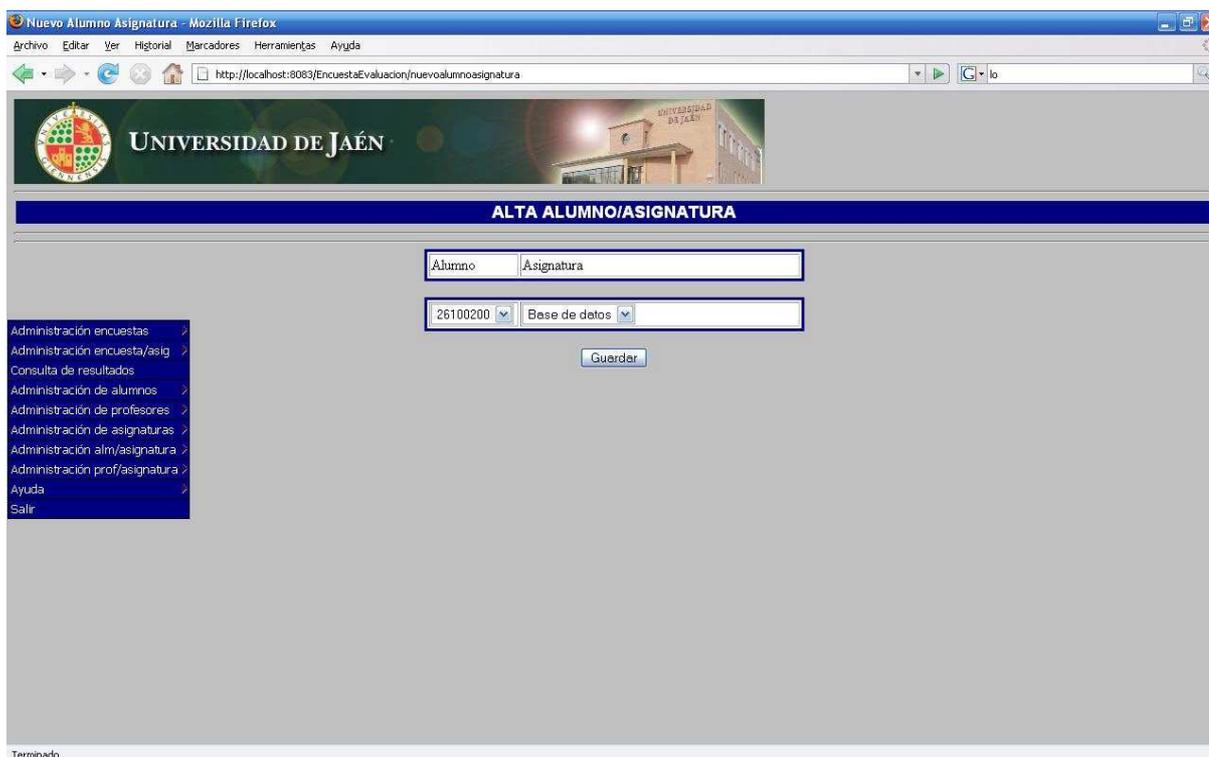


Figura III.30 Alta Alumnos/Asignatura

Modificación

Si queremos modificar la asignación de un alumno en una asignatura, debemos ir al apartado modificar, dentro de administración de Alm/Asignaturas.

En primer lugar nos aparece una lista con los nombres de las asignaturas del sistema. Seleccionamos la que nos interesa. A continuación nos aparece otra lista con los DNI de los alumnos que se encuentran asignados a esta asignatura, seleccionamos el DNI que nos interesa.

Por último nos aparecen los datos del alumno seleccionado (solo de lectura) y una lista con las asignaturas del sistema, figura III.31. Elegimos la nueva asignatura que queremos asignarle y pulsamos el botón Guardar Cambios.

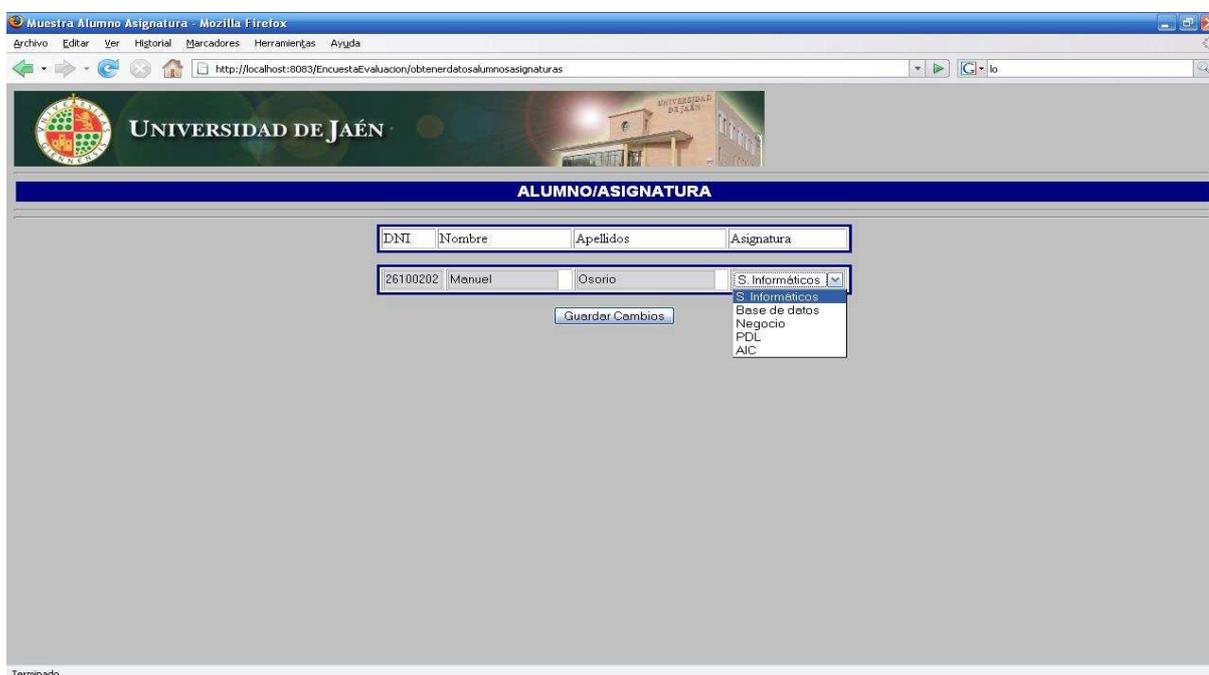


Figura III.31 Modificación Alumnos/Asignatura

Consulta

En este apartado podemos ver todas las asignaciones alumno-asignatura almacenadas en el sistema.

2.5 Administración de Prof/Asignaturas

Este apartado es similar al 2.4, por lo que no se muestran figuras, sólo el proceso para llevar a cabo el alta, y la modificación

Alta

Si pulsamos en Alta dentro de administración de Prof/Asignaturas, nos aparecerá una pantalla con dos listas desplegadas en las que elegir el DNI del profesor y la asignatura a la que queremos asignarle.

Una vez elegidos pulsamos el botón guardar, informándonos el sistema de que la operación se ha realizado de forma correcta. Si el profesor elegido estuviera ya asignado a esta asignatura, la aplicación no dejaría realizar la operación, indicándolo mediante un error.

Modificación

Si queremos modificar la asignación de un profesor en una asignatura, debemos ir al apartado modificar, dentro de administración de Prof/Asignaturas.

En primer lugar nos aparece una lista con los nombres de las asignaturas del sistema. Seleccionamos la que nos interesa. A continuación nos aparece otra lista con los DNI de los profesores que se encuentran asignados a esta asignatura, seleccionamos el DNI que nos interesa.

Por último nos aparecen los datos del profesor seleccionado (solo de lectura) y una lista con las asignaturas del sistema. Elegimos la nueva asignatura que queremos asignarle y pulsamos el botón Guardar Cambios.

Consulta

En este apartado podemos ver todas las asignaciones profesor-asignatura almacenadas en el sistema.

3. Información de la aplicación

3.1 Acerca de

En este apartado encontramos información acerca de la versión de la aplicación así como el nombre del autor.

3.2 Manual

Aquí podemos encontrar este mismo manual, en formato HTML.

Por último, el botón Salir nos permite abandonar la aplicación.

ANEXO IV

Manual de Usuario: Alumno

Este manual de usuario está organizado como una visita guiada por la aplicación, en su apartado de alumno. Antes de embarcarse en ella es conveniente que el usuario tenga claro lo siguiente:

- Ésta es una aplicación web optimizada para su visualización en un navegador Firefox (a ser posible su versión más reciente, la cual se puede descargar en <http://www.mozilla-europe.org/es/>) y con una resolución no inferior a 800x600 pixels. Si se utiliza otro navegador o una resolución inferior a la recomendada se pueden producir fallos de visualización aunque la funcionalidad de la aplicación esta completamente asegurada.

Una vez aclarado esto puede empezar la visita guiada.

El primer paso para utilizar la aplicación es abrir el Navegador (Firefox) y teclear la dirección en la que tenemos alojada esta, que en nuestro caso será:
<http://localhost:8083/EncuestaEvaluacion/index.jsp>

Al realizar lo anteriormente comentado, nos encontramos con la página de inicio, tal y como se ve en la figura IV.1

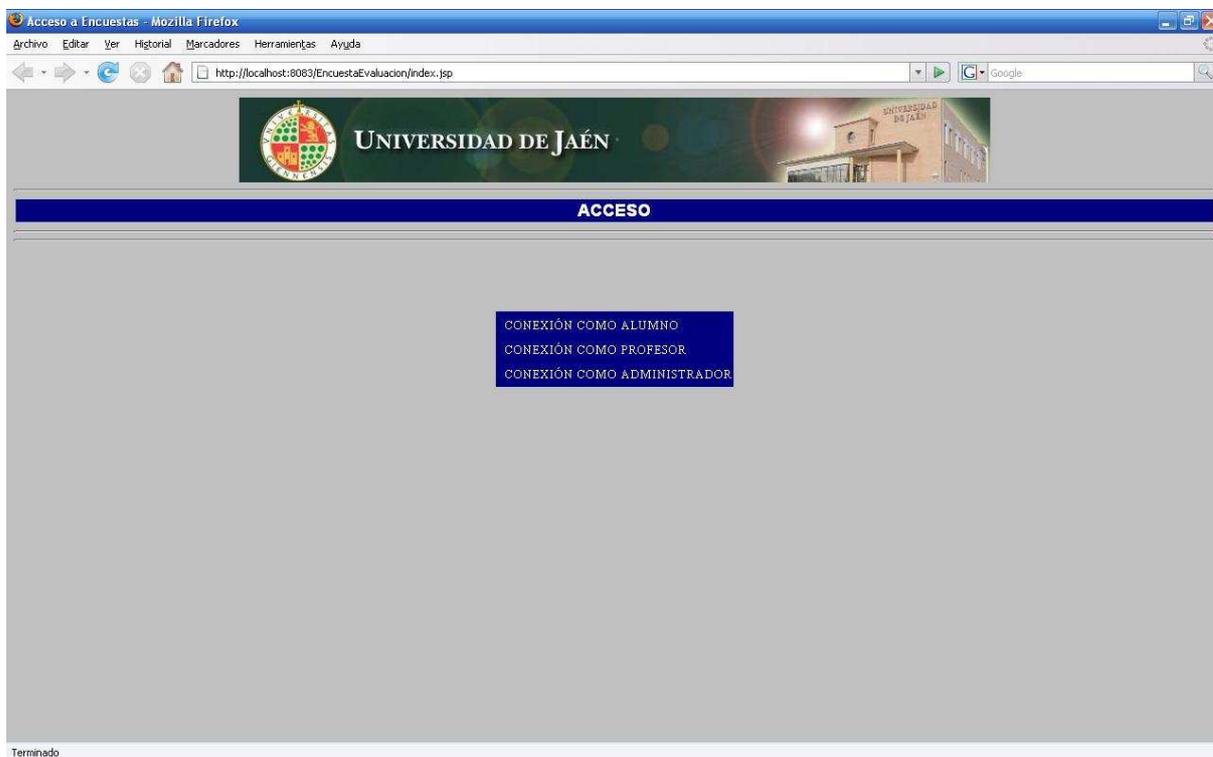


Figura IV. 1: Inicio de la Aplicación

Como es Alumno, debe hacer clic en el botón de Conexión como Alumno, accediendo a la pantalla de identificación, figura IV.2

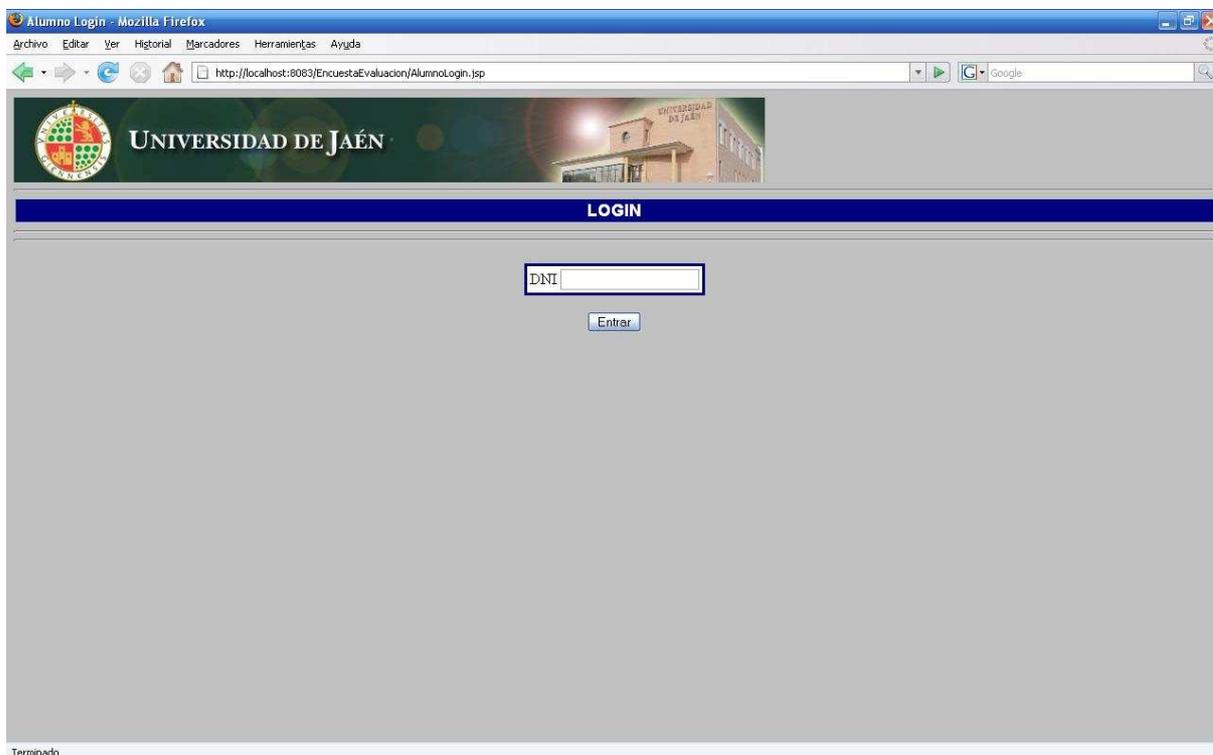


Figura IV.2 Identificación de Alumno

Debe introducir su DNI y pulsar Entrar. Si el DNI introducido no es correcto el sistema nos muestra la pantalla de la figura IV.3, indicando que el DNI introducido no se encuentra en la base de datos y le invita a una nueva identificación.

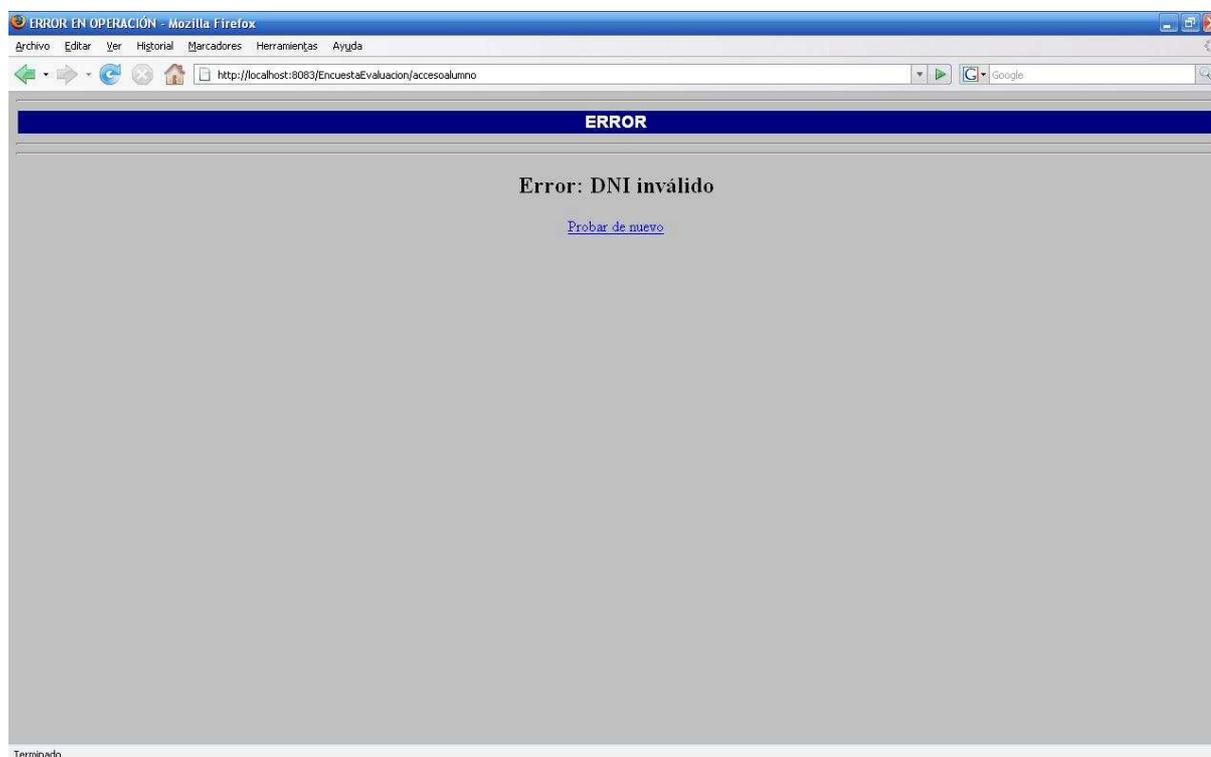


Figura IV.3 Identificación de Alumno incorrecta

En el caso de una identificación correcta aparece una pantalla similar a la figura IV.4. En esta pantalla podemos ver que en la cabecera aparece su nombre.

En esta pantalla inicial aparecen dos opciones:

- Seleccionar una asignatura: debe elegir de la lista, la asignatura que quiere evaluar.
- Salir del sistema, pulsando el botón Salir.

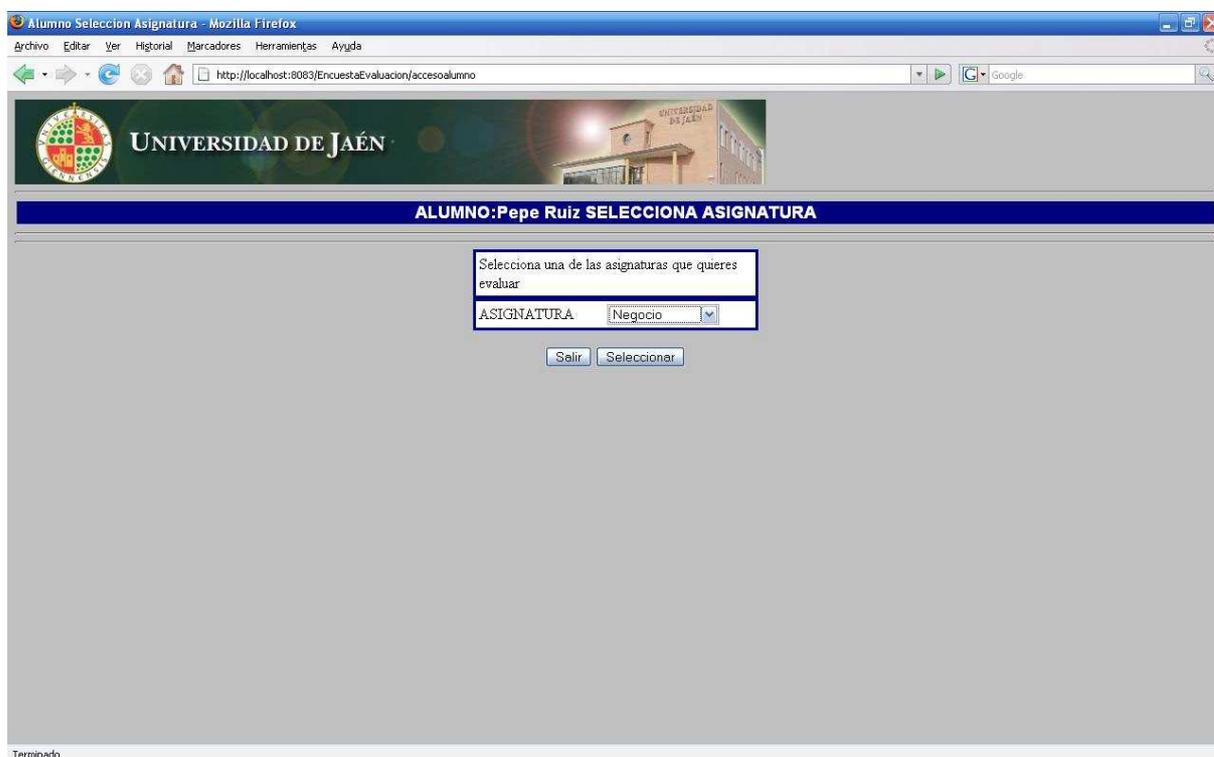


Figura IV.4 Menú inicial de Alumno

Si ha seleccionado una asignatura y pulsado el botón Seleccionar, le aparecerá una pantalla como la de la figura IV.5, en la que puede seleccionar las encuestas que aún no ha contestado relacionadas con la asignatura anteriormente elegida.

Si ya ha contestado a todas las encuestas de la asignatura o no hay ninguna encuesta asignada a esa asignatura aparece un mensaje en pantalla advirtiéndole de tal hecho, permitiéndole volver a seleccionar otra asignatura.

Si pulsa el botón Salir, abandonará el sistema.

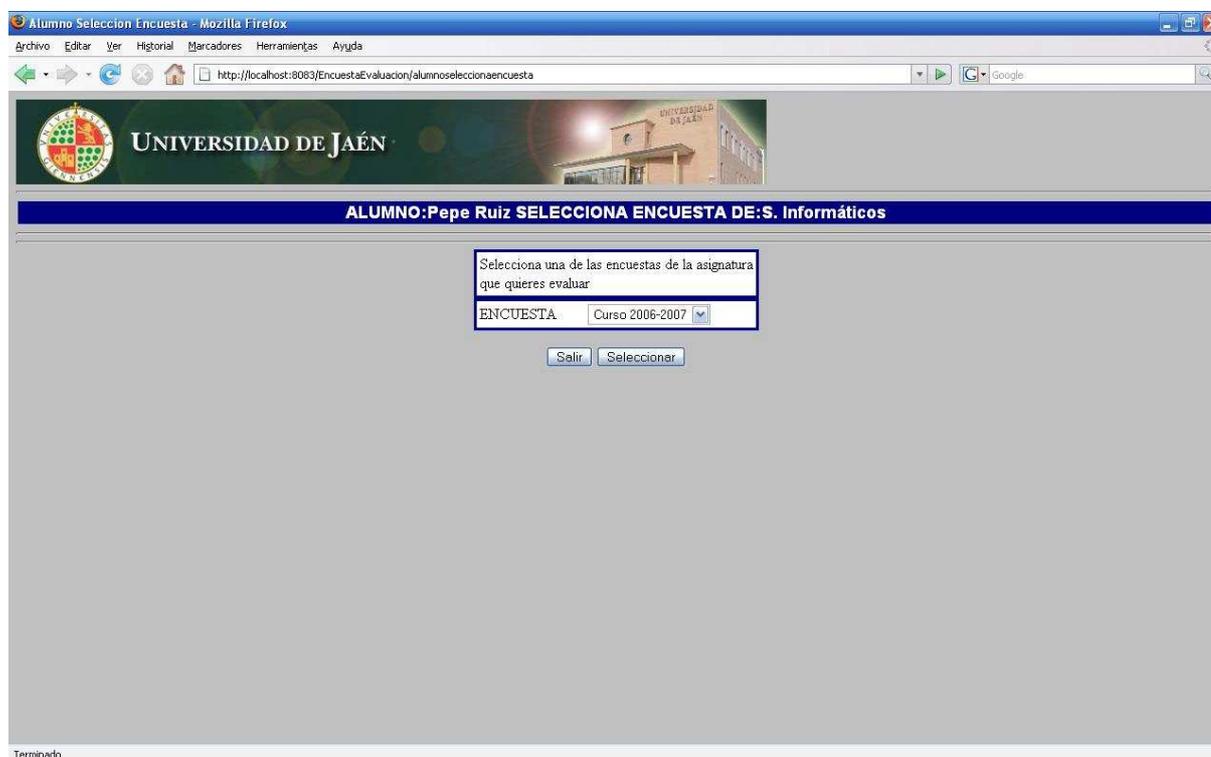


Figura IV.5 Selección de encuesta

La selección de la encuesta a rellenar se realiza de la misma forma que ha seleccionado anteriormente la asignatura, esto es, elige de la lista desplegable la encuesta a rellenar y pulsa el botón Seleccionar.

Si todo ocurre de forma correcta debe aparecer una pantalla similar a la figura IV.6.

Esta pantalla variará según el tipo de la primera pregunta, pero siempre existirá una parte que común que aparece independientemente del tipo de la pregunta.

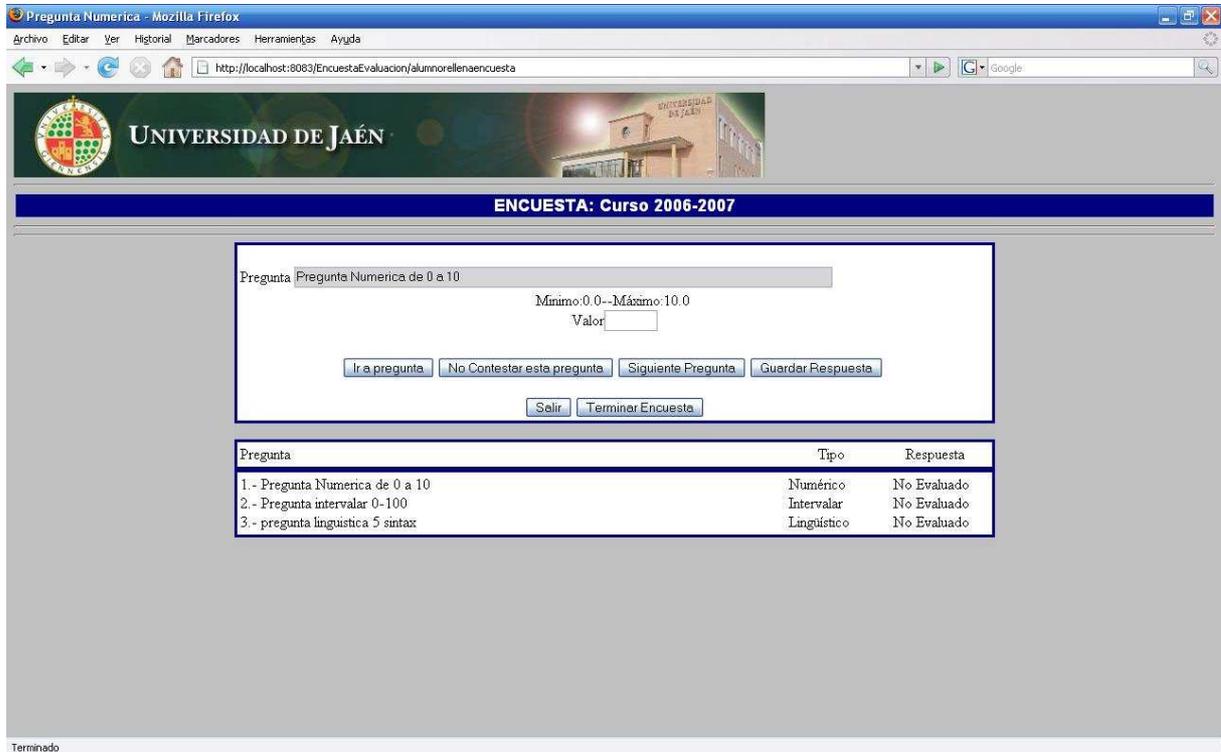


Figura IV.6 Rellenar encuesta

Esta parte común está formada por dos filas de botones y una tabla con las preguntas de la encuesta y las respuestas de cada una de ellas realizada hasta ahora.

La primera fila de botones contiene los siguientes:

- Ir a pregunta
- No contestar esta pregunta
- Siguiente pregunta
- Guardar Respuesta

- Ir a pregunta

Si pulsa este botón aparece una pantalla similar a la figura IV.7, en la que debe introducir el número de pregunta a la que quiere desplazarse. Pulsando el botón Ir a Pregunta, el sistema mostrará la pregunta que ha indicado.

Si por error introduce un número de pregunta inexistente o una letra el sistema indica el error y permite volver a introducir un número de pregunta válido.

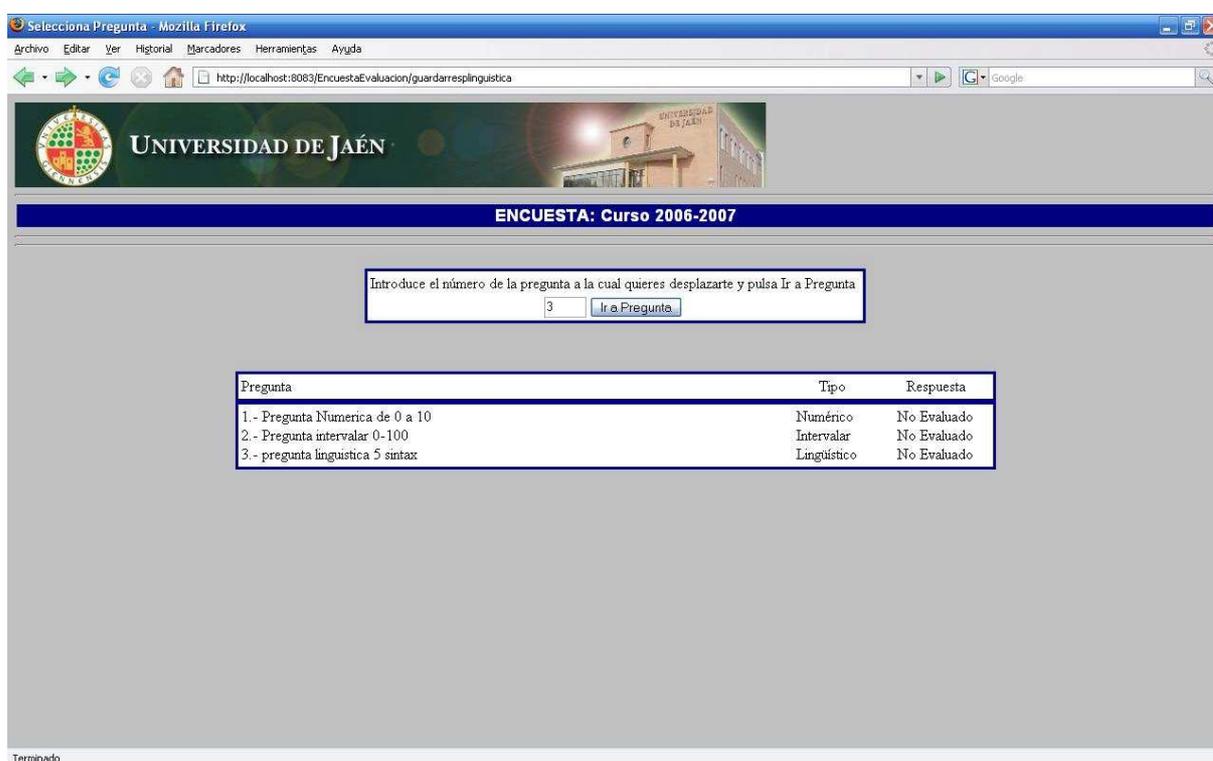


Figura IV.7 Ir a Pregunta

- No contestar esta pregunta

Si pulsa este botón la respuesta a la pregunta, en la que actualmente se encuentra, será No Evaluado. Si la pregunta tiene una respuesta, ésta será eliminada siendo su nuevo valor No Evaluado (Ns/Nc).

- Siguiente Pregunta

Este botón permite pasar la pregunta actual a la siguiente sin realizar ningún tipo de cambio en la respuesta actual.

- Guardar Respuesta

Para guardar una respuesta es necesario rellenar o elegir un valor del campo Valor. Según el tipo de pregunta, debe realizar una u otra acción.

- o Pregunta Numérica. Si la pregunta es de este tipo (figura IV.8) debe introducir, en la casilla de valor, un número real comprendido entre el mínimo y el máximo que aparecen justo encima de esta casilla y pulsar el botón Guardar Respuesta. Al guardar esta respuesta puede comprobar que el valor introducido aparece en la tabla de respuestas.

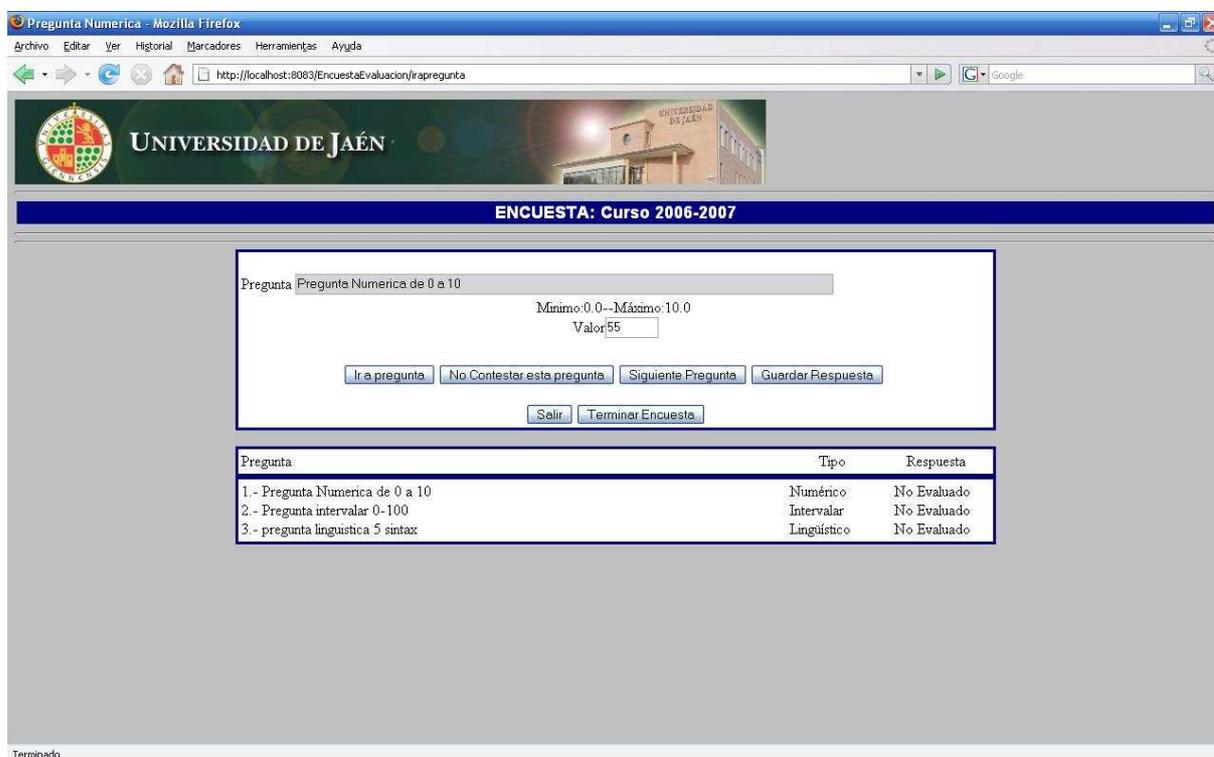


Figura IV.8 Pregunta Numérica

Si introduce un número erróneo o no introduce ningún valor, aparece un mensaje como el de la figura IV.9, indicando el error, permitiéndole insertar de nuevo un número correcto.



Figura IV.9 Error en Pregunta Numérica

- **Pregunta Intervalar.** Si la pregunta es de este tipo (figura IV.10) debe introducir dos valores reales comprendidos entre el intervalo de los valores mínimo y máximo. Como es lógico el valor mínimo debe ser menor que el máximo. Al pulsar Guardar Respuesta, si el intervalo es correcto, puede comprobar que el valor introducido aparece en la tabla de respuestas.

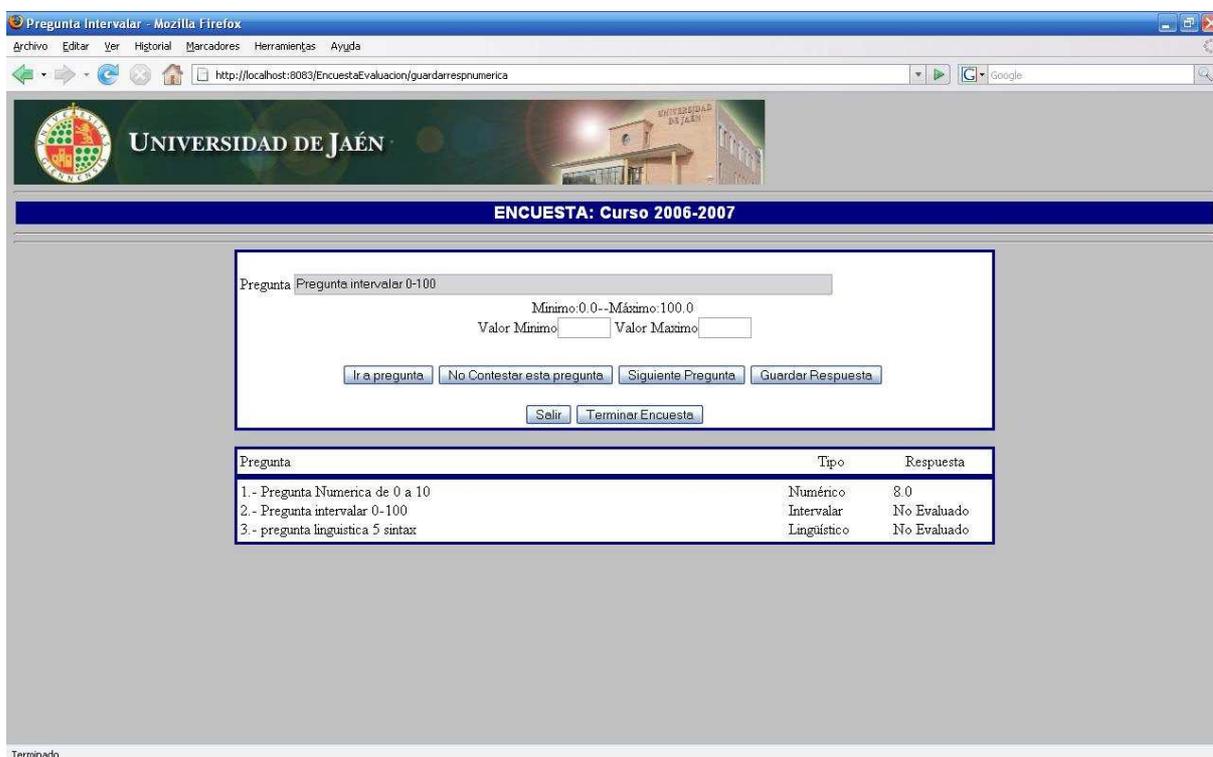


Figura IV.10 Pregunta Intervalar

En caso de introducir un intervalo incorrecto, pueden aparecer diversos errores:



Figura IV.11 Error en valor mínimo



Figura IV.12 Error en valor máximo



Figura IV.13 Valor mínimo introducido mayor que valor máximo

- Pregunta Lingüística. Si la pregunta es de este tipo (figura IV.14) debe seleccionar un valor de la lista de valores que aparece y pulsar Guardar Respuesta. Puede comprobar que el valor seleccionado aparece en la tabla de respuestas.

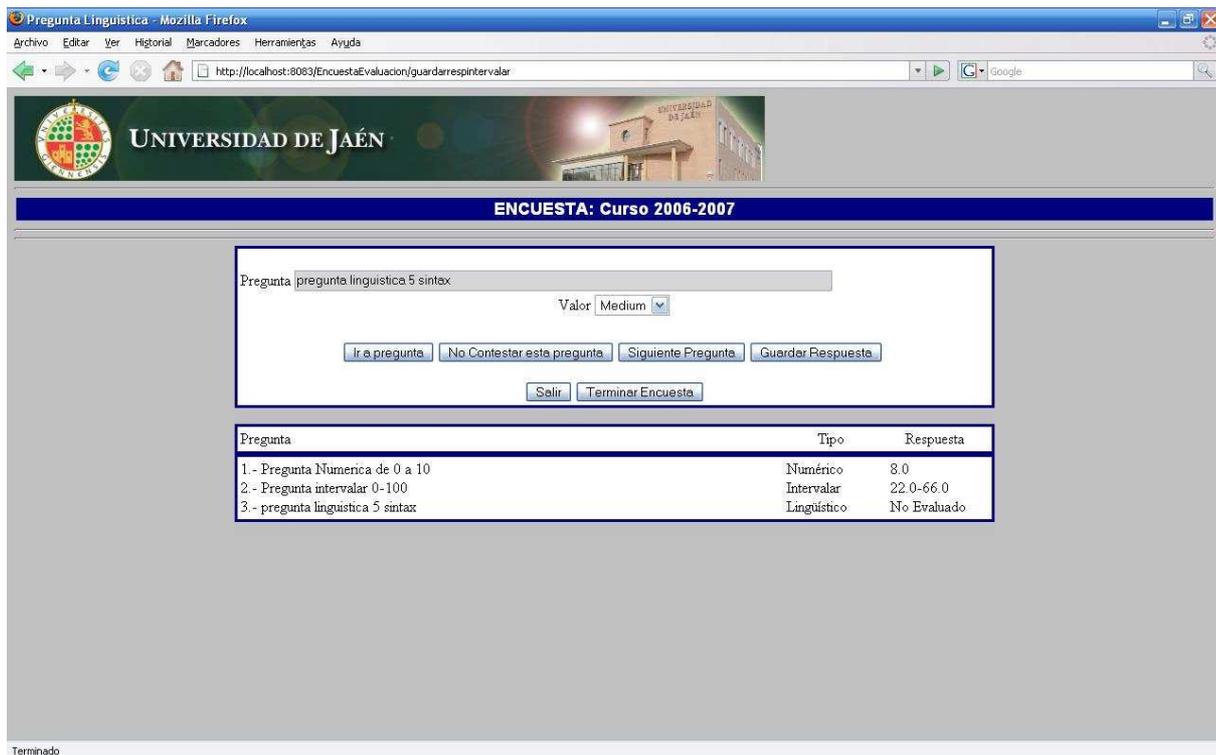
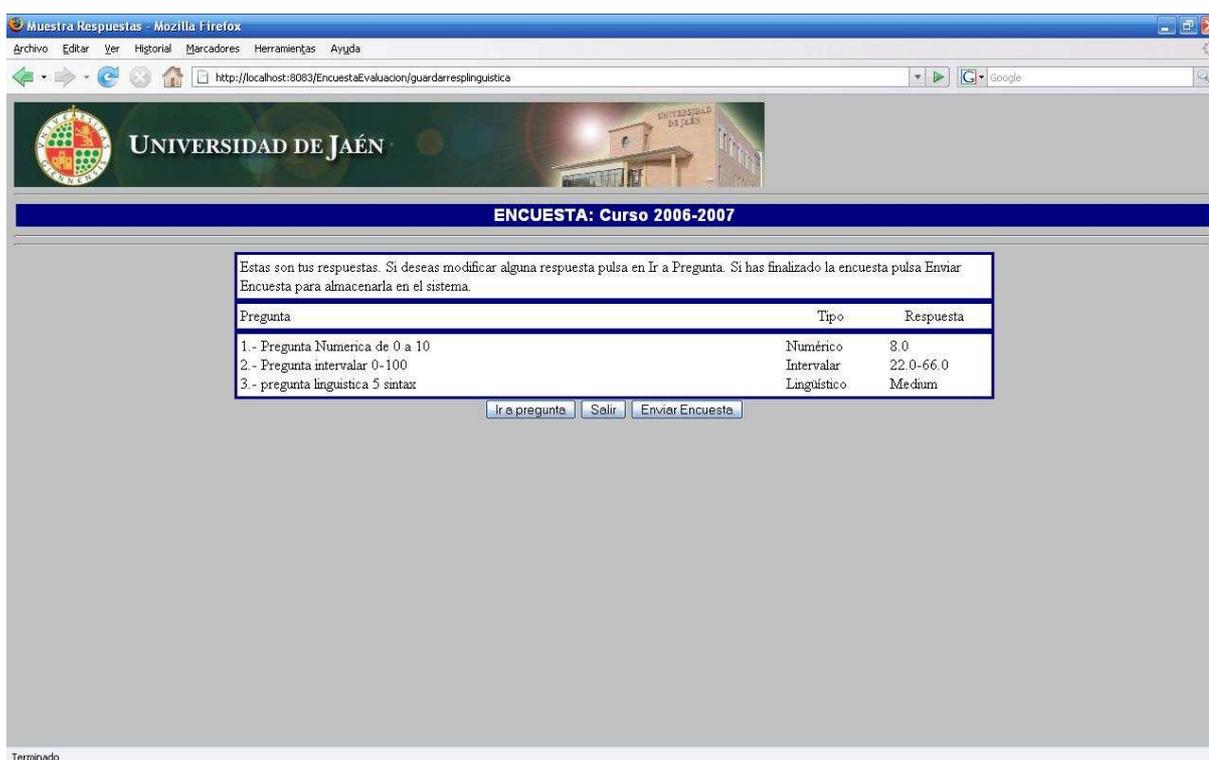


Figura IV.14 Pregunta Lingüística

En este tipo de pregunta no tiene cabida un error al contestar, puesto que no hay que introducir un valor, sino seleccionarlo de una lista.

Pasamos a la segunda fila de botones, formada por dos botones, Salir y Terminar Encuesta.

- Salir: abandona la aplicación. La encuesta no será enviada al sistema y le aparecerá como no contestada cuando inicie de nuevo la aplicación.
- Terminar Encuesta: si pulsa este botón aparecerá una pantalla resumen, figura IV.15, con las respuestas a las preguntas que ha ido contestando y tres botones en la parte inferior.



Muestra Respuestas - Mozilla Firefox

http://localhost:8083/EncuestaEvaluacion/guardarresplinguistica

UNIVERSIDAD DE JAÉN

ENCUESTA: Curso 2006-2007

Estas son tus respuestas. Si deseas modificar alguna respuesta pulsa en Ir a Pregunta. Si has finalizado la encuesta pulsa Enviar Encuesta para almacenarla en el sistema.

Pregunta	Tipo	Respuesta
1.- Pregunta Numerica de 0 a 10	Númérico	8.0
2.- Pregunta intervalar 0-100	Intervalar	22.0-66.0
3.- pregunta linguistica 5 syntax	Linguístico	Medium

[Ir a pregunta](#) [Salir](#) [Enviar Encuesta](#)

Terminado

Figura IV.15 Resumen de Encuesta

Estos tres botones tienen la siguiente funcionalidad:

- Ir a Pregunta: tiene el mismo funcionamiento que el explicado anteriormente (Figura IV.7)
- Salir: abandona la aplicación. La encuesta no será enviada al sistema y le aparecerá como no contestada cuando inicie de nuevo la aplicación.
- Enviar Encuesta: si todo es correcto, figura IV.16, sus respuestas serán almacenadas en el sistema de forma anónima.

La aplicación le da la opción de contestar otra encuesta o salir del sistema.

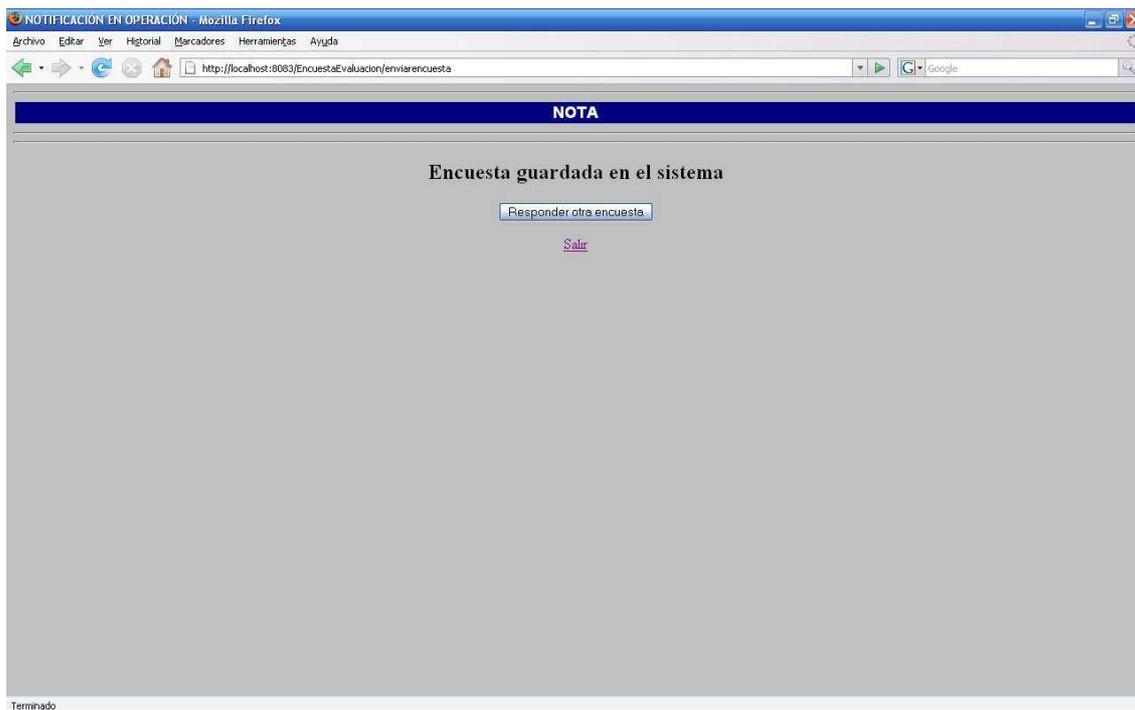


Figura IV.16 Encuesta Enviada

ANEXO V

Manual de Usuario: Profesor

Este manual de usuario está organizado como una visita guiada por la aplicación, en su apartado de profesor. Antes de embarcarse en ella es conveniente que el usuario tenga claro lo siguiente:

- Ésta es una aplicación web optimizada para su visualización en un navegador Firefox (a ser posible su versión más reciente, la cual se puede descargar en <http://www.mozilla-europe.org/es/>) y con una resolución no inferior a 800x600 pixels. Si se utiliza otro navegador o una resolución inferior a la recomendada se pueden producir fallos de visualización aunque la funcionalidad de la aplicación esta completamente asegurada.

Una vez aclarado esto puede empezar la visita guiada.

El primer paso para utilizar la aplicación es abrir el Navegador (Firefox) y teclear la dirección en la que tenemos alojada esta, que en nuestro caso será:
<http://localhost:8083/EncuestaEvaluacion/index.jsp>

Al realizar lo anteriormente comentado, nos encontramos con la página de inicio, tal y como se ve en la figura V.1



Figura V.1: Inicio de la Aplicación

Como es Profesor, debe hacer clic en el botón de Conexión como Profesor, accediendo a la pantalla de identificación, figura V.2

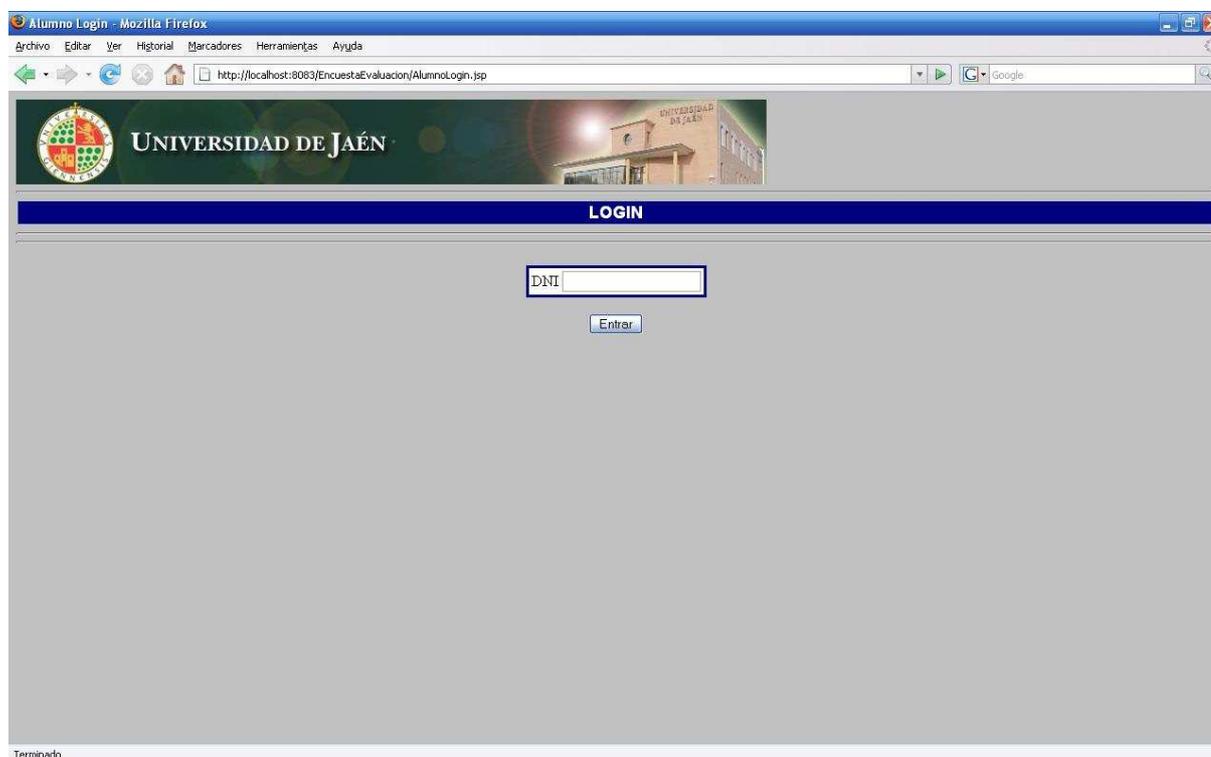


Figura V.2 Identificación de Profesor

Debe introducir su DNI y pulsar Entrar. Si el DNI introducido no es correcto el sistema nos muestra la pantalla de la figura V.3, indicando que el DNI introducido no se encuentra en la base de datos y le invita a una nueva identificación.

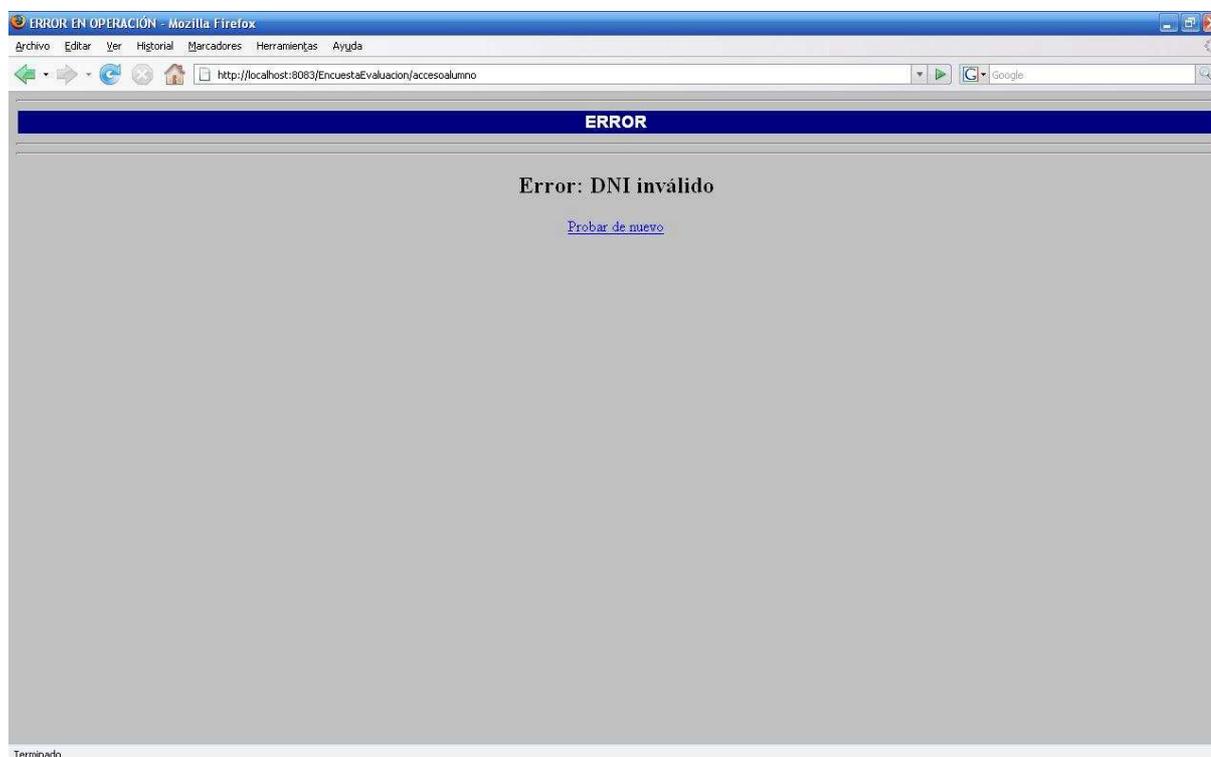


Figura IV.3 Identificación de Profesor incorrecta

En el caso de una identificación correcta aparece una pantalla similar a la figura V.4. En esta pantalla podemos ver que en la cabecera aparece su nombre.

En esta pantalla inicial aparecen dos opciones:

- Seleccionar una asignatura: debe elegir de la lista, la asignatura de la que quiere consultar los resultados
- Salir del sistema, pulsando el botón Salir.

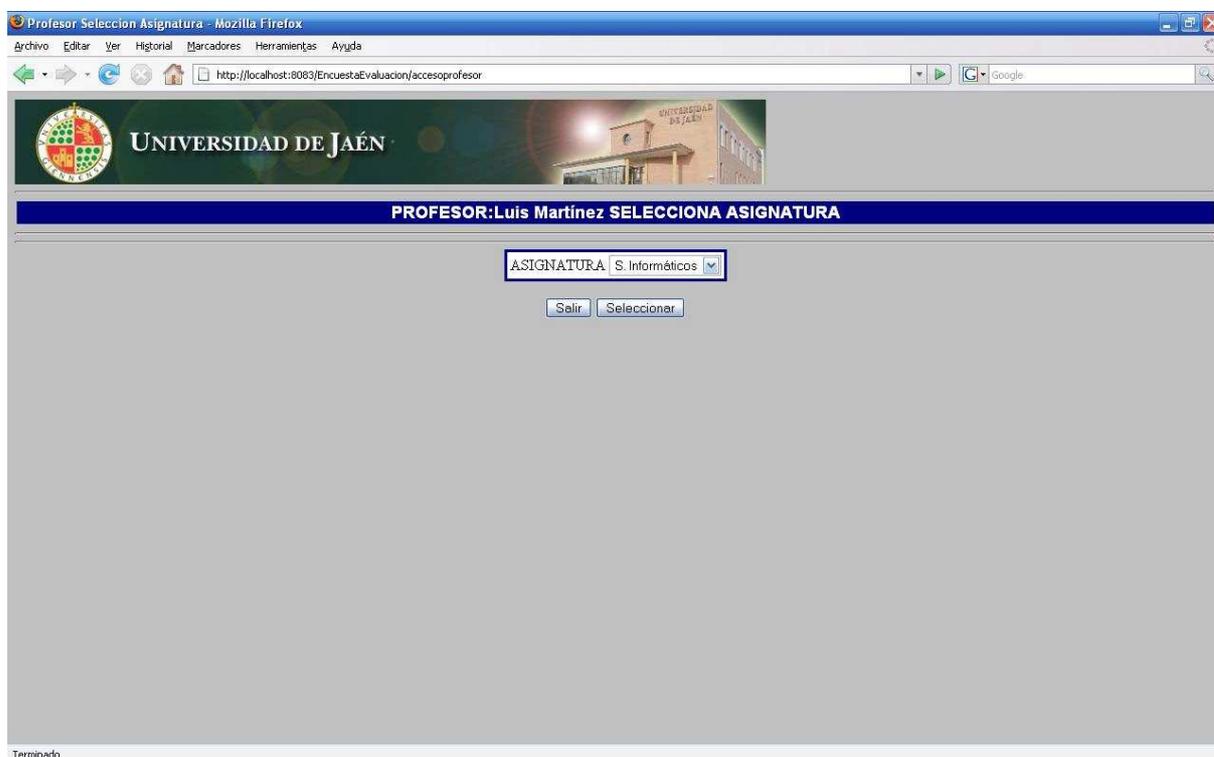


Figura V.4 Menú inicial de Profesor

Si ha seleccionado una asignatura y pulsado el botón Seleccionar, le aparecerá una pantalla como la de la figura V.5, en la que puede seleccionar las encuestas que ya se encuentran cerradas y que por tanto tienen los resultados calculados.

Si pulsa el botón Salir, abandonará el sistema.

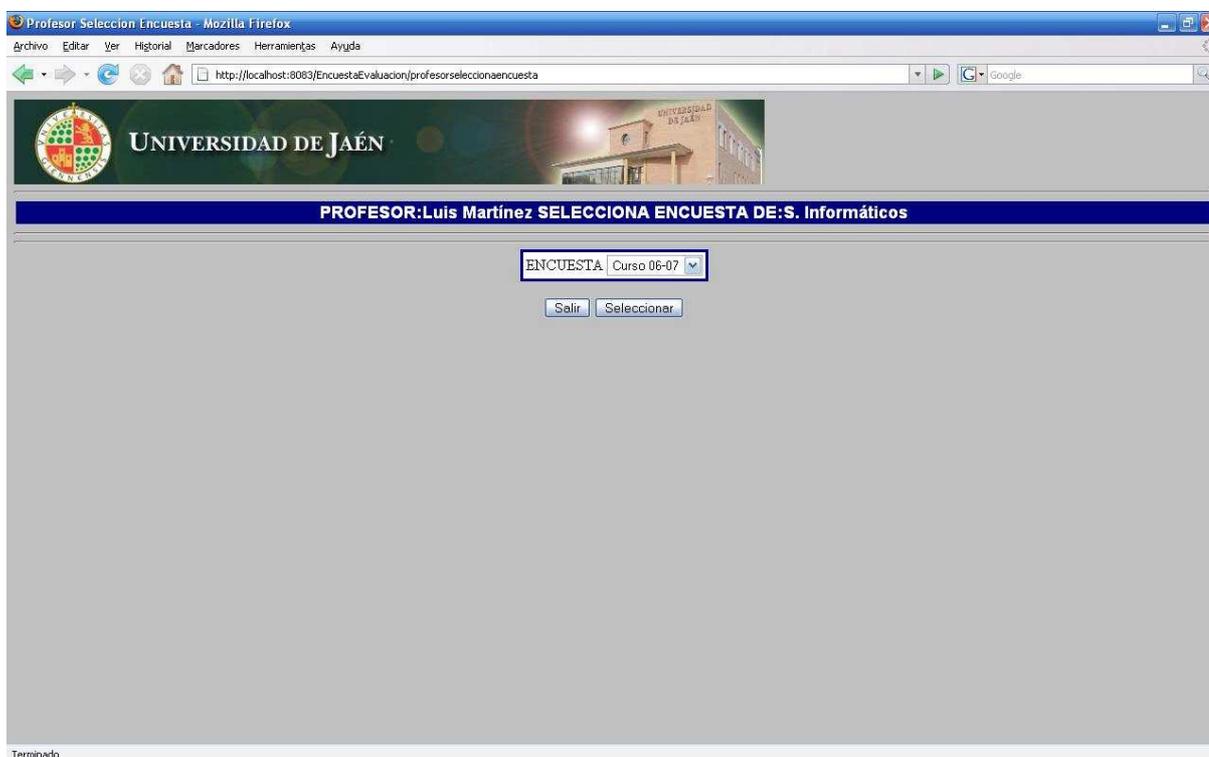


Figura V.5 Selección de encuesta

La selección de la encuesta a consultar se realiza de la misma forma que ha seleccionado anteriormente la asignatura, esto es, elige de la lista desplegable la encuesta a que quiere consultar y pulsa el botón Seleccionar.

Si todo ocurre de forma correcta debe aparecer una pantalla similar a la figura V.6. En la parte superior puede ver el nombre de la encuesta consultada y la asignatura a la que pertenece.

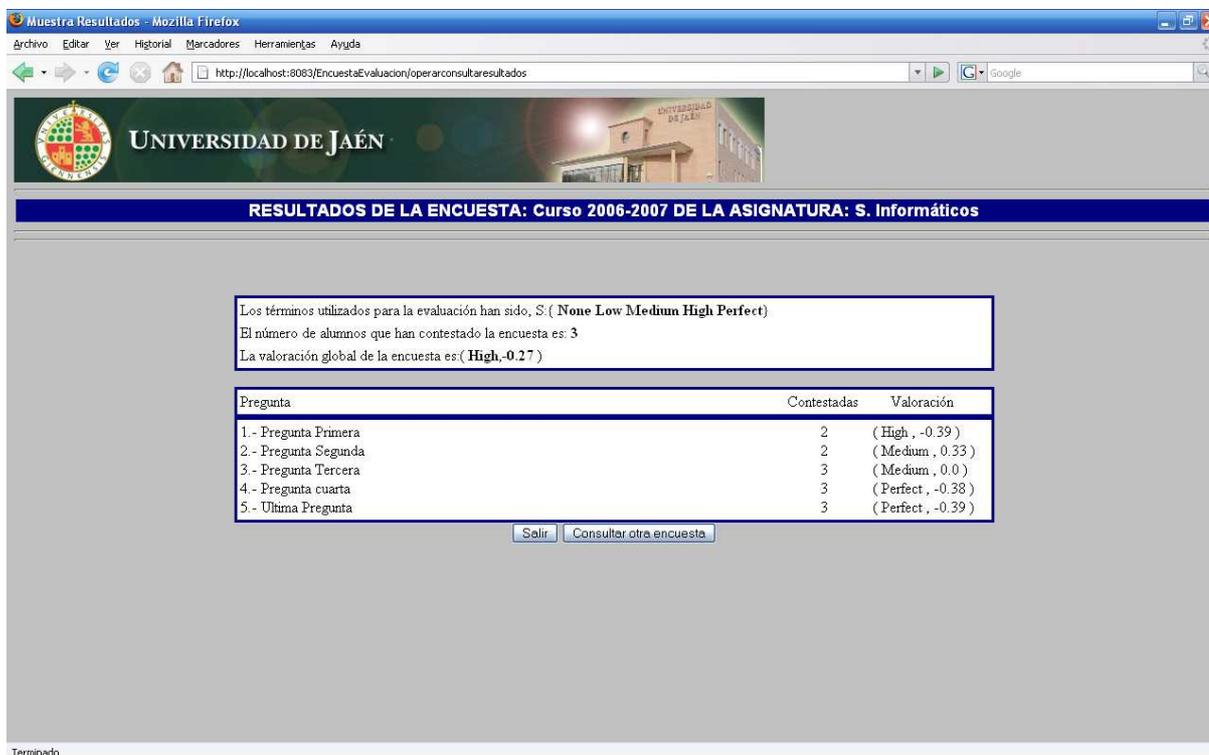


Figura IV.6 Resultados encuesta

En la primera tabla puede observar los resultados globales de la encuesta:

- Términos utilizados para la evaluación.
- Número de alumnos que han contestado esta encuesta.
- Valoración global de la encuesta.

En la segunda tabla puede ver los resultados obtenidos por pregunta.

Utilizando los botones inferiores puede salir del sistema (botón Salir) o volver al menú inicial para seleccionar una nueva consulta (Consultar otra encuesta).