



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

**APLICACIÓN WEB USANDO
STRUTS, HIBERNATE Y AJAX PARA
LA GESTIÓN INTEGRAL DE
ALUMNOS, PROFESORES Y
ASIGNATURAS**

**Realizado por
Noel Guillén Montilla, Fernando López Iglesias**

**Para la obtención del título de
INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN**

**Dirigido por
David Benavides Cuevas**

**Realizado en el departamento de
LENGUAJES Y SISTEMAS INFORMÁTICOS**

Convocatoria de Febrero, Curso 2006/2007

PARTE I: INTRODUCCIÓN	5
1 Introducción	6
1.1 Motivación	7
1.2 Objetivos del proyecto	8
1.2.1 Conocimientos	8
1.2.2 Formación	9
1.3 Desarrollo del proyecto	9
1.4 Estructura del documento	13
PARTE II: MATERIAS RELACIONADAS	16
2. Conceptos	17
3. Software	19
4. Herramientas	23
PARTE III: SISTEMA DESARROLLADO	27
5 Dominio del problema	28
5.1 Introducción	28
5.2 Glosario de términos	29
6 Visión general del sistema	30
6.1 Participantes en el proyecto	30
6.2 Objetivos del sistema	30
7 Requisitos del sistema	33
7.1 Requisitos de información	33
7.2 Reglas de negocio y restricciones	36
7.3 Requisitos funcionales	37
7.3.1 Diagramas de casos de uso	37
7.3.1.1 Subsistema gestión de tutorías y peticiones de cita	37
7.3.1.2 Subsistema gestión de usuarios y perfiles	38
7.3.1.3 Subsistema gestión de información de las asignaturas	39
7.3.1.4 Subsistema gestión de información de alumnos y profesores	39
7.3.1.5 Subsistema gestión administrativa vía Web	40
7.3.2 Definición de actores	41
7.3.3 Casos de uso del sistema	42
7.3.3.1 Subsistema gestión de tutorías y peticiones de cita	42
7.3.3.2 Subsistema gestión de usuarios y perfiles	48
7.3.3.3 Subsistema gestión de información de las asignaturas	51
7.3.3.4 Subsistema gestión de información de alumnos y profesores	54
7.3.3.5 Subsistema gestión administrativa vía Web	58
7.3.4 Otros requisitos funcionales	67
7.5 Requisitos no funcionales	68
7.6 Matriz de restreabilidad objetivos/requisitos	70
8. Análisis del sistema	72
8.1 Modelo estático del sistema	72
8.1.1 Diagramas	72
8.1.1.1 Subsistema gestión de usuarios y perfiles	72
8.1.1.2 Subsistema gestión de información de las asignaturas	73

8.1.1.3 Subsistema gestión administrativa vía Web	74
8.1.1.4 Subsistema gestión de información alumnos y profesores	75
8.1.1.5 Subsistema gestión de tutorías y peticiones de cita	76
8.1.2 Tipos	76
8.1.2.1 Subsistema gestión de usuarios y perfiles	76
8.1.2.2 Subsistema gestión de información de las asignaturas	78
8.1.2.3 Subsistema gestión de información alumnos y profesores	84
8.1.2.4 Subsistema gestión de tutorías y peticiones de cita	89
8.2 Modelo dinámico del sistema	92
8.2.1 Subsistema gestión de información alumnos y profesores	92
8.2.2 Subsistema gestión de información de las asignaturas	96
8.2.3 Subsistema gestión administrativa vía Web	99
8.2.4 Subsistema gestión de usuarios y perfiles	107
8.2.5 Subsistema gestión de tutorías y peticiones de cita	108
8.3 Modelo funcional del sistema	113
8.3.1 Subsistema gestión de información alumnos y profesores	113
8.3.2 Subsistema gestión de información de las asignaturas	117
8.3.3 Subsistema gestión administrativa vía Web	119
8.3.4 Subsistema gestión de usuarios y perfiles	125
8.3.5 Subsistema gestión de tutorías y peticiones de cita	127
8.3 Diseño del sistema	131
8.3.1 Patrones usados	131
8.3.2 Diseño de la aplicación	138
8.3.2.1 Capa de datos	139
8.3.2.2 Capa de aplicación	141
8.3.2 Diseño de la base de datos	148
8.4 Puntos de variación y evolución	153
8.4.1 Puntos de variación	153
8.4.2 Puntos de evolución	154
9 Arquitectura del sistema	157
10 Implementación del sistema	161
10.1 Entorno de desarrollo	161
10.1.1 Instalación del entorno	161
10.1.2 Configuración del repositorio RedIris	162
10.2 Componentes más relevantes	162
11 Pruebas del sistema	171
11.1 Pruebas de Interfaces y Contenidos	171
10.2 Pruebas de Funcionalidades y Operación	174
10.3 Pruebas de carga	176
12 Instalación del sistema	179
12.1 Requisitos	179
12.2 Pasos de la instalación	179
13 Manual de uso del sistema	181
13.1 Manual de funcionamiento	181
13.2 Manual de alumno	182
13.2.1 Consultar su ficha personal	182
13.2.2 Modificar su ficha personal	182
13.2.4 Solicitar una cita	184
13.2.5 Consultar el estado de sus citas	185
13.2.6 Ver asignaturas	185
13.3 Manual del profesor	186
13.3.1 Consultar su horario de tutorías.	186
13.3.2 Modificar su horario de tutorías.	187
13.3.3 Desactivar las tutorías	187
13.3.4 Consultar las fichas de los alumnos	188

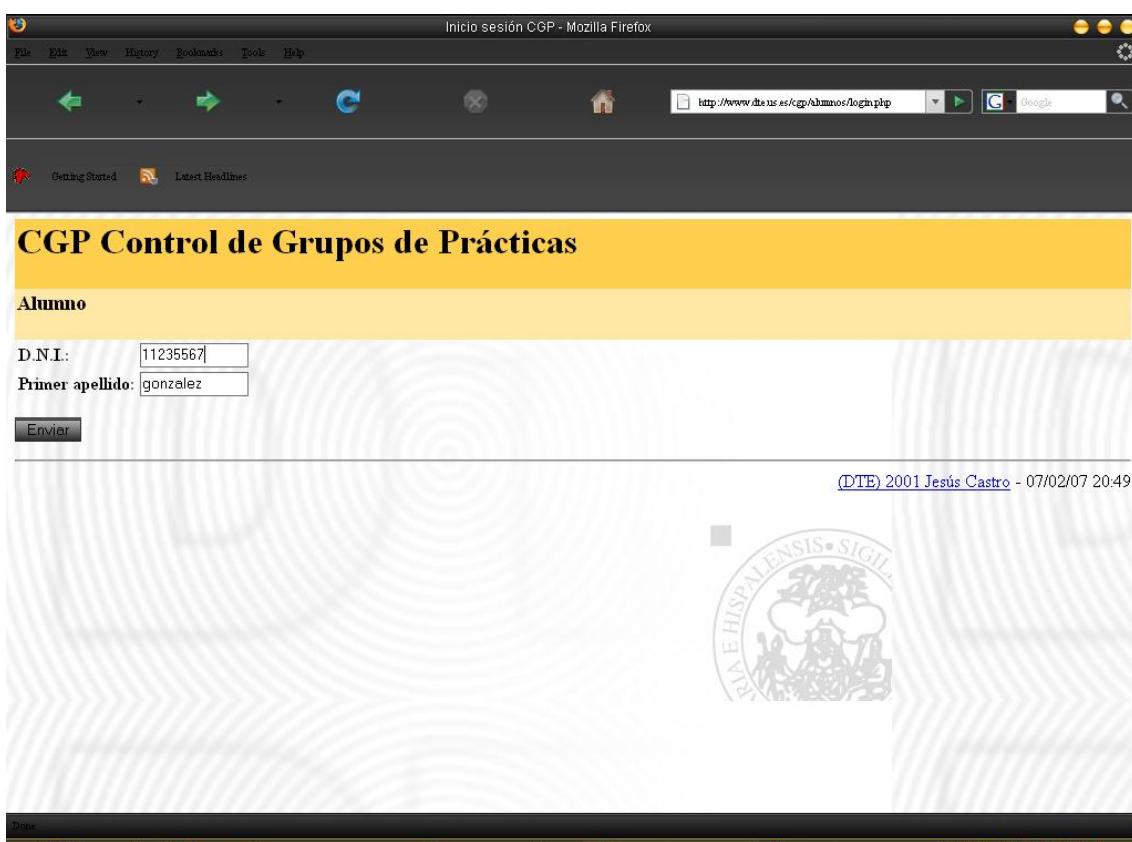
13.3.5 Cancelar citas	188
13.3.6 Responder peticiones de citas	188
13.3.7 Consultar su calendario de citas.	189
13.3.9 Modificar su información personal	189
13.3.10 Introducir noticias	190
13.3.11 Modificar noticias	190
13.3.12 Consultar noticias	191
13.4 Manual del administrador	192
13.4.1 Añadir Titulaciones.	193
13.4.2 Eliminar Titulaciones	193
13.4.3 Añadir Asignaturas	193
13.4.4 Eliminar Asignaturas.	194
13.4.5 Añadir Usuario	195
13.4.6 Eliminar Usuario	195
13.4.7 Vincular una asignatura a una titulación	195
13.4.8 Desvincular una asignatura a una titulación	196
13.4.9 Vincular un usuario a una asignatura	196
13.4.10 Desvincular un usuario de una asignatura	197
13.4.11 Importar una lista de usuarios	197
13.4.12 Importar una lista de asignaturas	200
13.4.13 Vaciar Lista asignatura	202
13.4.13 Añadir Departamento	202
13.4.13 Modificar Departamento	203
PARTE IV: CONCLUSIONES	204
14 Cumplimiento de los objetivos del proyecto	205
15 Esfuerzo empleado en el proyecto	206
15.1 Planificación	206
15.1 Estimación del coste del proyecto	206
16 Bibliografía y referencias	211

PARTE I: INTRODUCCIÓN

1 Introducción

Actualmente, el entorno de la enseñanza esta sufriendo grandes trasformaciones con el fin de automatizar numerosos procesos que se llevan a cabo en los distintos centros educativos. Esta transformación se esta desarrollando en los distintos niveles educativos de nuestra sociedad y esta permitiendo tanto a empleados como a usuarios interactuar de forma mas cómoda, sencilla y eficiente. Así, actualmente es posible realizar la matricula de la universidad a través de Internet o seleccionar un grupo de practicas para una determinada asignatura.

Pero, aunque actualmente la informatización en el mundo de la enseñanza ha avanzado considerablemente, existen aun muchos procedimientos que se llevan a cabo de forma manual con la consiguiente repercusión para los usuarios. Así, actualmente la información entre alumnos y profesores se intercambia de forma manual a partir de fichas personales en formato papel. Con el fin de informatizar este proceso, nació nuestro proyecto inicial “Gestión de Fichas” que permite intercambiar a profesores y alumnos su información personal a partir de un portal Web. Posteriormente se reconsidero la idea inicial y se replanteo abarcando más objetivos, tales como la gestión de las noticias de las asignaturas y las citas entre profesores y alumnos.



Ejemplo: Grupo de Practicas

GAVILAN RAMIREZ, JOSE

Tecnología Básica de las Comunicaciones: Grupo 3; Cerrado!

Grupos de 16 alumnos (2 por puesto)

Gr	Li	Práctica 1	Práctica 2	Práctica 3	Práctica 4
G1	0	J-26-Oct (M)	J-09-Nov (M)	J-23-Nov (M)	J-14-Dic (M)
G2	0	J-26-Oct (M)	J-09-Nov (M)	J-23-Nov (M)	J-14-Dic (M)
G3	2	V-27-Oct (M)	V-10-Nov (M)	V-24-Nov (M)	V-15-Dic (M)
G4	3	V-27-Oct (M)	V-10-Nov (M)	V-24-Nov (M)	V-15-Dic (M)
G5	3	V-03-Nov (M)	V-17-Nov (M)	V-01-Dic (M)	J-11-Ene (M)
G6	16	V-03-Nov (M)	V-17-Nov (M)	V-01-Dic (M)	J-11-Ene (M)

Leyenda: (M) = Mañana, (T) = Tarde, Gr = Grupo, Li = Plazas libres

Fecha límite de inscripción: 28/10/2006

Web de la asignatura: [Tecnología Básica de las Comunicaciones](#)

Práctica 1 - Sistema de Comunicaciones
 Práctica 2 - Comunicación Asíncrona
 Práctica 3 - Filtros
 Práctica 4 - Comunicaciones Digitales

Ejemplo: Grupo de practicas

1.1 Motivación

El principal objetivo de nuestra aplicación es facilitar la tarea tanto a alumnos, como a profesores. Actualmente, el intercambio de información entre profesores y alumnos se produce de manera manual. Esto conlleva las siguientes consecuencias:

- Los alumnos deben de acudir a recoger las fichas personalmente y a su posterior entrega.
- En caso de error en la ficha personal, el profesor debe contactar con el alumno para que realice las modificaciones oportunas, volviendo a entregar una nueva ficha personal.
- Los profesores son los encargados de mantener las fichas personales ordenadas y en ocasiones las búsquedas pueden resultar costosas.
- El transporte físico de las fichas también conlleva numerosos inconvenientes.

Por otra parte, actualmente las peticiones de tutorías entre alumnos y profesores se realizan a través de correos electrónicos o mediante una petición directa con el profesor en cuestión. Esto conlleva también numerosos inconvenientes:

- El profesor es el encargado de gestionar y actualizar sus horas de autorías.

- Generalmente, los alumnos desconocen las horas libres y ocupadas que posee un profesor dentro del horario de autoría para un determinado día.
- El envío de email puede ocasionar perdidas de peticiones o respuestas.

Nuestra aplicación se encarga de solucionar todos estos problemas que surgen en el sistema actual de nuestra universidad.

Así, la gestión de fichas y tutorías a partir de nuestro portal Web permitirá a los alumnos:

- Crear su ficha personal en cualquier momento desde cualquier lugar a través de una conexión a Internet.
- Modificar su ficha personal de manera rápida y sencilla.
- Consultar los horarios de tutoría, completamente actualizados.
- Comprobar las horas de tutoría libres para un día determinado.
- Solicitar tutorías a través del portal.
- Consultar el estado de las peticiones de tutorías realizadas.

Por otro lado, los profesores, a partir de nuestro portal podrán:

- Consultar las fichas de todos los alumnos que cursan sus asignaturas de forma ordenada y rápida.
- Consultar las fichas personales de los alumnos en cualquier momento, desde cualquier lugar, sin la necesidad de transportar ningún tipo de documentos.
- Responder a todas las peticiones recibidas de manera ordenada, cómoda y rápida.
- Consultar sus tutorías actuales aceptadas y cancelarlas en el caso de que surgiera una emergencia

1.2 Objetivos del proyecto

El principal objetivo del proyecto es diseñar un portal Web consistente que permita gestionar las fichas personales de los alumnos y las tutorías de manera eficiente.

Para el desarrollo del proyecto debemos tener en cuenta dos aspectos fundamentales: conocimientos y formación.

1.2.1 Conocimientos

El proyecto debe desarrollarse a partir de los conocimientos adquiridos a lo largo de la carrera. Debemos aplicar estos conocimientos tanto en el aspecto más técnico, como a la hora de enfrentarnos a los problemas que vayan surgiendo. Así, en nuestro proyecto hemos intentado reflejar y aplicar el mayor número de conocimientos adquiridos. Ejemplo de tales aplicaciones son el uso de patrones a la hora de desarrollar (Singleton, DAO...), el desarrollo correcto de la documentación...

1.2.2 Formación

En nuestro proyecto hemos querido dar un paso más y no utilizar solo los conocimientos adquiridos a lo largo de la carrera. Hemos pretendido complementar nuestra formación, haciendo uso de tecnologías más actuales con el fin de obtener un proyecto de mayor consideración. Así, en nuestro proyecto han intervenido tecnologías tales como STRUTS, HIBERNATE o AJAX. La integración de estas tecnologías en nuestro proyecto se ha llevado a cabo en distintas fases.

1. Recogida de información: Hemos recogido información en forma de manuales, principalmente a través de Internet.
2. Estudio de la información: Hemos llevado acabo un profundo estudio de la información recogida con el fin de comprender mejor el funcionamiento de las distintas tecnologías.
3. Prácticas: Hemos llevado a cabo diferentes prácticas con distintos grados de dificultad con el fin de familiarizarnos con la tecnología y de conocer sus limitaciones
4. Integración: Hemos realizado un estudio para determinar como se integrarían las distintas tecnologías en nuestra idea de proyecto y como interaccionarían entre si.
5. Desarrollo: Finalmente, hemos realizado el proyecto mediante el uso de estas tecnologías y de los conocimientos adquiridos a lo largo de la carrera.

El proyecto debe presentarse como una prueba final que nos permita adquirir experiencia a la hora de enfrentarnos a proyectos en la vida real.

1.3 Desarrollo del proyecto

Este proyecto ha sido realizado por un grupo de dos estudiantes que mantienen domicilios muy separados entre sí. Este hecho ha influido considerablemente en el desarrollo del proyecto y en la forma de abarcar el mismo.

Pero pese a los inconvenientes que en un principio puedan surgir, mayores han sido las ventajas que hemos obtenido.

Como el proyecto ha sido realizado entre dos personas, hemos tenido que familiarizarnos con el trabajo en equipo. Así, hemos llevado acabo numerosos esfuerzos de coordinación y hemos abordado los problemas que nos han ido surgiendo desde diferentes puntos de vista, resolviendo nuestras discrepancias y llegando a soluciones consensuadas. En definitiva, hemos aprendido a trabajar en grupo, acercándonos a la forma de desarrollar un proyecto en la vida real.

Como herramientas de ayuda para esta constante coordinación en el desarrollo de nuestro proyecto, hemos utilizado:

Trac

En el hemos ido anotando las distintas tareas ha realizar. Cada vez que se finalizaba una tarea se accedía al track y se asignaba una nueva tarea. La asignación de tareas evitaba el solapamiento de trabajo, ya que cuando el otro componente accedía, comprobaba que la tarea ya se estaba realizando y pasaba a realizar una tarea diferente. El ir determinando las tareas pendientes, realizadas o en proceso nos ha permitido trabajar de forma individual sin que existan conflictos, no era necesario que el otro componente estuviese trabajando en el proyecto para poder llevar a cabo su desarrollo.

También destacar, que el Trac permite establecer prioridad a las tareas, con el fin de su correcta ejecución.

Capa de presentación											
Ticket	Summary	Component	Status	Resolution	Version	Type	Priority	Owner	Modified		
#30	Calendario	component1	closed	fixed		enhancement	minor	Noel	01/15/07		

Capa de lógica de negocio											
Ticket	Summary	Component	Status	Resolution	Version	Type	Priority	Owner	Modified		
#32	Buscador	component1	new	None		enhancement	minor	somebody	12/28/06		

Análisis											
Ticket	Summary	Component	Status	Resolution	Version	Type	Priority	Owner	Modified		
#18	Prototipo de Interfaz de Usuario	component1	new	None		task	major	somebody	10/01/06		
#29	Diagramas de clase gestión de usuarios y roles	component1	closed	fixed		task	major	Fernando	12/28/06		
#28	Diagramas de clase gestión de tutorías	component1	closed	fixed		task	major	Fernando	12/28/06		
#27	Diagramas de clase gestión de fichas	component1	closed	fixed		task	major	Noel	12/28/06		
#15	Modelo Estático del Sistema	component1	closed	fixed		task	major	Noel	12/28/06		
#16	Modelo Dinámico del Sistema	component1	closed	fixed		task	major	Fernando	12/28/06		
#31	Buscador	component1	new	None		enhancement	minor	somebody	12/28/06		

Note: See [TracReports](#) for help on using and creating reports.

Download in other formats:
[XML](#) | [RSS Feed](#) | [Comma-delimited Text](#) | [Tab-delimited Text](#) | [SQL Query](#)


Powered by Trac 0.10.1
By Edgewall Software.

Visit the Trac open source project at
<http://trac.edgewall.com/>

Trac

Este método nos ha permitido trabajar de manera coordinada y eficiente.

El Trac se ha utilizado principalmente en el control de las tareas de desarrollo de la documentación

Repositorio Web

En él, hemos almacenado el código de nuestra aplicación, así como la documentación realizada y la información necesaria para el desarrollo del mismo.

El repositorio Web ha presentado la siguiente estructura:

- Documentación: contiene la documentación desarrollada para el proyecto.
- Gestión: contiene la implementación del código de nuestra aplicación.

- Manuales: contiene manuales de consulta para Struts e hibernate, con el fin de facilitar la tarea de desarrollo.
- Base de Datos: Contiene información sobre la base de datos de nuestra aplicación.

Así atendiendo a las distintas modificaciones, accedíamos a partir de determinadas herramientas a las distintas ubicaciones de nuestro repositorio.

La principal ventaja que nos ha facilitado el repositorio es el transporte de información entre nosotros. El transporte manual era imposible debido a la distancia que nos separa y demasiado costoso e ineficiente.

El repositorio era la mejor herramienta para mantener la información actualizada y de forma coherente entre nosotros.

El acceso al repositorio se ha llevado a cabo a partir de determinadas herramientas que se explican a continuación.

Revision 499: /

- [Gestion/](#)
- [Proyecto.war](#)
- [base de datos/](#)
- [documentacion/](#)
- [documento/](#)
- [temporal/](#)
- [tutoriales/](#)

Powered by [Subversion](#) version 1.4.2 (r22196).

Repository

TortoiseSVN

Es la herramienta utilizada para acceder al repositorio enfocada al manejo de la documentación del proyecto. Nos permitía compartir la información entre nosotros y combinar nuestros archivos.

Además, ha sido la herramienta encargada de detectar los distintos conflictos que se han ido produciendo. Para resolver estos conflictos nos permite

mantener el control de versiones de todos los documentos indicando la fecha y persona que los ha modificado.

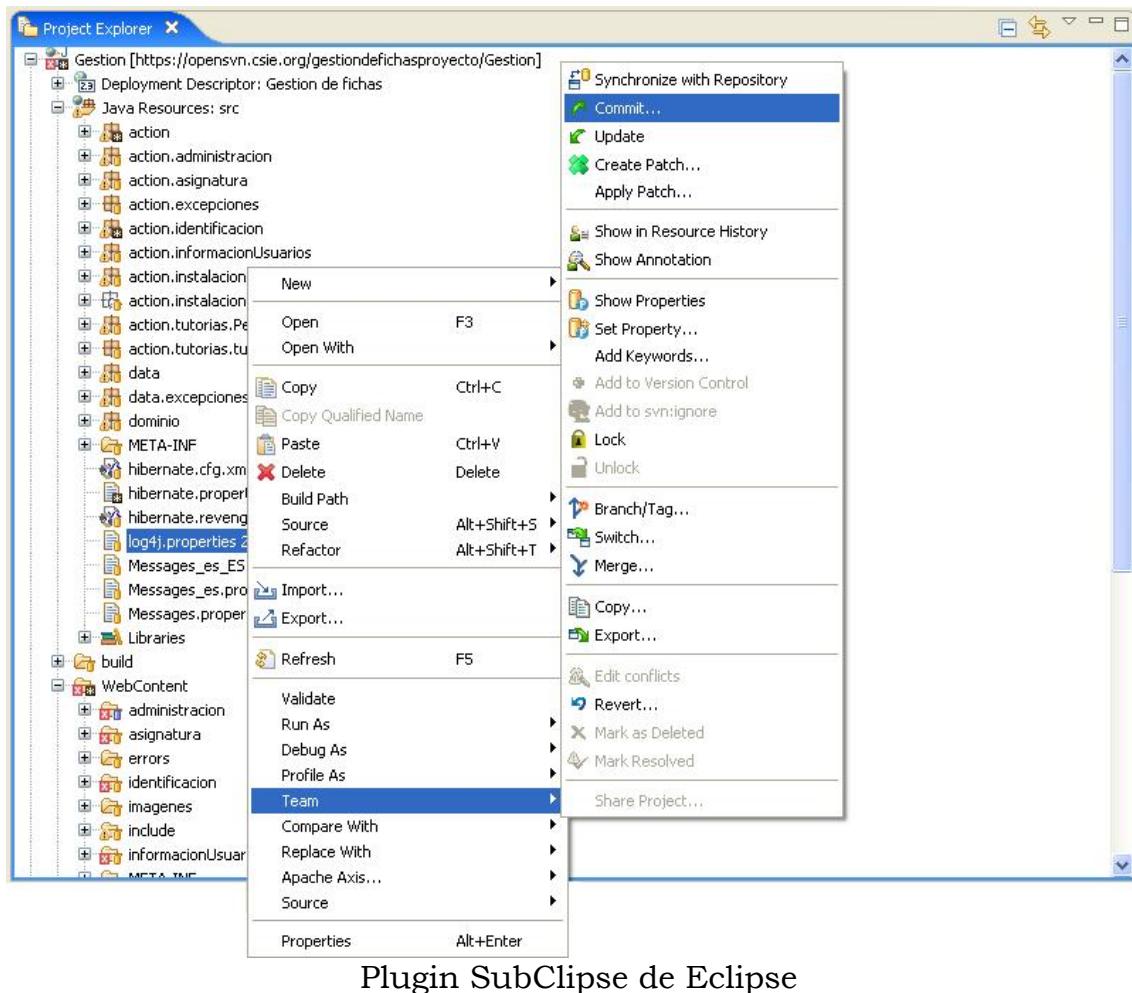


Plugin SubClipse de Eclipse

Herramienta encargada de acceder al repositorio, para obtener y contrastar el código de nuestra aplicación. El uso de esta herramienta ha sido fundamental ya que nos ha permitido reducir considerablemente el número de conflictos producidos. Destacar, que durante la implementación es cuando se producen el mayor numero de conflictos y que estos son los más difíciles de resolver. Pese que al reparto de tareas se ha realizado de la manera más independiente posible con el fin de permitir el trabajo en paralelo y de reducir los conflictos, existen archivos con alto riesgo de conflicto.

Estos archivos son: struts-config.xml, estilo.css, menú.jsp, Messages.properties, así como todos los Archivos Manager.

Para su correctas modificaciones ha sido indispensable está herramienta que nos ha permitido reducir los conflictos y actualizar los archivos de manera eficaz.



En el desarrollo de nuestro proyecto se pueden distinguir las siguientes fases:

1. Reunión para discutir el tema del proyecto.
2. Reunión para determinar los principales objetivos de nuestra aplicación y las tecnologías a desarrollar.
3. Búsqueda e intercambio de información de las tecnologías aplicadas al proyecto.
4. Reparto de las distintas tareas para la creación de la documentación del proyecto.
5. Reunión para determinar la arquitectura principal de nuestra aplicación.
6. Configuración del entorno de desarrollo, común para ambos.
7. Reparto de las distintas tareas para la implementación del proyecto.
8. Revisión completa del funcionamiento de la aplicación
9. Reparación de los errores encontrados.

1.4 Estructura del documento

El documento esta estructurado de la siguiente forma:

BLOQUE 1. Introducción

Es el bloque actual y contendrá los siguientes subapartados:

- 1.1 Introducción
- 1.2 Motivación
- 1.3 Objetivos del proyecto
- 1.4 Desarrollo del proyecto
- 1.5 Estructura del documento

BLOQUE 2. Materias relacionadas

En este bloque describiremos los conceptos necesarios para comprender mejor nuestra aplicación, así como las herramientas y software empleados para el desarrollo de nuestro proyecto. Contendrá los siguientes subapartados:

- 2 Conceptos
- 3 Software
- 4 Herramientas

BLOQUE 3. Sistema Desarrollado

Es el bloque más amplio, contara con los siguientes apartados:

- 5 Dominio del problema:
Consiste en una descripción del sistema actual.

- 5.1 Introducción al problema
- 5.2 Glosario de términos

- 6 Visión general del sistema:
En este apartado definiremos los objetivos de nuestra aplicación

- 6.1 Objetivos del sistema
- 6.2 Actores del sistema

- 7 Requisitos del sistema:
Aquí va Documento de Requisitos del Sistema.

- 7.1 Requisitos de información
- 7.2 Reglas de negocio/restricciones
- 7.3 Requisitos funcionales/casos de uso
- 7.4 Requisitos de interfaz con otros sistemas
- 7.5 Requisitos no funcionales
- 7.6 Rastreabilidad requisitos – objetivos

- 8 Análisis del sistema:
Aquí se mostrará el análisis realizado para desarrollar el sistema

- 9 Arquitectura del sistema:
En este apartado se explican los componentes con los que esta formando la aplicación y como se relacionan

- 10 Implementación del sistema:

Aspectos más relevantes de la implementación del sistema

- 10.1 Entorno de desarrollo
- 10.2 Componentes más relevantes

11 Pruebas del sistema

Distintas pruebas realizadas al sistema para evaluar su funcionamiento, y rendimiento

12 Instalación del sistema

Manual de instalación explicando como realizar la instalación

13 Manual de usuario del sistema

Guía de las distintas acciones que puede realizar un usuario

BLOQUE 4. Conclusiones

En este bloque se estudiaran si se han alcanzado los objetivos previstos y se determinara el esfuerzo empleado en el proyecto a partir de estimaciones de temporales y de coste.

- 14 Cumplimiento de los objetivos del proyecto
- 15 Esfuerzo empleado en el proyecto

BLOQUE 5. Bibliografía consultada

En este bloque se indicaran las fuentes de información que se han utilizado para llevar a cabo el proyecto.

PARTE II: MATERIAS RELACIONADAS

2. Conceptos

Framework

“Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Un framework esta orientado a facilitar el desarrollo de software, permitiendo a los diseñadores y programadores dedicar más tiempo en la lógica de negocio del sistema que en detalles tediosos de bajo nivel”.

Wikipedia – <http://en.wikipedia.org/wiki/Framework>

Patrón MVC.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

- *Modelo: Esta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos; por ejemplo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o portes en un carrito de la compra.*
- *Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.*
- *Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.*
- *Muchas aplicaciones utilizan un mecanismo de almacenamiento persistente (como puede ser una base de datos) para almacenar los datos. MVC no menciona específicamente esta capa de acceso a datos.*

Wikipedia – http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

ORM (Object to Relational Mapping)

Hay una diferencia entre el modelo de objetos y el de bases de datos relacionales. El modelo relacional trata con relaciones, y conjuntos es muy matemático por naturaleza. Sin embargo, el paradigma orientado a objetos trata con objetos, sus atributos y asociaciones de unos con otros. Cuando se quiere que los objetos sean persistentes se observa que hay una diferencia entre los dos paradigmas, un ORM, o un mapeador de objetos a relaciones, nos ayuda a evitar esa diferencia, interviniendo entre los dos, y transformándolos mediante un archivo de mapeo.

AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

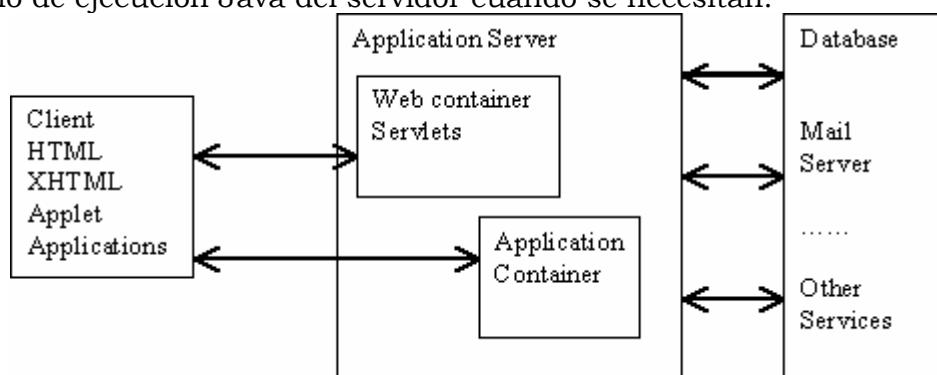
- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asíncronicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Ventajas de la tecnología AJAX

- Recuperación asíncrona de datos, el usuario no tienen que esperar después de una petición.
- Acercamiento de la metáfora de escritorio a la Web. Más cerca de crear realmente "Aplicaciones Web".
- No requiere plugins.
- Menos carga del servidor (menos transferencia).
- Se reduce el tamaño de la información intercambiada.

Servlet

Los Servlets son componentes del servidor. Estos componentes pueden ser ejecutados en cualquier plataforma o en cualquier servidor debido a la tecnología Java que se usa para implementarlos. Los Servlets incrementan la funcionalidad de una aplicación web. Se cargan de forma dinámica por el entorno de ejecución Java del servidor cuando se necesitan.



Arquitectura de componentes

Ventajas que ofrece la tecnología Servlet

- Los servlets añaden comportamiento dinámico a los servidores.
- La API de programación de los servlets hace muy fácil la escritura de servicios complejos para aplicaciones basadas en Web, sin tener que centrarse en los detalles de bajo nivel de los protocolos HTTP, formatos de petición, y cabeceras.
- Puesto que los servlets están implementados mediante Java, son independientes de la plataforma tanto en el cliente como en el servidor

JSP(Java Server Pages)

JSP es una tecnología Java desarrollada por la compañía Sun Microsystems que permite generar contenido dinámico en la Web en forma de documentos HTML, XML, o de otro tipo. JSP's permite código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento Web.

Antes de ejecutarse una pagina en JSP, el servidor web compila el script y genera un servlet. Por lo tanto, se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada.

Ventajas

- 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.
- Al generarse a través del script JSP un servlet, se provee de más rapidez ya que el servidor trabaja con código ya compilado.
- Otra de las ventajas que ofrece JSP al estar desarrollado en java es que es portable.

3. Software

STRUTS (versión 1.3.5)

Struts es un framework para el desarrollo de aplicaciones Web que implementa el patrón MVC, el cual se utiliza ampliamente y es considerado de gran solidez. Funciona bajo la plataforma J2EE.

Funcionalidades

- Soporte para la internacionalización (i18n) a través de ficheros de recursos y Java Locales
- Un gran conjunto de etiquetas personalizadas(taglibraries) JSP que cooperan con el controlador para su uso en la capa view de MVC
- Varias opciones para la validación de entrada de usuario en formularios HTML: ActionForm o Validator Framework
- Mecanismos para el manejo y reporte de errores
- Soporte para fuentes de datos

Beneficios de usar Struts

Te quita un gran trozo del pastel a la hora de desarrollar la aplicación Web, tiene implementadas muchas funcionalidades, como las expuestas anteriormente.

Su carácter de software libre y su compatibilidad con todas las plataformas en que Java Enterprise esté disponible, lo convierte en una herramienta altamente disponible.

Es un Framework muy usado, tiene una gran comunidad de desarrolladores y por tanto muchos manuales de su uso, ejemplos, solución de errores y una infinidad de herramientas para su utilización. Además tiene muchos plugins para su uso con otras tecnologías como por ejemplo AjaxTags, que hemos usado en el proyecto y que permite la integración de Ajax mediante taglibraries.

¿Por qué no hemos dado el salto hacia Struts 2?

A pesar de que Struts 2 tiene más funcionalidades, es un proyecto aun nuevo del que se disponen de pocos manuales. Además la migración de Struts 1 a Struts 2 aunque no es muy compleja, supone un gran cambio en la estructura y mucho tiempo para las mejoras que supondría su uso.

HIBERNATE

Hibernate es un ORM para la plataforma Java (aunque también hay un proyecto para c++) que ayuda en el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Hibernate no solo realiza esta transformación sino que nos proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que nos reducen el tiempo de desarrollo.



Ventajas de su uso:

Productividad: El código relacionado con la persistencia resulta tedioso para las aplicaciones de Java. Hibernate elimina la mayor parte de este y permite concentrarse principalmente en el problema, con lo cual aumenta la productividad. Según algunos estudios reduce aproximadamente el **95%** de las tareas que un programador tenía que hacer para realizar tareas comunes de acceso a datos.

Mantenimiento: Menor número de líneas de código hacen un sistema más comprensible, haciendo más fácil su mantenimiento. Automatizando la persistencia de los objetos se reduce sustancialmente las líneas de código y de esta manera se reduce la complejidad de la aplicación

Independencia del gestor de BD: Hibernate facilita la migración de sistemas entre diferentes motores de base de datos.

Bases de datos soportadas por Hibernate:

- Oracle 8i, 9i, 10g
- DB2 7.1, 7.2, 8.1
- Microsoft SQL Server 2000
- Sybase 12.5 (JConnect 5.5)
- MySQL 3.23, 4.0, 4.1, 5.0
- PostgreSQL 7.1.2, 7.2, 7.3, 7.4, 8.0, 8.1
- TimesTen 5.1, 6.0
- HypersonicSQL 1.61, 1.7.0, 1.7.2, 1.7.3, 1.8
- SAP DB 7.3
- InterSystems Cache 2007.1

Ver <http://www.hibernate.org/80.html> para ver más BDs soportadas

MYSQL

Sistema de gestión de base de datos, multihilo y multiusuario.

¿Por qué MySQL y no otros?

Dado que usamos Hibernate podríamos haber escogido cualquiera de los gestores de bases de datos mencionados en la lista de soportados. Pero nos decantamos por MySQL debido a los siguientes puntos:

- **Portabilidad:** MySQL puede correr en la inmensa mayoría de sistemas operativos, por lo que junto a otro lenguaje de programación de lado de servidor de alta portabilidad como Java nos permite el desarrollo de aplicaciones Web fáciles de migrar y el acceso y copia de los datos desde cualquier Sistema Operativo
- **Velocidad:** Es de los gestores de base de datos mas rápidos y de mejor rendimiento.

- **Documentación:** hay mucha documentación y tutoriales en Internet sobre él, y dispone de una gran comunidad de usuarios. Según las cifras del fabricante, existirían más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos.

4. Herramientas

REM 1.2.2

Herramienta para el desarrollo de la elicitation de requisitos y análisis del sistema. Desarrollado en nuestra escuela de informática por el profesor Amador Durán del Toro.

Apache Tomcat (versión 5.5)

Funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Se le considera un servidor de aplicaciones.

MySQL Server 5.0

Servidor del gestor de base de datos MySQL.

MySQL Administrador

MySQL Administrador es una herramienta que permite realizar tareas administrativas sobre servidores de MySQL incluyendo:

- la configuración de las opciones de inicio de los servidores
- inicio y detención de servidores
- monitorización de conexiones al servidor
- administración de usuarios
- monitorización del estado del servidor, incluyendo estadísticas de uso
- visualización de los logs de servidor
- gestión de copias de seguridad y recuperaciones
- visualización de catálogos de datos.

MySQL Query Browser

Es una utilidad para trabajar con la base de datos MySQL. Es un editor de sentencias SQL visual, que además incorpora herramientas para optimizar las consultas. Dispone también de un editor de tablas y registros, que permite crear nuevas tablas o cambiar las existentes, también añadirles o modificar sus atributos. Además es gratuito y es de los desarrolladores del gestor de Base de datos.

ArgoUML

Es herramienta para realizar los diagramas de UML escrita en Java y publicada bajo la Licencia BSD open source que además está traducida en español. Dado que es una aplicación Java, está disponible en cualquier plataforma soportada por Java. Entre sus características principales está la de generación de código a través del diagrama de clases, y opciones de ingeniería inversa.

JMeter

Aplicación realizada 100% en Java con una interfaz gráfica intuitiva, y además traducido al español. Esta aplicación nos permite definir comportamientos para casos de test y medir su rendimiento. Además es válido tanto para contenido estático como dinámico. Puede simular una gran carga en el servidor mediante multihilos y permite registrar los sucesos con completas gráficas e informes.



Captura de pantalla de la aplicación JMeter

ECLIPSE IDE LOMBOZ 3.2

Entorno de desarrollo eclipse unido al plugin Lomboz diseñado para el desarrollo de aplicaciones Web en Java.

¿Por qué Eclipse IDE LOMBOZ?

Hemos elegido el entorno Eclipse porque es gratuito, está siendo de los más utilizados por sus funcionalidades, incluso por las empresas, y por lo tanto tiene una infinidad de plugins para adaptarlo a las necesidades de cada proyecto.

La elección de Lomboz ha sido basada en las buenas experiencias de uso, y porque tiene funcionalidades relacionadas con entornos Webs como: editor de xml, soporte de servidores Web más usuales y multitud de tipos de proyecto. Además en sus últimas versiones soporta proyectos Web con struts.

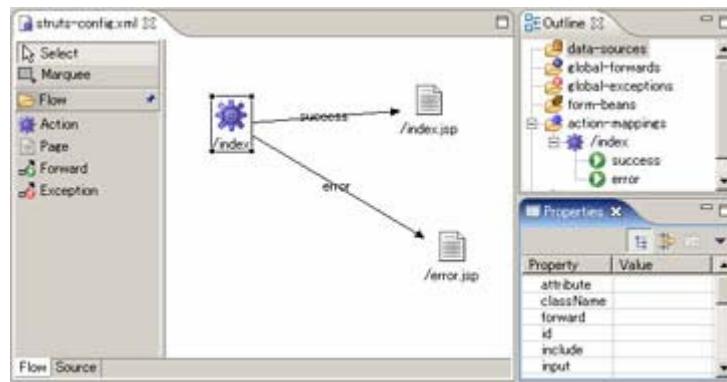
Plugins de Eclipse:

Subversion (<http://subversion.tigris.org/>)

Permite la conexión entre el servidor SVN y eclipse, para llevar un control de versiones. Es simple y fácil de usar.

Amateras StrutsIDE 2.0.2

Plugin para eclipse que permite trabajar de manera gráfica con Struts. Entre sus funciones más destacables está la de modificar el fichero de configuración de Struts (struts-config.xml) de forma grafica, mediante flujos, lo cual permite llevar un mejor control de las acciones, además sirve en cierto modo de diagrama de navegación.



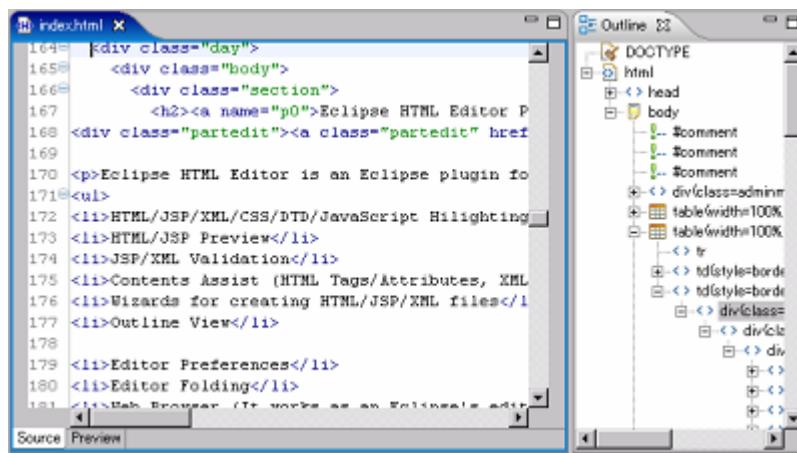
Entorno gráfico para la configuración de struts-config.xml

Más información sobre StrutsIDE 2.0.2 en:

http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=StrutsIDE

Amateras Eclipse HTML Editor

Junto con el plugin StrutsIDE, permite reconocer las taglibraries, opciones de autocompletado y validación del código Jsp.



Entorno para el desarrollo de las vistas en JSP usando las taglibraries

Más información sobre StrutsIDE 2.0.2 en:

http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=EclipseHTMLEditor

Hibernate Tools (<http://tools.hibernate.org/>)

Herramienta desarrollada por Hibernate, que permite generar tanto el código necesario, como el script SQL, a partir del mapeo en XML de la base de datos, lo cual supone un ahorro sustancial de tiempo.

PARTE III: SISTEMA DESARROLLADO

5 Dominio del problema

Actualmente los datos de los alumnos se encuentran almacenados en la secretaría virtual de la universidad de Sevilla. Sin embargo, a la hora de solicitar los profesores datos de los alumnos, la información se recoge en forma de ficha o mediante correo electrónico.

No existe ningún sistema que recoja los datos de los alumnos de forma ordenada o que no conlleve un proceso de transformación por parte del profesorado. El profesor debe reorganizar los e-mail que recibe de los alumnos y transformar la información de las fichas de formato papel a formato digital, lo cual a veces puede resultar una tarea tediosa. Por otra parte, tampoco existe ningún sistema que permita al profesor mantener un control y un orden de las tutorías solicitadas por los alumnos. El alumno tiene que comunicarse con el profesor personalmente, por e-mail, por teléfono, etc. y el profesor es el encargado de comprobar la disponibilidad y de anotar la petición de tutoría.

De esta idea, de recoger los datos de los alumnos de forma ordenada y digital, a nacido nuestro proyecto "Gestión de Fichas", que además permite a los profesores mantener un control de todas las peticiones de tutorías realizadas por los alumnos, verlas de una forma gráfica y llevar una planificación. En definitiva, nuestro proyecto pretende simplificar la labor del docente.

5.1 Introducción

El proyecto consistirá en una aplicación Web relacionado con la enseñanza y los servicios ofrecidos por el proyecto se detallan más concretamente a continuación:

Llevar acabo la gestión de los datos de los alumnos por fichas para que el profesor pueda contactar con sus alumnos. También los datos referentes a un profesor para que el alumno a su vez pueda contactar con los profesores.

- Permitir mostrar las tutorías del profesor, y en el caso de que lo deseé, podrá activar la citación electrónica que ofrece la aplicación Web, con la cual se pueden llevar a cabo las citas entre los alumnos y profesores en un horario concreto.
- Ofrecer información sobre las asignaturas, y las noticias referentes a éstas introducidas por los profesores
- Ofrecer la figura del administrador que velará por el buen funcionamiento de la Web. Será el encargado de inicializar la base de datos de los alumnos y profesores, mediante un fichero de texto, además de modificarlos, dar de alta y baja a un usuario.

En conclusión la aplicación Web ofrecerá servicios que ayudarán a llevar con más facilidad la tarea del profesorado.

5.2 Glosario de términos

Alumno: Toda aquella persona que se encuentra matriculada en al menos una de las asignaturas que componen el sistema

Profesor: Toda aquella persona que imparte al menos una de las asignaturas que se encuentran registradas en el sistema.

Asignatura: Cada una de las materias que se enseñan en un centro docente o de que consta una carrera o un plan de estudios

Administrador: Persona encargada del correcto funcionamiento del sistema. Tiene pleno control sobre todos los elementos del sistema y es el encargado de resolver los posibles conflictos que puedan surgir.

Cita: petición de cita que ha sido aceptada previamente por el profesor

Petición de cita: Resguardo de la acción realizada por el alumno para obtener una cita con un profesor. Sus valores son, aceptada, en proceso, y cancelada.

Tutorías: conjunto de horarios que un profesor tiene asignado para los alumnos

6 Visión general del sistema

6.1 Participantes en el proyecto

Participante	Fernando López Iglesias
Organización	Desarrolladores Open Source
Rol	Desarrollador
Es desarrollador	Sí
Es cliente	No
Es usuario	No

Participante	Noel Guillén Montilla
Organización	Desarrolladores Open Source
Rol	Desarrollador
Es desarrollador	Sí
Es cliente	No
Es usuario	No

Organización	Departamento de lenguajes y sistemas informáticos
Dirección	Avda. Reina Mercedes s/n. 41012 Sevilla
Teléfono	954 557 139
Fax	954 557 139
Comentarios	E-mail: lsi@lsi.us.es Web: www.lsi.us.es

Participante	David Benavides Cuevas
Organización	Departamento de lenguajes y sistemas informáticos
Rol	Tutor
Es desarrollador	No
Es cliente	Sí
Es usuario	Sí

6.2 Objetivos del sistema

OBJ-0001	Gestionar toda la información referente a los alumnos y profesores
Versión	1.0 (01/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Descripción	El sistema deberá <i>facilitar al usuario profesor</i> toda la información del <i>alumno</i> . <i>El sistema también facilitará la información de los profesores a los alumnos.</i>
Subobjetivos	Ninguno
Importancia	vital

VISIÓN GENERAL DEL SISTEMA

Urgencia	inmediatamente
Estado	validado
Estabilidad	alta

OBJ-0005	Controlar el proceso de citación entre alumnos y profesores
Versión	1.0 (01/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Descripción	El sistema deberá ofrecer un mecanismo que permita a profesores y alumnos llegar a un acuerdo en la petición de citas para tutoría, y que el profesor pueda mostrar el horario sus tutorías.
Subobjetivos	Ninguno
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta

OBJ-0009	Permitir perfiles de usuarios y gestionar sus datos
Versión	1.0 (07/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Descripción	El sistema deberá permitir a los usuarios registrados en el sistema iniciar sesión como usuario alumno , profesor o administrador , aplicando las restricciones que recaen sobre cada uno de ellos
Subobjetivos	[OBJ-0010] Permitir la protección de datos del alumno: El sistema deberá impedir que un usuario alumno tenga acceso a la información personal de los demás alumnos mediante un sistema de identificación Alumno / profesor .
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta

OBJ-0011	Establecer una administración vía Web
Versión	1.0 (09/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla

VISIÓN GENERAL DEL SISTEMA

Descripción	El sistema deberá permitir la gestión y control de todos los usuarios que acceden al portal mediante un <u>administrador</u> Web. El sistema deberá también permitir al <u>administrador</u> , la gestión de las asignaturas y titulaciones del sistema
Subobjetivos	<ul style="list-style-type: none"> ▪ [OBJ-0012] Control de usuarios: El sistema deberá permitir al <u>administrador</u> añadir o eliminar usuarios al portal ya sean de tipo <u>alumno</u> o <u>profesor</u>. (Los usuarios podrán ser añadidos de forma individual o a partir de un fichero, plano o XML.)
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta

OBJ-0014	Gestionar la información de las asignaturas
Versión	1.0 (09/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Descripción	El sistema deberá almacenar la información de cada <u>asignatura</u> así como todos los profesores y alumnos asociados a ella, las noticias introducidas por los profesores que imparten la <u>asignatura</u> .
Subobjetivos	Ninguno
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta

7 Requisitos del sistema

Denominamos requisitos a aquellas características que solicita el cliente y que desea que se cumplan en el sistema. A continuación detallaremos los que hemos recogido para el desarrollo de la aplicación.

7.1 Requisitos de información

IRQ-0001	Información sobre Fichas
Versión	1.0 (01/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores • [CRQ-0001] Número de peticiones de cita
Descripción	El sistema deberá almacenar la información correspondiente a <i>las fichas de los alumnos de una asignatura</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> ▪ Datos de dirección de residencia real, y mientras estudia: Domicilio, código postal ,país ▪ Datos personales: Nombre, apellidos, fecha de nacimiento ▪ Datos de contacto: teléfono, móvil, correo electrónico ▪ Datos académicos: Asignaturas aprobadas
Importancia	vital
Urgencia	hay presión
Estado	validado
Estabilidad	alta

IRQ-0002	Información sobre las tutorías
Versión	1.0 (01/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores
Descripción	El sistema deberá almacenar la información correspondiente a <i>las tutorías ofrecidas por el profesor en su horario</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> ▪ Datos de la tutoría: día de la semana, hora de inicio y hora de fin ▪ Datos del estado: estado(activada, desactivada)
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

REQUISITOS DEL SISTEMA

IRQ-0003	Información sobre las citaciones	
Versión	1.0 (01/08/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 	
Descripción	<p>El sistema deberá almacenar la información correspondiente a <i>las citas pedidas por los alumnos hacia el profesor</i>. En concreto:</p>	
Datos específicos	<ul style="list-style-type: none"> ▪ Datos de la cita: breve descripción, estado, motivo de la denegación (si es denegada) ▪ Datos de la fecha: fecha de la cita, hora de inicio. 	
Tiempo de vida	Medio	Máximo
	5 día(s)	10 día(s)
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

IRQ-0004	Información sobre los usuarios	
Versión	1.0 (01/08/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores • [OBJ-0009] Permitir perfiles de usuarios y gestionar sus datos • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores • [OBJ-0010] Permitir la protección de datos del alumno • [OBJ-0011] Establecer una administración vía Web • [OBJ-0012] Control de usuarios 	
Descripción	<p>El sistema deberá almacenar la información correspondiente a <i>los usuarios que utilizan la aplicación, es decir alumnos y profesores</i>. En concreto:</p>	
Datos específicos	<ul style="list-style-type: none"> ▪ Datos del usuario: usuario(DNI), clave de acceso, rol y correo electrónico 	
Importancia	vital	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	

IRQ-0005	Información sobre las asignaturas	
Versión	1.0 (01/08/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	

REQUISITOS DEL SISTEMA

Dependencias	<ul style="list-style-type: none"> • [OBJ-0014] Gestionar la información de las asignaturas
Descripción	El sistema deberá almacenar la información correspondiente a <i>la asignatura</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> ▪ Datos de la asignatura: código, descripción, departamento, titulación a la que pertenece, créditos, duración, tipo(optativa, obligatoria, troncal o de libre configuración), nombre de la asignatura, profesores y coordinador de la asignatura.
Importancia	importante
Urgencia	hay presión
Estado	pendiente de validación
Estabilidad	alta

IRQ-0006	Información sobre las noticias
Versión	1.0 (13/09/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [OBJ-0014] Gestionar la información de las asignaturas
Descripción	El sistema deberá almacenar la información correspondiente a <i>las noticias que sean necesarias por los profesores</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> ▪ Datos de la fecha: fecha y hora, tanto de la fecha de la primera inserción como de la modificación ▪ Datos de la noticia: Titulo, texto de la noticia y asignatura a la que corresponde
Importancia	quedaría bien
Urgencia	puede esperar
Estado	pendiente de validación
Estabilidad	media

IRQ-0007	Información sobre los profesores
Versión	1.0 (15/09/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores • [OBJ-0009] Permitir perfiles de usuarios y gestionar sus datos • [OBJ-0012] Control de usuarios • [OBJ-0014] Gestionar la información de las asignaturas
Descripción	El sistema deberá almacenar la información correspondiente a <i>los profesores que imparten las asignaturas</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> ▪ Datos de contacto: teléfono, pagina personal, despacho ▪ Datos personales: Nombre y categoría
Importancia	importante
Urgencia	inmediatamente

Estado	pendiente de validación
Estabilidad	alta

7.2 Reglas de negocio y restricciones

CRQ-0001	Numero de peticiones de cita
Versión	1.0 (09/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>Ningún alumno podrá solicitar mas de dos citas para un mismo profesor para un mismo día.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta

CRQ-0002	Unicidad de perfiles
Versión	1.0 (09/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	Ninguno
Descripción	La información almacenada por el sistema deberá satisfacer la siguiente restricción: <i>Un usuario solo podrá poseer el perfil de <u>alumno</u> o de <u>profesor</u>, nunca el de ambos.</i>
Importancia	vital
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta

7.3 Requisitos funcionales

7.3.1 Diagramas de casos de uso

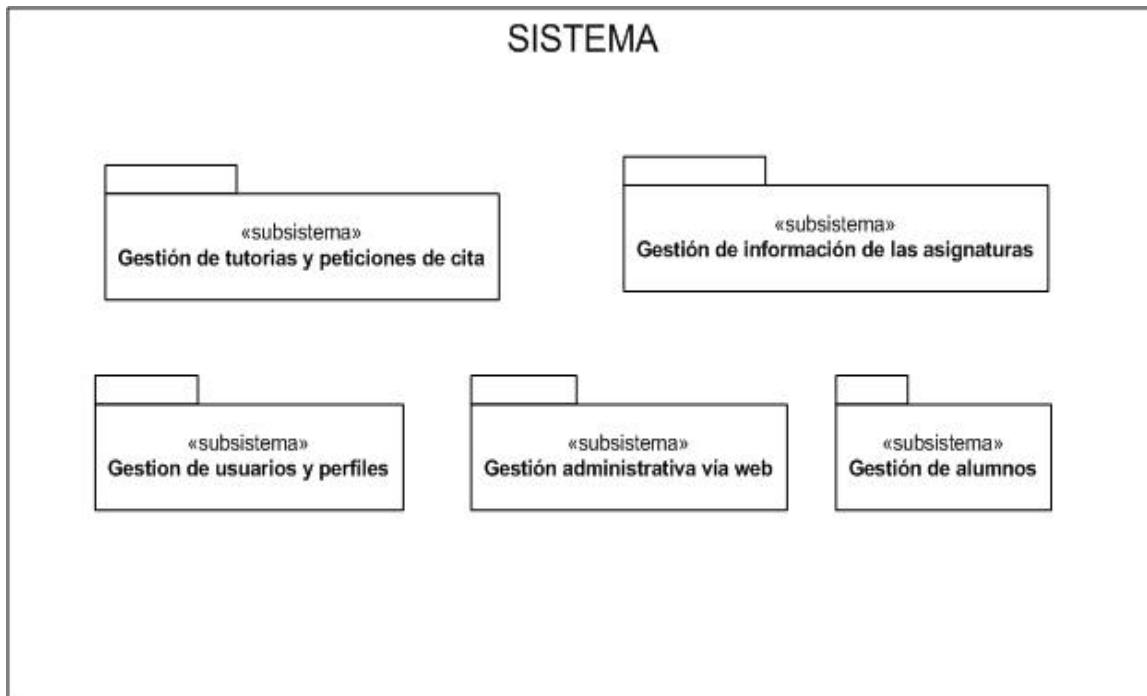


Figura 1: Sistema

7.3.1.1 Subsistema gestión de tutorías y peticiones de cita

En este subsistema se gestionará todo lo referente a la comunicación entre los alumnos y profesores. Por una parte se controla la exposición del horario de tutorías del [profesor](#), el cual podrá modificar el horario de tutorías, desactivarlas para que no se muestren por un motivo y volverlas activar. Por otra parte se gestionará el sistema que permitirá concertar citas entre profesores y alumnos, con el cual se podrán realizar peticiones de citas, denegarlas con su correspondiente motivo y consultar las citas aceptadas actualmente.

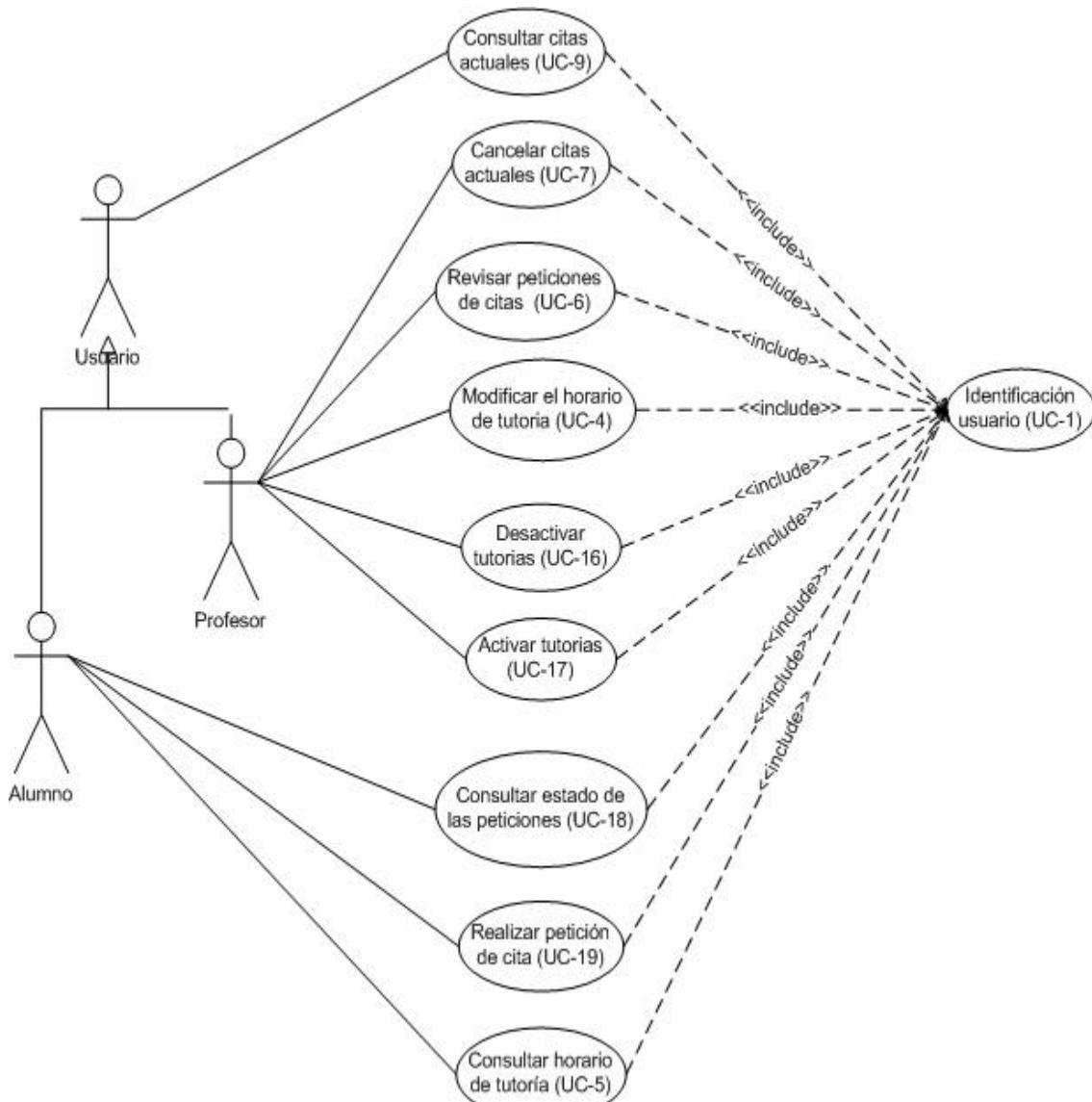
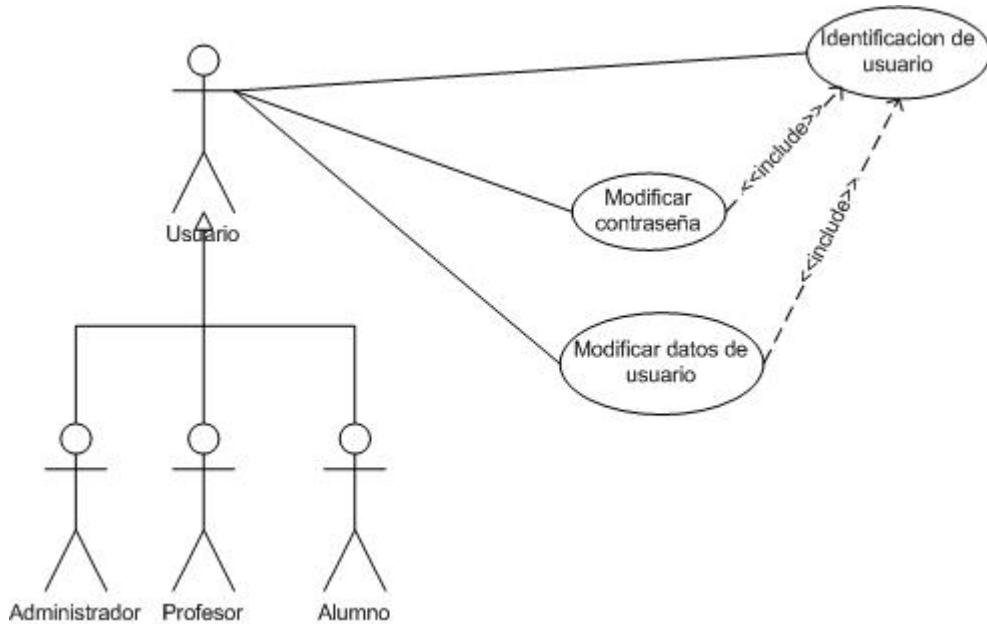


Figura 1: Gestión de tutorías y peticiones de citas

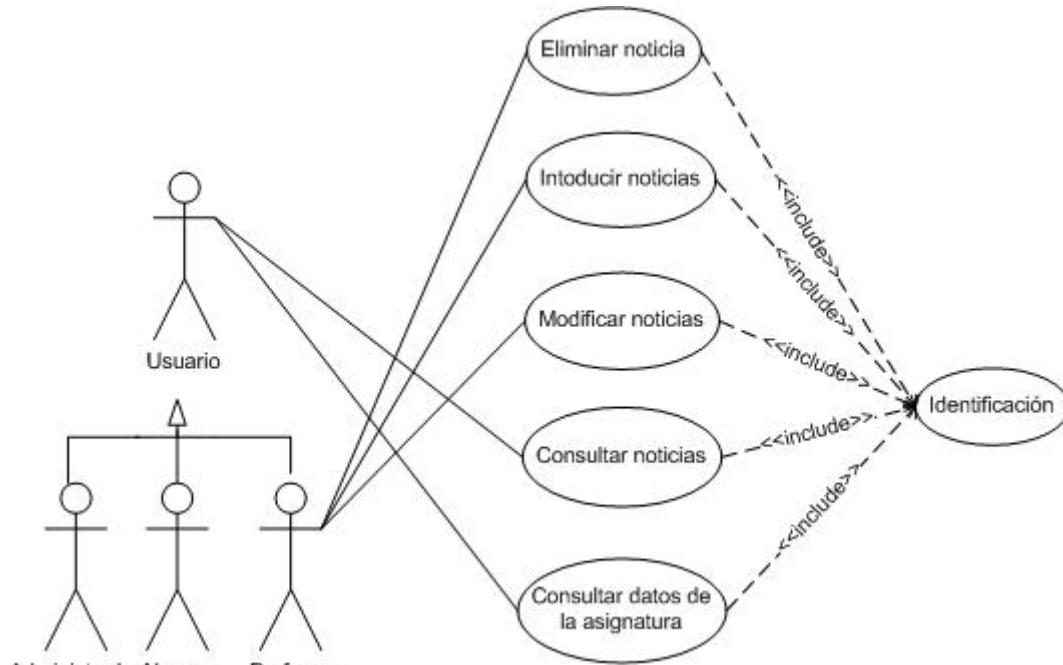
7.3.1.2 Subsistema gestión de usuarios y perfiles

En este subsistema se gestionará a los distintos usuarios que utilizan la aplicación Web. Estos tendrán que identificarse si quiere modificar la contraseña o los datos propios del usuario.


Figura 1: Gestión de usuarios

7.3.1.3 Subsistema gestión de información de las asignaturas

En el siguiente subsistema se gestiona la información de las asignaturas. Se controlará el sistema de noticias con el cual el [profesor](#) tendrá un método para avisar a los alumnos.


Figura 1: Gestión de información de las asignaturas

7.3.1.4 Subsistema gestión de información de alumnos y profesores

En el siguiente subsistema se gestiona la información de los alumnos, así como la información que los profesores hacen pública

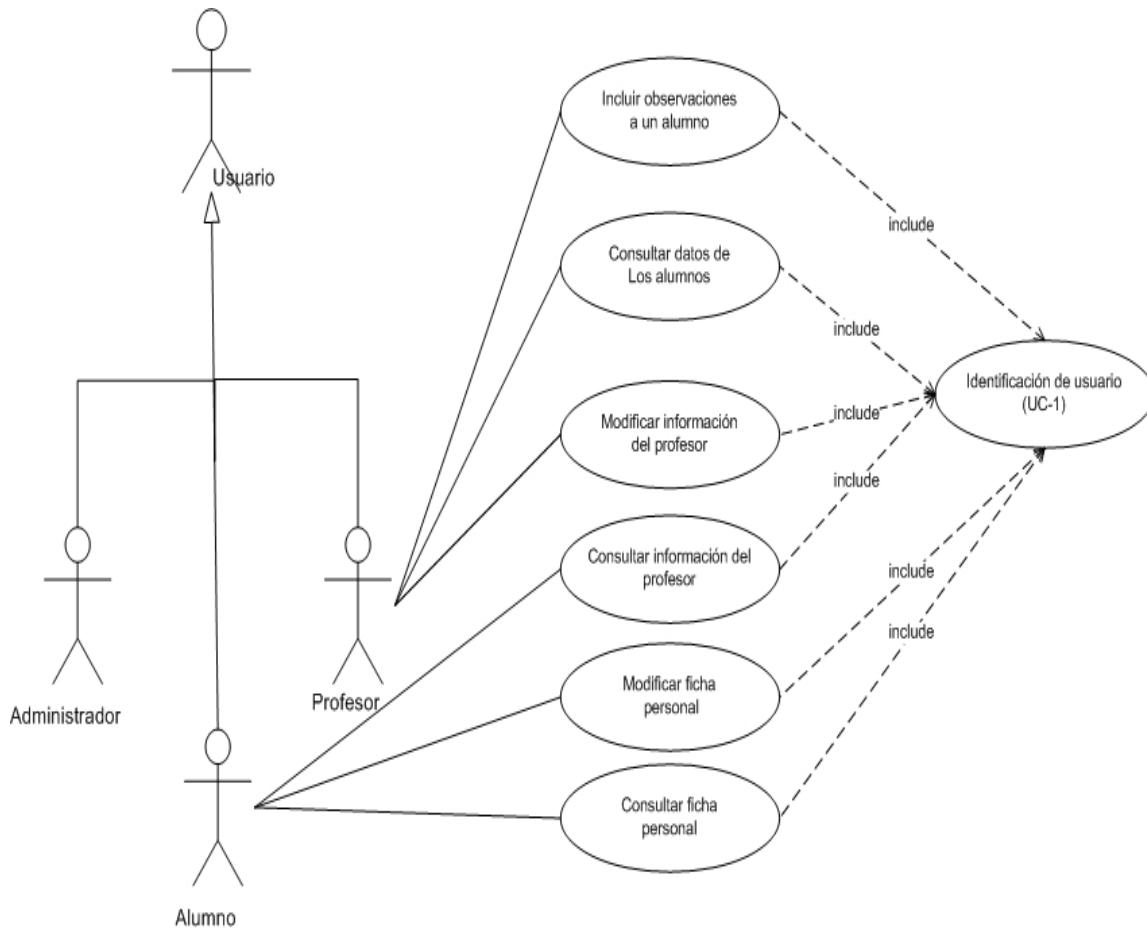


Figura 1: Gestión de información de alumnos y profesores

7.3.1.5 Subsistema gestión administrativa vía Web

En el siguiente subsistema se gestionan las asignaturas, los usuarios y las titulaciones, así como las relaciones que existen entre ellos. Se permitirá añadir o eliminar cualquiera de los elementos anteriores y establecer o deshacer las asociaciones pertinentes entre ellos.

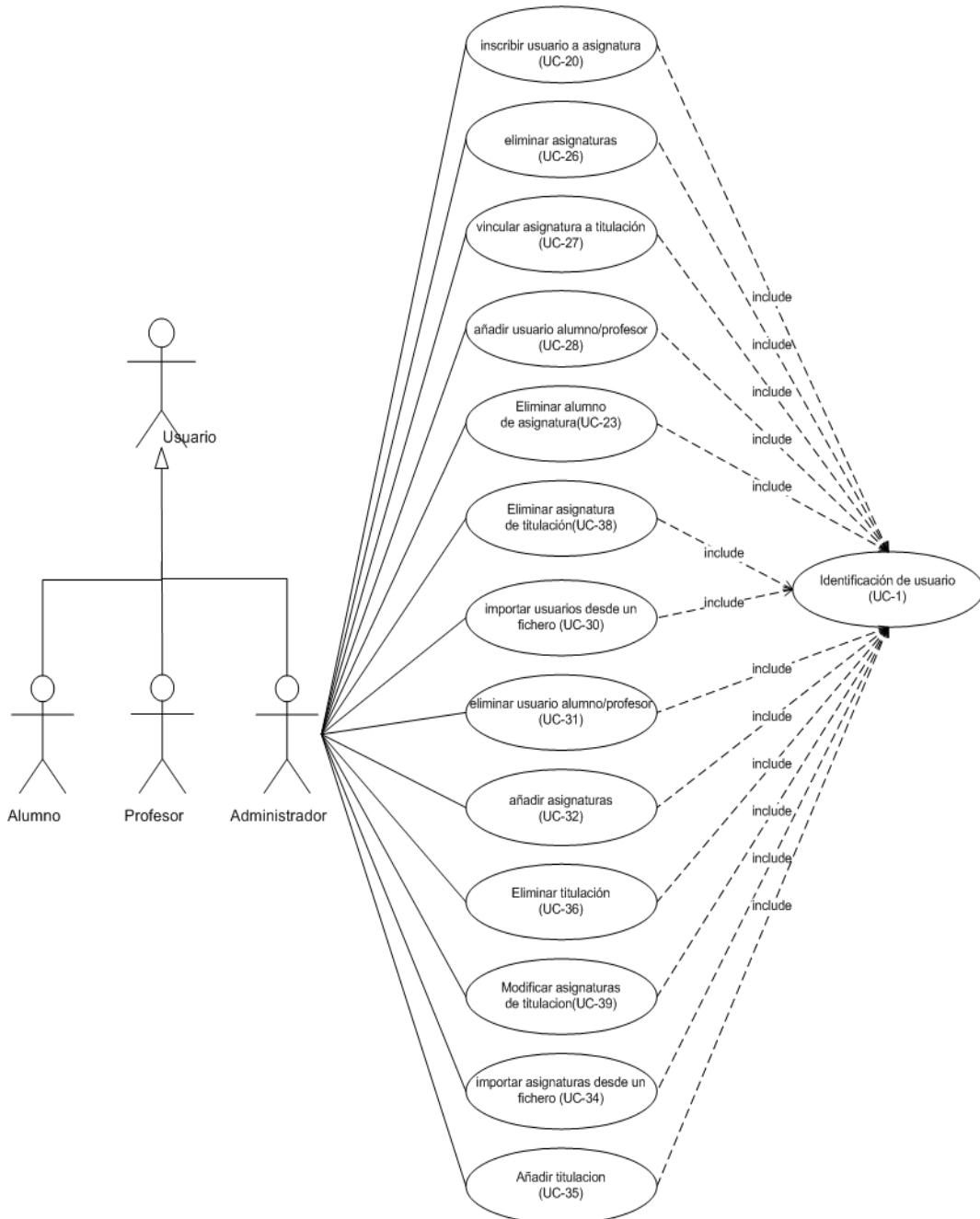


Figura 1: Gestión administrativa vía Web

7.3.2 Definición de actores

ACT-0001	<u>Administrador</u>
Versión	1.0 (01/08/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Descripción	Este actor representa a la persona encargada de gestionar la pagina Web
Comentarios	Este usuario tiene derechos ilimitados, aunque no participa en el portal Web

REQUISITOS DEL SISTEMA

	activamente, sólo lo administra y lo gestiona para su buen funcionamiento
--	---

ACT-0002	<u>Alumno</u>
Versión	1.0 (01/08/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Descripción	Este actor representa <i>al usuario que es alumno de las asignaturas que el portal contiene</i>
Comentarios	Estos usuarios tienen derechos restringidos, solo pueden modificar su ficha, visualizar los datos referentes a las asignaturas y los profesores que las imparten. Participan activamente en la Web.

ACT-0003	<u>Profesor</u>
Versión	1.0 (01/08/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Descripción	Este actor representa <i>al usuario que es profesor de las asignaturas contenidas en el portal</i>
Comentarios	Estos usuarios tienen derechos limitados y participan activamente en la Web.

ACT-0004	<u>Usuario</u>
Versión	1.0 (09/08/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Descripción	Este actor representa <i>al usuario de la aplicación Web</i>

7.3.3 Casos de uso del sistema

7.3.3.1 Subsistema gestión de tutorías y peticiones de cita

UC-0004	<u>Modificar horario de tutorías</u>
Versión	1.0 (10/08/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	<ul style="list-style-type: none"> • <u>[IRQ-0002] Información sobre las tutorías</u> • <u>[OBJ-0005] Controlar el proceso de citación entre alumnos y profesores</u>

REQUISITOS DEL SISTEMA

Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desea cambiar el horario de tutorías	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra el horario de tutorías actual por días de la semana, hora de inicio y fin.
	3	El actor Profesor (ACT-0003) modifica, añadiendo o eliminando días de la semana con la hora de inicio y fin.
	4	El sistema valida los datos de los horarios de tutoría
	5	El sistema muestra por pantalla el horario elegido por el profesor y da la opción de modificar de nuevo, o aceptarla
	6	Si el profesor elige modificar los datos referentes a los horarios de tutorías, el sistema vuelve al paso 2
	7	Si está de acuerdo con las tutorías, el sistema almacena las tutorías
Postcondición	Se ha cambiado el horario de tutorías y se ha almacenado	
Excepciones	Paso	Acción
	4	Si la hora de ingreso no es valida, es decir, hay días repetidos, horas fuera de rango, el sistema muestra el error por pantalla y da la opción de modificarlo, a continuación este caso de uso queda sin efecto
	7	Si no se ha podido almacenar, el sistema se muestra el error por pantalla y se da la opción de volver a intentarlo, a continuación este caso de uso queda sin efecto
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

UC-0005	Consultar el horario de tutorías	
Versión	1.0 (10/08/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> [IRQ-0002] Información sobre las tutorías [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el profesor desea ver el horario de tutorías	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra las asignaturas
	3	El actor Alumno (ACT-0002) elige la asignatura
	4	El sistema muestra a los profesores de la asignatura

REQUISITOS DEL SISTEMA

	5	El actor Alumno (ACT-0002) elige el profesor deseado
	6	El sistema muestra por pantalla el horario de las tutorías
Postcondición	Ninguna	
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	media	

UC-0006	Revisar peticiones de citas	
Versión	1.0 (10/08/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el profesor desea revisar las peticiones de citas de tutorías de los alumnos	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra por orden de fecha de envío y por alumno las peticiones de citas de tutorías
	3	El actor Profesor (ACT-0003) selecciona una petición
	4	El sistema muestra los datos de la cita y da la opción de aceptarla o rechazarla
	5	Si el profesor acepta la petición, el sistema rechaza las peticiones en el mismo horario con el motivo: horario ocupado. Luego salta al paso 8
	6	Si el profesor acepta la petición de cita, el sistema solicita al usuario profesor , los motivos de la cancelación
	7	El actor Profesor (ACT-0003) introduce el motivo, o elige uno de los ya predefinidos
	8	El sistema modifica el estado de la petición del alumno afectado
Postcondición	El sistema ha modificado el estado actual de las peticiones modificadas	
Excepciones	Paso	Acción
	2	Si el sistema no puede acceder a los datos de la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

REQUISITOS DEL SISTEMA

UC-0007	Cancelar citas actuales																			
Versión	1.0 (10/08/2006)																			
Autores	Fernando López Iglesias Noel Guillén Montilla																			
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 																			
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario profesor</i> desee ver <i>las citas aceptadas y cancelarlas por algún motivo</i>																			
Precondición	Hay citas aceptadas anteriormente																			
Secuencia normal	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Paso</th> <th style="text-align: left; padding: 2px;">Acción</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td style="padding: 2px;">2</td> <td style="padding: 2px;">El sistema <i>solicita al usuario que elija uno de los días ofrecidos</i></td> </tr> <tr> <td style="padding: 2px;">3</td> <td style="padding: 2px;">El actor Profesor (ACT-0003) <i>elige el día deseado</i></td> </tr> <tr> <td style="padding: 2px;">4</td> <td style="padding: 2px;">El sistema <i>muestra las citas para el día elegido, con el motivo, la hora, y el alumno, ordenado por la hora</i></td> </tr> <tr> <td style="padding: 2px;">5</td> <td style="padding: 2px;">El actor Profesor (ACT-0003) <i>selecciona las citas que desea cancelar</i></td> </tr> <tr> <td style="padding: 2px;">6</td> <td style="padding: 2px;">El sistema <i>solicita un motivo de cancelación al usuario</i></td> </tr> <tr> <td style="padding: 2px;">7</td> <td style="padding: 2px;">El actor Profesor (ACT-0003) <i>introduce el motivo de la cancelación de citas</i></td> </tr> <tr> <td style="padding: 2px;">8</td> <td style="padding: 2px;">El sistema <i>modifica el estado de las cita, e informa a los alumnos afectados de la cancelación</i></td> </tr> </tbody> </table>		Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema <i>solicita al usuario que elija uno de los días ofrecidos</i>	3	El actor Profesor (ACT-0003) <i>elige el día deseado</i>	4	El sistema <i>muestra las citas para el día elegido, con el motivo, la hora, y el alumno, ordenado por la hora</i>	5	El actor Profesor (ACT-0003) <i>selecciona las citas que desea cancelar</i>	6	El sistema <i>solicita un motivo de cancelación al usuario</i>	7	El actor Profesor (ACT-0003) <i>introduce el motivo de la cancelación de citas</i>	8	El sistema <i>modifica el estado de las cita, e informa a los alumnos afectados de la cancelación</i>
Paso	Acción																			
1	Se realiza el caso de uso Identificación de usuario (UC-0001)																			
2	El sistema <i>solicita al usuario que elija uno de los días ofrecidos</i>																			
3	El actor Profesor (ACT-0003) <i>elige el día deseado</i>																			
4	El sistema <i>muestra las citas para el día elegido, con el motivo, la hora, y el alumno, ordenado por la hora</i>																			
5	El actor Profesor (ACT-0003) <i>selecciona las citas que desea cancelar</i>																			
6	El sistema <i>solicita un motivo de cancelación al usuario</i>																			
7	El actor Profesor (ACT-0003) <i>introduce el motivo de la cancelación de citas</i>																			
8	El sistema <i>modifica el estado de las cita, e informa a los alumnos afectados de la cancelación</i>																			
Postcondición	Se han modificado las citas seleccionadas, y se le ha notificado a los afectados por la cancelación																			
Excepciones	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Paso</th> <th style="text-align: left; padding: 2px;">Acción</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">8</td> <td style="padding: 2px;">Si no puede acceder a la base de datos, el sistema da la opción de reintentarlo, a continuación este caso de uso queda sin efecto</td> </tr> </tbody> </table>		Paso	Acción	8	Si no puede acceder a la base de datos, el sistema da la opción de reintentarlo, a continuación este caso de uso queda sin efecto														
Paso	Acción																			
8	Si no puede acceder a la base de datos, el sistema da la opción de reintentarlo, a continuación este caso de uso queda sin efecto																			
Importancia	quedaría bien																			
Urgencia	puede esperar																			
Estado	validado																			
Estabilidad	media																			

UC-0009	Consultar citas actuales			
Versión	1.0 (11/08/2006)			
Autores	Fernando López Iglesias Noel Guillén Montilla			
Dependencias	<ul style="list-style-type: none"> • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores • [IRQ-0003] Información sobre las citaciones 			
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario deseja ver las citas de una determinada fecha</i>			
Precondición	Hay citas aceptadas			
Secuencia	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Paso</th> <th style="text-align: left; padding: 2px;">Acción</th> </tr> </thead> </table>		Paso	Acción
Paso	Acción			

REQUISITOS DEL SISTEMA

normal	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra los posibles días a elegir para la consulta de citas
	3	El actor Usuario (ACT-0004) elige el día deseado
	4	El sistema muestra las citas aceptadas por nombre y hora, ordenadas por la hora.
Postcondición	Se ha mostrado al usuario las citas actuales	
Excepciones	Paso	Acción
	4	Si no se ha podido tener acceso a la base de datos, el sistema informa del error y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	validado	
Estabilidad	media	

UC-0016	Desactivar tutorías temporalmente	
Versión	1.0 (16/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> [IRQ-0002] Información sobre las tutorías [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el profesor desee que no se muestre su horario de tutorías y se muestre su motivo	
Precondición	Hay tutorías introducidas anteriormente	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra un formulario para llenar con el motivo para que este se muestre en vez de las tutorías
	3	El actor Profesor (ACT-0003) Introduce el motivo de la desactivación
	4	El sistema muestra que se ha realizado correctamente
Postcondición	Se han desactivado las tutorías	
Excepciones	Paso	Acción
	4	Si no ha podido desactivar las tutorías correctamente, el sistema informa del error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	pendiente de validación	
Estabilidad	media	

REQUISITOS DEL SISTEMA

UC-0017	Activar tutorías	
Versión	1.0 (16/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0002] Información sobre las tutorías • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el profesor desee activar las tutorías que han sido desactivadas y por tanto han dejado de ser mostradas</p>	
Precondición	<p>El mostrado de las tutorías está desactivado</p>	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra la opción de activarla
	3	El actor Profesor (ACT-0003) acepta
	4	El sistema muestra que se ha realizado correctamente la activación
Postcondición	<p>Se ha activado el mostrado de las tutorías</p>	
Excepciones	Paso	Acción
	4	Si no se ha podido realizar correctamente la activación, el sistema informa del error y da la posibilidad de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	pendiente de validación	
Estabilidad	media	

UC-0018	Consultar estado de las peticiones	
Versión	1.0 (16/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 	
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el alumno desee consultar el estado de las peticiones de cita realizadas</p>	
Precondición	<p>Hay peticiones</p>	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El actor Alumno (ACT-0002) solicita el estado de las peticiones
	3	El sistema muestra las peticiones ordenadas por fechas, junto con su estado
Postcondición	<p>Se ha mostrado el estado de las peticiones realizadas por el alumno</p>	
Excepciones	Paso	Acción

REQUISITOS DEL SISTEMA

	3	Si no se ha podido tener acceso a la base de datos, el sistema informa del error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	importante	
Urgencia	hay presión	
Estado	pendiente de validación	
Estabilidad	alta	

UC-0019	Realizar petición de cita	
Versión	1.0 (16/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <i>alumno</i> desee realizar una petición de cita	
Precondición	No ha realizado mas de dos peticiones el mismo día	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema ofrece formularios para que el usuario los rellene con el motivo, y la fecha de la cita deseada
	3	El actor Usuario (ACT-0004) rellena el formulario
	4	El sistema muestra la petición y pide la confirmación del usuario
	5	Si el <i>alumno</i> confirma, el sistema notifica que se ha realizado correctamente la petición de la cita
	6	Si el <i>alumno</i> no confirma la tutoría y desea modificar la petición, el sistema muestra la petición con la opción de modificarlo y salta al paso 3
Postcondición	Se notifica al profesor de la petición	
Importancia	importante	
Urgencia	inmediatamente	
Estado	pendiente de validación	
Estabilidad	alta	

7.3.3.2 Subsistema gestión de usuarios y perfiles

UC-0001	Identificación de usuario
Versión	1.0 (09/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla

REQUISITOS DEL SISTEMA

Dependencias	<ul style="list-style-type: none"> • [OBJ-0010] Permitir la protección de datos del alumno • [OBJ-0012] Control de usuarios • [IRQ-0004] Información sobre los usuarios 										
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso abstracto durante la realización de los siguientes casos de uso: [UC-0002] Modificar contraseña, [UC-0004] Modificar horario de tutorías, [UC-0005] Consultar el horario de tutorías, [UC-0006] Revisar peticiones de citas, [UC-0007] Cancelar citas actuales, [UC-0008] Consultar datos de alumnos, [UC-0009] Consultar citas actuales, [UC-0010] Introducir noticias, [UC-0011] Modificar noticias, [UC-0012] Consultar noticias, [UC-0013] Consultar datos de la asignatura, [UC-0014] Modificar datos del usuario, [UC-0015] Eliminar noticia, [UC-0016] Desactivar tutorías temporalmente, [UC-0017] Activar tutorías, [UC-0018] Consultar estado de las peticiones, [UC-0019] Realizar petición de cita, [UC-0020] Inscribir usuario a asignatura, [UC-0023] Eliminar usuario de asignatura, [UC-0026] Eliminar asignaturas, [UC-0027] Vincular asignatura a titulación, [UC-0028] Añadir usuario alumno/profesor, [UC-0030] Importar usuarios a partir de un fichero, [UC-0031] Eliminar usuario alumno/profesor, [UC-0032] Añadir asignatura, [UC-0034] Importar asignatura a partir de fichero, [UC-0035] Añadir titulación, [UC-0036] Eliminar titulación, [UC-0038] Eliminar asignatura de titulación, [UC-0039] modificar asignaturas de titulación, [UC-0040] Consultar ficha personal, [UC-0041] Modificar ficha personal, [UC-0042] Consultar información del profesor, [UC-0043] Modificar información del profesor, [UC-0044] Incluir observaciones a un alumno</p>										
Precondición	Ninguna										
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El sistema solicita los datos de identificación al usuario</td></tr> <tr> <td>2</td><td>El actor Usuario (ACT-0004) introduce los datos de identificación</td></tr> <tr> <td>3</td><td>El sistema valida los datos proporcionados por el usuario</td></tr> <tr> <td>4</td><td>El sistema confirma los datos del usuario</td></tr> </tbody> </table>	Paso	Acción	1	El sistema solicita los datos de identificación al usuario	2	El actor Usuario (ACT-0004) introduce los datos de identificación	3	El sistema valida los datos proporcionados por el usuario	4	El sistema confirma los datos del usuario
Paso	Acción										
1	El sistema solicita los datos de identificación al usuario										
2	El actor Usuario (ACT-0004) introduce los datos de identificación										
3	El sistema valida los datos proporcionados por el usuario										
4	El sistema confirma los datos del usuario										
Postcondición	El usuario es reconocido por el sistema y se le aplican las correspondientes restricciones según el tipo de usuario.										
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>3</td><td>Si los datos ofrecidos por el usuario no son correctos, el sistema ofrece 3 intentos al usuario, al tercer intento se bloquea, a continuación este caso de uso queda sin efecto</td></tr> <tr> <td>3</td><td>Si no tiene acceso a la base de datos de los usuarios registrados, el sistema muestra el error y ofrecerle la opción de reintentarlo, a continuación este caso de uso continúa</td></tr> <tr> <td>3</td><td>Si uno de los datos introducidos por el usuario no son correctos, el sistema muestra el error y permite modificar los datos de acceso introducidos, a continuación este caso de uso continúa</td></tr> </tbody> </table>	Paso	Acción	3	Si los datos ofrecidos por el usuario no son correctos, el sistema ofrece 3 intentos al usuario, al tercer intento se bloquea, a continuación este caso de uso queda sin efecto	3	Si no tiene acceso a la base de datos de los usuarios registrados, el sistema muestra el error y ofrecerle la opción de reintentarlo, a continuación este caso de uso continúa	3	Si uno de los datos introducidos por el usuario no son correctos, el sistema muestra el error y permite modificar los datos de acceso introducidos, a continuación este caso de uso continúa		
Paso	Acción										
3	Si los datos ofrecidos por el usuario no son correctos, el sistema ofrece 3 intentos al usuario, al tercer intento se bloquea, a continuación este caso de uso queda sin efecto										
3	Si no tiene acceso a la base de datos de los usuarios registrados, el sistema muestra el error y ofrecerle la opción de reintentarlo, a continuación este caso de uso continúa										
3	Si uno de los datos introducidos por el usuario no son correctos, el sistema muestra el error y permite modificar los datos de acceso introducidos, a continuación este caso de uso continúa										
Importancia	vital										
Urgencia	inmediatamente										
Estado	validado										
Estabilidad	alta										

UC-0002	Modificar contraseña
Versión	1.0 (09/08/2006)

REQUISITOS DEL SISTEMA

Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0009] Permitir perfiles de usuarios y gestionar sus datos • [IRQ-0004] Información sobre los usuarios 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario solicita al sistema un cambio de contraseña</i>	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema <i>pide al usuario que introduzca una nueva contraseña, dos veces para evitar equivocaciones</i>
	3	El actor Usuario (ACT-0004) <i>introduce los datos de la nueva contraseña</i>
	4	El sistema <i>cambia la contraseña asignada al usuario y muestra que se ha realizado correctamente</i>
Postcondición	El sistema ha cambiado la contraseña asignada al usuario	
Excepciones	Paso	Acción
	4	Si <i>usuario no introduce una contraseña de una longitud mínima o las dos contraseñas no coinciden</i> , el sistema <i>muestra el error por pantalla y se le da la opción de introducir la contraseña al usuario</i> , a continuación este caso de uso queda sin efecto
Importancia	importante	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	

UC-0014	Modificar datos del usuario	
Versión	1.0 (13/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0004] Información sobre los usuarios • [OBJ-0009] Permitir perfiles de usuarios y gestionar sus datos 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea modificar sus datos</i>	
Precondición	Ninguno	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema <i>muestra los valores actuales de los datos del usuario y da la opción de cambiar el email y la clave</i>
	3	El actor Usuario (ACT-0004) <i>introduce los datos</i>
	4	El sistema <i>Muestra por pantalla los nuevos valores de los datos de usuario y da la opción de aceptarla o modificarla de nuevo</i>
Postcondición	Ninguna	

Importancia	importante
Urgencia	hay presión
Estado	pendiente de validación
Estabilidad	alta

7.2.3.3 Subsistema gestión de información de las asignaturas

UC-0010	Introducir noticias																					
Versión	1.0 (13/09/2006)																					
Autores	Fernando López Iglesias Noel Guillén Montilla																					
Dependencias	<ul style="list-style-type: none"> • [OBJ-0014] Gestionar la información de las asignaturas • [IRQ-0006] Información sobre las noticias 																					
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <i>profesor</i> desea introducir una noticia en una <i>asignatura concreta</i>																					
Precondición	Ninguna																					
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema muestra las asignaturas impartidas por el <i>profesor</i></td> </tr> <tr> <td>3</td> <td>El actor Profesor (ACT-0003) elige la <i>asignatura</i> deseada</td> </tr> <tr> <td>4</td> <td>El sistema muestra un formulario con los campos respectivos a rellenar para una noticia, es decir, <i>título</i> y <i>texto</i></td> </tr> <tr> <td>5</td> <td>El actor Profesor (ACT-0003) rellena el formulario</td> </tr> <tr> <td>6</td> <td>El sistema muestra de nuevo la noticia con todos los datos y queda pendiente de la validación del <i>profesor</i></td> </tr> <tr> <td>7</td> <td>Si el <i>profesor</i> quiere modificar la noticia, el sistema vuelve al paso 4</td> </tr> <tr> <td>8</td> <td>Si el <i>profesor</i> esta de acuerdo, el sistema almacena la noticia</td> </tr> <tr> <td>9</td> <td>El sistema avisa de que se ha realizado correctamente la inserción de la nueva noticia</td> </tr> </tbody> </table>		Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema muestra las asignaturas impartidas por el <i>profesor</i>	3	El actor Profesor (ACT-0003) elige la <i>asignatura</i> deseada	4	El sistema muestra un formulario con los campos respectivos a rellenar para una noticia, es decir, <i>título</i> y <i>texto</i>	5	El actor Profesor (ACT-0003) rellena el formulario	6	El sistema muestra de nuevo la noticia con todos los datos y queda pendiente de la validación del <i>profesor</i>	7	Si el <i>profesor</i> quiere modificar la noticia, el sistema vuelve al paso 4	8	Si el <i>profesor</i> esta de acuerdo, el sistema almacena la noticia	9	El sistema avisa de que se ha realizado correctamente la inserción de la nueva noticia
Paso	Acción																					
1	Se realiza el caso de uso Identificación de usuario (UC-0001)																					
2	El sistema muestra las asignaturas impartidas por el <i>profesor</i>																					
3	El actor Profesor (ACT-0003) elige la <i>asignatura</i> deseada																					
4	El sistema muestra un formulario con los campos respectivos a rellenar para una noticia, es decir, <i>título</i> y <i>texto</i>																					
5	El actor Profesor (ACT-0003) rellena el formulario																					
6	El sistema muestra de nuevo la noticia con todos los datos y queda pendiente de la validación del <i>profesor</i>																					
7	Si el <i>profesor</i> quiere modificar la noticia, el sistema vuelve al paso 4																					
8	Si el <i>profesor</i> esta de acuerdo, el sistema almacena la noticia																					
9	El sistema avisa de que se ha realizado correctamente la inserción de la nueva noticia																					
Postcondición	Se ha almacenado la noticia.																					
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>Si el sistema no ha podido guardar las noticias porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto</td> </tr> </tbody> </table>		Paso	Acción	8	Si el sistema no ha podido guardar las noticias porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto																
Paso	Acción																					
8	Si el sistema no ha podido guardar las noticias porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto																					
Importancia	quedaría bien																					
Urgencia	puede esperar																					
Estado	pendiente de validación																					
Estabilidad	media																					

REQUISITOS DEL SISTEMA

UC-0011	Modificar noticias	
Versión	1.0 (13/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0014] Gestionar la información de las asignaturas • [IRQ-0006] Información sobre las noticias 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <i>profesor</i> desea modificar una noticia	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra por pantalla las asignaturas que imparte el <i>profesor</i>
	3	El actor Profesor (ACT-0003) elige la asignatura deseada
	4	El sistema el sistema muestra todas las noticias de la asignatura ordenadas por fecha
	5	El actor Profesor (ACT-0003) elige la noticia deseada
	6	El sistema muestra la noticia con campos que permitan su modificación
	7	El actor Profesor (ACT-0003) modifica la noticia
	8	El sistema muestra la noticia y da la opción de modificarla o aceptarla
Postcondición	Se ha modificado la noticia y actualizado la fecha de modificación	
Excepciones	Paso	Acción
	8	Si no se ha podido modificar la noticia, el sistema muestra el error y ofrece la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	pendiente de validación	
Estabilidad	media	

UC-0012	Consultar noticias	
Versión	1.0 (13/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0014] Gestionar la información de las asignaturas • [IRQ-0006] Información sobre las noticias 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <i>profesor</i> desea consultar las noticias referentes a una asignatura	
Precondición	Hay noticias ya actualmente	
Secuencia	Paso	Acción

REQUISITOS DEL SISTEMA

normal	Paso	1 Se realiza el caso de uso Identificación de usuario (UC-0001)
	Paso	2 El sistema muestra las asignaturas relacionadas con el usuario
	Paso	3 El actor Usuario (ACT-0004) elige la asignatura deseada
	Paso	4 El sistema muestra las noticias ordenadas por fecha y título
Postcondición	Se ha mostrado las noticias por pantalla	
Excepciones	Paso	Acción
	2	Si ocurre un error con la base de datos, el sistema muestra el tipo de error y ofrecerle la opción de reintentarla, a continuación este caso de uso continúa
	4	Si no tiene acceso a la base de datos, el sistema muestra el tipo de error y ofrecerle la opción de reintentarla, a continuación este caso de uso queda sin efecto
Importancia	quedaría bien	
Urgencia	hay presión	
Estado	pendiente de validación	
Estabilidad	media	

UC-0013	Consultar datos de la asignatura	
Versión	1.0 (13/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0005] Información sobre las asignaturas • [OBJ-0014] Gestionar la información de las asignaturas 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desea consultar los datos referentes a una asignatura	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra las asignaturas relacionadas con el usuario
	3	El actor Usuario (ACT-0004) elige la asignatura deseada
	4	El sistema muestra la información referente a la asignatura seleccionada
Postcondición	Se ha mostrado por pantalla la información de la asignatura	
Excepciones	Paso	Acción
	2	Si el sistema no tiene acceso a la base de datos, el sistema muestra el error y ofrece la opción de reintentarla, a continuación este caso de uso continúa
	4	Si no se tiene acceso a la base de datos, el sistema da la opción de reintentarla, a continuación este caso de uso queda sin efecto
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	pendiente de validación	
Estabilidad	media	

UC-0015	Eliminar noticia	
Versión	1.0 (16/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0006] Información sobre las noticias • [OBJ-0014] Gestionar la información de las asignaturas 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <i>profesor</i> desee <i>eliminar una noticia ya existente</i>	
Precondición	Ninguna	
Secuencia normal	Paso Acción	1 Se realiza el caso de uso Identificación de usuario (UC-0001) 2 El sistema <i>Muestra las asignaturas que imparte el profesor</i> 3 El actor Profesor (ACT-0003) escoge la asignatura deseada 4 El sistema <i>muestra las noticias por título al usuario</i> 5 El actor Profesor (ACT-0003) <i>elige la noticia deseada</i> 6 El sistema <i>indica que el proceso se ha realizado correctamente</i>
Postcondición		
Excepciones		
Importancia		
Urgencia		
Estado		
Estabilidad		

7.3.3.4 Subsistema gestión de información de alumnos y profesores

UC-0008	Consultar datos de alumnos	
Versión	1.0 (10/08/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores • [IRQ-0001] Información sobre Fichas 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <i>usuario profesor</i> desea la <i>información de alumno</i>	
Precondición	Ninguna	
Secuencia normal	Paso Acción	1 Se realiza el caso de uso Identificación de usuario (UC-0001)

REQUISITOS DEL SISTEMA

	2	El sistema muestra las asignaturas impartidas por el profesor y solicita que el usuario elija la deseada
	3	El actor Profesor (ACT-0003) elije la asignatura deseada
	4	El sistema muestra los alumnos asignados a esa asignatura por DNI y nombre, ordenados alfabéticamente. Se muestra también si tienen ficha y se da la opción de seleccionar a un usuario que si tenga
	5	El actor Profesor (ACT-0003) selecciona el alumno deseado
	6	El sistema muestra la ficha del alumno seleccionado
Postcondición		Se ha mostrado por pantalla la información del alumno deseado
Importancia		importante
Urgencia		hay presión
Estado		validado
Estabilidad		alta

UC-0040	Consultar ficha personal	
Versión	1.0 (29/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el alumno desee consultar su ficha personal	
Precondición		
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El actor Alumno (ACT-0002) solicita ver su ficha personal
	3	El sistema muestra la ficha personal
Postcondición	El alumno ha visualizado su ficha personal	
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	media	

UC-0041	Modificar ficha personal	
Versión	1.0 (29/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas 	

REQUISITOS DEL SISTEMA

	<ul style="list-style-type: none"> • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores 										
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <u>alumno</u> desee realizar una modificación de su ficha										
Precondición	Existen usuarios alumnos en el sistema										
Secuencia normal	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema muestra la ficha que hay almacenada actualmente</td> </tr> <tr> <td>3</td> <td>El actor <u>Alumno (ACT-0002)</u> rellena y modifica los campos necesarios</td> </tr> <tr> <td>4</td> <td>El sistema valida la información y registra los cambios</td> </tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema muestra la ficha que hay almacenada actualmente	3	El actor <u>Alumno (ACT-0002)</u> rellena y modifica los campos necesarios	4	El sistema valida la información y registra los cambios
Paso	Acción										
1	Se realiza el caso de uso Identificación de usuario (UC-0001)										
2	El sistema muestra la ficha que hay almacenada actualmente										
3	El actor <u>Alumno (ACT-0002)</u> rellena y modifica los campos necesarios										
4	El sistema valida la información y registra los cambios										
Postcondición	Se ha modificado la ficha										
Importancia	importante										
Urgencia	hay presión										
Estado	validado										
Estabilidad	alta										

UC-0042	Consultar información del profesor														
Versión	1.0 (29/09/2006)														
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>														
Dependencias	<ul style="list-style-type: none"> • [IRQ-0007] Información sobre los profesores • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores 														
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <u>alumno</u> desee consultar la información de un <u>profesor</u>														
Precondición	Existen profesores en el sistema														
Secuencia normal	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema muestra la lista de asignaturas del <u>alumno</u></td> </tr> <tr> <td>3</td> <td>El actor <u>Alumno (ACT-0002)</u> selecciona la <u>asignatura</u> deseada</td> </tr> <tr> <td>4</td> <td>El sistema muestra los profesores asociados</td> </tr> <tr> <td>5</td> <td>El actor <u>Alumno (ACT-0002)</u> selecciona el <u>profesor</u> deseado</td> </tr> <tr> <td>6</td> <td>El sistema muestra la información del <u>profesor</u></td> </tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema muestra la lista de asignaturas del <u>alumno</u>	3	El actor <u>Alumno (ACT-0002)</u> selecciona la <u>asignatura</u> deseada	4	El sistema muestra los profesores asociados	5	El actor <u>Alumno (ACT-0002)</u> selecciona el <u>profesor</u> deseado	6	El sistema muestra la información del <u>profesor</u>
Paso	Acción														
1	Se realiza el caso de uso Identificación de usuario (UC-0001)														
2	El sistema muestra la lista de asignaturas del <u>alumno</u>														
3	El actor <u>Alumno (ACT-0002)</u> selecciona la <u>asignatura</u> deseada														
4	El sistema muestra los profesores asociados														
5	El actor <u>Alumno (ACT-0002)</u> selecciona el <u>profesor</u> deseado														
6	El sistema muestra la información del <u>profesor</u>														
Postcondición	El <u>alumno</u> ha consultado la información requerida														
Importancia	importante														
Urgencia	hay presión														
Estado	validado														
Estabilidad	media														

REQUISITOS DEL SISTEMA

UC-0043	Modificar información del profesor											
Versión	1.0 (29/09/2006)											
Autores	Fernando López Iglesias Noel Guillén Montilla											
Dependencias	<ul style="list-style-type: none"> • [IRQ-0007] Información sobre los profesores • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores • [OBJ-0009] Permitir perfiles de usuarios y gestionar sus datos 											
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el profesor desee <i>modificar la información que muestra a los alumnos</i></p>											
Precondición	Existen profesores en el sistema											
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema <i>muestra la información que tiene almacenada actualmente</i></td> </tr> <tr> <td>3</td> <td>El actor Profesor (ACT-0003) rellena y modifica los campos necesarios</td> </tr> <tr> <td>4</td> <td>El sistema <i>valida la información y registra los cambios</i></td> </tr> </tbody> </table>		Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema <i>muestra la información que tiene almacenada actualmente</i>	3	El actor Profesor (ACT-0003) rellena y modifica los campos necesarios	4	El sistema <i>valida la información y registra los cambios</i>
Paso	Acción											
1	Se realiza el caso de uso Identificación de usuario (UC-0001)											
2	El sistema <i>muestra la información que tiene almacenada actualmente</i>											
3	El actor Profesor (ACT-0003) rellena y modifica los campos necesarios											
4	El sistema <i>valida la información y registra los cambios</i>											
Postcondición	Se ha modificado la información											
Importancia	importante											
Urgencia	hay presión											
Estado	validado											
Estabilidad	media											

UC-0044	Incluir observaciones a un alumno															
Versión	1.0 (05/10/2006)															
Autores	Fernando López Iglesias Noel Guillén Montilla															
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores 															
Descripción	<p>El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el profesor desee <i>incluir una observación de un determinado alumno</i></p>															
Precondición	Existen alumnos en el sistema															
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema <i>muestra las asignaturas que son impartidas por el profesor</i></td> </tr> <tr> <td>3</td> <td>El actor Profesor (ACT-0003) selecciona la asignatura deseada</td> </tr> <tr> <td>4</td> <td>El sistema <i>muestra la lista de los alumnos pertenecientes a dicha asignatura</i></td> </tr> <tr> <td>5</td> <td>El actor Profesor (ACT-0003) selecciona el alumno</td> </tr> <tr> <td>6</td> <td>El sistema <i>muestra un formulario con el campo observación</i></td> </tr> </tbody> </table>		Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema <i>muestra las asignaturas que son impartidas por el profesor</i>	3	El actor Profesor (ACT-0003) selecciona la asignatura deseada	4	El sistema <i>muestra la lista de los alumnos pertenecientes a dicha asignatura</i>	5	El actor Profesor (ACT-0003) selecciona el alumno	6	El sistema <i>muestra un formulario con el campo observación</i>
Paso	Acción															
1	Se realiza el caso de uso Identificación de usuario (UC-0001)															
2	El sistema <i>muestra las asignaturas que son impartidas por el profesor</i>															
3	El actor Profesor (ACT-0003) selecciona la asignatura deseada															
4	El sistema <i>muestra la lista de los alumnos pertenecientes a dicha asignatura</i>															
5	El actor Profesor (ACT-0003) selecciona el alumno															
6	El sistema <i>muestra un formulario con el campo observación</i>															

	7	El actor Profesor (ACT-0003) introduce los datos de la observación
	8	El sistema almacena la nueva observación junto con la hora a la que se ha realizado y el profesor que la ha llevado a cabo
Postcondición		Se ha realizado la observación sobre el alumno
Importancia		quedaría bien
Urgencia		puede esperar
Estado		validado
Estabilidad		alta

7.2.3.5 Subsistema gestión administrativa vía Web

UC-0035	Añadir titulación	
Versión	1.0 (28/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0011] Establecer una administración vía Web 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee añadir una nueva titulación	
Precondición		
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra las titulaciones existentes y solicita el nombre de la nueva titulación
	3	El actor Administrador (ACT-0001) introduce los datos necesarios
	4	El sistema comprueba que no existe la nueva titulación en el sistema
	5	El sistema almacena la nueva titulación
Postcondición	Se ha añadido una nueva titulación	
Excepciones	Paso	Acción
	4	Si la titulación ya existe en el sistema, el sistema solicita el nombre de una nueva titulación, a continuación este caso de uso continúa
	5	Si el sistema no ha podido guardar la titulación porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

UC-0036	Eliminar titulación
----------------	----------------------------

REQUISITOS DEL SISTEMA

Versión	1.0 (29/09/2006)								
Autores	Fernando López Iglesias Noel Guillén Montilla								
Dependencias	• [OBJ-0011] Establecer una administración vía Web								
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee eliminar una titulación								
Precondición	Existen titulaciones en el sistema								
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema muestra todas las titulaciones existentes en el sistema</td> </tr> <tr> <td>3</td> <td>El actor Administrador (ACT-0001) selecciona la titulación a eliminar</td> </tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema muestra todas las titulaciones existentes en el sistema	3	El actor Administrador (ACT-0001) selecciona la titulación a eliminar
Paso	Acción								
1	Se realiza el caso de uso Identificación de usuario (UC-0001)								
2	El sistema muestra todas las titulaciones existentes en el sistema								
3	El actor Administrador (ACT-0001) selecciona la titulación a eliminar								
Postcondición	Se han eliminado las titulaciones deseadas								
Importancia	importante								
Urgencia	hay presión								
Estado	validado								
Estabilidad	alta								

UC-0028	Añadir usuario alumno/profesor															
Versión	1.0 (28/09/2006)															
Autores	Fernando López Iglesias Noel Guillén Montilla															
Dependencias	<ul style="list-style-type: none"> [IRQ-0004] Información sobre los usuarios [IRQ-0007] Información sobre los profesores [OBJ-0011] Establecer una administración vía Web 															
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee añadir un nuevo usuario															
Precondición	Ninguna															
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema pregunta si se trata de un usuario alumno o profesor</td> </tr> <tr> <td>3</td> <td>El actor Administrador (ACT-0001) selecciona la opción deseada</td> </tr> <tr> <td>4</td> <td>El sistema muestra el formulario a llenar de acuerdo al tipo de usuario que se ha seleccionado</td> </tr> <tr> <td>5</td> <td>El sistema comprueba que no existe el usuario en el sistema</td> </tr> <tr> <td>6</td> <td>El sistema almacena el nuevo usuario</td> </tr> </tbody> </table>		Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema pregunta si se trata de un usuario alumno o profesor	3	El actor Administrador (ACT-0001) selecciona la opción deseada	4	El sistema muestra el formulario a llenar de acuerdo al tipo de usuario que se ha seleccionado	5	El sistema comprueba que no existe el usuario en el sistema	6	El sistema almacena el nuevo usuario
Paso	Acción															
1	Se realiza el caso de uso Identificación de usuario (UC-0001)															
2	El sistema pregunta si se trata de un usuario alumno o profesor															
3	El actor Administrador (ACT-0001) selecciona la opción deseada															
4	El sistema muestra el formulario a llenar de acuerdo al tipo de usuario que se ha seleccionado															
5	El sistema comprueba que no existe el usuario en el sistema															
6	El sistema almacena el nuevo usuario															
Postcondición	Se ha añadido un nuevo usuario															
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>Si el usuario ya existe en el sistema, el sistema solicita un nuevo usuario, a continuación este caso de uso continúa</td> </tr> <tr> <td>6</td> <td>Si el sistema no ha podido guardar al usuario porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin</td> </tr> </tbody> </table>		Paso	Acción	5	Si el usuario ya existe en el sistema, el sistema solicita un nuevo usuario, a continuación este caso de uso continúa	6	Si el sistema no ha podido guardar al usuario porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin								
Paso	Acción															
5	Si el usuario ya existe en el sistema, el sistema solicita un nuevo usuario, a continuación este caso de uso continúa															
6	Si el sistema no ha podido guardar al usuario porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin															

REQUISITOS DEL SISTEMA

	efecto
Importancia	importante
Urgencia	hay presión
Estado	validado
Estabilidad	alta

UC-0032	Añadir asignatura	
Versión	1.0 (28/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0005] Información sobre las asignaturas • [OBJ-0011] Establecer una administración vía Web • [OBJ-0014] Gestionar la información de las asignaturas 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee añadir una nueva asignatura	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra las asignaturas existentes y solicita el nombre de la nueva asignatura
	3	El actor Administrador (ACT-0001) introduce los datos necesarios
	4	El sistema comprueba que no existe la nueva asignatura en el sistema
	5	El sistema almacena la nueva asignatura
Postcondición	Se ha añadido una nueva asignatura	
Excepciones	Paso	Acción
	4	Si la asignatura ya existe en el sistema, el sistema solicita el nombre de una nueva asignatura , a continuación este caso de uso continúa
	5	Si el sistema no ha podido guardar la asignatura porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

UC-0038	Eliminar asignatura de titulación
Versión	1.0 (29/09/2006)
Autores	Fernando López Iglesias

REQUISITOS DEL SISTEMA

	<u>Noel Guillén Montilla</u>														
Dependencias	<ul style="list-style-type: none"> • [OBJ-0011] Establecer una administración vía Web 														
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee eliminar una asignatura de una titulación														
Precondición	Existen asignaturas y titulaciones en el sistema														
Secuencia normal	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Paso</th><th style="text-align: left; padding: 2px;">Acción</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;">1</td><td style="padding: 2px;">Se realiza el caso de uso Identificación de usuario (UC-0001)</td></tr> <tr> <td style="padding: 2px;">2</td><td style="padding: 2px;">El sistema muestra las titulaciones existentes</td></tr> <tr> <td style="padding: 2px;">3</td><td style="padding: 2px;">El actor Administrador (ACT-0001) selecciona la titulación deseada</td></tr> <tr> <td style="padding: 2px;">4</td><td style="padding: 2px;">El sistema muestra una lista con las asignaturas pertenecientes a la titulación</td></tr> <tr> <td style="padding: 2px;">5</td><td style="padding: 2px;">El actor Administrador (ACT-0001) selecciona la asignatura a eliminar</td></tr> <tr> <td style="padding: 2px;">6</td><td style="padding: 2px;">El sistema informa de que la asignatura ha sido eliminada correctamente</td></tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema muestra las titulaciones existentes	3	El actor Administrador (ACT-0001) selecciona la titulación deseada	4	El sistema muestra una lista con las asignaturas pertenecientes a la titulación	5	El actor Administrador (ACT-0001) selecciona la asignatura a eliminar	6	El sistema informa de que la asignatura ha sido eliminada correctamente
Paso	Acción														
1	Se realiza el caso de uso Identificación de usuario (UC-0001)														
2	El sistema muestra las titulaciones existentes														
3	El actor Administrador (ACT-0001) selecciona la titulación deseada														
4	El sistema muestra una lista con las asignaturas pertenecientes a la titulación														
5	El actor Administrador (ACT-0001) selecciona la asignatura a eliminar														
6	El sistema informa de que la asignatura ha sido eliminada correctamente														
Postcondición	Se ha eliminado la asignatura de la titulación														
Excepciones	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; padding: 2px;">Paso</td><td style="width: 90%; padding: 2px;">Acción</td></tr> <tr> <td style="padding: 2px;">6</td><td style="padding: 2px;">Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto</td></tr> </table>	Paso	Acción	6	Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto										
Paso	Acción														
6	Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto														
Importancia	importante														
Urgencia	hay presión														
Estado	validado														
Estabilidad	alta														

UC-0031	Eliminar usuario alumno/profesor												
Versión	1.0 (28/09/2006)												
Autores	Fernando López Iglesias Noel Guillén Montilla												
Dependencias	<ul style="list-style-type: none"> • [IRQ-0004] Información sobre los usuarios • [OBJ-0011] Establecer una administración vía Web 												
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desea eliminar un usuario en concreto del sistema												
Precondición	Existen usuarios en la aplicación												
Secuencia normal	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Paso</th><th style="text-align: left; padding: 2px;">Acción</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;">1</td><td style="padding: 2px;">Se realiza el caso de uso Identificación de usuario (UC-0001)</td></tr> <tr> <td style="padding: 2px;">2</td><td style="padding: 2px;">El sistema pregunta si se trata de un usuario alumno o profesor</td></tr> <tr> <td style="padding: 2px;">3</td><td style="padding: 2px;">El actor Administrador (ACT-0001) selecciona la opción deseada</td></tr> <tr> <td style="padding: 2px;">4</td><td style="padding: 2px;">El sistema muestra los usuarios existentes y un formulario con el campo <i>DNI</i></td></tr> <tr> <td style="padding: 2px;">5</td><td style="padding: 2px;">El actor Administrador (ACT-0001) introduce el <i>DNI</i> de la persona a eliminar o lo selecciona de la lista</td></tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema pregunta si se trata de un usuario alumno o profesor	3	El actor Administrador (ACT-0001) selecciona la opción deseada	4	El sistema muestra los usuarios existentes y un formulario con el campo <i>DNI</i>	5	El actor Administrador (ACT-0001) introduce el <i>DNI</i> de la persona a eliminar o lo selecciona de la lista
Paso	Acción												
1	Se realiza el caso de uso Identificación de usuario (UC-0001)												
2	El sistema pregunta si se trata de un usuario alumno o profesor												
3	El actor Administrador (ACT-0001) selecciona la opción deseada												
4	El sistema muestra los usuarios existentes y un formulario con el campo <i>DNI</i>												
5	El actor Administrador (ACT-0001) introduce el <i>DNI</i> de la persona a eliminar o lo selecciona de la lista												

REQUISITOS DEL SISTEMA

	6	El sistema informa de que el usuario ha sido eliminado correctamente
Postcondición		Se ha eliminado el usuario
Excepciones	Paso	Acción
	6	Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia		importante
Urgencia		hay presión
Estado		validado
Estabilidad		alta

UC-0026	Eliminar asignaturas	
Versión	1.0 (28/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0005] Información sobre las asignaturas • [OBJ-0011] Establecer una administración vía Web • [OBJ-0014] Gestionar la información de las asignaturas 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee eliminar una asignatura	
Precondición	Existen asignaturas en el sistema	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema muestra todas las asignaturas existentes en el sistema por orden alfabético
	3	El actor Administrador (ACT-0001) selecciona la asignatura a eliminar
	4	El sistema informa de que la asignatura ha sido eliminada correctamente
Postcondición	Se ha eliminado la asignatura	
Excepciones	Paso	Acción
	4	Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

UC-0023	Eliminar usuario de asignatura
Versión	1.0 (28/09/2006)
Autores	Fernando López Iglesias

REQUISITOS DEL SISTEMA

	<u>Noel Guillén Montilla</u>																			
Dependencias	<ul style="list-style-type: none"> • [OBJ-0011] Establecer una administración vía Web • [OBJ-0014] Gestionar la información de las asignaturas 																			
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee eliminar un usuario de una asignatura																			
Precondición	Existen usuarios asociados a asignaturas																			
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema pregunta si se trata de un usuario alumno o profesor</td> </tr> <tr> <td>3</td> <td>El actor Administrador (ACT-0001) selecciona la opción deseada</td> </tr> <tr> <td>4</td> <td>El sistema muestra las asignaturas existentes en el sistema</td> </tr> <tr> <td>5</td> <td>El actor Administrador (ACT-0001) selecciona la asignatura deseada</td> </tr> <tr> <td>6</td> <td>El sistema muestra una lista con los usuarios de la asignatura</td> </tr> <tr> <td>7</td> <td>El actor Profesor (ACT-0003) selecciona el usuario a eliminar</td> </tr> <tr> <td>8</td> <td>El sistema informa de que el usuario ha sido eliminado correctamente</td> </tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema pregunta si se trata de un usuario alumno o profesor	3	El actor Administrador (ACT-0001) selecciona la opción deseada	4	El sistema muestra las asignaturas existentes en el sistema	5	El actor Administrador (ACT-0001) selecciona la asignatura deseada	6	El sistema muestra una lista con los usuarios de la asignatura	7	El actor Profesor (ACT-0003) selecciona el usuario a eliminar	8	El sistema informa de que el usuario ha sido eliminado correctamente	
Paso	Acción																			
1	Se realiza el caso de uso Identificación de usuario (UC-0001)																			
2	El sistema pregunta si se trata de un usuario alumno o profesor																			
3	El actor Administrador (ACT-0001) selecciona la opción deseada																			
4	El sistema muestra las asignaturas existentes en el sistema																			
5	El actor Administrador (ACT-0001) selecciona la asignatura deseada																			
6	El sistema muestra una lista con los usuarios de la asignatura																			
7	El actor Profesor (ACT-0003) selecciona el usuario a eliminar																			
8	El sistema informa de que el usuario ha sido eliminado correctamente																			
Postcondición	El alumno ha sido eliminado de la asignatura																			
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto</td> </tr> </tbody> </table>	Paso	Acción	8	Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto															
Paso	Acción																			
8	Si no se tiene acceso a la base de datos u ocurre un problema al eliminar, el sistema muestra el error al usuario y le da la opción de reintentarlo, a continuación este caso de uso queda sin efecto																			
Importancia	importante																			
Urgencia	hay presión																			
Estado	validado																			
Estabilidad	alta																			

UC-0020	Inscribir usuario a asignatura							
Versión	1.0 (28/09/2006)							
Autores	Fernando López Iglesias Noel Guillén Montilla							
Dependencias	<ul style="list-style-type: none"> • [OBJ-0014] Gestionar la información de las asignaturas • [OBJ-0011] Establecer una administración vía Web 							
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el profesor desee añadir un usuario a una asignatura							
Precondición	Existen usuarios y asignaturas en el sistema							
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se realiza el caso de uso Identificación de usuario (UC-0001)</td> </tr> <tr> <td>2</td> <td>El sistema muestra las asignaturas existentes en el sistema</td> </tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso Identificación de usuario (UC-0001)	2	El sistema muestra las asignaturas existentes en el sistema	
Paso	Acción							
1	Se realiza el caso de uso Identificación de usuario (UC-0001)							
2	El sistema muestra las asignaturas existentes en el sistema							

REQUISITOS DEL SISTEMA

	3	El actor Administrador (ACT-0001) selecciona la asignatura a la que quiere añadir un nuevo usuario
	4	El sistema pregunta si se trata de un usuario alumno o profesor
	5	El actor Administrador (ACT-0001) selecciona la opción deseada
	6	El sistema muestra un formulario con el campo <i>DNI</i>
	7	El actor Administrador (ACT-0001) introduce el <i>DNI</i> del usuario
	8	El sistema almacena los datos y los muestra por pantalla, indicando que el proceso a finalizado correctamente
Postcondición	Se ha inscrito el usuario en la asignatura .	
Excepciones	Paso	Acción
	8	Si el usuario ya está inscrito en dicha asignatura , el sistema muestra el error por pantalla acompañado de los datos del usuario, a continuación este caso de uso queda sin efecto
	8	Si el sistema no ha podido guardar al usuario porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto
	8	Si el usuario no existe en el sistema, el sistema muestra un error por pantalla y solicita un nuevo usuario, a continuación este caso de uso continúa
Importancia	importante	
Urgencia	hay presión	
Estado	pendiente de validación	
Estabilidad	alta	

UC-0030	Importar usuarios a partir de un fichero	
Versión	1.0 (28/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0004] Información sobre los usuarios • [IRQ-0007] Información sobre los profesores • [OBJ-0011] Establecer una administración vía Web 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee añadir usuarios a partir de un fichero	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema solicita la ruta del archivo y muestra información sobre como debe estar elaborado dicho archivo
	3	El actor Administrador (ACT-0001) especifica la dirección solicitada
	4	El sistema almacena todos los nuevos usuarios y muestra los posibles errores que han sucedido
	5	El sistema muestra una lista con todas las asignaturas existentes en el sistema

REQUISITOS DEL SISTEMA

	6	El actor Administrador (ACT-0001) selecciona la asignatura a la que quiere importar la lista de usuarios
	7	El sistema añade todos los nuevos usuarios a la asignatura e informa de los posibles errores que se hayan producido
Postcondición	Se han añadido los usuarios al sistema	
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

UC-0034	Importar asignatura a partir de fichero	
Versión	1.0 (28/09/2006)	
Autores	Fernando López Iglesias Noel Guillén Montilla	
Dependencias	<ul style="list-style-type: none"> • [IRQ-0005] Información sobre las asignaturas • [OBJ-0011] Establecer una administración vía Web • [OBJ-0014] Gestionar la información de las asignaturas 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador desee añadir asignaturas a partir de un fichero	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Identificación de usuario (UC-0001)
	2	El sistema solicita la ruta del archivo y muestra información sobre como debe estar elaborado dicho archivo
	3	El actor Administrador (ACT-0001) especifica la dirección solicitada
	4	El sistema almacena todas las nuevas asignaturas y muestra los posibles errores que han sucedido
	5	El sistema muestra una lista con todas las titulaciones existentes en el sistema
	6	El actor Administrador (ACT-0001) selecciona la titulación a la que quiere importar la lista de asignaturas
	7	El sistema añade todas las nuevas asignaturas a la titulación e informa de los posibles errores que se hayan producido
Postcondición	Se han añadido las asignaturas al sistema	
Importancia	importante	
Urgencia	hay presión	
Estado	validado	
Estabilidad	alta	

UC-0027	Vincular asignatura a titulación
Versión	1.0 (28/09/2006)

REQUISITOS DEL SISTEMA

Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>														
Dependencias	• [OBJ-0011] Establecer una administración vía Web														
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <u>administrador</u> desee añadir una <u>asignatura</u> a una titulación														
Precondición	Existen asignaturas y titulaciones en el sistema														
Secuencia normal	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Paso</th> <th style="text-align: left; padding: 2px;">Acción</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">Se realiza el caso de uso <u>Identificación de usuario (UC-0001)</u></td> </tr> <tr> <td style="padding: 2px;">2</td> <td style="padding: 2px;">El sistema muestra las titulaciones existentes</td> </tr> <tr> <td style="padding: 2px;">3</td> <td style="padding: 2px;">El actor <u>Administrador (ACT-0001)</u> selecciona la titulación deseada</td> </tr> <tr> <td style="padding: 2px;">4</td> <td style="padding: 2px;">El sistema solicita el nombre o código de la <u>asignatura</u></td> </tr> <tr> <td style="padding: 2px;">5</td> <td style="padding: 2px;">El actor <u>Administrador (ACT-0001)</u> introduce los datos requeridos</td> </tr> <tr> <td style="padding: 2px;">6</td> <td style="padding: 2px;">El sistema añade la <u>asignatura</u> a la titulación e informa de que el proceso se ha realizado correctamente</td> </tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso <u>Identificación de usuario (UC-0001)</u>	2	El sistema muestra las titulaciones existentes	3	El actor <u>Administrador (ACT-0001)</u> selecciona la titulación deseada	4	El sistema solicita el nombre o código de la <u>asignatura</u>	5	El actor <u>Administrador (ACT-0001)</u> introduce los datos requeridos	6	El sistema añade la <u>asignatura</u> a la titulación e informa de que el proceso se ha realizado correctamente
Paso	Acción														
1	Se realiza el caso de uso <u>Identificación de usuario (UC-0001)</u>														
2	El sistema muestra las titulaciones existentes														
3	El actor <u>Administrador (ACT-0001)</u> selecciona la titulación deseada														
4	El sistema solicita el nombre o código de la <u>asignatura</u>														
5	El actor <u>Administrador (ACT-0001)</u> introduce los datos requeridos														
6	El sistema añade la <u>asignatura</u> a la titulación e informa de que el proceso se ha realizado correctamente														
Postcondición	Se ha añadido la <u>asignatura</u> a la titulación														
Excepciones	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">Paso</td> <td style="width: 90%;">Acción</td> </tr> <tr> <td style="padding: 2px;">6</td> <td style="padding: 2px;">Si el sistema no ha podido guardar los cambios porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto</td> </tr> <tr> <td style="padding: 2px;">6</td> <td style="padding: 2px;">Si la <u>asignatura</u> ya está inscrita en dicha titulación , el sistema muestra el error por pantalla acompañado de los datos de la <u>asignatura</u> , a continuación este caso de uso queda sin efecto</td> </tr> <tr> <td style="padding: 2px;">6</td> <td style="padding: 2px;">Si la <u>asignatura</u> no existe en el sistema, el sistema muestra un error por pantalla y solicita una nueva <u>asignatura</u> , a continuación este caso de uso continúa</td> </tr> </table>	Paso	Acción	6	Si el sistema no ha podido guardar los cambios porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto	6	Si la <u>asignatura</u> ya está inscrita en dicha titulación , el sistema muestra el error por pantalla acompañado de los datos de la <u>asignatura</u> , a continuación este caso de uso queda sin efecto	6	Si la <u>asignatura</u> no existe en el sistema, el sistema muestra un error por pantalla y solicita una nueva <u>asignatura</u> , a continuación este caso de uso continúa						
Paso	Acción														
6	Si el sistema no ha podido guardar los cambios porque no tiene acceso a la base de datos, el sistema muestra el error por pantalla y da la opción de reintentarlo, a continuación este caso de uso queda sin efecto														
6	Si la <u>asignatura</u> ya está inscrita en dicha titulación , el sistema muestra el error por pantalla acompañado de los datos de la <u>asignatura</u> , a continuación este caso de uso queda sin efecto														
6	Si la <u>asignatura</u> no existe en el sistema, el sistema muestra un error por pantalla y solicita una nueva <u>asignatura</u> , a continuación este caso de uso continúa														
Importancia	importante														
Urgencia	hay presión														
Estado	validado														
Estabilidad	alta														

UC-0039	modificar asignaturas de titulación						
Versión	1.0 (29/09/2006)						
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>						
Dependencias	• [OBJ-0011] Establecer una administración vía Web						
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el <u>administrador</u> desee modificar las asignaturas pertenecientes a las titulaciones						
Precondición	Existen asignaturas en el sistema						
Secuencia normal	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Paso</th> <th style="text-align: left; padding: 2px;">Acción</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">Se realiza el caso de uso <u>Identificación de usuario (UC-0001)</u></td> </tr> <tr> <td style="padding: 2px;">2</td> <td style="padding: 2px;">El sistema muestra las titulaciones existentes</td> </tr> </tbody> </table>	Paso	Acción	1	Se realiza el caso de uso <u>Identificación de usuario (UC-0001)</u>	2	El sistema muestra las titulaciones existentes
Paso	Acción						
1	Se realiza el caso de uso <u>Identificación de usuario (UC-0001)</u>						
2	El sistema muestra las titulaciones existentes						

	3	El actor Administrador (ACT-0001) selecciona la titulación deseada
	4	El sistema muestra una lista con todas las asignaturas del sistema precedidas con una casilla que indica si la asignatura pertenece actualmente a la titulación
	5	El actor Administrador (ACT-0001) modifica la lista de las asignaturas, marcando o no la correspondiente casilla.
	6	El sistema almacena todas las modificaciones que haya sufrido la lista de asignaturas
Postcondición		Se han modificado las asignaturas asociadas a las titulaciones
Importancia		importante
Urgencia		hay presión
Estado		validado
Estabilidad		alta

7.4 Otros requisitos funcionales

FRQ-0001	Imprimir horario de tutorías
Versión	1.0 (29/09/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0002] Información sobre las tutorías • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores
Descripción	El sistema deberá elaborar un horario de tutorías de forma que pueda ser imprimido a petición del profesor
Importancia	quedaría bien
Urgencia	puede esperar
Estado	validado
Estabilidad	media

FRQ-0002	Elaborar calendario semanal de citas
Versión	1.0 (29/09/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores
Descripción	El sistema deberá elaborar un calendario con todas las citas previstas para una determinada semana de forma que se pueda imprimir a petición del profesor .
Importancia	quedaría bien
Urgencia	puede esperar

REQUISITOS DEL SISTEMA

Estado	pendiente de validación
Estabilidad	media

FRQ-0003	Imprimir ficha personal
Versión	1.0 (29/09/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [OBJ-0001] Gestionar toda la información referente a los alumnos y profesores
Descripción	El sistema deberá permitir imprimir al profesor la ficha personal de los alumnos
Importancia	quedaría bien
Urgencia	puede esperar
Estado	pendiente de validación
Estabilidad	alta

FRQ-0004	Imprimir resguardo de petición
Versión	1.0 (29/09/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [OBJ-0005] Controlar el proceso de citación entre alumnos y profesores
Descripción	El sistema deberá permitir a los usuarios imprimir un resguardo de la peticiones de citas realizadas
Importancia	quedaría bien
Urgencia	puede esperar
Estado	validado
Estabilidad	media

7.5 Requisitos no funcionales

NFR-0001	Seguridad
Versión	1.0 (04/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla

REQUISITOS DEL SISTEMA

Dependencias	Ninguno
Descripción	El sistema deberá proveer de mecanismos de seguridad para un acceso seguro y proteger la información de los usuarios. Un alumno no podrá ver la ficha de otro alumno por motivos de privacidad, solo personal autorizado como el profesor puede hacerlo.
Importancia	vital
Urgencia	hay presión
Estado	validado
Estabilidad	alta
Comentarios	Haremos especial hincapié a las principales vulnerabilidades de las aplicaciones Web. Parámetros sin validar, desbordamientos de memoria intermedia, gestión modificada de sesión y de cuentas, problemas en la gestión de errores, configuración inadecuada de un servidor de aplicaciones y de Web.

NFR-0002	Portabilidad
Versión	1.0 (04/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	Ninguno
Descripción	El sistema deberá poder ser accedido por los navegadores más actuales y más utilizados como son Mozilla FireFox, Opera y Internet Explorer
Importancia	importante
Urgencia	hay presión
Estado	validado
Estabilidad	alta

NFR-0003	Internacionalización
Versión	1.0 (04/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	Ninguno
Descripción	El sistema deberá ofrecer la información mas relevante en inglés, además de en castellano para que así puedan acceder a él alumnos extranjeros
Importancia	quedaría bien
Urgencia	puede esperar
Estado	pendiente de validación
Estabilidad	media

NFR-0004	Escalabilidad
Versión	1.0 (04/08/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	Ninguno
Descripción	El sistema deberá proveer de mecanismos para no ver comprometido el rendimiento de la aplicación con el acceso de muchos usuarios registrados simultáneos a la aplicación Web
Importancia	importante
Urgencia	hay presión
Estado	validado
Estabilidad	media

7.6 Matriz de restreabilidad objetivos/requisitos

TRM-0001	OBJ-0001	OBJ-0005	OBJ-0009	OBJ-0010	OBJ-0011	OBJ-0012	OBJ-0014
IRQ-0001	↑	-	-	-	-	-	-
IRQ-0002	-	↑	-	-	-	-	-
IRQ-0003	-	↑	-	-	-	-	-
IRQ-0004	↑	↑	↑	↑	↑	↑	-
IRQ-0005	-	-	-	-	-	-	↑
IRQ-0006	-	-	-	-	-	-	↑
IRQ-0007	-	↑	↑	-	-	↑	↑

Matriz de rastreabilidad: Matriz de rastreabilidad Objetivos/requisitos de información

REQUISITOS DEL SISTEMA

TRM-0002	OBJ-0001	OBJ-0005	OBJ-0009	OBJ-0010	OBJ-0011	OBJ-0012	OBJ-0014
UC-0001	-	-	-	↑	-	↑	-
UC-0002	-	-	↑	-	-	-	-
UC-0004	-	↑	-	-	-	-	-
UC-0005	-	↑	-	-	-	-	-
UC-0006	-	↑	-	-	-	-	-
UC-0007	-	↑	-	-	-	-	-
UC-0008	↑	-	-	-	-	-	-
UC-0009	-	↑	-	-	-	-	-
UC-0010	-	-	-	-	-	-	↑
UC-0011	-	-	-	-	-	-	↑
UC-0012	-	-	-	-	-	-	↑
UC-0013	-	-	-	-	-	-	↑
UC-0014	-	-	↑	-	-	-	-
UC-0015	-	-	-	-	-	-	↑
UC-0016	-	↑	-	-	-	-	-
UC-0017	-	↑	-	-	-	-	-
UC-0018	-	↑	-	-	-	-	-
UC-0019	-	↑	-	-	-	-	-
UC-0020	-	-	-	-	↑	-	↑
UC-0023	-	-	-	-	↑	-	↑
UC-0026	-	-	-	-	↑	-	↑
UC-0027	-	-	-	-	↑	-	-
UC-0028	-	-	-	-	↑	-	-
UC-0030	-	-	-	-	↑	-	-
UC-0031	-	-	-	-	↑	-	-
UC-0032	-	-	-	-	↑	-	↑
UC-0034	-	-	-	-	↑	-	↑
UC-0035	-	-	-	-	↑	-	-
UC-0036	-	-	-	-	↑	-	-
UC-0038	-	-	-	-	↑	-	-
UC-0039	-	-	-	-	↑	-	-
UC-0040	↑	-	-	-	-	-	-
UC-0041	↑	-	-	-	-	-	-
UC-0042	↑	-	-	-	-	-	-
UC-0043	-	↑	↑	-	-	-	-
UC-0044	↑	-	-	-	-	-	-

Matriz de rastreabilidad: Matriz de rastreabilidad Objetivos/requisitos de funcionalidad

8. Análisis del sistema

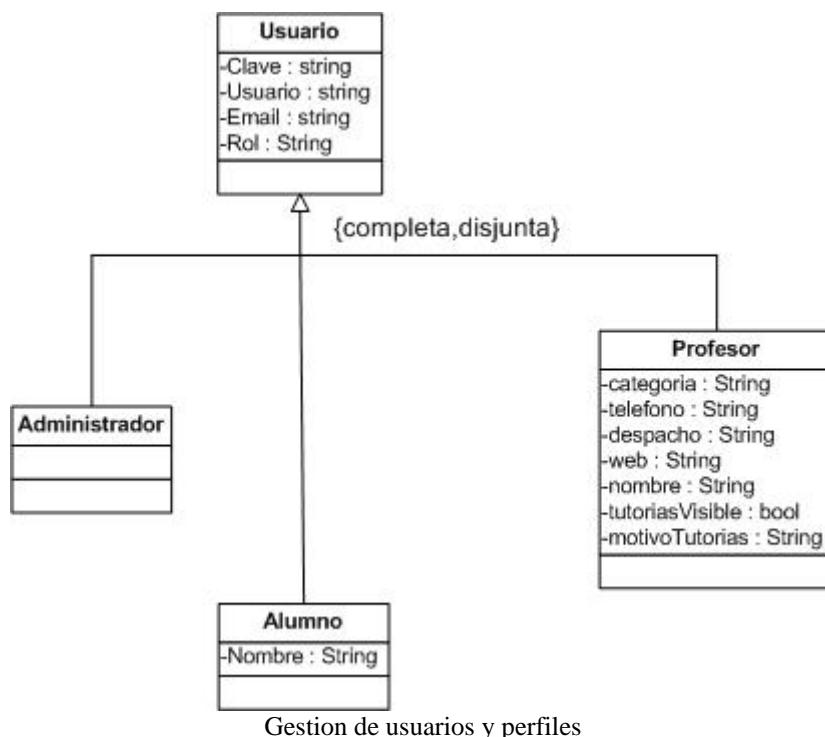
En este documento realizaremos el análisis de nuestra aplicación, obteniendo una visión más concreta de nuestra aplicación, mediante clases conceptuales del mundo real.

8.1 Modelo estático del sistema

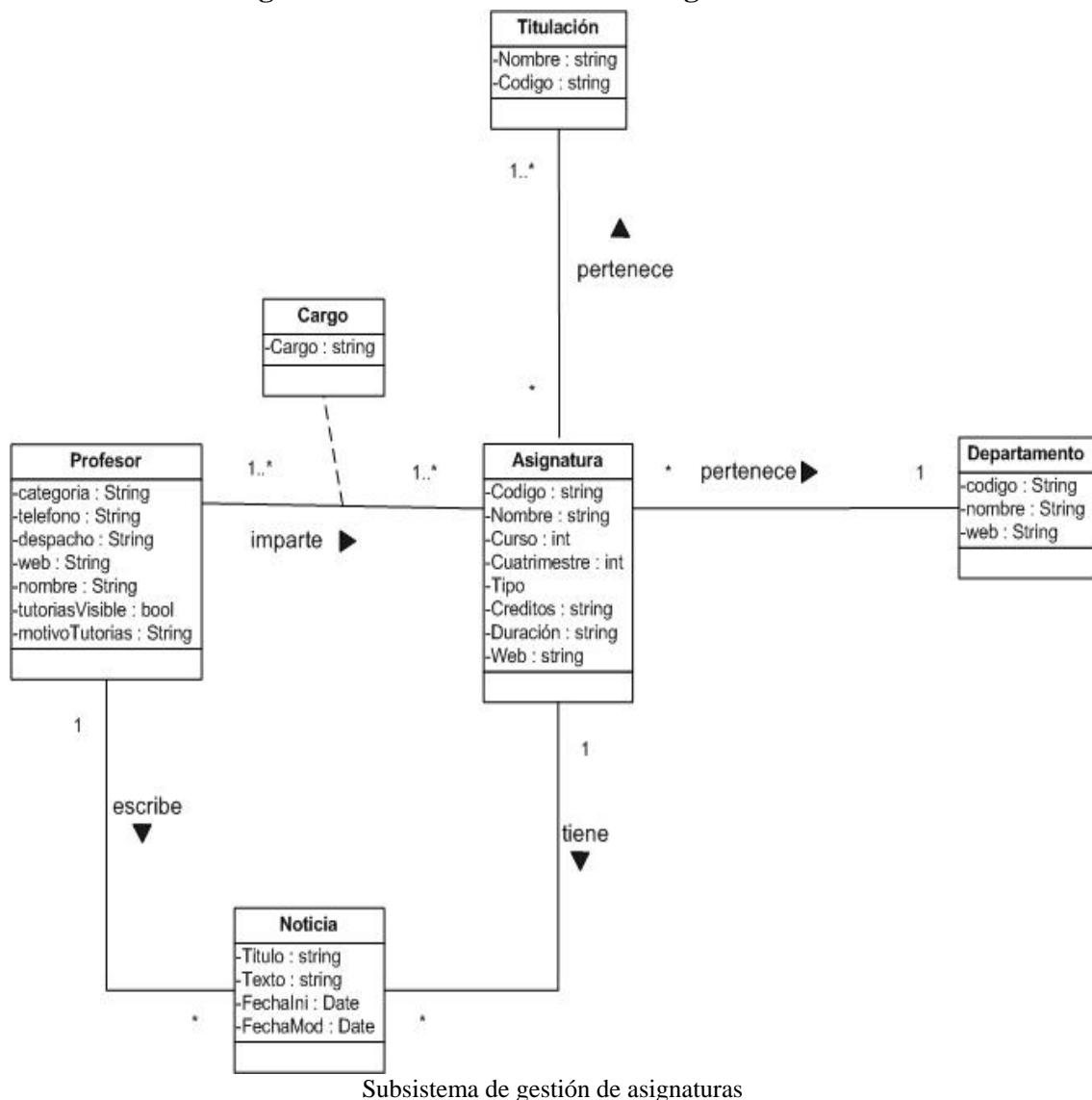
Esta sección describe el modelo estático del sistema, es decir, los tipos de objetos y las asociaciones entre tipos de objetos que componen el estado del sistema y que se han identificado durante la elicitudación.

8.1.1 Diagramas

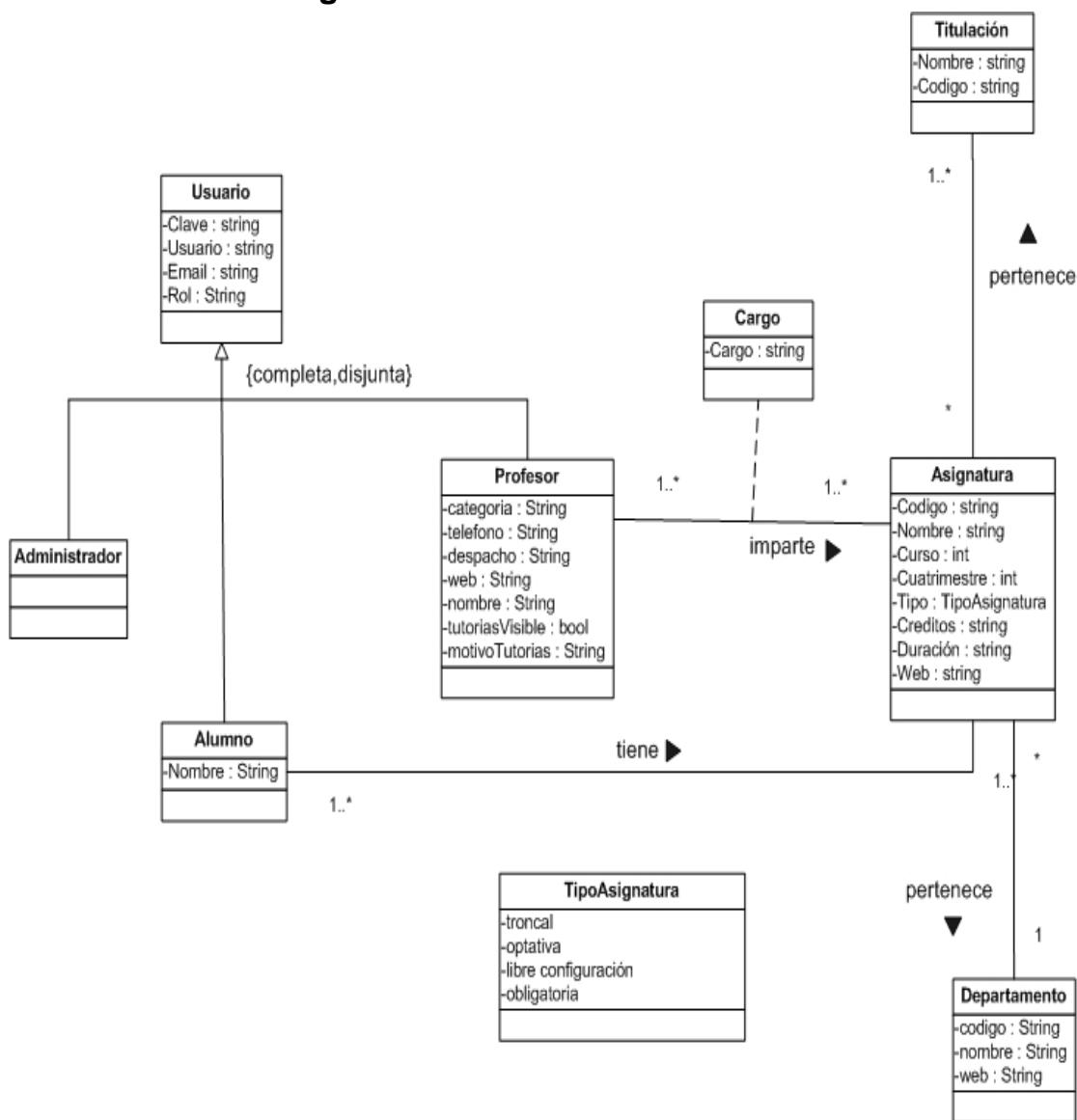
8.1.1.1 Subsistema gestión de usuarios y perfiles



8.1.1.2 Subsistema gestión de información de las asignaturas

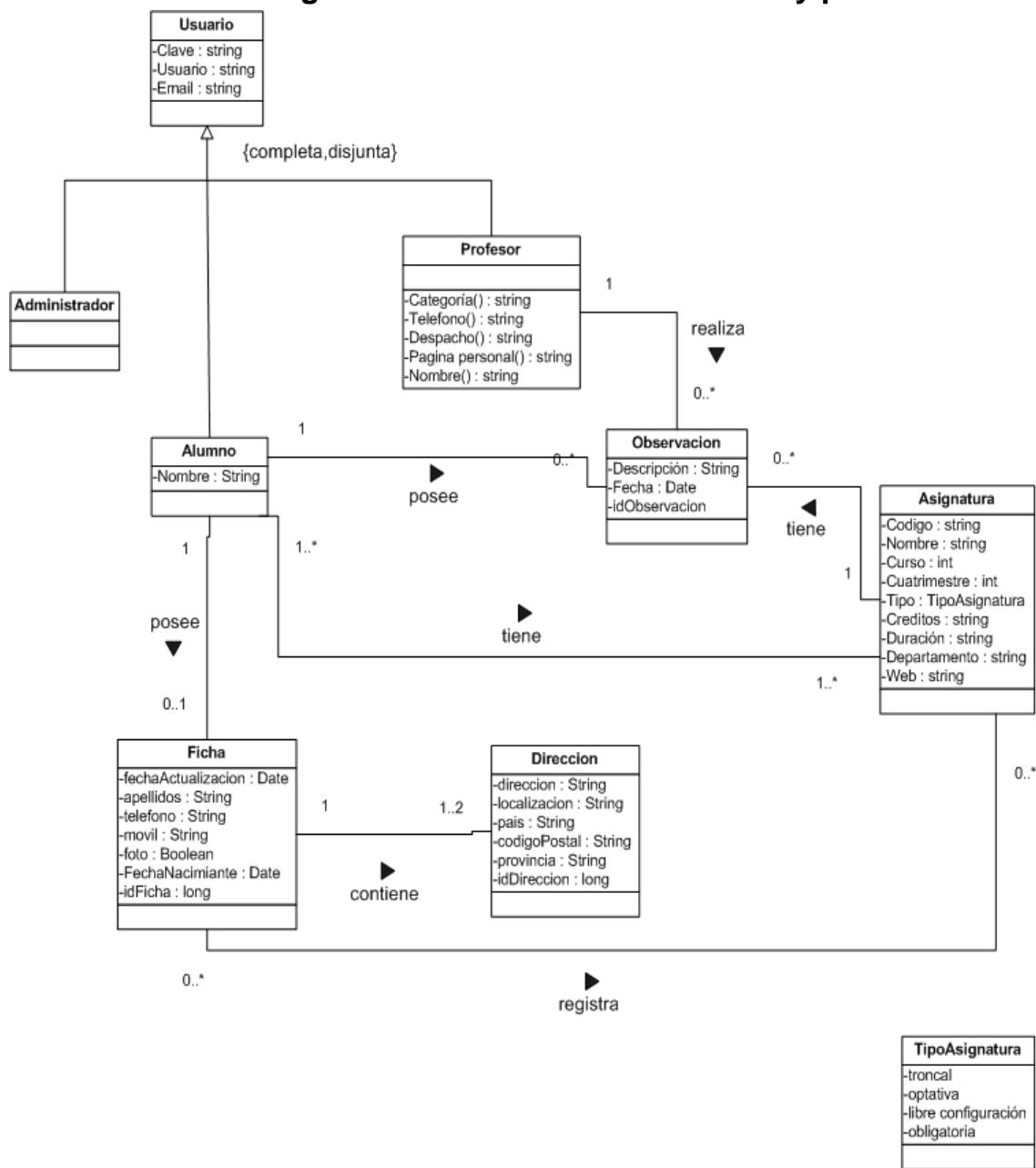


8.1.1.3 Subsistema gestión administrativa vía Web



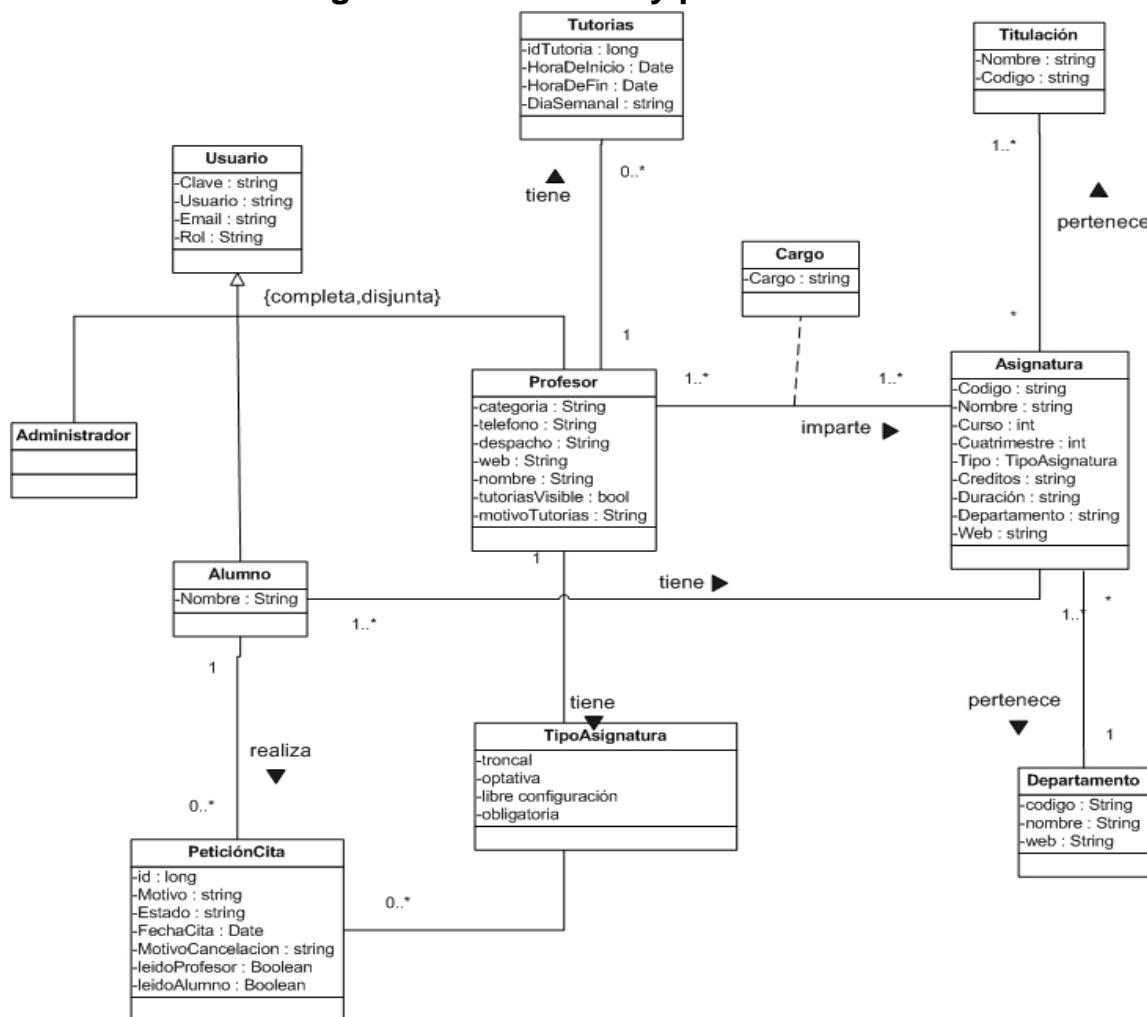
Gestión administrativa via web

8.1.1.4 Subsistema gestión de información alumnos y profesores



Gestión de información alumnos y profesores

8.1.1.5 Subsistema gestión de tutorías y peticiones de cita



Subsistema gestión de tutorías y peticiones de cita

8.1.2 Tipos

8.1.2.1 Subsistema gestión de usuarios y perfiles

TYP-0002	Usuario
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> [IRQ-0004] Información sobre los usuarios
Descripción	Este tipo de objetos representa <i>usuario de la aplicación</i>
Subtipos (disjuntos)	<ul style="list-style-type: none"> [TYP-0004] Profesor [TYP-0007] Alumno

Atributo variable	Usuario:: usuario
Descripción	Este atributo representa <i>representa al usuario</i>

Tipo	String
Comentarios	en nuestra aplicación este será su dni

Atributo variable	Usuario:: clave
Descripción	Este atributo representa <i>clave que permite entrar con el perfil de usuario</i>
Tipo	String
Comentarios	Ninguno

Atributo variable	Usuario:: rol
Descripción	Este atributo representa <i>al rol del usuario, es decir el tipo de usuario</i>
Tipo	String
Comentarios	Ninguno

Atributo variable	Usuario:: email
Descripción	Este atributo representa <i>al correo electrónico del usuario</i>
Tipo	String
Comentarios	Ninguno

TYP-0004	Profesor
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0004] Información sobre los usuarios • [IRQ-0007] Información sobre los profesores
Descripción	Este tipo de objetos representa <i>al profesor que usa la aplicación</i>
Supertipo	[TYP-0002] Usuario

Atributo variable	Profesor:: categoría
Descripción	Este atributo representa <i>al tipo de profesor que es</i>
Tipo	String

Atributo variable	Profesor:: teléfono
Descripción	Este atributo representa <i>teléfono del despacho del profesor</i>
Tipo	String

Atributo variable	Profesor:: despacho
Descripción	Este atributo representa <i>dirección en la que se encuentra el despacho del profesor</i>
Tipo	String

Atributo variable	Profesor:: web
Descripción	Este atributo representa <i>Dirección web donde se encuentra la página personal del profesor</i>
Tipo	String

Atributo variable	Profesor:: nombre
Descripción	Este atributo representa <i>Nombre completo del profesor</i>
Tipo	String

Atributo variable	Profesor:: tutoriaVisible
Descripción	Este atributo representa <i>al estado de la presentación de las tutorías, si está activado que se muestren o no</i>
Tipo	Natural

Atributo variable	Profesor:: motivoTutoria
Descripción	Este atributo representa <i>al motivo por el que no se muestra el horario de tutorías</i>
Tipo	String

TYP-0007	Alumno
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0004] Información sobre los usuarios
Descripción	Este tipo de objetos representa <i>al usuario alumno de la aplicación</i>
Supertipo	[TYP-0002] Usuario

Atributo variable	Alumno:: Nombre
Descripción	Este atributo representa <i>al nombre completo del usuario</i>
Tipo	String

8.1.2.2 Subsistema gestión de información de las asignaturas

TYP-0001	Asignatura
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de objetos representa <i>las asignaturas que registra el sistema</i>

ANÁLISIS DEL SISTEMA

Componentes	Nombre	Tipo	Multiplicidad
	tipo	TipoAsignatura	1
Atributo variable	Asignatura:: codigo		
Descripción	Este atributo representa el <i>identificador de la asignatura</i>		
Tipo	Integer		

Atributo variable	Asignatura:: nombre
Descripción	Este atributo representa el <i>nombre de la asignatura</i>
Tipo	String

Atributo variable	Asignatura:: curso
Descripción	Este atributo representa el <i>curso en el que se imparte la asignatura</i>
Tipo	Natural

Atributo variable	Asignatura:: cuatrimestre
Descripción	Este atributo representa <i>cuatrimestre en el que empieza la asignatura</i>
Tipo	Natural

Atributo variable	Asignatura:: creditos
Descripción	Este atributo representa <i>creditos que le corresponden a la asignatura</i>
Tipo	Real

Expresión de invariante	Asignatura::Unicidad del codigo de la asignatura
Descripción	no puede haber dos asignaturas con el mismo codigo
Expresión OCL	
Comentarios	el codigo es la clave

TYP-0003	Noticia
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0006] Información sobre las noticias
Descripción	Este tipo de objetos representa <i>representa a la noticia asociada a una asignatura y escrita por un profesor</i>

Atributo variable	Noticia:: idNoticia
Descripción	Este atributo representa al <i>identificador de la noticia</i>
Tipo	Integer

Atributo variable	Noticia:: idNoticia
--------------------------	----------------------------

ANÁLISIS DEL SISTEMA

Descripción	Este atributo representa <i>al identificador de la noticia</i>
Tipo	Integer

Atributo variable	Noticia:: texto
Descripción	Este atributo representa <i>el texto de la noticia</i>
Tipo	String

Atributo variable	Noticia:: fechaMod
Descripción	Este atributo representa <i>la última fecha en la que se ha modificado la noticia</i>
Tipo	Date

Atributo variable	Noticia:: fechalni
Descripción	Este atributo representa <i>la fecha en la que se ha creado la noticia</i>
Tipo	Date

Expresión de invariante	Noticia::Unicidad de IdNoticia
Descripción	No puede existir dos noticias con el mismo IdNoticia
Comentarios	es la clave

TYP-0008	Titulación
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de objetos representa <i>a la titulación que se imparte</i>

Atributo variable	Titulación:: nombre
Descripción	Este atributo representa <i>nombre completo de la titulación que se imparte</i>
Tipo	String

Atributo variable	Titulación:: codigo
Descripción	Este atributo representa <i>al codigo que tiene la titulación</i>
Tipo	Integer

Expresión de invariante	Titulación::Unicidad del código de la titulación
Descripción	No pueden existir dos titulaciones con el mismo código

TYP-0012	TipoAsignatura
-----------------	-----------------------

Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de objetos representa <i>a los distintos tipos de asignaturas que hay</i>

TYP-0013	Departamento
Versión	1.0 (10/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de objetos representa <i>al departamento que imparte asignaturas</i>

Atributo variable	Departamento:: nombre
Descripción	Este atributo representa <i>nombre del departamento</i>
Tipo	String

Atributo variable	Departamento:: web
Descripción	Este atributo representa <i>dirección web del departamento</i>
Tipo	String

Atributo variable	Departamento:: idDepartamento
Descripción	Este atributo representa <i>identificador del departamento</i>
Tipo	String

ASO-0001	tiene(Asignatura, Noticia)
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0005] Información sobre las asignaturas • [IRQ-0006] Información sobre las noticias
Descripción	Este tipo de asociación representa el hecho de que <i>de que las asignaturas podrán tener noticias asociadas</i>

Rol variable	tiene(Asignatura, Noticia):: asignatura
Descripción	Este rol representa <i>a la asignatura que tiene noticias</i>
Tipo	Asignatura
Multiplicidad	1

Rol variable	tiene(Asignatura, Noticia):: noticia
Descripción	Este rol representa <i>la noticia que tiene una asignatura</i>
Tipo	Set(Noticia)
Multiplicidad	0.. *

ASO-0002	escribe(Profesor, Noticia)
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0006] Información sobre las noticias • [IRQ-0007] Información sobre los profesores
Descripción	Este tipo de asociación representa el hecho de que <i>que un profesor escribe una noticia para una asignatura</i>

Rol variable	escribe(Profesor, Noticia):: profesor
Descripción	Este rol representa <i>al profesor que escribe una noticia</i>
Tipo	Profesor
Multiplicidad	1

Rol variable	escribe(Profesor, Noticia):: noticia
Descripción	Este rol representa <i>a la noticia que escribe un profesor</i>
Tipo	Set(Noticia)
Multiplicidad	0.. *

ASO-0006	imparte(Profesor, Asignatura)
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0005] Información sobre las asignaturas • [IRQ-0007] Información sobre los profesores
Descripción	Este tipo de asociación representa el hecho de que <i>que un profesor imparte una asignatura</i>

Rol variable	imparte(Profesor, Asignatura):: profesor
Descripción	Este rol representa <i>que el profesor imparte asignaturas</i>
Tipo	Set(Profesor)
Multiplicidad	1.. *

Rol variable	imparte(Profesor, Asignatura):: asignatura
Descripción	Este rol representa <i>a las asignaturas que imparte un profesor</i>

ANÁLISIS DEL SISTEMA

Tipo	Asignatura
Multiplicidad	1

Atributo variable	imparte(Profesor, Asignatura):: cargo
Descripción	Este atributo representa <i>al tipo de cargo que tiene en esa asignatura</i> (coordinador, profesor de practicas...)
Tipo	String

ASO-0013	pertenece(Asignatura, Titulación)
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de asociación representa el hecho de que <i>una titulación tiene una asignatura</i> y <i>una asignatura pertenece a una o varias titulaciones</i>

Rol variable	pertenece(Asignatura, Titulación):: asignatura
Descripción	Este rol representa <i>a la asignatura que pertenece a una titulación</i>
Tipo	Asignatura
Multiplicidad	1

Rol variable	pertenece(Asignatura, Titulación):: titulacion
Descripción	Este rol representa <i>la titulación que esta compuesta de asignaturas</i>
Tipo	Titulación
Multiplicidad	1

ASO-0014	pertenece(Asignatura, Departamento)
Versión	1.0 (10/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de asociación representa el hecho de que <i>de qe una asignatura pertenece a un departamento</i>

Rol variable	pertenece(Asignatura, Departamento):: asignatura
Descripción	Este rol representa <i>a la asignatura que pertenece a un departamento</i>
Tipo	Asignatura
Multiplicidad	1

Rol variable	pertenece(Asignatura, Departamento):: departamento
---------------------	---

Descripción	Este rol representa <i>al departamento que imparte asignaturas</i>
Tipo	Departamento
Multiplicidad	1

8.1.2.3 Subsistema gestión de información alumnos y profesores

TYP-0009	Dirección
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0001] Información sobre Fichas
Descripción	Este tipo de objetos representa <i>las dirección que posee un alumno</i>

Atributo variable	Dirección:: direccion
Descripción	Este atributo representa <i>la dirección del alumno</i> (calle y numero)
Tipo	String

Atributo variable	Dirección:: localización
Descripción	Este atributo representa <i>a la localidad a la que pertenece o reside el alumno</i>
Tipo	String

Atributo variable	Dirección:: pais
Descripción	Este atributo representa <i>al pais al que pertenece la localidad del alumno</i>
Tipo	String

Atributo variable	Dirección:: codigoPostal
Descripción	Este atributo representa <i>el codigo postal de la localidad</i>
Tipo	String

Atributo variable	Dirección:: provincia
Descripción	Este atributo representa <i>a la provincia a la que pertenece la localidad del alumno</i>
Tipo	String

Atributo variable	Dirección:: IdDireccion
Descripción	Este atributo representa <i>al identificador de dirección</i>
Tipo	Integer

Expresión de invariante	Dirección::Unicidad del atributo idDirección
Descripción	No pueden existir dos direcciones con un mismo idDirección

TYP-0010	Ficha
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0001] Información sobre Fichas
Descripción	Este tipo de objetos representa <i>la ficha que recoge los datos personales del alumno</i>

Atributo variable	Ficha:: fechaActualización
Descripción	Este atributo representa <i>la fecha de la ultima vez que fue modificada la ficha</i>
Tipo	Date

Atributo variable	Ficha:: apellidos
Descripción	Este atributo representa <i>los apellidos del alumno</i>
Tipo	String

Atributo variable	Ficha:: telefono
Descripción	Este atributo representa <i>el telefono de contacto del alumno</i>
Tipo	String

Atributo variable	Ficha:: movil
Descripción	Este atributo representa <i>el telefono movil de contacto del alumno</i>
Tipo	String

Atributo variable	Ficha:: foto
Descripción	Este atributo representa <i>si el alumno posee o no fotografía personal</i>
Tipo	Integer
Comentarios	Puede tomar dos valores. Un "0" indica que no posee ficha y un "1" indica que si.

Atributo variable	Ficha:: Fecha de nacimiento
Descripción	Este atributo representa <i>la fecha de nacimiento del alumno</i>
Tipo	Date

Atributo variable	Ficha:: idFicha
Descripción	Este atributo representa <i>al identificador de la ficha personal</i>

Tipo	Integer
-------------	---------

Expresión de invariante	Ficha::Unicidad del atributo idFicha
Descripción	No pueden existir dos fichas que tengan el mismo idFicha

TYP-0011	Observación
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0001] Información sobre Fichas
Descripción	Este tipo de objetos representa <i>a una observación que ha realizado un profesor con respecto a un alumno</i>

Atributo variable	Observación:: descripción
Descripción	Este atributo representa <i>la información extra que el profesor a elaborado del alumno</i>
Tipo	String

Atributo variable	Observación:: fecha
Descripción	Este atributo representa <i>la fecha en la que se ha realizado la observación</i>
Tipo	Date

Atributo variable	Observación:: idObservacion
Descripción	Este atributo representa <i>al identificador de la observación</i>
Tipo	Integer

Expresión de invariante	Observación::Unicidad del atributo idObservacion
Descripción	No pueden existir dos observaciones con el mismo idObservacion

ASO-0007	contiene(Ficha, Dirección)
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0001] Información sobre Fichas
Descripción	Este tipo de asociación representa el hecho de que <i>una ficha puede tener registrada una dirección y una dirección de residencia de un alumno</i>

Rol variable	contiene(Ficha, Dirección):: ficha
Descripción	Este rol representa <i>a la ficha que contiene a las direcciones</i>
Tipo	Ficha

Multiplicidad	1
----------------------	---

Rol variable	contiene(Ficha, Dirección):: direccion
Descripción	Este rol representa <i>la dirección que tiene una ficha</i>
Tipo	Set(Dirección)
Multiplicidad	1.. 2

ASO-0008	registra(Ficha, Asignatura)
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de asociación representa el hecho de que <i>una ficha registra las asignaturas que han sido aprobadas por el alumno</i> .

Rol variable	registra(Ficha, Asignatura):: Ficha
Descripción	Este rol representa <i>la ficha que contiene las asignaturas aprobadas</i>
Tipo	Set(Ficha)
Multiplicidad	0.. *

Rol variable	registra(Ficha, Asignatura):: asignaturas
Descripción	Este rol representa <i>las asignaturas aprobadas que se recogen en las fichas</i>
Tipo	Set(Asignatura)
Multiplicidad	0.. *

ASO-0009	Posee(Alumno, Ficha)
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [IRQ-0004] Información sobre los usuarios
Descripción	Este tipo de asociación representa el hecho de que <i>un alumno posee una ficha</i>

Rol variable	Posee(Alumno, Ficha):: Alumno
Descripción	Este rol representa <i>al alumno que posee una ficha</i>
Tipo	Alumno
Multiplicidad	1

Rol variable	Posee(Alumno, Ficha):: Ficha
Descripción	Este rol representa <i>la ficha que posee un alumno</i>
Tipo	Set(Ficha)
Multiplicidad	0.. 1

ASO-0010	Posee(Alumno, Observación)
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [IRQ-0004] Información sobre los usuarios
Descripción	Este tipo de asociación representa el hecho de que <i>que un alumno puede poseer una observación con respecto a una asignatura</i>

Rol variable	Posee(Alumno, Observación):: Alumno
Descripción	Este rol representa <i>el alumno que posee una observación</i>
Tipo	Alumno
Multiplicidad	1

Rol variable	Posee(Alumno, Observación):: Observación
Descripción	Este rol representa <i>la observación que posee un alumno</i>
Tipo	Set(Observación)
Multiplicidad	0.. *

ASO-0011	Realiza(Profesor, Observación)
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [IRQ-0007] Información sobre los profesores
Descripción	Este tipo de asociación representa el hecho de que <i>que un profesor puede realizar una observación sobre un alumno</i>

Rol variable	Realiza(Profesor, Observación):: Profesor
Descripción	Este rol representa <i>el profesor que realiza una observación</i>
Tipo	Profesor
Multiplicidad	1

Rol variable	Realiza(Profesor, Observación):: Observación
Descripción	Este rol representa <i>la observación que realiza un profesor</i>

Tipo	Set(Observación)
Multiplicidad	0.. *

ASO-0012	Tiene(Asignatura, Observación)
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0001] Información sobre Fichas • [IRQ-0005] Información sobre las asignaturas
Descripción	Este tipo de asociación representa el hecho de que se puede realizar una observación para una asignatura para un alumno

Rol variable	Tiene(Asignatura, Observación):: Asignatura
Descripción	Este rol representa la asignatura de la cual se realiza la observación
Tipo	Asignatura
Multiplicidad	1

Rol variable	Tiene(Asignatura, Observación):: Observación
Descripción	Este rol representa la observación que se realiza de una asignatura
Tipo	Set(Observación)
Multiplicidad	0.. *

8.1.2.4 Subsistema gestión de tutorías y peticiones de cita

TYP-0005	PeticionCita
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones
Descripción	Este tipo de objetos representa la petición de cita que realiza un alumno a un profesor

Atributo variable	PeticionCita:: motivo
Descripción	Este atributo representa una descripción breve del motivo de la petición de la tutoría
Tipo	String

Atributo variable	PeticionCita:: fechaCita
Descripción	Este atributo representa la fecha y hora del comienzo de la cita
Tipo	Date

Atributo variable	PeticionCita:: motivoCancelacion
Descripción	Este atributo representa <i>al motivo de cancelación de la cita, o el motivo de la denegación</i>
Tipo	String

Atributo variable	PeticionCita:: estado
Descripción	Este atributo representa <i>al estado de la cita</i>
Tipo	String
Comentarios	sus valores son: en curso, denegada, cancelada, aceptada

Atributo variable	PeticionCita:: id
Descripción	Este atributo representa <i>al identificador de la petición de cita</i>
Tipo	Integer

Atributo variable	PeticionCita:: leidoProfesor
Descripción	Este atributo representa <i>al hecho de si un profesor ha leído la petición o no</i>
Tipo	Natural

Atributo variable	PeticionCita:: leidoAlumno
Descripción	Este atributo representa <i>al hecho de si un alumno ha leído la modificación de la petición o no</i>
Tipo	Natural

Expresión de invariante	PeticionCita::Unicidad en el atributo id
Descripción	No puede haber dos tipos de PeticionCita con el mismo id

TYP-0006	Tutoria
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [IRQ-0002] Información sobre las tutorías
Descripción	Este tipo de objetos representa <i>al horario de tutorías de un día de la semana en concreto</i>

Atributo variable	Tutoria:: id
Descripción	Este atributo representa <i>al identificador de la tutoría</i>
Tipo	Integer

Atributo variable	Tutoria:: diaSemanal
--------------------------	-----------------------------

ANÁLISIS DEL SISTEMA

Descripción	Este atributo representa <i>al día de la semana que corresponde</i>
Tipo	String

Atributo variable	Tutoria:: horalnicio
Descripción	Este atributo representa <i>a la hora de inicio de la tutoría</i>
Tipo	Time

Atributo variable	Tutoria:: horaFin
Descripción	Este atributo representa <i>hora a la que acaba el horario de tutorías</i>
Tipo	Time

Expresión de invariante	Tutoria::Unicidad del atributo id
Descripción	No puede existir dos Tutorías con el mismo id

ASO-0003	tiene(Profesor, Tutoria)
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0002] Información sobre las tutorías • [IRQ-0007] Información sobre los profesores
Descripción	Este tipo de asociación representa el hecho de que <i>que un profesor tiene un horario de tutorías</i>

Rol variable	tiene(Profesor, Tutoria):: profesor
Descripción	Este rol representa <i>al profesor que tiene tutorías</i>
Tipo	Profesor
Multiplicidad	1

Rol variable	tiene(Profesor, Tutoria):: tutoria
Descripción	Este rol representa <i>las tutorías que tiene un profesor</i>
Tipo	Set(Tutoria)
Multiplicidad	0.. *

ASO-0004	realiza(Alumno, PeticionCita)
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [IRQ-0004] Información sobre los usuarios
Descripción	Este tipo de asociación representa el hecho de que <i>un alumno desea enviar</i>

<i>una petición de cita a un profesor</i>

Rol variable	realiza(Alumno, PeticionCita):: alumno
Descripción	Este rol representa <i>al usuario que realiza una petición de cita</i>
Tipo	Alumno
Multiplicidad	1

Rol variable	realiza(Alumno, PeticionCita):: peticion
Descripción	Este rol representa <i>a la petición que realiza un alumno</i>
Tipo	Set(PeticionCita)
Multiplicidad	0.. *

ASO-0005	tiene(Profesor, PeticionCita)
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [IRQ-0003] Información sobre las citaciones • [IRQ-0007] Información sobre los profesores
Descripción	Este tipo de asociación representa el hecho de que <i>que un profesor tiene peticiones de citas</i>

Rol variable	tiene(Profesor, PeticionCita):: profesor
Descripción	Este rol representa <i>al profesor que tiene una cita</i>
Tipo	Profesor
Multiplicidad	1

Rol variable	tiene(Profesor, PeticionCita):: cita
Descripción	Este rol representa <i>a la cita que tiene un profesor</i>
Tipo	PeticionCita
Multiplicidad	1

8.2 Modelo dinámico del sistema

8.2.1 Subsistema gestión de información alumnos y profesores

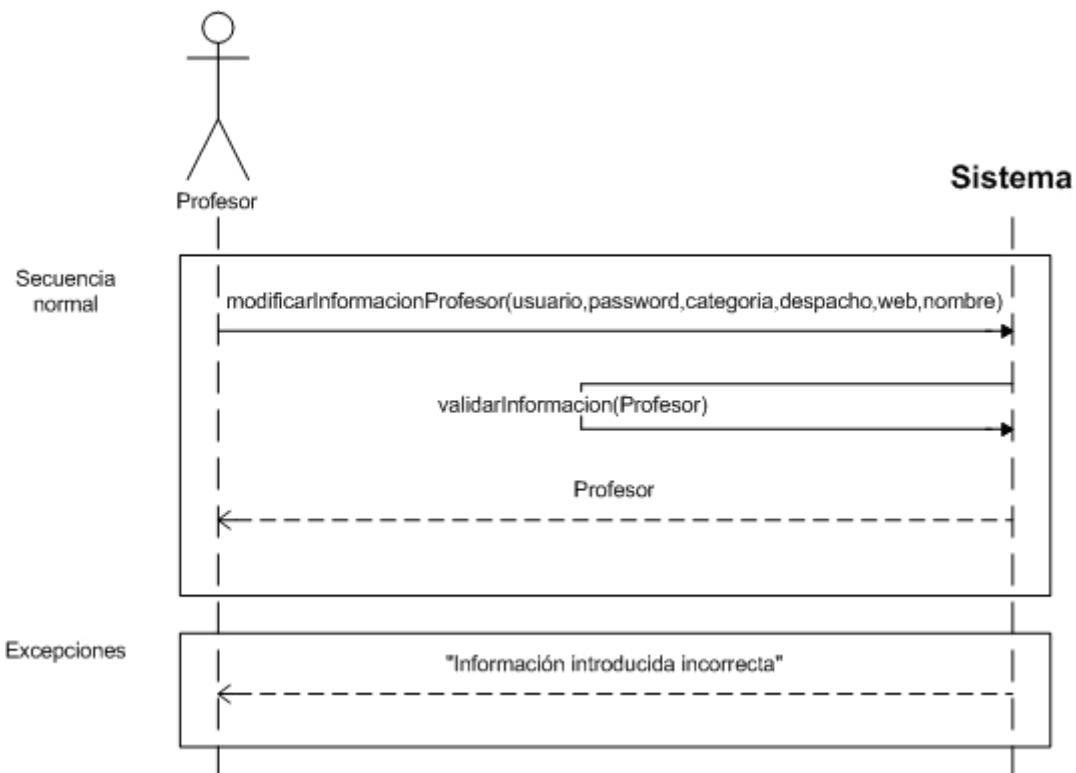


Figura 1:
modificarInformacionProfesor(usuario,password, categoria, despacho, web, nombre)

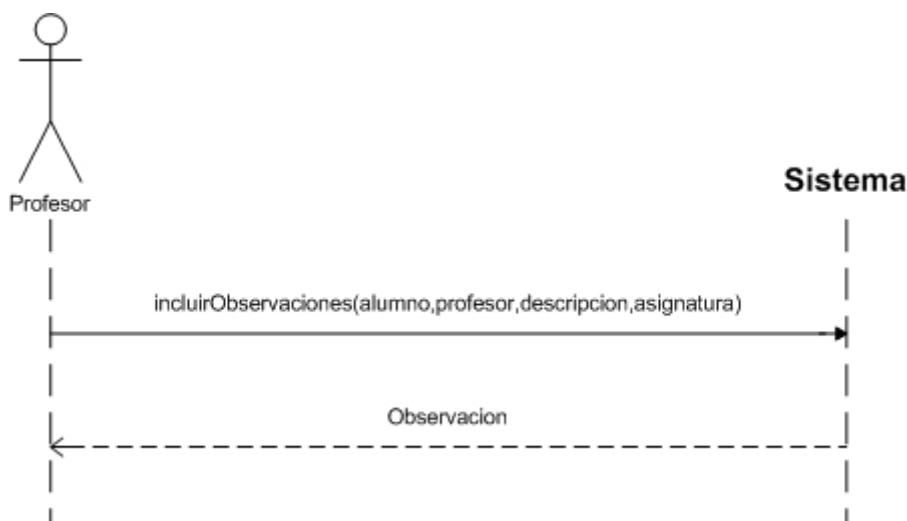


Figura 2: incluirObservaciones(alumno,profesor,descripcion,asignatura)

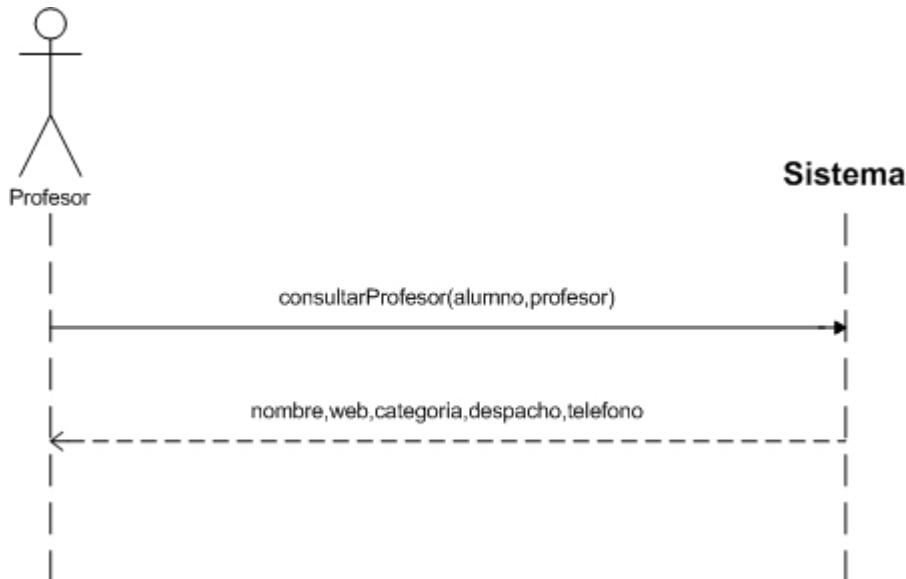
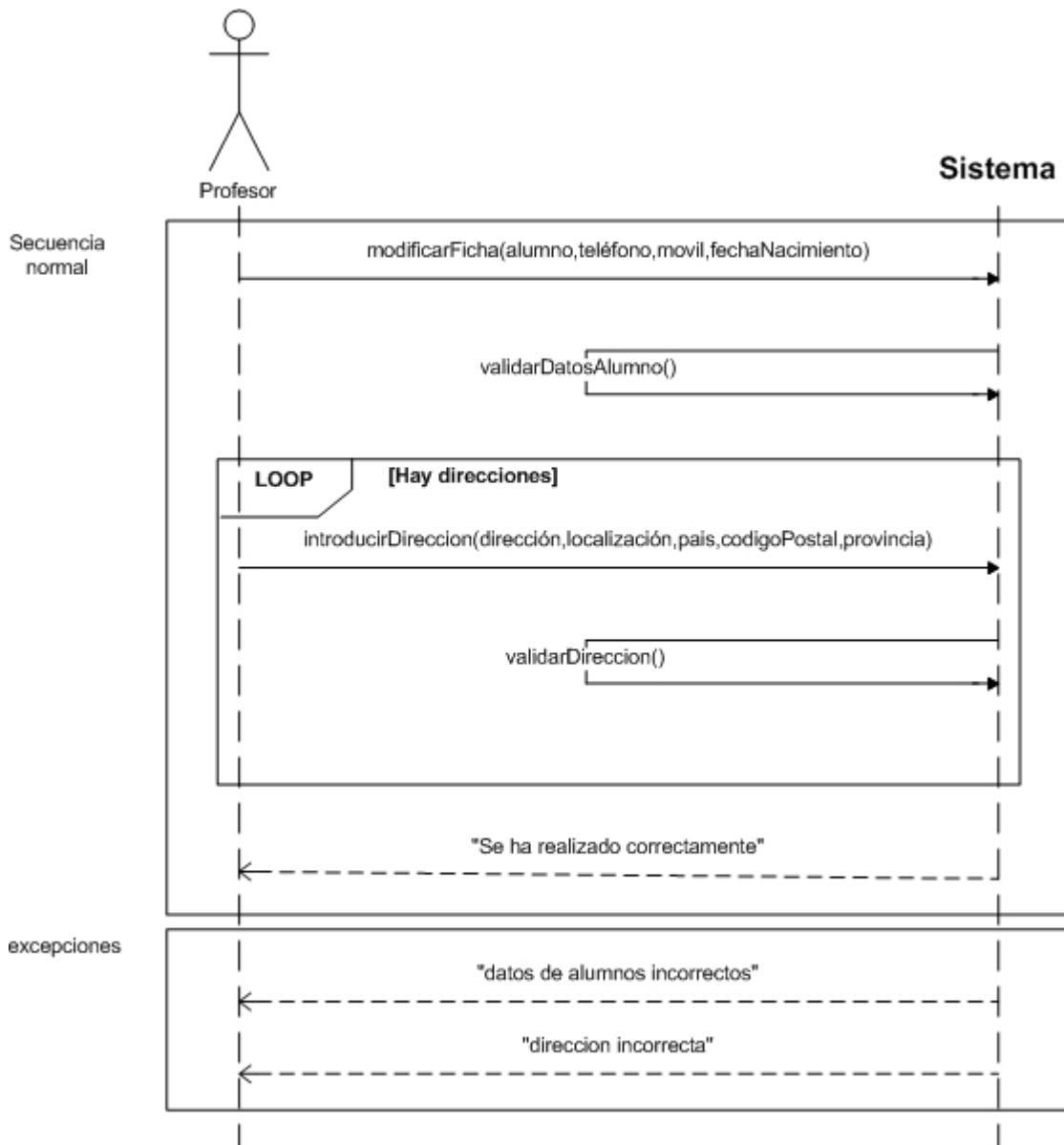


Figura 3: consultarProfesor(alumno,profesor)



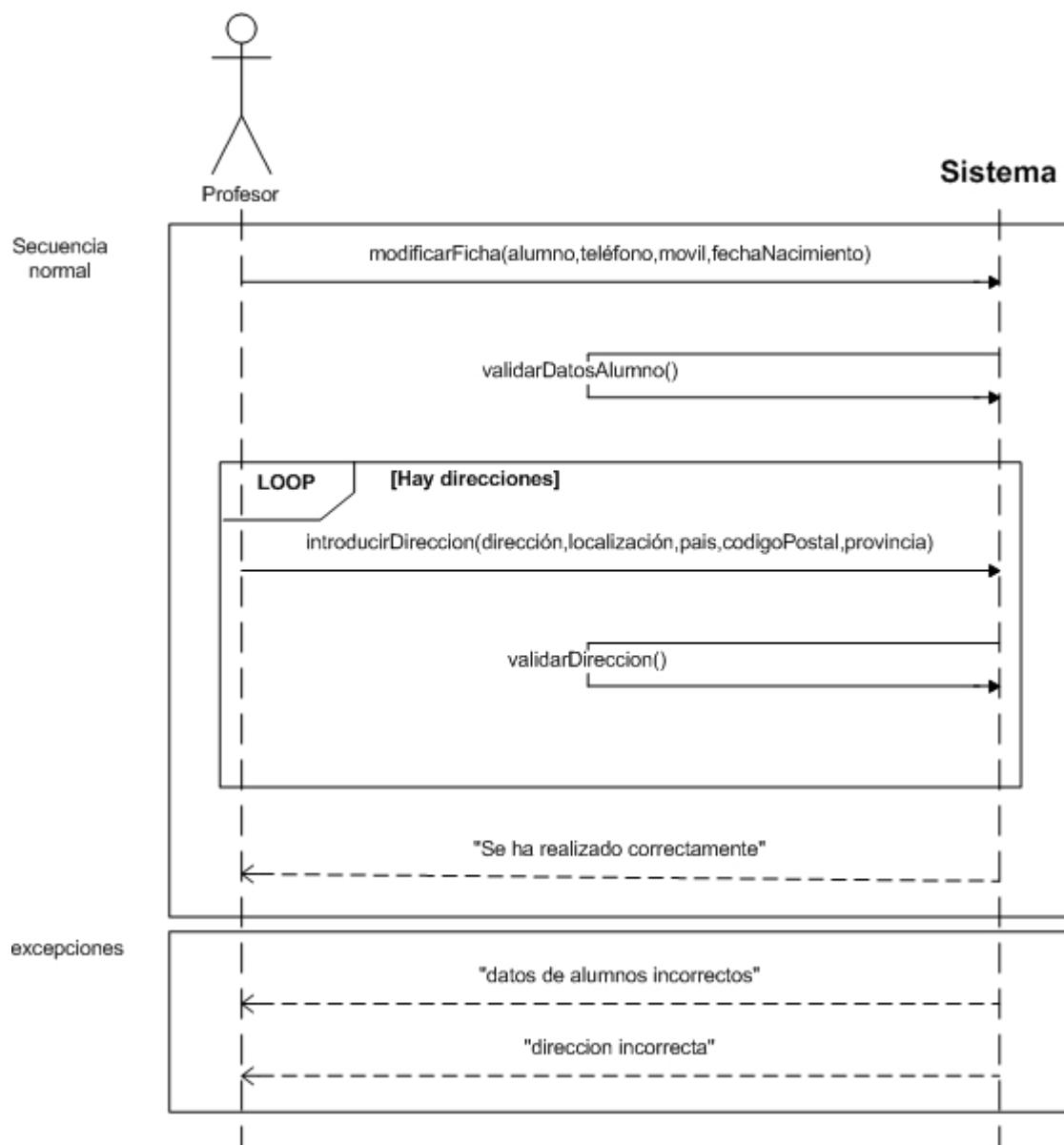


Figura 4: modificarFicha()

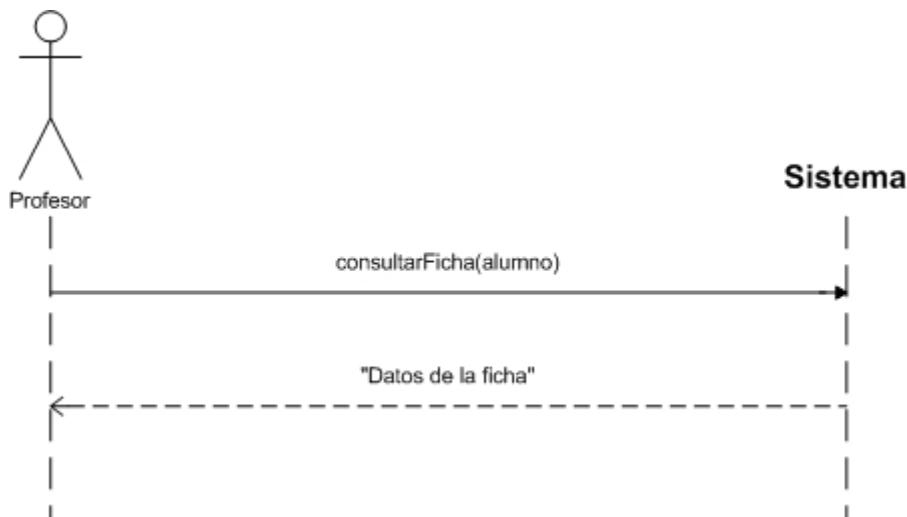


Figura 5: consultarFicha(alumno)

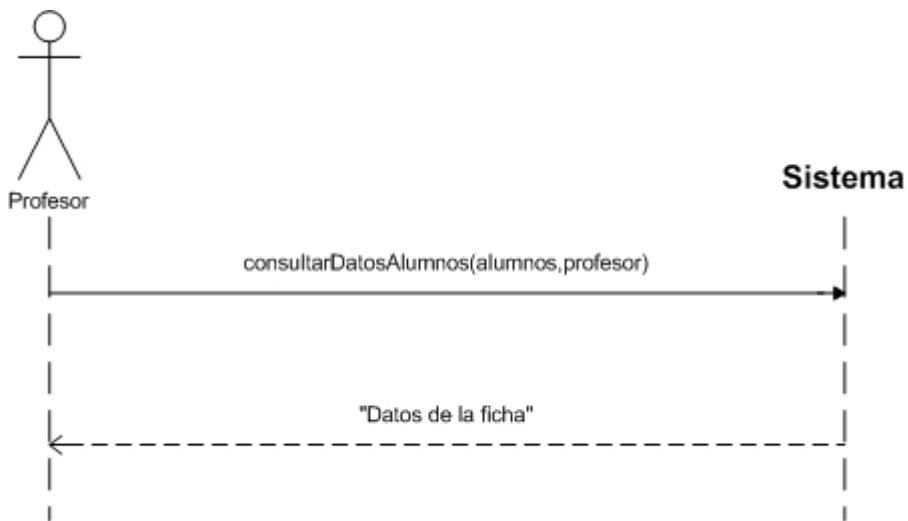


Figura 6: consultarDatosAlumno(alumno,profesor)

8.2.2 Subsistema gestión de información de las asignaturas

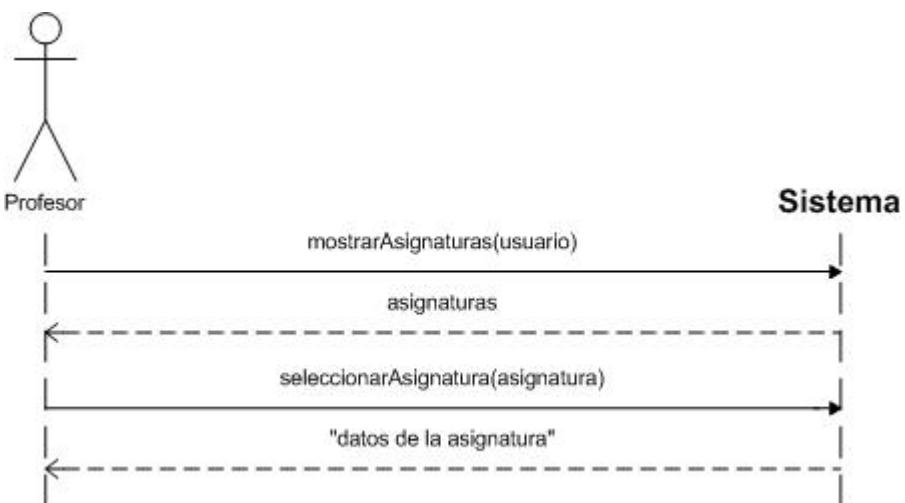
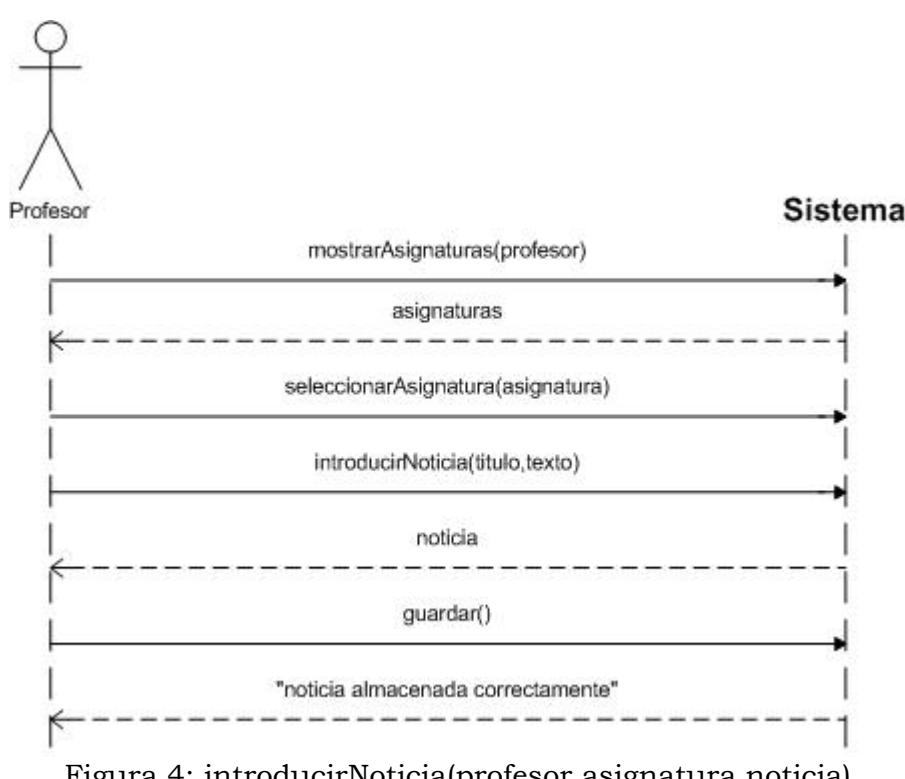
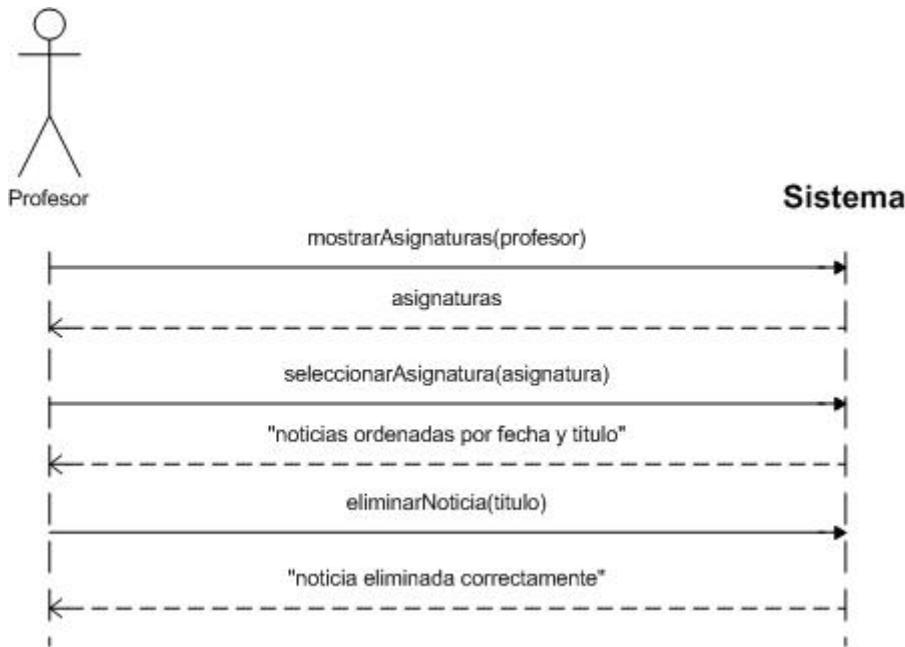


Figura 1: consultarDatosAsignatura(asignatura,usuario)



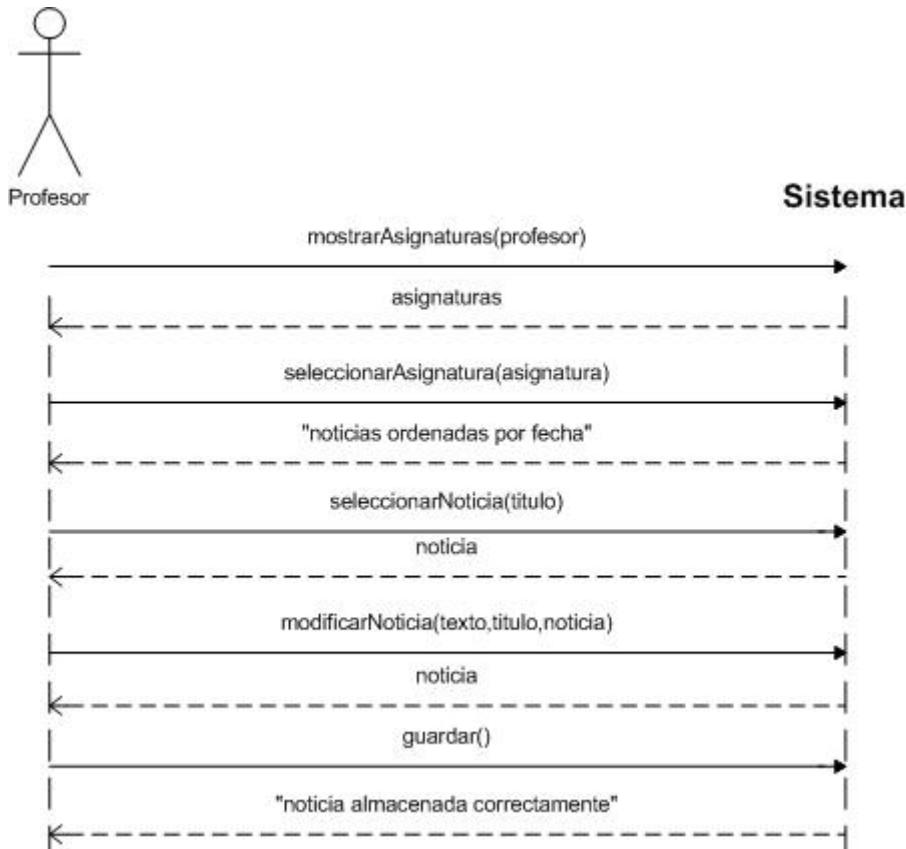


Figura 5: modificarNoticia(profesor,asignatura,noticia)

8.2.3 Subsistema gestión administrativa vía Web

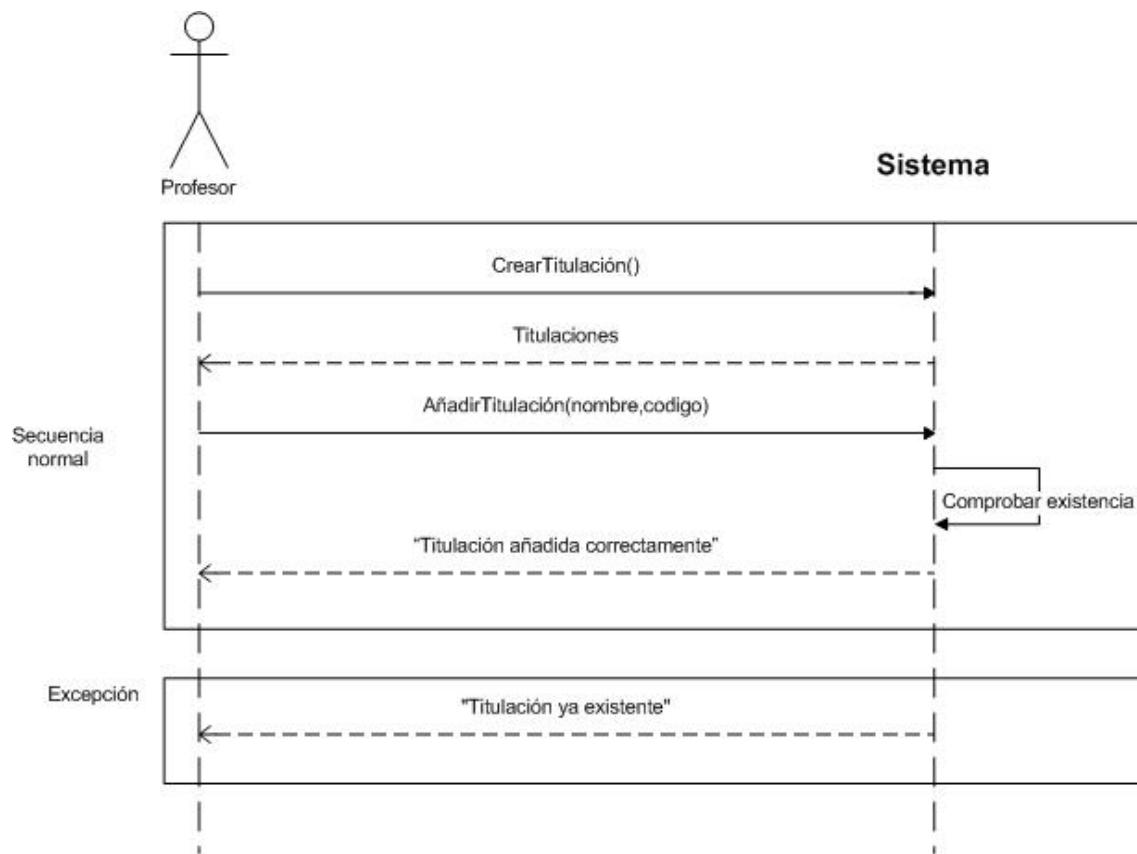


Figura 1: añadirTitulacion(nombre,código)

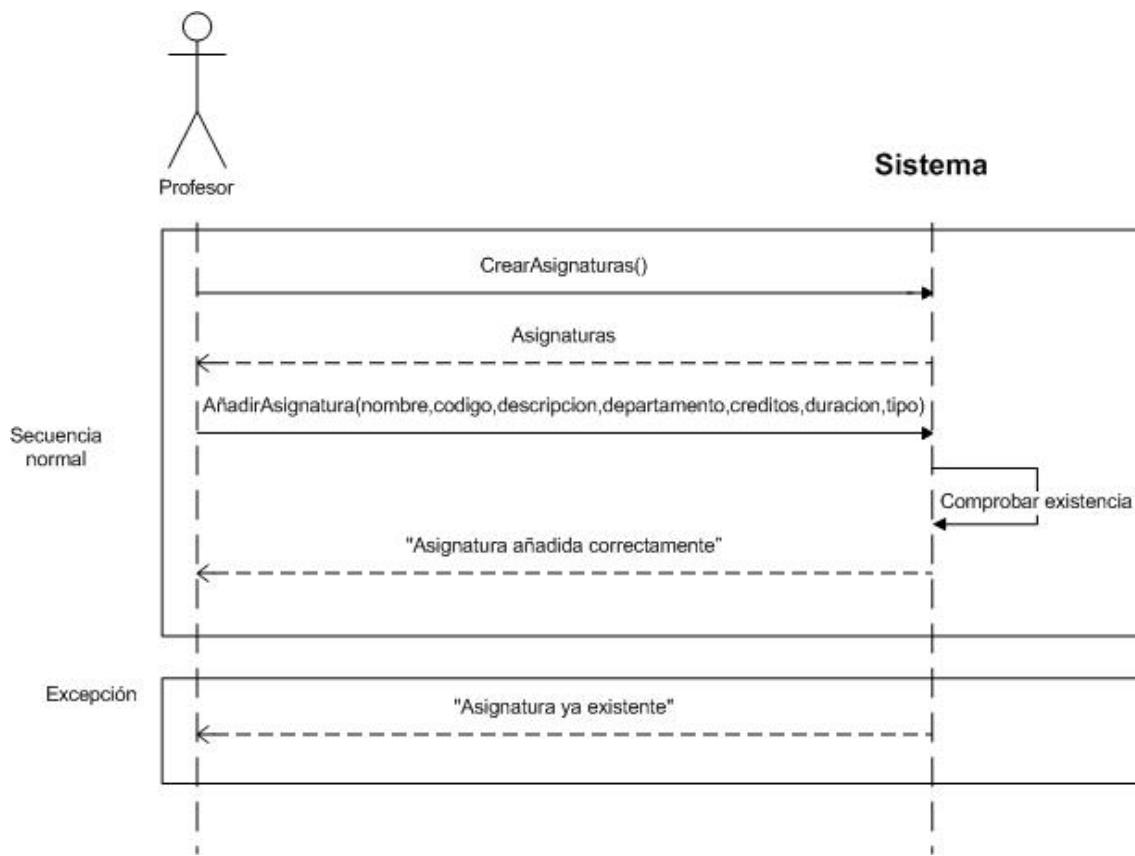


Figura 2:
AñadirAsignatura(nombre,codigo,descripcion,departamento,creditos,duracion,tipo)

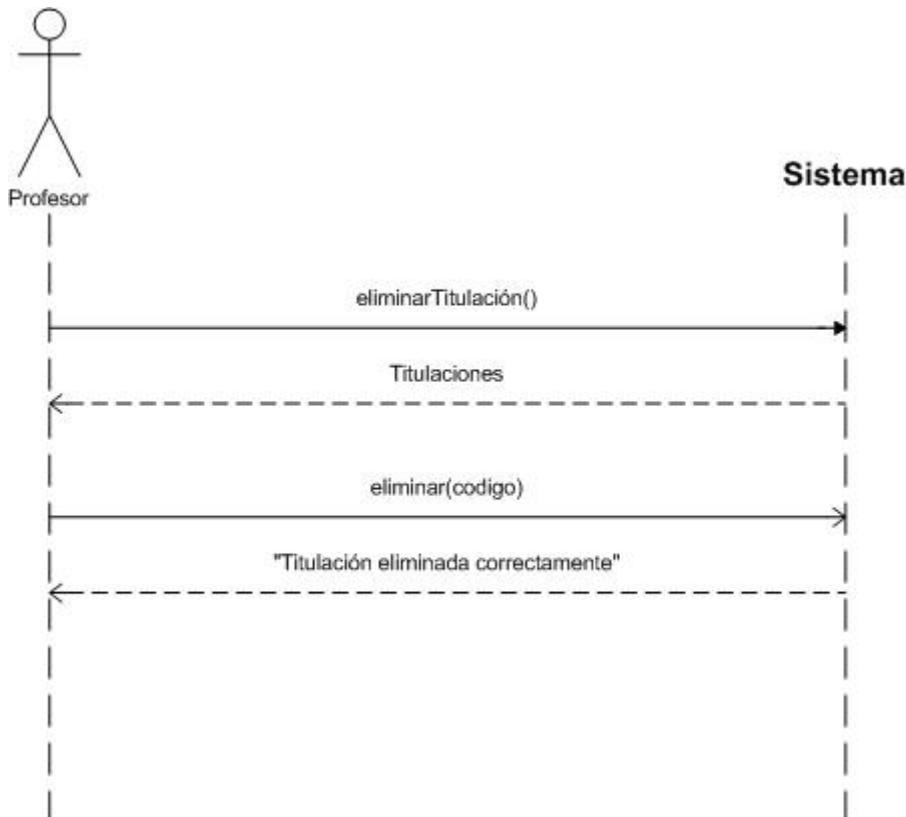


Figura 3: eliminarTitulacion(codigo)

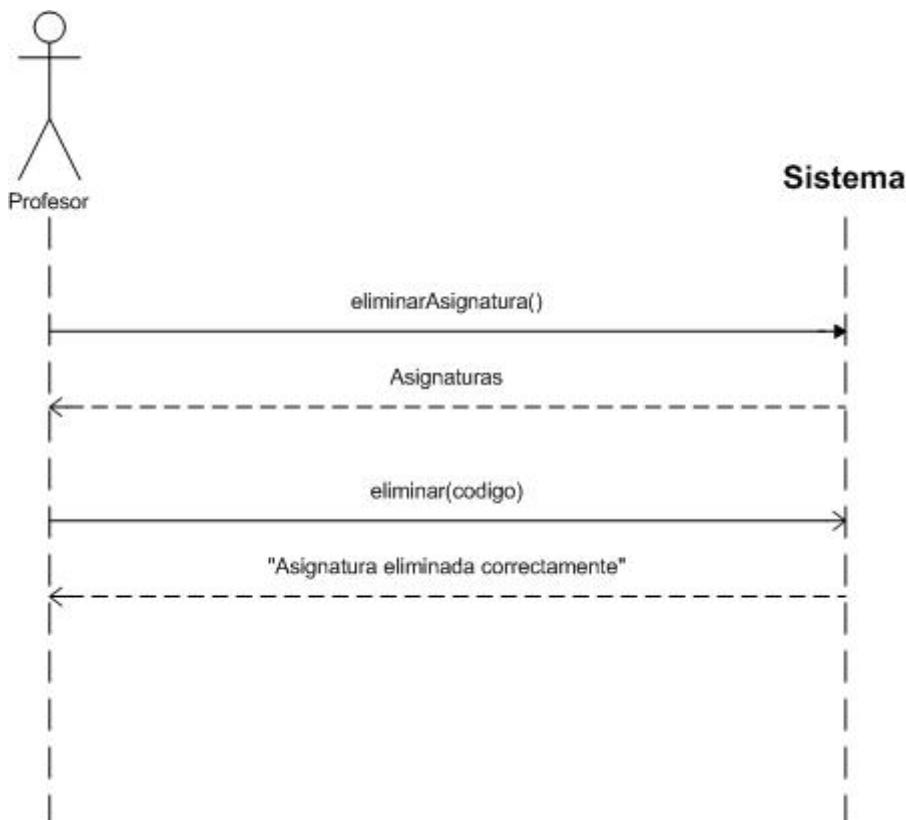


Figura 4: eliminarAsignatura(codigo)

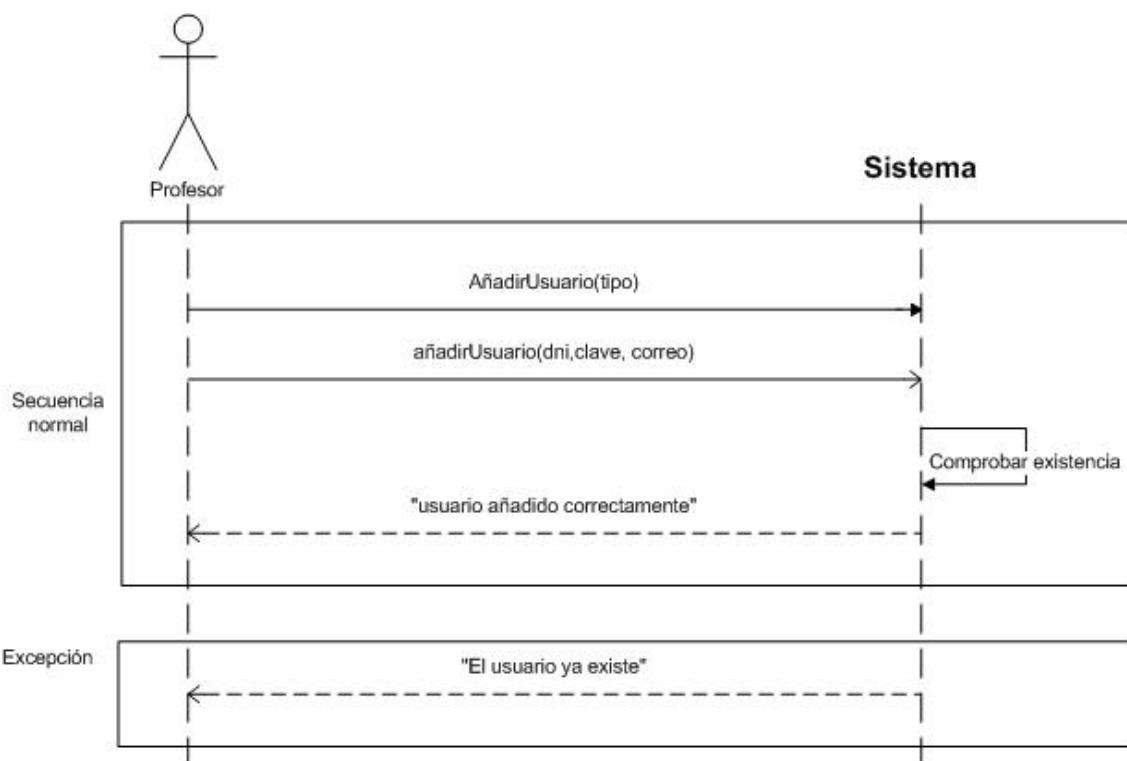


Figura 5: añadirUsuario(dni,clave,tipo,correo)

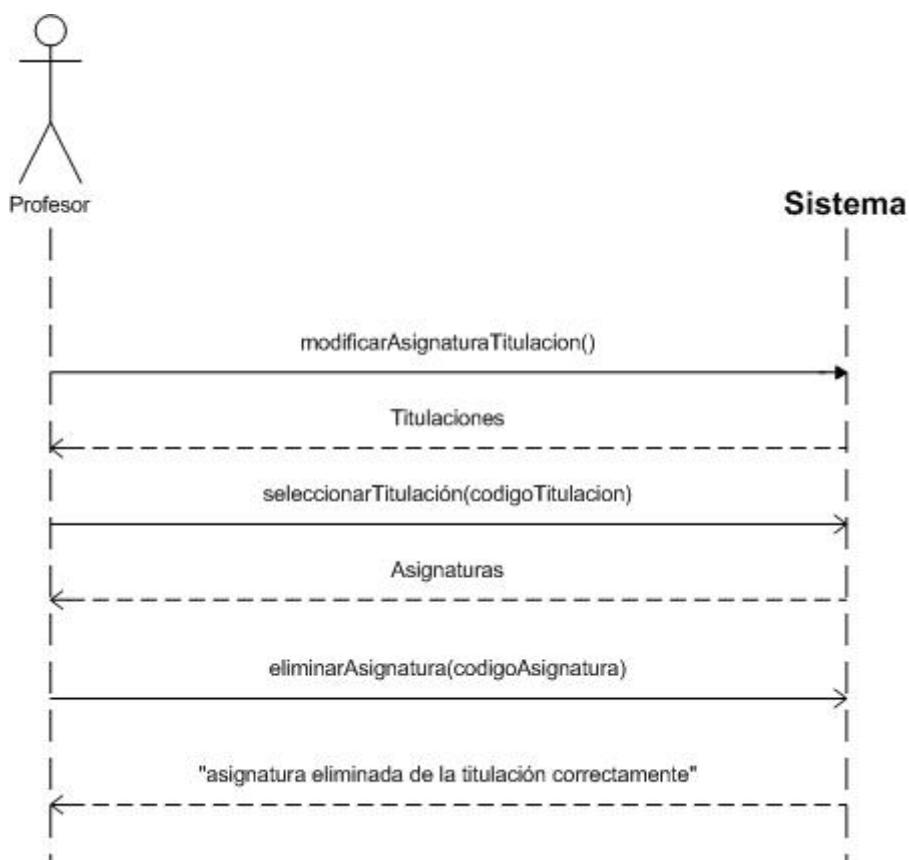


Figura 6: eliminarAsignaturaTitulacion(asignatura,titulacion)

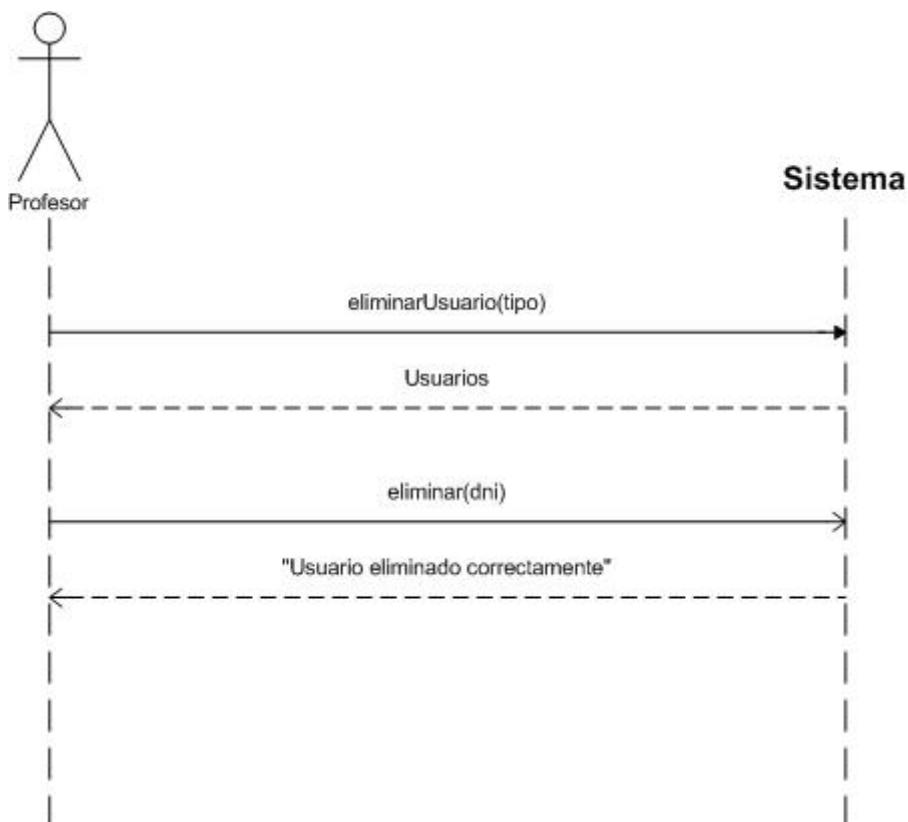


Figura 7: eliminarUsuario(dni,tipo)

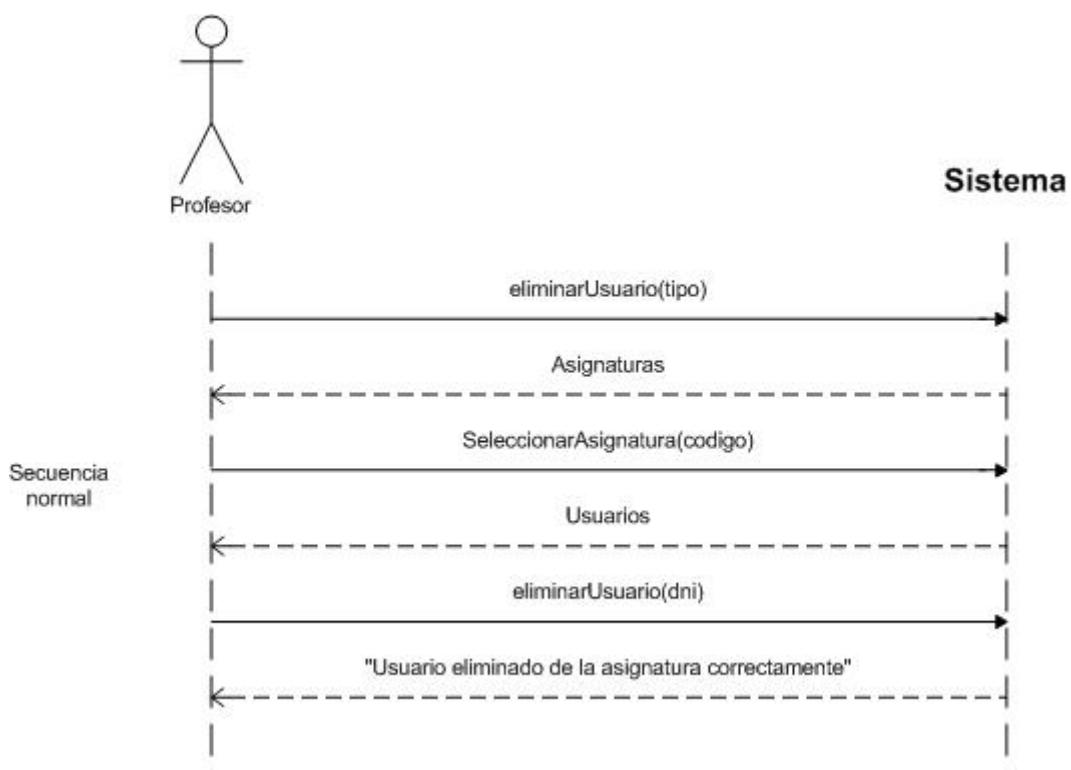
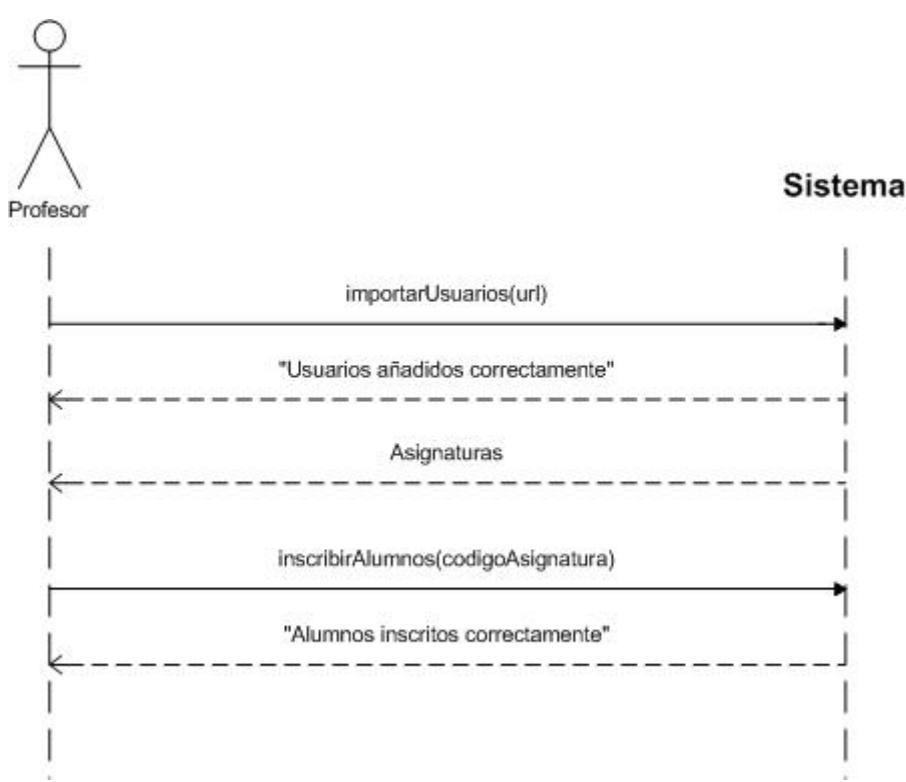
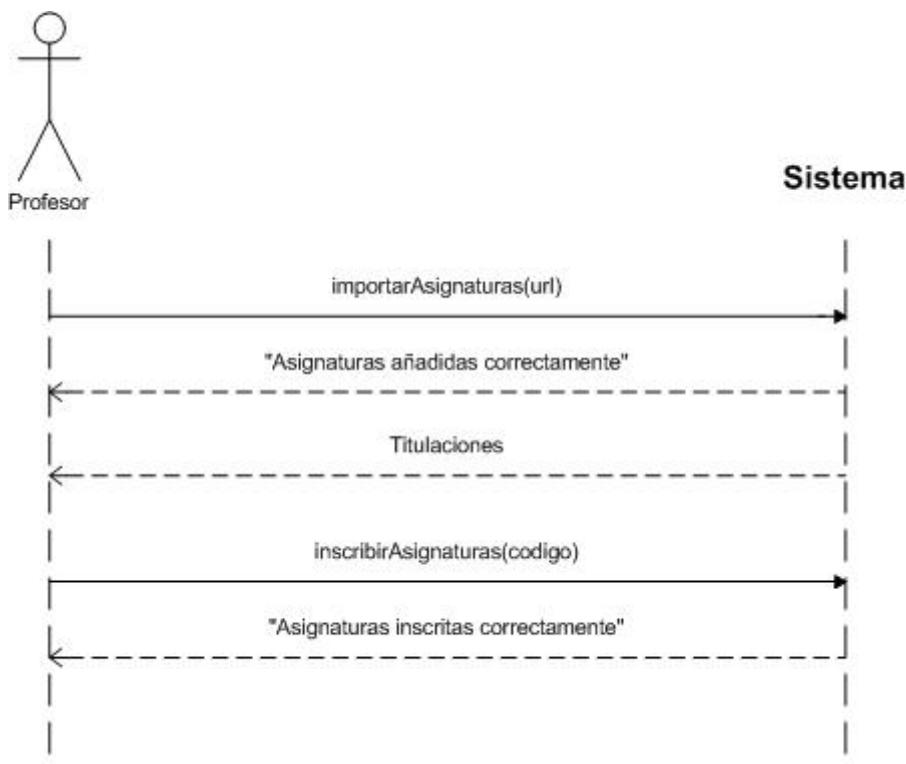


Figura 8: eliminarUsuarioAsignatura(usuario,asignatura)



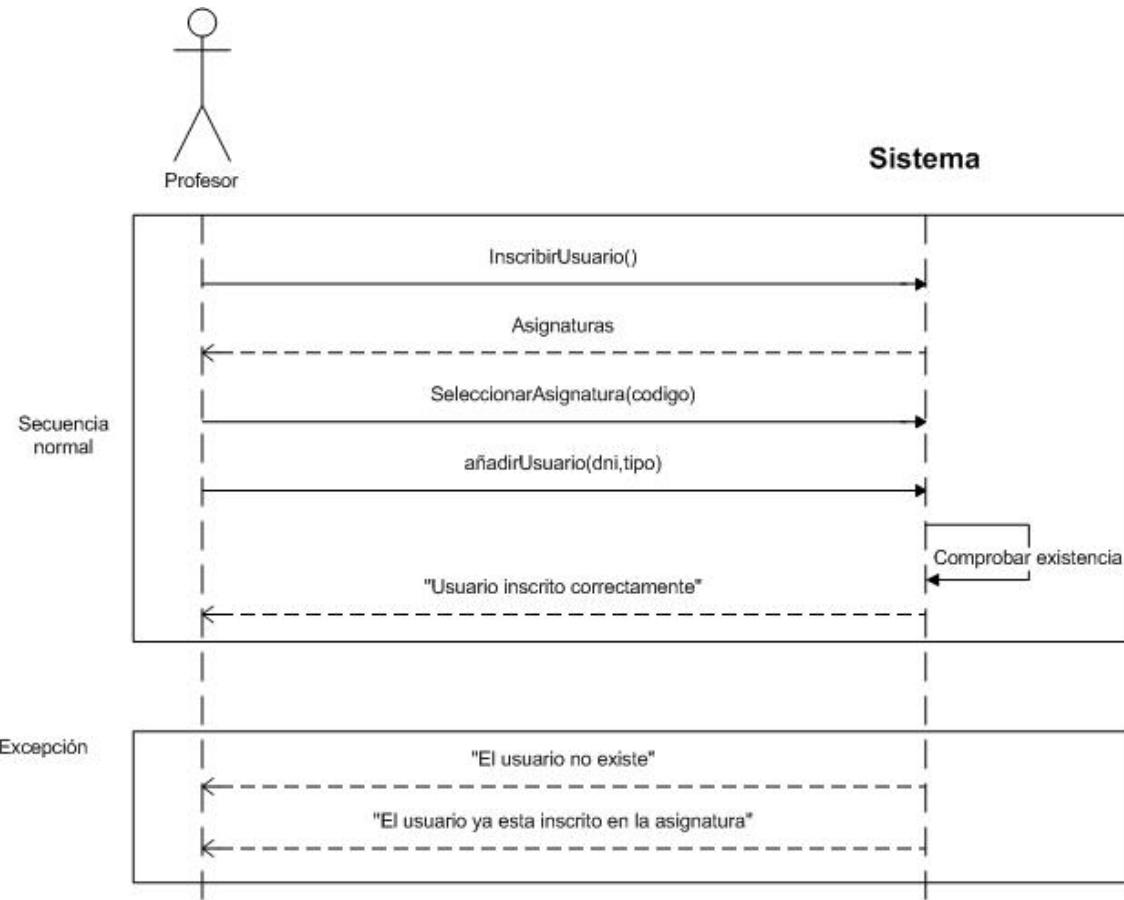


Figura 11: InscribirUsuarioAsignatura(usuario,asignatura)

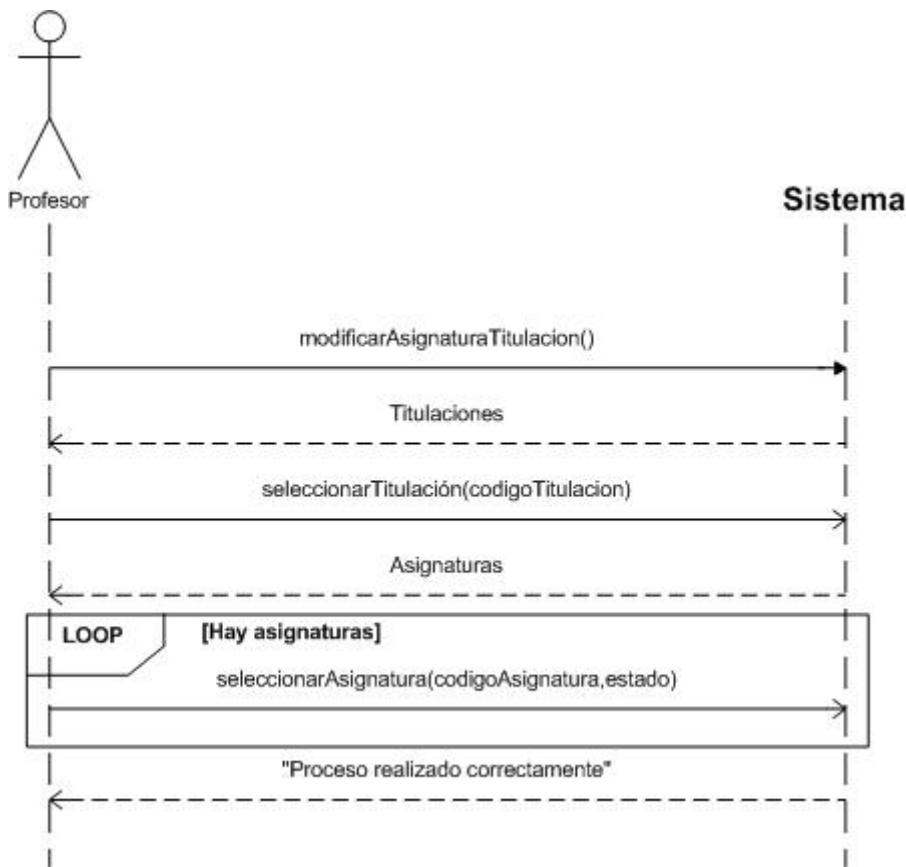


Figura 12: modificarAsignaturaTitulacion(titulacion,asignatura)

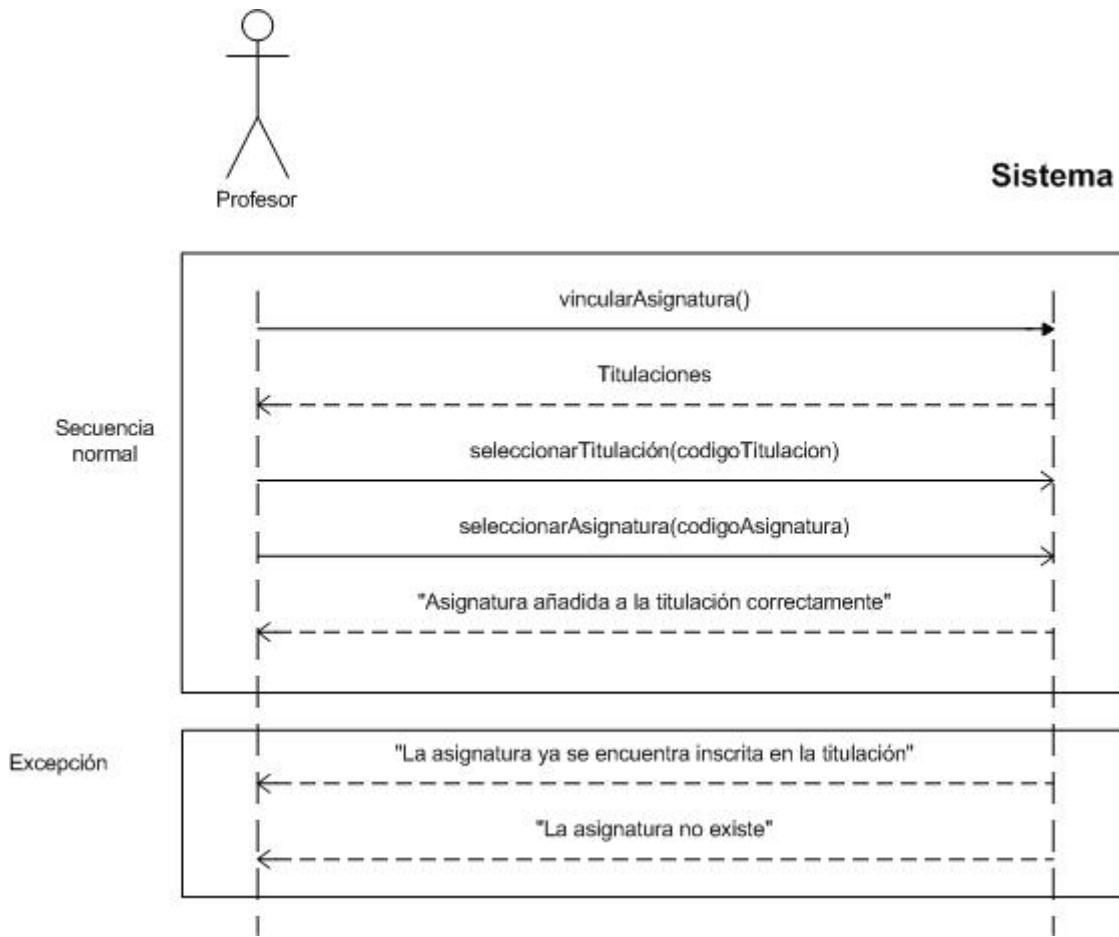


Figura 13: VincularAsignaturaTitulacion(asignatura,titulacion)

8.2.4 Subsistema gestión de usuarios y perfiles

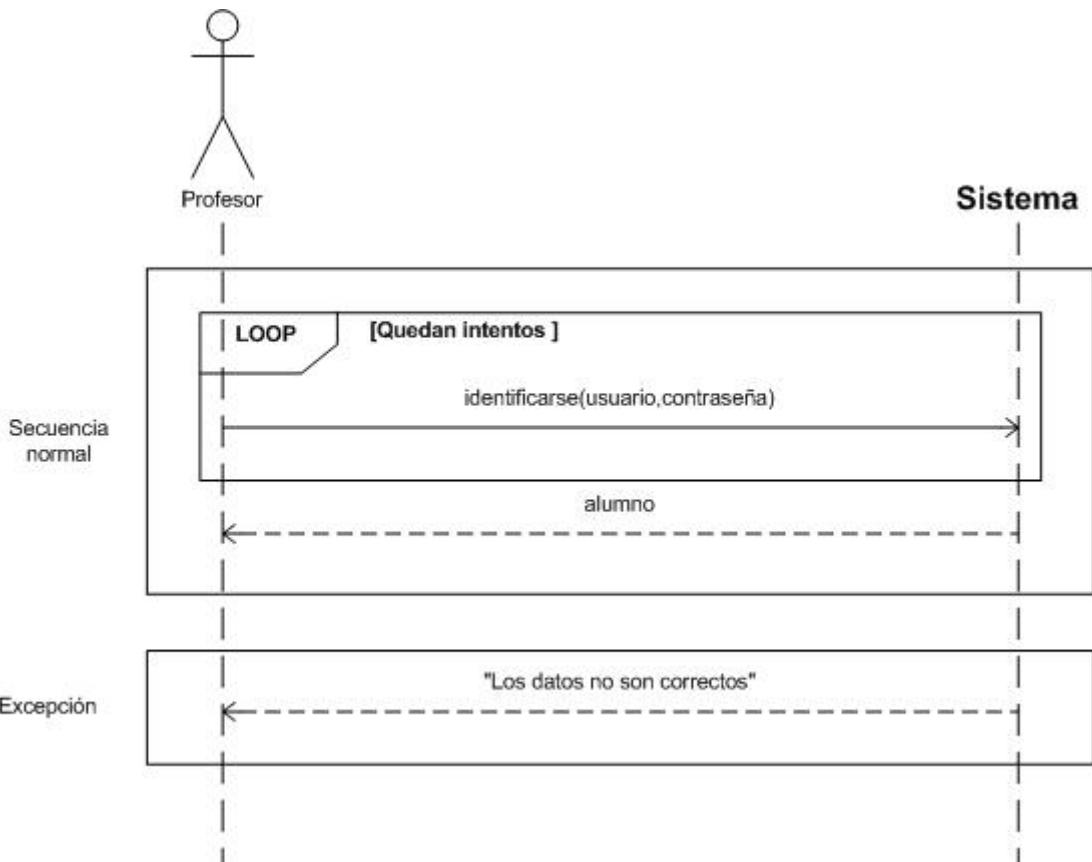


Figura 1: `identificarUsuario(usuario,contraseña)`

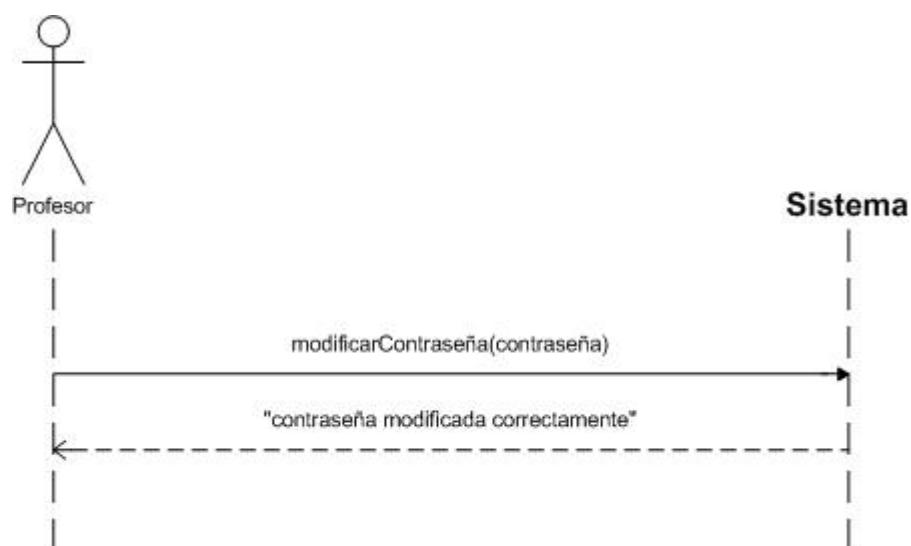


Figura 2: `modificarContraseña(contraseña)`

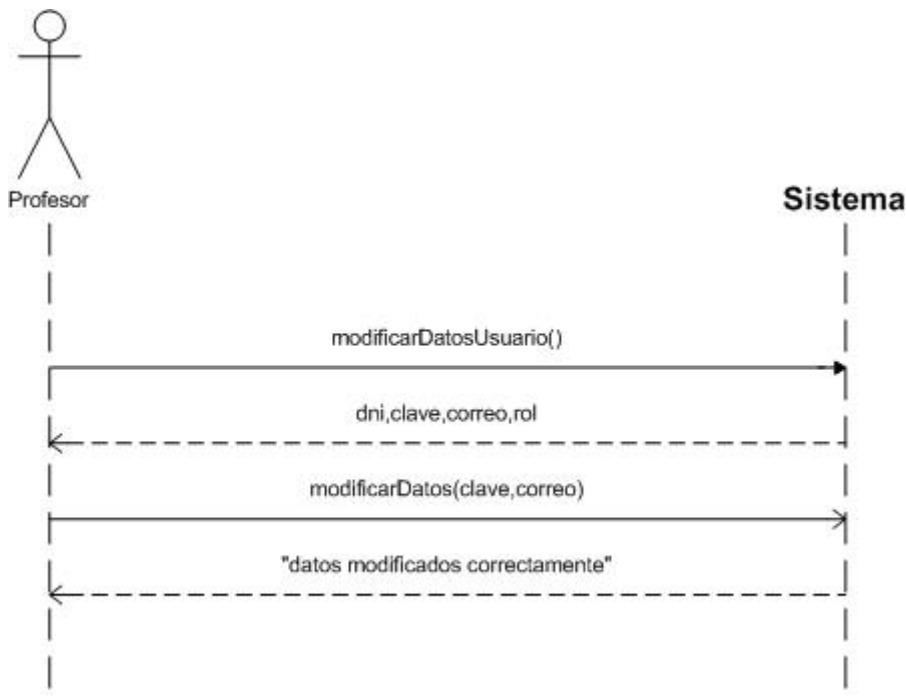


Figura 3: modificarDatosUsuario(Usuario,correo,contraseña)

8.2.5 Subsistema gestión de tutorías y peticiones de cita

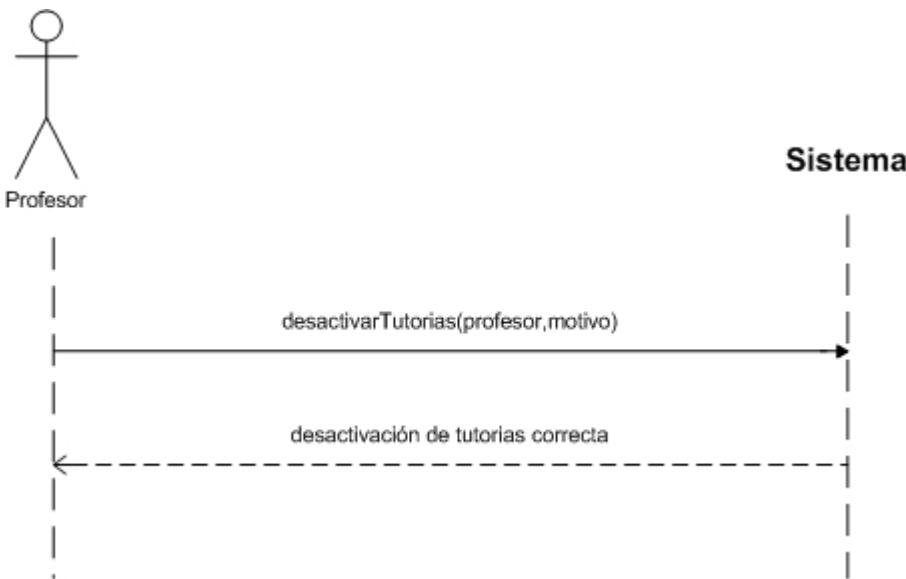


Figura 1: desactivarTutorias(profesor,motivo)

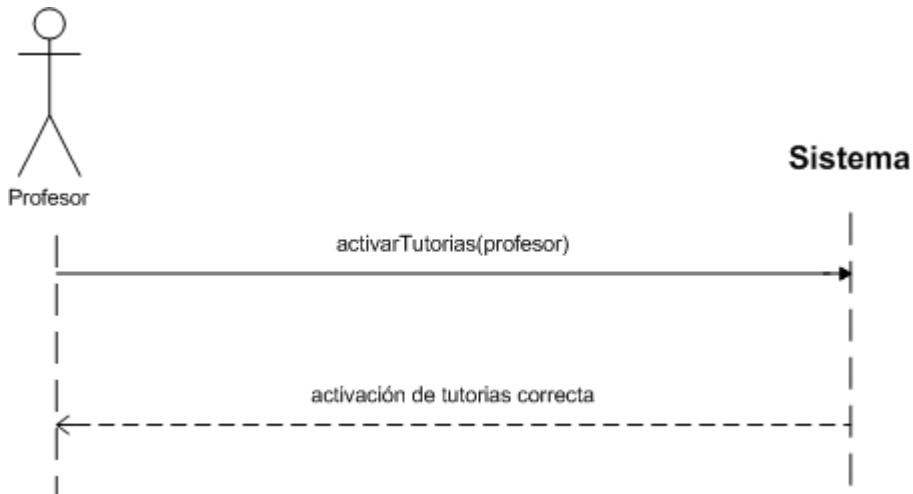


Figura 2: activarTutoria(profesor)

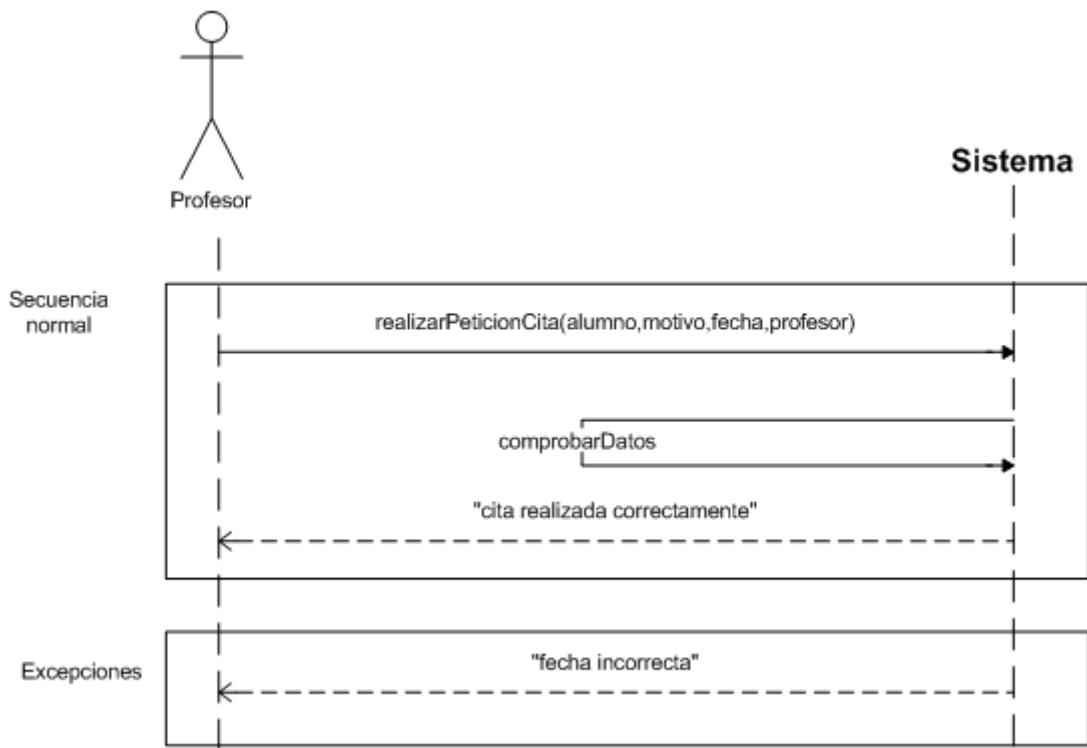


Figura 3: realizarPeticionCita(alumno,motivo,fecha,profesor)

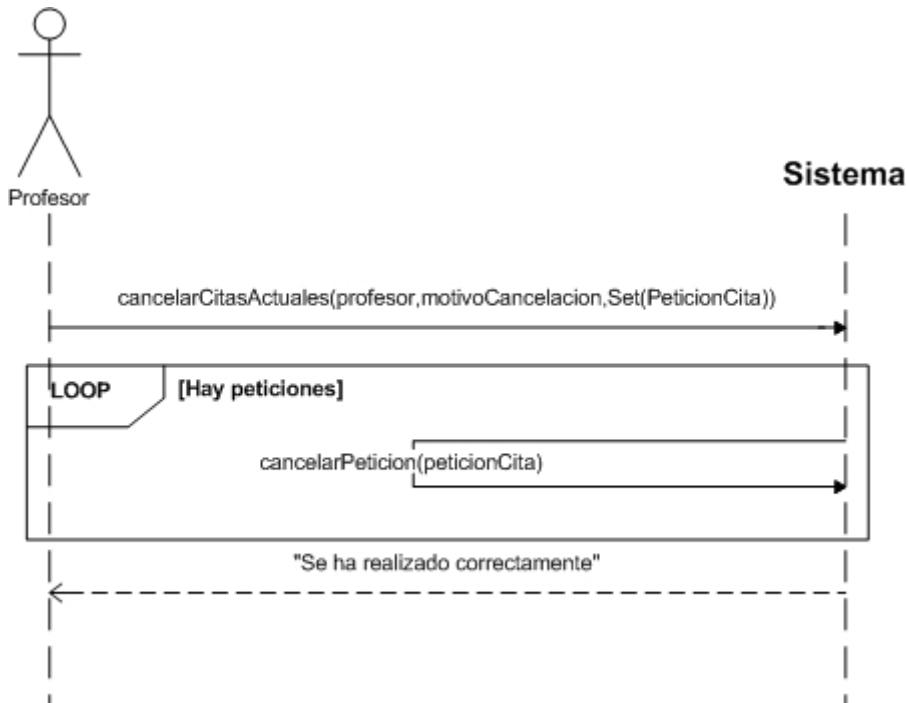


Figura 4: cancelarCitasActuales(profesor,peticiones,motivoCancelacion)

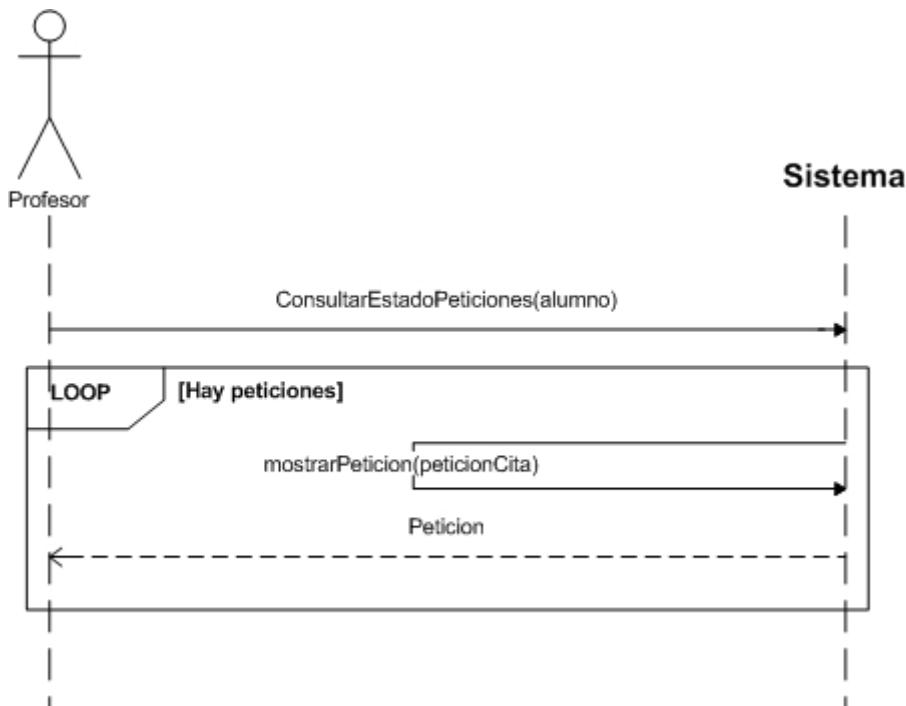


Figura 5: consultarEstadoPeticiones(alumno)

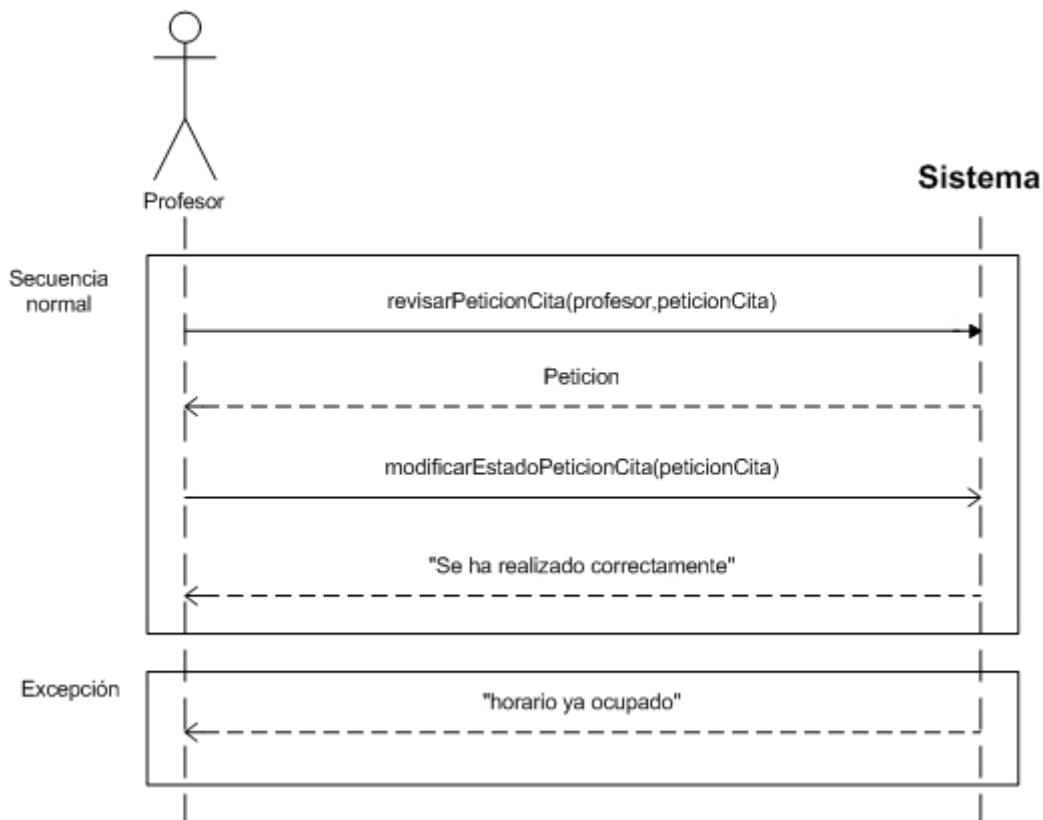


Figura 6: revisarPeticionCita(profesor,peticionCita)

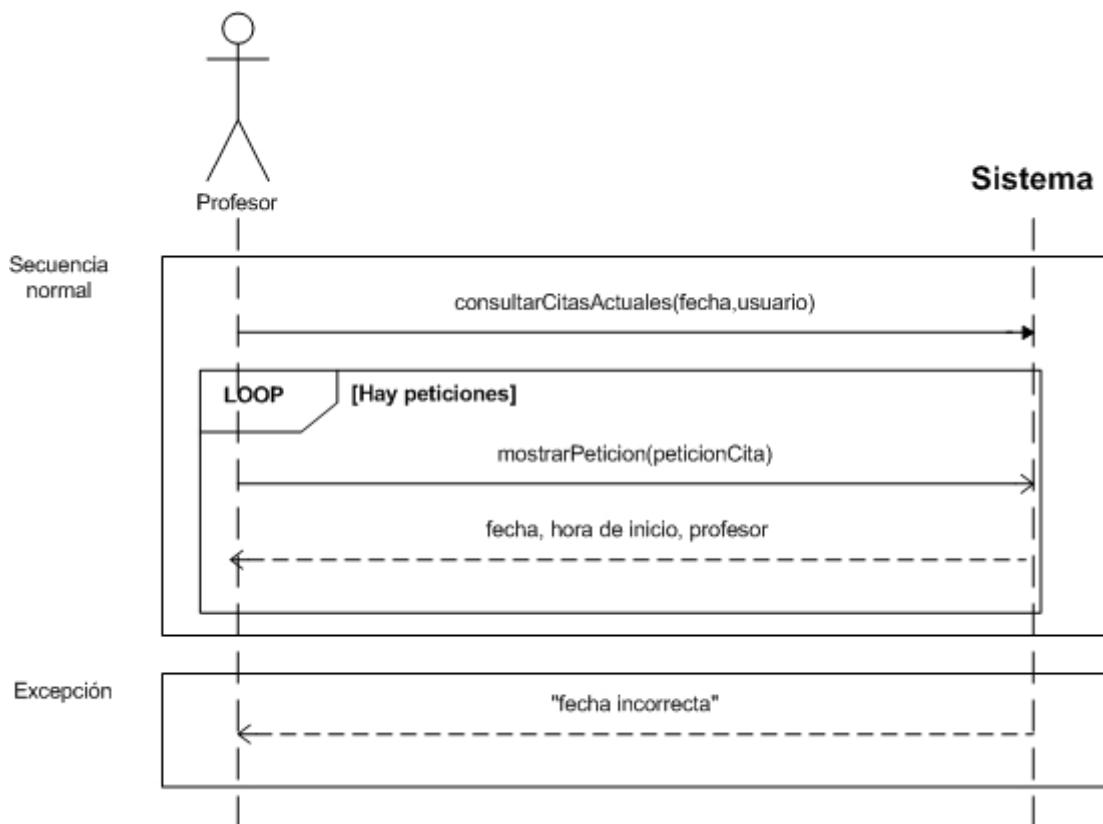


Figura 7: consultarCitasActuales(fecha,usuario)

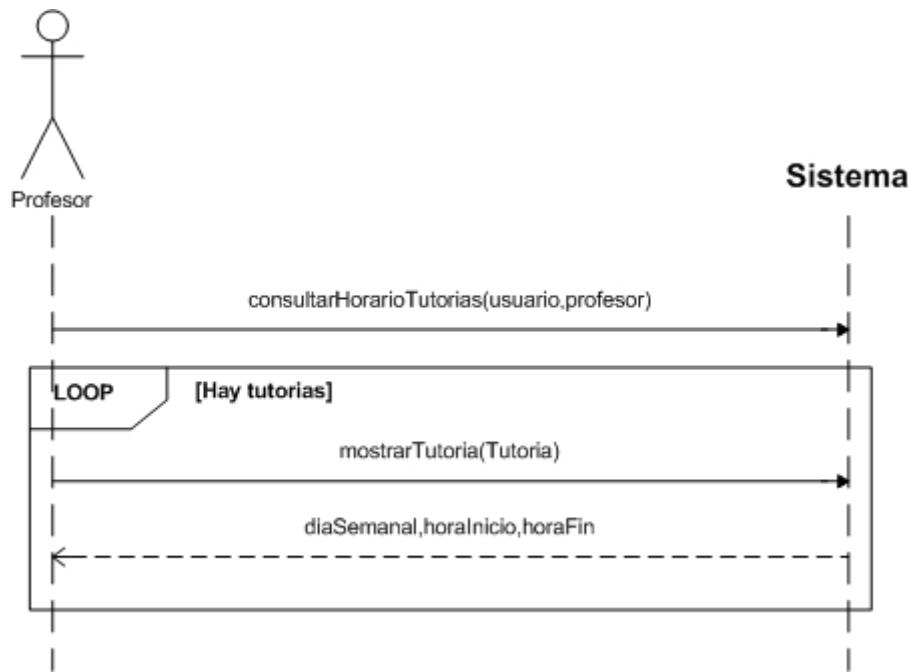


Figura 8: consultarHorarioTutorias(alumno,profesor)

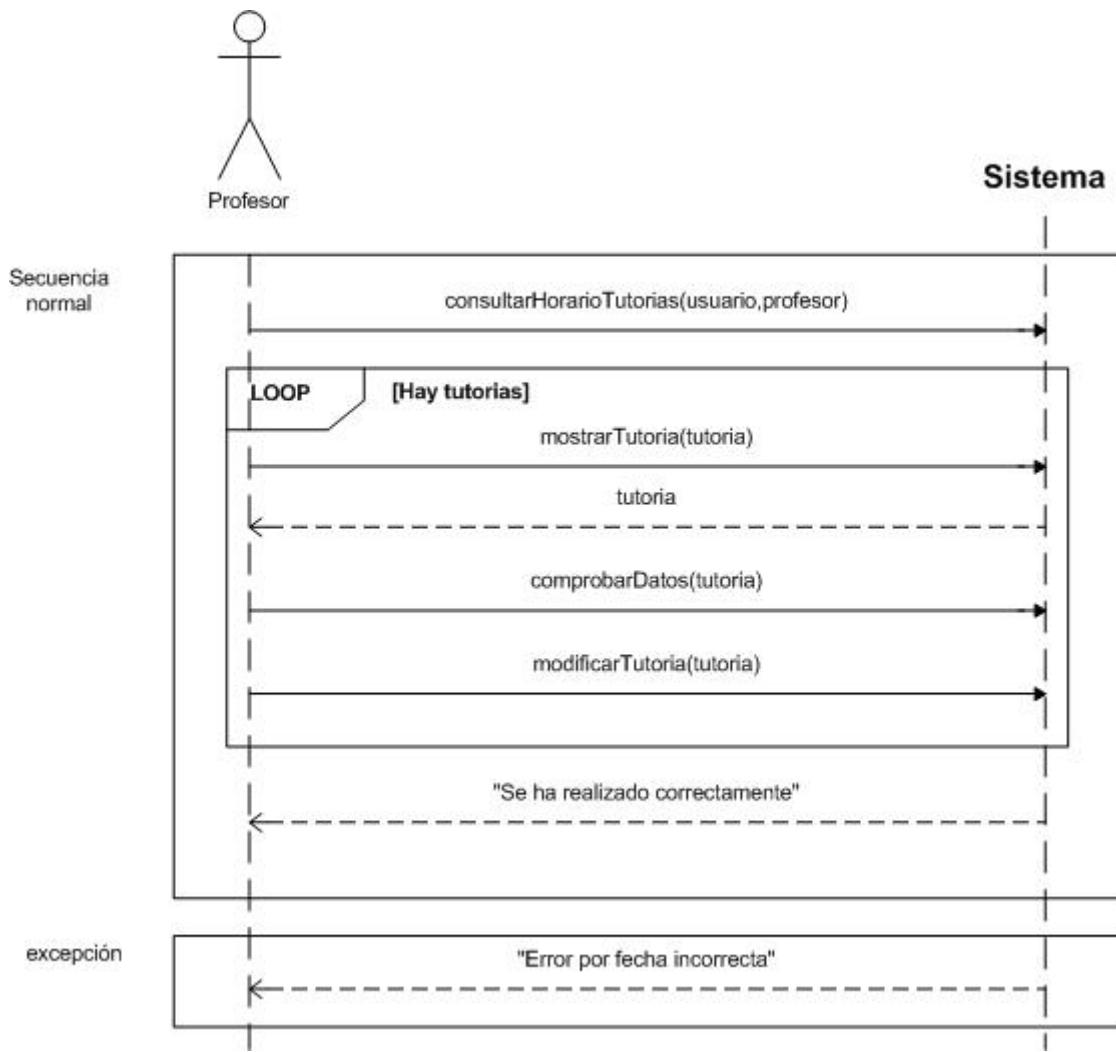


Figura 9: modificarHorarioTutorias(profesor)

8.3 Modelo funcional del sistema

8.3.1 Subsistema gestión de información alumnos y profesores

SOP-0017	modificarInformacionProfesor
Tipo del resultado	Profesor- devuelve el profesor modificado
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0043] Modificar información del profesor • [UC-0001] Identificación de usuario
Descripción	El profesor desea modificar sus datos de usuario
Parámetros	usuario : String -- usuario profesor que desea modificar sus datos

	<p>password : String -- contraseña que identifica al usuario en el sistema categoria : String -- tipo de <u>profesor</u> que es despacho : String -- localización del despacho del <u>profesor</u> web : String -- dirección personal del <u>profesor</u> nombre : String -- nombre completo de <u>profesor</u></p>
Expresiones de precondición	pre1: El usuario está registrado en el sistema
Expresiones de postcondición	post1: Los datos usuario se han modificado con los parámetros de entrada post2: Sólo se han modificado los datos requeridos por el usuario post3: El sistema notifica que se ha realizado correctamente

Excepción	modificarInformacionProfesor::excep1
Condición	La información introducida por el usuario no es correcta
Expresión	El sistema notifica del error y solicita los datos correctos

SOP-0018	incluirObservaciones
Versión	1.0 (05/10/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	<ul style="list-style-type: none"> • <u>[UC-0001] Identificación de usuario</u> • <u>[UC-0020] Inscribir usuario a asignatura</u>
Descripción	El usuario <u>profesor</u> desea incluir una observación a un <u>alumno</u> concreto
Parámetros	<p>alumno : Alumno -- el <u>alumno</u> al que se desea incluir una observación profesor : Profesor -- <u>profesor</u> identificado que desea realizar la observación al <u>alumno</u> descripción : String -- texto de la observación asignatura : Asignatura -- <u>asignatura</u> en la que se le desea incluir una observación al <u>alumno</u></p>
Expresiones de precondición	pre1: El usuario está registrado en el sistema
Expresiones de postcondición	post1: Se ha creado una observación con los datos de entrada, con la fecha actual y se le ha asignado un id post2: Se ha asociado la observación al <u>alumno</u> , a una <u>asignatura</u> y a un <u>profesor</u> post3: Se ha notificado al usuario que se ha realizado correctamente

SOP-0019	consultarProfesor
Tipo del	nombre: String - nombre completo del profesor web: String

ANÁLISIS DEL SISTEMA

resultado	- dirección de la web personal categoria:String- tipo de profesor telefono:Integer - número de teléfono del despacho del profesor despacho:String - dirección del despacho
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0001] Identificación de usuario • [UC-0042] Consultar información del profesor
Descripción	El usuario alumno desea consultar la información del profesor
Parámetros	<p>profesor : Profesor -- profesor <i>del que se desea consultar la información</i></p> <p>alumno : Alumno -- alumno <i>que desea consultar la información del profesor</i></p>
Expresiones de precondición	pre1: El usuario esta identificado
Expresiones de postcondición	post1: Se ha mostrado la información del profesor

SOP-0020	modificarFicha
Tipo del resultado	ficha:Ficha - ficha modificada del usuario
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0001] Identificación de usuario • [UC-0041] Modificar ficha personal
Descripción	El alumno desea modificar la información de su ficha personal
Parámetros	<p>alumno : Alumno -- alumno <i>que desea modificar su ficha</i></p> <p>telefono : Integer -- <i>teléfono fijo del alumno</i></p> <p>movil : String -- <i>teléfono móvil del alumno</i></p> <p>fechaNacimiento : String -- <i>fecha de nacimiento del usuario</i></p> <p>dirección : String -- <i>dirección de su domicilio real</i></p> <p>localización : String -- <i>localización de su domicilio real</i></p> <p>pais : String -- <i>país de residencia</i></p> <p>codigoPostal : String -- <i>codigo postal de su residencia real</i></p> <p>provincia : String -- <i>provincia de la residencia real</i></p> <p>dirección1 : String -- <i>dirección de su domicilio mientras estudia</i></p> <p>localización1 : String -- <i>localización de su domicilio mientras estudia</i></p> <p>pais1 : String -- <i>país de residencia mientras estudia</i></p> <p>codigoPostali1 : String -- <i>codigo postal de su residencia mientras estudia</i></p>

	provincial : String -- provincia de la residencia mientras estudia
Expresiones de precondition	pre1: El usuario está identificado
Expresiones de postcondición	post1: Se ha modificado la ficha con los datos pasados por parámetros post2: El sistema muestra que se ha realizado correctamente
Excepción	modificarFicha::excep1
Condición	Los datos del usuario introducidos no son correctos
Expresión	Se notifica del error al usuario y se le piden los datos correctos

Excepción	modificarFicha::excep2
Condición	Las direcciones del usuario no son válidas
Expresión	se muestra el error y se piden los datos correctos al usuario

SOP-0021	consultarFicha
Tipo del resultado	ficha:Ficha-devuelve la ficha para mostrarsela al usuario
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0001] Identificación de usuario • [UC-0040] Consultar ficha personal
Descripción	El usuario desea consultar su ficha
Parámetros	alumno : Alumno -- <i>alumno que realiza la consulta</i>
Expresiones de precondition	pre1: El usuario esta identificado
Expresiones de postcondición	post1: El sistema muestra su ficha

SOP-0022	consultarDatosAlumno
Tipo del resultado	ficha:Ficha-- ficha que se muestra al profesor observaciones:Set(Observacion)-- observaciones realizadas al alumnos
Versión	1.0 (05/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0008] Consultar datos de alumnos

Descripción	El usuario <u>profesor</u> desea consultar la ficha de un <u>alumno</u> y sus observaciones
Parámetros	alumno : Alumno -- <u>alumno</u> del que se desea consultar su ficha profesor : Profesor -- <u>profesor</u> que desea consultar la ficha del <u>alumno</u>
Expresiones de precondición	pre1: El usuario está identificado
Expresiones de postcondición	post1: El sistema muestra los datos

8.3.2 Subsistema gestión de información de las asignaturas

SOP-0004	registrarNoticia
Tipo del resultado	Noticia- noticia que se ha creado con los parámetros
Versión	1.0 (04/10/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	• <u>[UC-0010] Introducir noticias</u>
Descripción	El <u>profesor</u> desea añadir una noticia a una <u>asignatura</u> que imparte
Parámetros	profesor : Profesor -- <u>profesor</u> que realiza la noticia asignatura : Asignatura -- <u>asignatura</u> a la que corresponde la noticia titulo : String -- <u>título deseado para la noticia</u> texto : String -- <u>descripción de la noticia</u>
Expresiones de precondición	pre1: El usuario se ha registrado y su rol es <u>profesor</u>
Expresiones de postcondición	post1: Existe una nueva noticia con los parámetros dados asociada a una <u>asignatura</u> post2: El sistema notifica que se ha realizado correctamente

Excepción	registrarNoticia::excep1
Condición	los datos no han sido correctamente introducidos
Expresión	se informa del error y se piden de nuevo

SOP-0005	eliminarNoticia
Versión	1.0 (04/10/2006)
Autores	<u>Fernando López Iglesias</u>

	<u>Noel Guillén Montilla</u>
Dependencias	• [UC-0015] Eliminar noticia
Descripción	El usuario <u>profesor</u> desea eliminar una noticia que ha creado él
Parámetros	noticia : Noticia -- <i>a la noticia que se desea eliminar</i> profesor : Profesor -- <i>al <u>profesor</u> identificado por el sistema que desea eliminar la noticia</i>
Expresiones de precondición	pre1: El usuario se ha identificado correctamente
Expresiones de postcondición	post1: Se ha eliminado la relación de la noticia con la <u>asignatura</u> y el <u>profesor</u> post2: Se ha eliminado la noticia post3: Se ha notificado al <u>profesor</u> que se ha realizado correctamente

SOP-0006	modificarNoticia
Tipo del resultado	Noticia-noticia con sus datos modificados
Versión	1.0 (04/10/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	• [UC-0011] Modificar noticias
Descripción	El usuario <u>profesor</u> desea modificar una noticia
Parámetros	noticia : Noticia -- <i>a la noticia que se desea modificar</i> profesor : Profesor -- <i>al <u>profesor</u> identificado que quiere cambiar la noticia</i> titulo : String -- <i>al titulo de la noticia</i> texto : String -- <i>a la descripción de la noticia</i>
Expresiones de precondición	pre1: El usuario <u>profesor</u> se ha identificado correctamente pre2: La noticia existe previamente en el sistema
Expresiones de postcondición	post1: Sólo se han modificado los datos introducidos por el usuario post2: El sistema informa de que la operación se ha realizado correctamente post3: Se ha modificado la fecha de modificación por la actual

SOP-0007	consultarNoticias
Tipo del resultado	Set(Noticia)-noticias que se desean ver
Versión	1.0 (04/10/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	• [UC-0012] Consultar noticias

Descripción	El usuario <u>profesor</u> desea consultar una noticia
Parámetros	<p>profesor : Profesor -- <i>usuario profesor identificado que desea ver la noticia</i></p> <p>asignatura : Asignatura -- <u>asignatura</u> a la que corresponde la noticia</p>
Expresiones de precondición	pre1: El usuario se ha identificado
Expresiones de postcondición	post1: Se han mostrado las noticias

8.3.3 Subsistema gestión administrativa vía Web

SOP-0023	añadirAsignatura
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0032] Añadir asignatura
Descripción	El <u>administrador</u> desea añadir una nueva <u>asignatura</u> al sistema
Parámetros	<p>Asignatura : Asignatura -- <i>la nueva asignatura que se desea añadir</i></p> <p>administrador : Usuario -- <i>al usuario, que se ha registrado como administrador y desea añadir una asignatura</i></p>
Expresiones de precondición	pre1: El usuario se ha identificado como <u>administrador</u>
Expresiones de postcondición	pos1: Se ha añadido una nueva <u>asignatura</u>

Excepción	añadirAsignatura::excep1
Condición	La <u>asignatura</u> ya existe en el sistema
Expresión	se Informa del error y se solicita una nueva <u>asignatura</u>

SOP-0024	añadirTitulacion
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0035] Añadir titulación
Descripción	El <u>administrador</u> desea añadir una nueva titulación al sistema

ANÁLISIS DEL SISTEMA

Parámetros	Administrador : Usuario -- <i>al usuario, identificado como administrador que desea añadir una nueva titulación</i> titulacion : Titulación -- <i>la nueva titulación que se desea añadir</i>
Expresiones de precondición	pre1: El usuario se ha identificado como administrador
Expresiones de postcondición	pos1: Se ha añadido una nueva titulación

Excepción	añadirTitulacion::excep1
Condición	La titulación ya existe en el sistema
Expresión	se Informa del error y se solicita una nueva titulación

SOP-0025	añadirUsuario
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0028] Añadir usuario alumno/profesor
Descripción	El administrador desea añadir un nuevo usuario al sistema
Parámetros	Administrador : tiene -- <i>al usuario, identificado como administrador que desea añadir un nuevo alumno</i> usuario : Usuario -- <i>al nuevo usuario que se va añadir al sistema</i>
Expresiones de precondición	pre1: El usuario se ha identificado como administrador
Expresiones de postcondición	pos1: Se ha añadido un nuevo usuario

Excepción	añadirUsuario::excep1
Condición	El usuario ya existe en el sistema
Expresión	se Informa del error y se solicita un nuevo usuario

SOP-0026	eliminarAsignaturaTitulación
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla

Dependencias	<ul style="list-style-type: none"> • [UC-0038] Eliminar asignatura de titulación
Descripción	El <u>administrador</u> desea eliminar una <u>asignatura</u> de una titulación
Parámetros	<p>Titulación : Titulación -- <i>la titulación de la que se eliminara la asignatura</i></p> <p>asignatura : Asignatura -- <i>la asignatura que será eliminada de la titulación</i></p> <p>Administrador : Usuario -- <i>al usuario, identificado como administrador que desea eliminar una asignatura de una titulación</i></p>
Expresiones de precondición	<p>pre1: El usuario se ha identificado como <u>administrador</u></p> <p>pre2: Existen asignaturas asociadas a la titulación</p>
Expresiones de postcondición	<p>pos1: Se ha eliminado la <u>asignatura</u> de la titulación</p>

SOP-0027	eliminarAsignatura
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0026] Eliminar asignaturas
Descripción	El <u>administrador</u> desea eliminar una <u>asignatura</u>
Parámetros	<p>Asignatura : Asignatura -- <i>la asignatura que se desea eliminar</i></p> <p>Administrador : Usuario -- <i>al usuario, identificado como administrador que desea eliminar una asignatura del sistema</i></p>
Expresiones de precondición	<p>pre1: El usuario se ha identificado como <u>administrador</u></p> <p>pre2: Existen asignaturas en el sistema</p>
Expresiones de postcondición	<p>pos1: Se ha eliminado la <u>asignatura</u> del sistema</p>

SOP-0028	eliminarTitulacion
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0036] Eliminar titulación
Descripción	El <u>administrador</u> desea eliminar una titulación
Parámetros	<p>Titulación : Titulación -- <i>la titulación que se desea eliminar</i></p> <p>Administrador : Usuario -- <i>al usuario, identificado como administrador que desea eliminar una titulación del sistema</i></p>
Expresiones de precondición	<p>pre1: El usuario se ha identificado como <u>administrador</u></p> <p>pre2: Existen titulaciones en el sistema</p>
Expresiones de postcondición	<p>pos1: Se ha eliminado la titulación del sistema</p>

SOP-0029	eliminarUsuario
Versión	1.0 (09/10/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	• [UC-0031] Eliminar usuario alumno/profesor
Descripción	El <u>administrador</u> desea eliminar un usuario
Parámetros	usuario : Usuario -- <i>al usuario que se desea eliminar</i> Administrador : Usuario -- <i>al usuario, identificado como administrador que desea eliminar un usuario del sistema</i>
Expresiones de precondición	pre1: El usuario se ha identificado como <u>administrador</u> pre2: Existen usuarios en el sistema
Expresiones de postcondición	pos1: Se ha eliminado el usuario del sistema

SOP-0030	eliminarUsuarioAsignatura
Versión	1.0 (09/10/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	• [UC-0023] Eliminar usuario de asignatura
Descripción	El <u>administrador</u> desea eliminar un usuario de una <u>asignatura</u>
Parámetros	usuario : Usuario -- <i>al usuario que se desea eliminar de la asignatura</i> Administrador : Usuario -- <i>al usuario, identificado como administrador que desea eliminar un usuario de una asignatura</i> asignatura : Asignatura -- <i>la asignatura de la que se quiere eliminar el alumno</i>
Expresiones de precondición	pre1: El usuario se ha identificado como <u>administrador</u> pre2: Existen usuarios asociados a la <u>asignatura</u>
Expresiones de postcondición	pos1: Se ha eliminado el usuario de la <u>asignatura</u>

SOP-0031	importarAsignaturaFichero
Versión	1.0 (09/10/2006)
Autores	<u>Fernando López Iglesias</u> <u>Noel Guillén Montilla</u>
Dependencias	• [UC-0034] Importar asignatura a partir de fichero
Descripción	El <u>administrador</u> desea importar asignaturas desde un fichero

Parámetros	Administrador : Usuario -- <i>al usuario, identificado como administrador que desea importar la lista de asignaturas</i> Titulación : Titulación -- <i>la titulación a la que se desea añadir la lista de asignaturas importadas</i> url : String -- <i>la dirección donde se encuentra el archivo con las asignaturas a importar</i>
Expresiones de precondición	pre1: El usuario se ha identificado como administrador
Expresiones de postcondición	pos1: Se han incorporado las asignaturas al sistema

SOP-0032	importarUsuarioFichero
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0030] Importar usuarios a partir de un fichero
Descripción	El administrador desea importar usuarios desde un fichero
Parámetros	Administrador : Usuario -- <i>al usuario, identificado como administrador que desea importar la lista de usuarios</i> url : String -- <i>la dirección donde se encuentra el archivo con los usuarios a importar</i> Asignatura : Asignatura -- <i>la asignatura a la que se desea importar la lista de usuarios</i>
Expresiones de precondición	pre1: El usuario se ha identificado como administrador
Expresiones de postcondición	pos1: Se han incorporado los usuarios al sistema

SOP-0033	InscribirUsuarioAsignatura
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0020] Inscribir usuario a asignatura
Descripción	El administrador desea inscribir un usuario en una asignatura
Parámetros	Usuario : Usuario -- <i>el usuario que se desea añadir a la asignatura</i> Asignatura : Asignatura -- <i>la asignatura a la que se desea añadir alumnos</i> Administrador : Usuario -- <i>al usuario, identificado como administrador que desea añadir un alumno a una asignatura</i>

Expresiones de precondición	pre1: El usuario se ha identificado como administrador pre2: Existen usuarios en el sistema pre3: Existen asignaturas en el sistema
Expresiones de postcondición	pos1: Se ha añadido el usuario a la asignatura

Excepción	InscribirUsuarioAsignatura::excep1
Condición	El usuario no existe en el sistema
Expresión	Se informa del error y se solicita un nuevo usuario

Excepción	InscribirUsuarioAsignatura::excep2
Condición	El usuario ya esta inscrito en la asignatura
Expresión	Se informa del error y se aborta la operación

SOP-0034	ModificarAsignaturasTitulación
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0039] modificar asignaturas de titulación
Descripción	El administrador desea editar la lista de asignaturas de una titulación
Parámetros	Titulación : Titulación -- <i>la titulación de la que se desea modificar su lista de asignaturas</i> Administrador : Usuario -- <i>al usuario, identificado como administrador que desea modificar la lista de asignaturas de una titulación</i> Asignaturas : Set(Asignatura) -- <i>las asignaturas que se añadiran o se eliminaran de la titulación</i>
Expresiones de precondición	pre1: El usuario se ha identificado como administrador pre2: Existen asignaturas en el sistema pre3: existen titulaciones en el sistema
Expresiones de postcondición	pos1: Se ha modificado la lista de asignaturas de la titulación

SOP-0035	VincularAsignaturaTitulación
Versión	1.0 (09/10/2006)
Autores	Fernando López Iglesias

	<u>Noel Guillén Montilla</u>
Dependencias	• [UC-0027] Vincular asignatura a titulación
Descripción	El administrador desea añadir una asignatura a una titulación
Parámetros	<p>Asignaturas : Asignatura -- <i>la asignatura que se añade a la titulación</i></p> <p>Administrador : Usuario -- <i>al usuario, identificado como administrador que desea añadir una asignatura a una titulación</i></p> <p>Titulación : Titulación -- <i>la titulación a la que se desea añadir una asignatura</i></p>
Expresiones de precondición	<p>pre1: El usuario se ha identificado como administrador</p> <p>pre2: Existen asignaturas en el sistema</p> <p>pre3: existen titulaciones en el sistema</p>
Expresiones de postcondición	pos1: Se ha añadido la asignatura a la titulación

Excepción	VincularAsignaturaTitulación::excep1
Condición	La asignatura no existe
Expresión	Se informa del error y se solicita una nueva asignatura

Excepción	VincularAsignaturaTitulación::excep2
Condición	La asignatura ya se encuentra añadida a la titulación
Expresión	Se informa del error y se aborta el proceso

8.3.4 Subsistema gestión de usuarios y perfiles

SOP-0001	modificarContraseña
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0001] Identificación de usuario • [UC-0002] Modificar contraseña
Descripción	el usuario desea modificar su contraseña
Parámetros	<p>contraseña : String -- <i>la contraseña del usuario que desea cambiar la contraseña</i></p> <p>usuario : String -- <i>usuarios que quiere cambiar la contraseña</i></p> <p>nuevaContraseña : String -- <i>nueva contraseña del usuarios</i></p>
Expresiones de	pre1: el usuario esta identificado en el sistema

ANÁLISIS DEL SISTEMA

precondición	
Expresiones de postcondición	post1: la contraseña ha sido modificada por la pasada por parametro post2: el sistema ha notificado que todo se ha realizado correctamente

Excepción	modificarContraseña::excep1
Condición	si la contraseña nueva no es correcta, es decir no cumple el mínimo número de caracteres o tiene caracteres no aceptados por el sistema
Expresión	el sistema muestra el tipo de error y pide los datos de nuevo

SOP-0002	modificarDatosUsuario
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0001] Identificación de usuario • [UC-0014] Modificar datos del usuario
Descripción	el usuario desea modificar datos del usuarios
Parámetros	<p>contraseña : String -- <i>contraseña del usuario</i></p> <p>usuarios : String -- <i>usuario que desea modificar sus datos</i></p> <p>correo : String -- <i>correo que desea introducir</i></p> <p>contraseñaNueva : String -- <i>contraseña nueva por la que cambia la anterior</i></p>
Expresiones de precondición	pre1: el usuario está identificado
Expresiones de postcondición	post1: se han modificados los datos del usuario post2: se ha notificado al usuario que se ha realizado correctamente

Excepción	modificarDatosUsuario::excep1
Condición	los datos del usuario no son correctos
Expresión	se informa al usuario del error y se piden de nuevo los datos

SOP-0003	identificarUsuario
Tipo del resultado	Usuario--Usuario que se ha identificado
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla

Dependencias	• [UC-0001] Identificación de usuario
Descripción	el usuario desea identificarse en el sistema
Parámetros	usuario : String -- <i>el usuario que desea identificarse</i> contraseña : String -- <i>contraseña del usuario</i>
Expresiones de precondición	pre1: El usuario ha introducido algún dato
Expresiones de postcondición	post1: Se ha identificado al usuario post2: Se ha informado al usuario de que se ha realizado correctamente

Excepción	identificarUsuario::excep1
Condición	los datos introducidos no son correctos
Expresión	se muestra el error y se piden los datos de nuevo

8.3.5 Subsistema gestión de tutorías y peticiones de cita

SOP-0008	desactivarTutorias
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0016] Desactivar tutorías temporalmente
Descripción	El usuario profesor quiere desactivar las tutorías para que no se muestren y en su lugar se muestra el motivo
Parámetros	profesor : Profesor -- <i>profesor identificado que desea desactivar las tutorías</i> motivo : String -- <i>motivo que se mostrará al desactivarse las tutorías</i>
Expresiones de precondición	pre1: El usuario está debidamente identificado pre2: Las tutorías no están desactivadas anteriormente
Expresiones de postcondición	post1: Se ha modificado el atributo de estado de las tutorías del profesor post2: Se ha modificado el motivo con el pasado por parámetro post3: Se ha notificado de que se ha realizado correctamente

SOP-0009	activarTutorias
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0017] Activar tutorias
Descripción	El profesor desea activar su horario de tutorías
Parámetros	profesor : Profesor -- <i>profesor autenticado que desea activar</i>

	<i>la visualización de las tutorías</i>
Expresiones de precondición	pre1: El usuario se ha identificado correctamente pre2: Las tutorías están desactivadas
Expresiones de postcondición	post1: Se ha modificado el estado de visualización de las tutorías para que se muestren post2: El sistema notifica al usuario que se ha realizado correctamente

SOP-0010	realizarPeticionCita
Tipo del resultado	PeticionCita- petición de cita que se muestra al usuario
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0019] Realizar petición de cita
Descripción	El usuario quiere pedirle una cita a un profesor
Parámetros	alumno : Alumno -- <i>alumno identificado que realiza la petición de cita</i> motivo : String -- <i>descripción del motivo por el que el quiere realizar la cita</i> fecha : Date -- <i>fecha que desea que realice la tutoría</i> profesor : Profesor -- <i>profesor al que se desea pedir una cita</i>
Expresiones de precondición	pre1: El usuario alumno está debidamente identificado por el sistema
Expresiones de postcondición	post1: Se ha registrado la petición post2: Se notifica al usuario que se ha realizado correctamente

Excepción	realizarPeticionCita::excep1
Condición	la fecha no esta dentro de un rango posible
Expresión	se muestra el error y el sistema pide al usuario que ingrese la fecha correcta

SOP-0011	cancelarCitasActuales
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0007] Cancelar citas actuales
Descripción	El usuario profesor quiere cancelar las citas de un dia
Parámetros	profesor : Profesor -- <i>usuario profesor debidamente identificado que desea cancelar las citas</i>

	<p>peticiones : Set(PeticionCita) -- <i>conjunto de peticiones que desea cancelar</i></p> <p>motivoCancelacion : String -- <i>al motivo de la cancelación</i></p>
Expresiones de precondición	pre1: El usuario está identificado por el sistema
Expresiones de postcondición	<p>post1: Se ha modificado motivoCancelacion de las peticiones con el introducido por parámetro</p> <p>post2: Se ha modificado el estado del conjunto de peticiones a cancelado</p> <p>post3: Se notifica al usuario que se ha realizado correctamente</p>

SOP-0012	consultarEstadoPeticiones
Tipo del resultado	Set(PeticionesCita)-devuelve el conjunto de peticiones para que el usuario pueda ver su estado
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0018] Consultar estado de las peticiones
Descripción	El usuario alumno desea consultar el estado de sus peticiones realizadas
Parámetros	alumno : Alumno -- <i>al alumno que quiere consultar el estado de las peticiones</i>
Expresiones de precondición	pre1: El usuario se ha identificado
Expresiones de postcondición	<p>post1: El sistema ha mostrado las peticiones junto con su estado</p> <p>post2: Si la petición ha sido modificada y el usuario la ha leído, el estado de la lectura se modifica, es decir los atributos leidoProfesor o leidoAlumno según el usuario</p>

SOP-0013	revisarPeticionCita
Tipo del resultado	peticionCita
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	<ul style="list-style-type: none"> • [UC-0006] Revisar peticiones de citas
Descripción	El usuario profesor desea revisar las peticiones de cita que le han sido enviadas
Parámetros	profesor : Profesor -- <i>usuario profesor que se ha identificado y desea revisar las peticiones</i> peticion : PeticionCita -- <i>peticion revisada por el profesor</i>
Expresiones de precondición	pre1: El usuario profesor está identificado

Expresiones de postcondición	<p>post1: Se ha modificado el estado de las petición de cita según el pasado por parámetro y el motivo de cancelación si ha sido cancelada.</p> <p>post2: El sistema notifica que se ha realizado correctamente</p> <p>post3: Se modifica el estado de lectura de la petición de cita</p>
-------------------------------------	--

Excepción	revisarPeticionCita::excep1
Condición	La cita que ha aceptado no es posible porque ya hay una aceptada en la misma hora
Expresión	se muestra el error

SOP-0014	consultarCitasActuales
Tipo del resultado	Set(PeticionCita)-Conjunto de citas aceptadas
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0009] Consultar citas actuales
Descripción	El usuario desea consultar las citas actuales
Parámetros	<p>usuario : Usuario -- <i>usuario que desea consultar sus citas asociadas</i></p> <p>fecha : Date -- <i>fecha en la que se desea consultar las peticiones de cita</i></p>
Expresiones de precondición	pre1: El usuario se ha identificado
Expresiones de postcondición	post1: Se ha mostrado las peticiones de cita por fecha, hora de inicio y el profesor

Excepción	consultarCitasActuales::excep1
Condición	El usuario ha introducido una fecha errónea
Expresión	se muestra el error y se vuelve a pedir la fecha al usuario

SOP-0015	consultarHorarioTutorias
Tipo del resultado	Set(Tutorias)- conjunto de tutorías que forman el horario del profesor
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias

	<u>Noel Guillén Montilla</u>
Dependencias	• [UC-0005] Consultar el horario de tutorías
Descripción	El alumno desea consultar las tutorías del profesor
Parámetros	alumno : Alumno -- alumno que realiza la consulta profesor : Profesor -- profesor del que se desean ver las tutorías
Expresiones de precondición	pre1: El usuario está identificado
Expresiones de postcondición	post1: El sistema ha mostrado las tutorías del profesor

SOP-0016	modificarTutorias
Tipo del resultado	Set(Tutoria)-- Las tutorias modificadas por el profesor
Versión	1.0 (04/10/2006)
Autores	Fernando López Iglesias Noel Guillén Montilla
Dependencias	• [UC-0004] Modificar horario de tutorías
Descripción	El profesor desea modificar o añadir tutorías a su horario
Parámetros	profesor : Profesor -- profesor que desea modificar sus tutorías
Expresiones de precondición	pre1: El usuario esta autenticado
Expresiones de postcondición	post1: Se han añadido o modificado las tutorías post2: El sistema notifica que se ha realizado correctamente
Comentarios	Ninguno

Excepción	modificarTutorias::excep1
Condición	El horario no tiene rangos de horarios validos
Expresión	Se muestra el error y se pide al usuario que cambie los datos

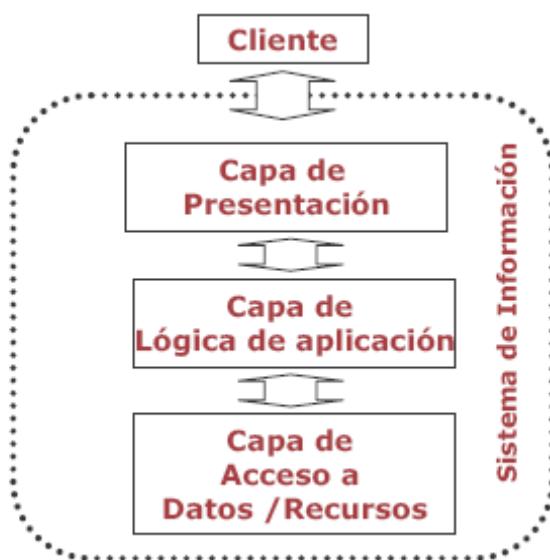
8.3 Diseño del sistema

8.3.1 Patrones usados

A continuación, se explicarán los distintos patrones usados para el diseño del sistema, y donde ha sido aplicado

Patrón Arquitectónico Orientado a Capas (Capas)

Uno de los diseños para aplicaciones Web más empleados es realizar una división de la arquitectura de las aplicaciones en niveles. El patrón arquitectónico orientado a capas es aquel que divide una aplicación en diferentes capas funcionales independientes. En inglés, suele usarse el término *tier* o *layer*, que en este documento ha sido traducido por capas. De manera particular, se puede hablar del *patrón arquitectónico 3-tier* o de 3 capas, que es aquel que se divide en 3 capas o niveles funcionales. Cada una de estas capas se encarga de implementar un grupo de funciones independientes.



Esquema de la arquitectura de 3 capas

A continuación, pasaremos a detallar cómo se ha seguido el diseño de cada una de las capas en la aplicación:

Capa de presentación

Esta capa es la que se encarga de mostrar información del sistema al usuario. Así, estará formada por páginas JSP's con los que interacciona el usuario y cuya lógica se basa en las funciones que ofrece la capa de presentación implementadas mediante taglibraries de struts.

Capa de aplicación (lógica de negocio o dominio)

La capa de aplicación se encargará de recibir los datos del usuario, llamará a las funciones necesarias y de devolver los resultados a la capa de presentación.

En nuestro caso la información la envía y reciben los actions, usan los actionForm como medio. La lógica se encuentra en las clases manager

Capa de datos

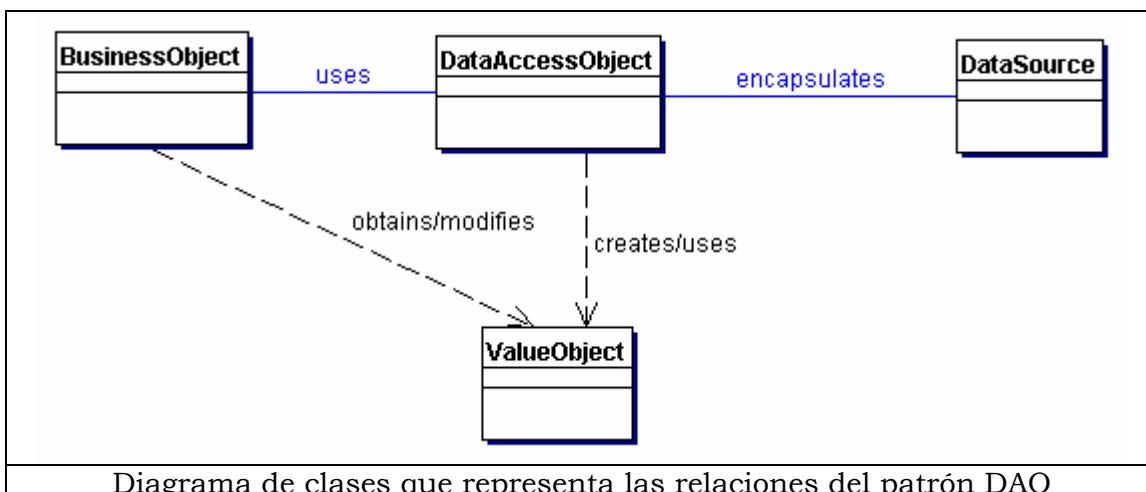
Esta capa ofrecerá los métodos para acceder a los datos, permitirá que la capa de aplicación interactúe con las tablas del SGBD y acceda a información contenida en el mismo.

Patrón DAO

El acceso a los datos varía dependiendo de la fuente de los datos. El acceso al almacenamiento persistente, como una base de datos, varía en gran medida dependiendo del tipo de almacenamiento (bases de datos relacionales, bases de datos orientadas a objetos, ficheros planos, etc.) y de la implementación del vendedor.

El DAO implementa el mecanismo de acceso requerido para trabajar con la fuente de datos. El DAO oculta completamente los detalles de implementación de la fuente de datos a sus clientes. Como el interface expuesto por el DAO no cambia cuando cambia la implementación de la fuente de datos subyacente, este patrón permite al DAO adaptarse a diferentes esquemas de almacenamiento sin que esto afecte a sus clientes o componentes de negocio. Esencialmente, el DAO actúa como un adaptador entre el componente y la fuente de datos.

En este gráfico se puede ver cómo es el funcionamiento del patrón:



Descripción de los objetos

BusinessObject: representa los datos del cliente. Es el objeto que requiere el acceso a la fuente de datos para obtener y almacenar datos. Podríamos implementar un BusinessObject como un bean.

DataAccessObject: es el objeto principal de este patrón. DataAccessObject abstrae la implementación del acceso a datos subyacente al BusinessObject para permitirle un acceso transparente a la fuente de datos. El BusinessObject también delega las operaciones de carga y almacenamiento en el DataAccessObject.

DataSource: representa la implementación de la fuente de datos.

TransferObject: representa un Transfer Object utilizado para el transporte de datos. DataAccessObject podría utilizar un Transfer Object para devolver los datos al cliente. El DataAccessObject también podría recibir datos desde el cliente en un Transfer Object para actualizar los datos en la fuente de datos.

Ventajas de su uso

- Reduce la Complejidad del Código de los Objetos de Negocio :
Como los DAOs manejan todas las complejidades del acceso a los datos, se simplifica el código de los objetos de negocio y de otros clientes que utilizan los DAOs. Todo el código relacionado con la implementación (como las sentencias SQL) está dentro del DAO y no en el objeto de negocio. Esto mejora la lectura del código y la productividad del desarrollo.
- Permite la Transparencia :
Los objetos de negocio puede utilizar la fuente de datos sin conocer los detalles específicos de su implementación. El acceso es transparente porque los detalles de la implementación se ocultan dentro del DAO.

Aplicación en el proyecto

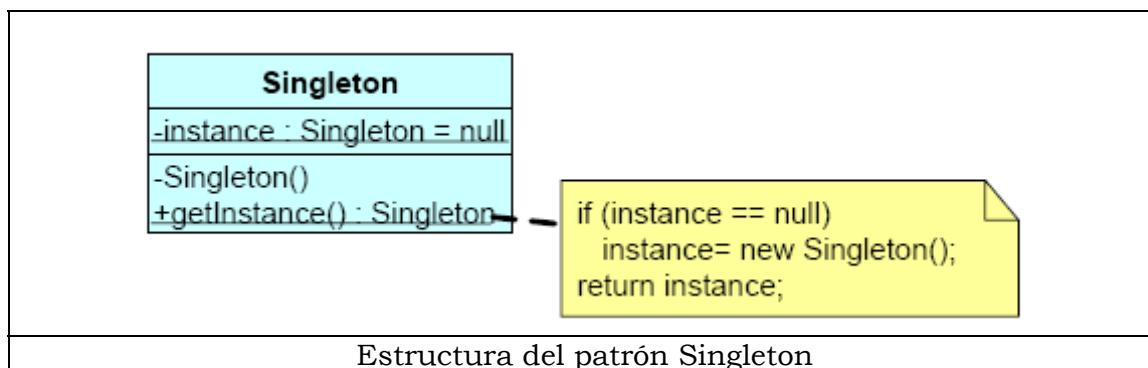
En nuestro caso, hemos aplicado el patrón aunque con unas leves diferencias debido al uso de hibernate. Los DAO no acceden mediante JDBC directamente como lo harían normalmente en una aplicación similar, hay una capa intermedia debido a que Hibernate media entre los dos. Por tanto en los DAO se usa la funcionalidad de Hibernate, con una notable reducción de código.

Patrón Singleton

Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón singleton se implementa creando en nuestra clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado).



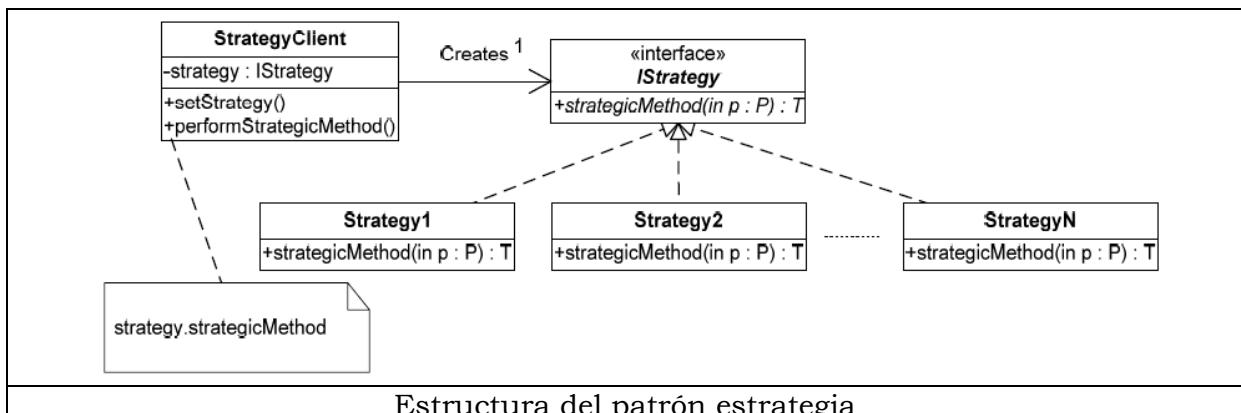
En nuestro caso hemos requerido de este patrón en dos ocasiones por los siguientes motivos:

- HibernatePool: Es la clase que maneja las conexiones y necesitábamos que se controlara el objeto sessionFactory que es el encargado de manejar los objetos de la conexión y debe de ser único, ya que la creación otro objeto de ese tipo implica la inicialización de Hibernate que es muy costosa.
- LectorScript: Esta clase es la encargada de leer del archivo properties y leer del fichero de script apropiado. Es singleton, ya que no era apropiado que leyera del fichero de properties continuamente, si no que lo hiciera sólo una vez y se cargaran las properties de forma estática.

Patrón Estrategia

El patrón Strategy permite mantener un conjunto de algoritmos de los que el objeto cliente puede elegir aquel que le conviene e intercambiarlo según sus necesidades.

Cuando tenemos un algoritmo que puede cambiar, por cualquier motivo, bien en tiempo de ejecución, bien cuando lo implementemos en otro sitio, y no queremos modificar el código que usa ese algoritmo, viene bien usar el patrón estrategia.



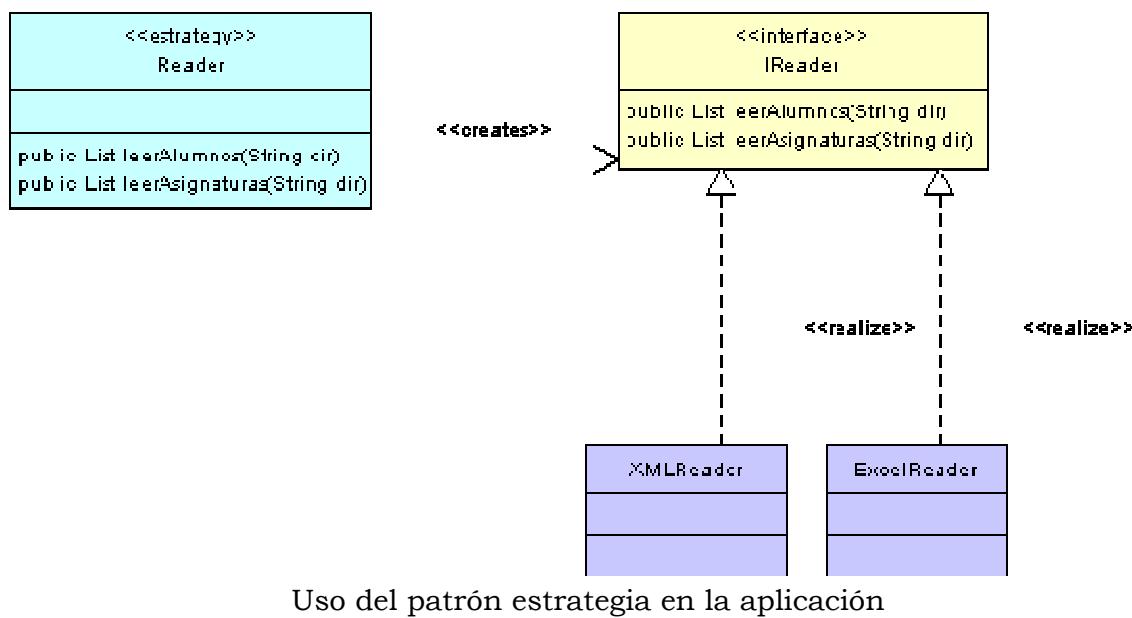
Consecuencias

- Factoriza aspectos comunes de una familia de algoritmos y utilizarlos en las clases base de la jerarquía
- Aumenta cohesión del cliente
- Sistematiza el uso de implementaciones alternativas
- El cliente es el responsable de crear estrategias, por tanto debe comprender las posibilidades que ofrecen, esto es, debe ser relevante para el contexto del cliente
- Menor eficiencia. Aumenta el número de objetos creados

Aplicación

En nuestro caso teníamos el problema de que necesitamos leer de varios tipos de ficheros para realizar la importación de usuario y asignaturas. Además era posible que se añadieran más tipos a la aplicación en un futuro y era necesario que se eligiera el tipo de implementación según el tipo de archivo en tiempo de ejecución.

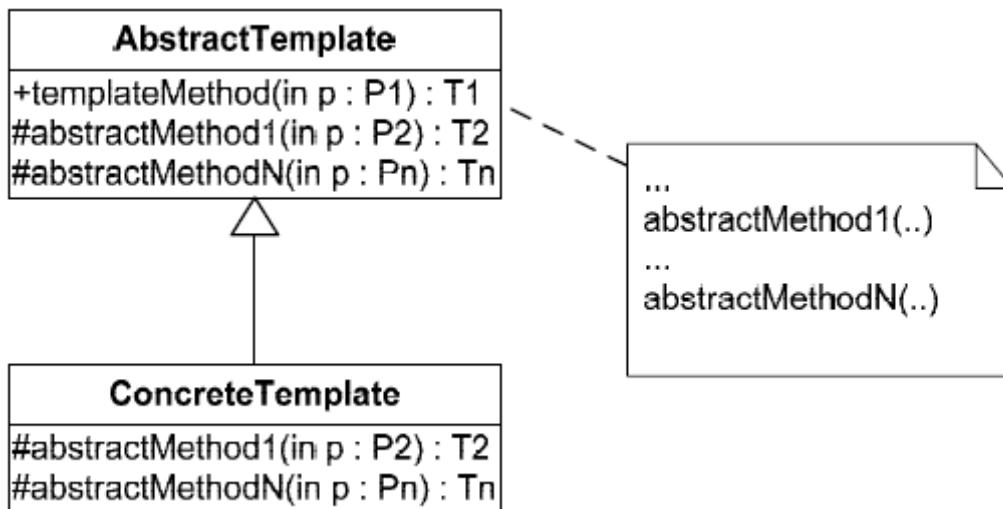
Este problema es solucionado a través del uso del patrón estrategia mediante el cual definimos una interfaz que deben de implementar las clases encargadas de leer los ficheros. También creamos la clase Reader que será la encargada de crear la clase correspondiente según la extensión del archivo que se le pase por parámetro.



Template Method (método plantilla)

Un Template Method es un patrón de diseño que define una estructura algorítmica en la super clase, delegando la implementación a las subclases. Es decir, define una serie de pasos, en donde los pasos serán redefinidos en las subclases.

Este patrón permite controlar las ampliaciones de las subclases, convirtiendo en métodos plantillas aquéllos métodos que pueden ser redefinidos. En este caso, los métodos abstractos se les denominan métodos de enganche (*hook*)



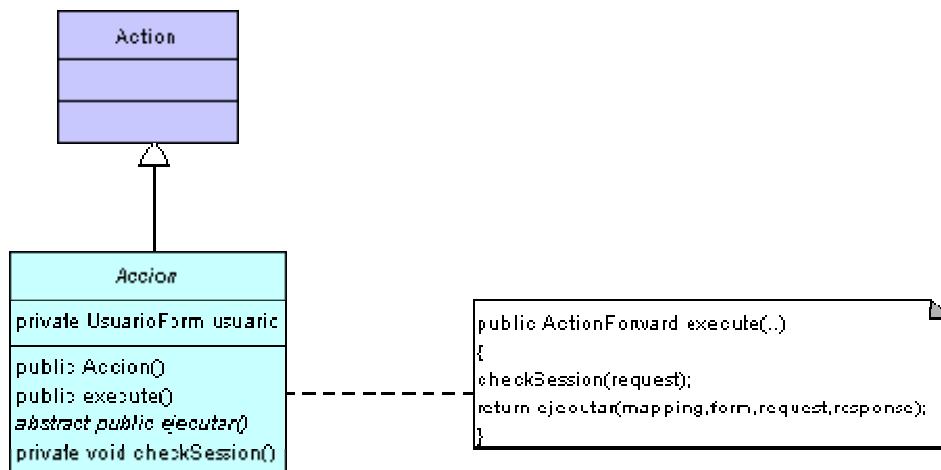
Implementación del patrón Template Method

Consecuencias (ventajas e inconvenientes)

- Favorece la reutilización del código. Muy útiles para construir bibliotecas, pues ayuda a factorizar el comportamiento común de las clases
- Lleva a una estructura de control invertido (Principio de Hollywood): la superclase base invoca los métodos de las subclases

En nuestro caso, queríamos crear una política de seguridad de acceso en las acciones, para que un usuario no identificado pudiera acceder a ellos. Con lo cual se presentaba el problema de que todas las clases tenían que comprobar los datos ingresados en la sesión y no queríamos duplicar código ya que complica el mantenimiento de la aplicación.

Por eso usamos este patrón y con ello creamos la clase *Accion*, que hereda de la clase *Action* de Struts, y su método de enganche *ejecutar* que es el que tienen que redefinir todas las clases que requieran que el usuario este identificado. El método de enganche es llamado a su vez por *execute* realizando previamente una comprobación de la sesión. Con lo cual ya no se replica código.



Aplicación del patrón Template Method

Composite view

Este patrón hace que la representación de vistas sea más manejable ya que gestiona los diferentes elementos de una página por medio de una plantilla. Frecuentemente, las páginas Webs contienen una combinación de contenido dinámico y elementos estáticos, tales como cabeceras, pies, logos, imágenes de fondo, etc. La parte dinámica es particular para cada página, pero los elementos estáticos suelen ser los mismos para todas las páginas. La plantilla de la vista compuesta captura estas características comunes. La integración debe ser dinámica, siendo el Composite View básicamente un layout (diseño, esquema) que componga dicha página.

En nuestro caso hemos dividido la página Web en 4 partes, cabecera, pie, menú y head que contiene todo el contenido de head. Esta solución nos ha ayudado a mejorar la facilidad de mantenimiento, ya que no es necesario modificar una a una cada página en jsp para cambiar elementos comunes como pueden ser la cabecera.

8.3.2 Diseño de la aplicación

En esta sección se mostrarán y explicarán los diagramas de clases a partir de la solución obtenida a través del uso de los patrones elegidos en la etapa anterior.

En primer lugar se presentará un diagrama general para tener una visión global, además separadas por las tres capas principales de la aplicación: presentación, aplicación y datos.

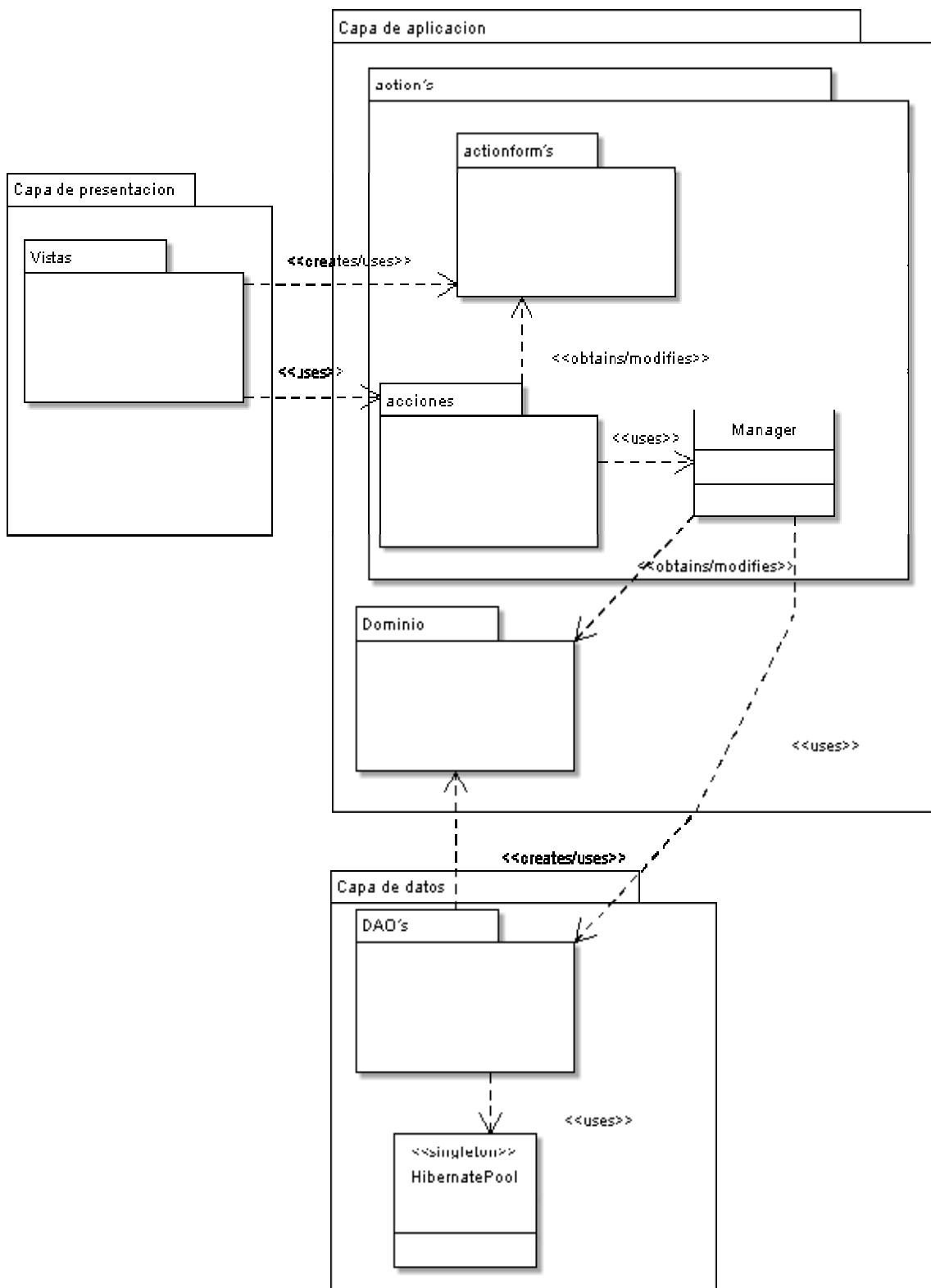


Diagrama general del diseño de la aplicación

Posteriormente se mostrarán los diagramas divididos por capas

8.3.2.1 Capa de datos

Esta se encuentra recogida en el paquete data, en el centro está la clase HibernatePool que es la encargada de devolver las conexiones a la base de datos mediante Hibernate. A su alrededor se encuentran las clases que implementan el patrón DAO con sus respectivas interfaces. Las clases DAO usan las conexiones de HibernatePool y manejan Beans del paquete dominio.

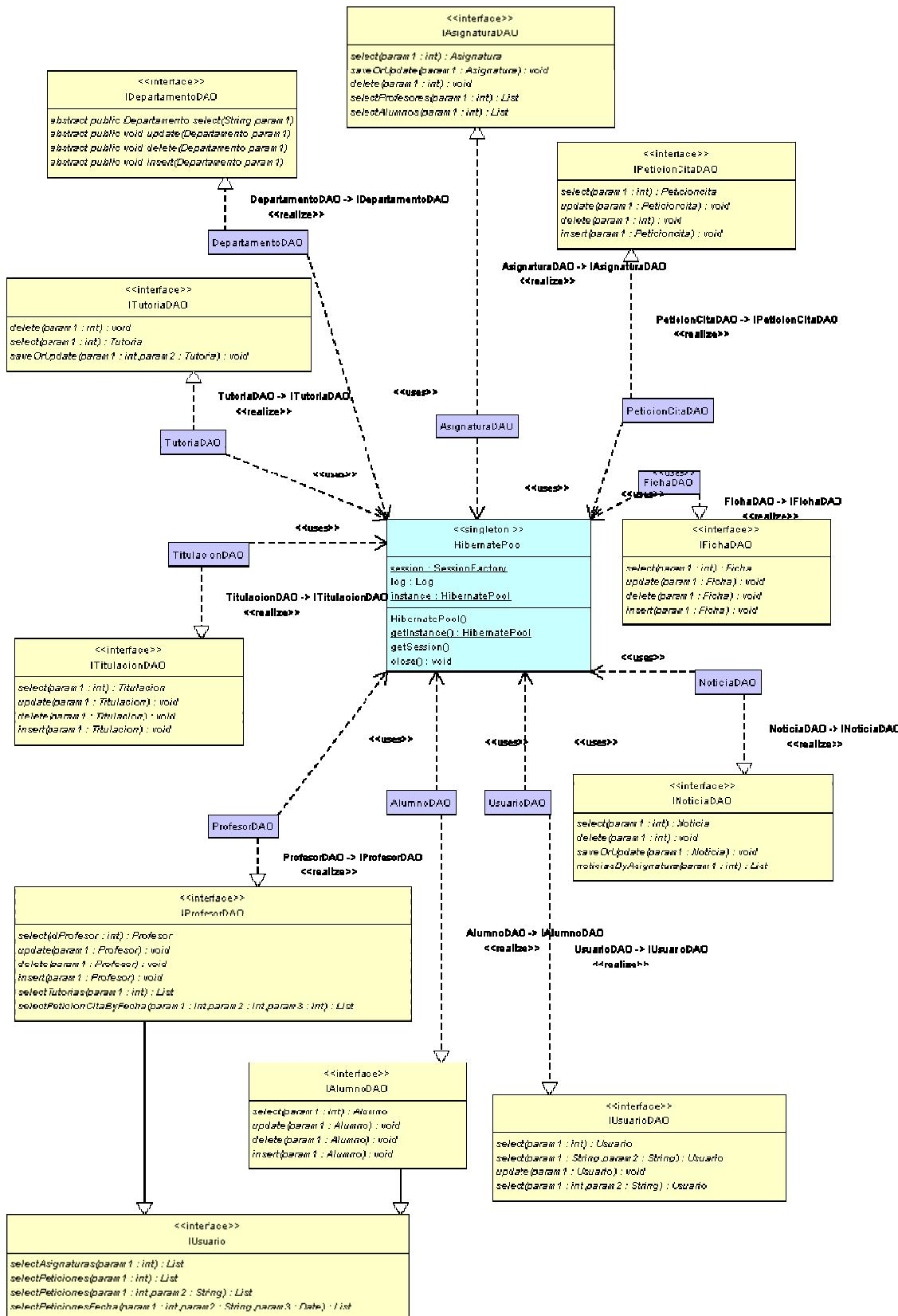
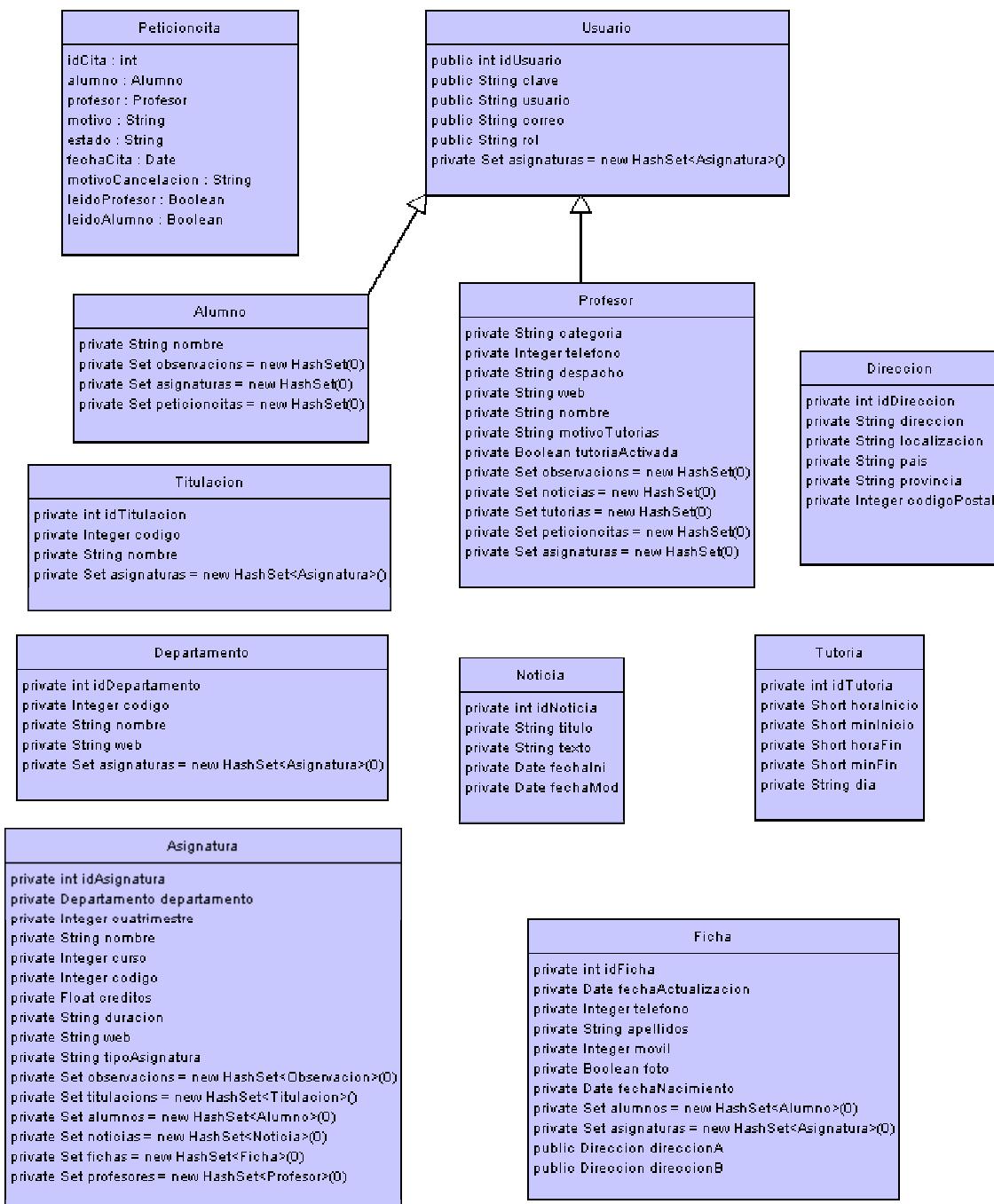


Diagrama de las clases de la capa de datos

8.3.2.2 Capa de aplicación

Clases del dominio de la aplicación

A continuación se mostrará el diagrama de las clases del dominio, creado a través del diagrama del conceptual, éstas representan a los ValueObjects en el patrón DAO, son los objetos que están entre la lógica de la aplicación y la capa de datos. Como son Bean tienen métodos set y get por cada atributo los cuales no se han incluido por motivos de legibilidad.



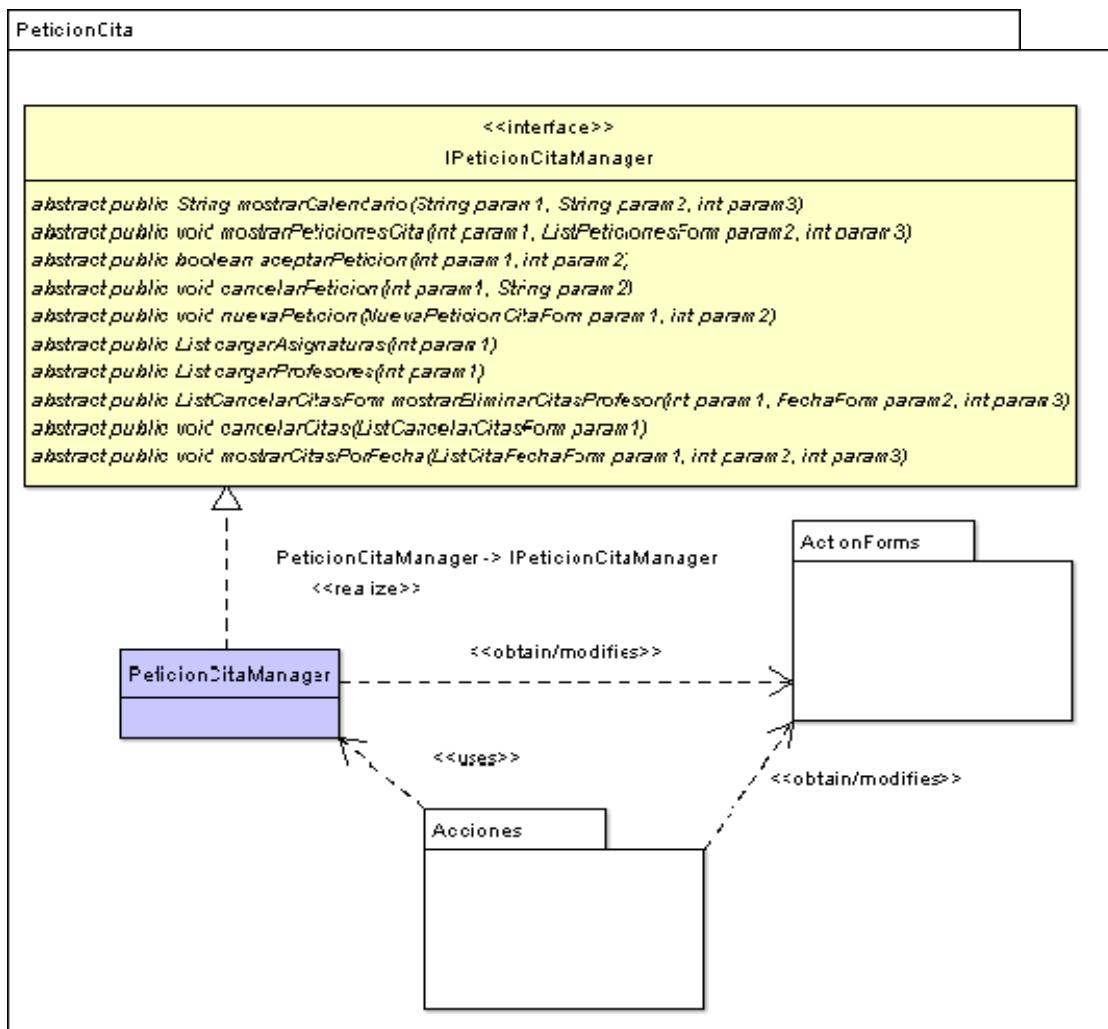
Clases del dominio

Clases de los módulos de la aplicación

Posteriormente se describirán los distintos módulos en los que se han separado la aplicación y se ilustrará mediante diagramas de clase su contenido y relaciones entre sus clases, con la consideración de que no se mostrarán las acciones ni los actionsforms debido a que el diagrama no sería manejable por el número de clases. Además las acciones son clases con un sólo método y se encargan de redireccionar y llamar a la funcionalidad de la clase manager correspondiente del módulo.

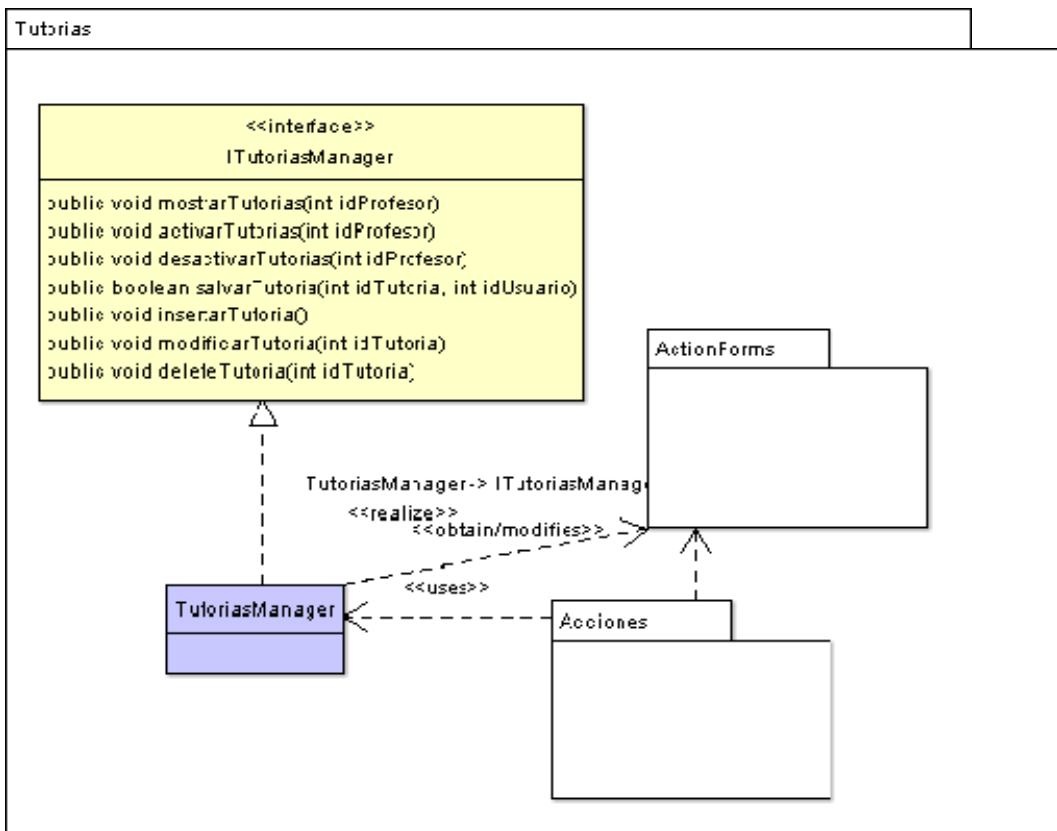
Módulo Tutorías

Este módulo se ha separado en dos submodulos debido a que se podía agrupar su funcionalidad en dos tipos. El primero está relacionado con la funcionalidad de las citas y peticiones de citas de los alumnos y de los profesores.



Lógica de aplicación del módulo PeticionCita

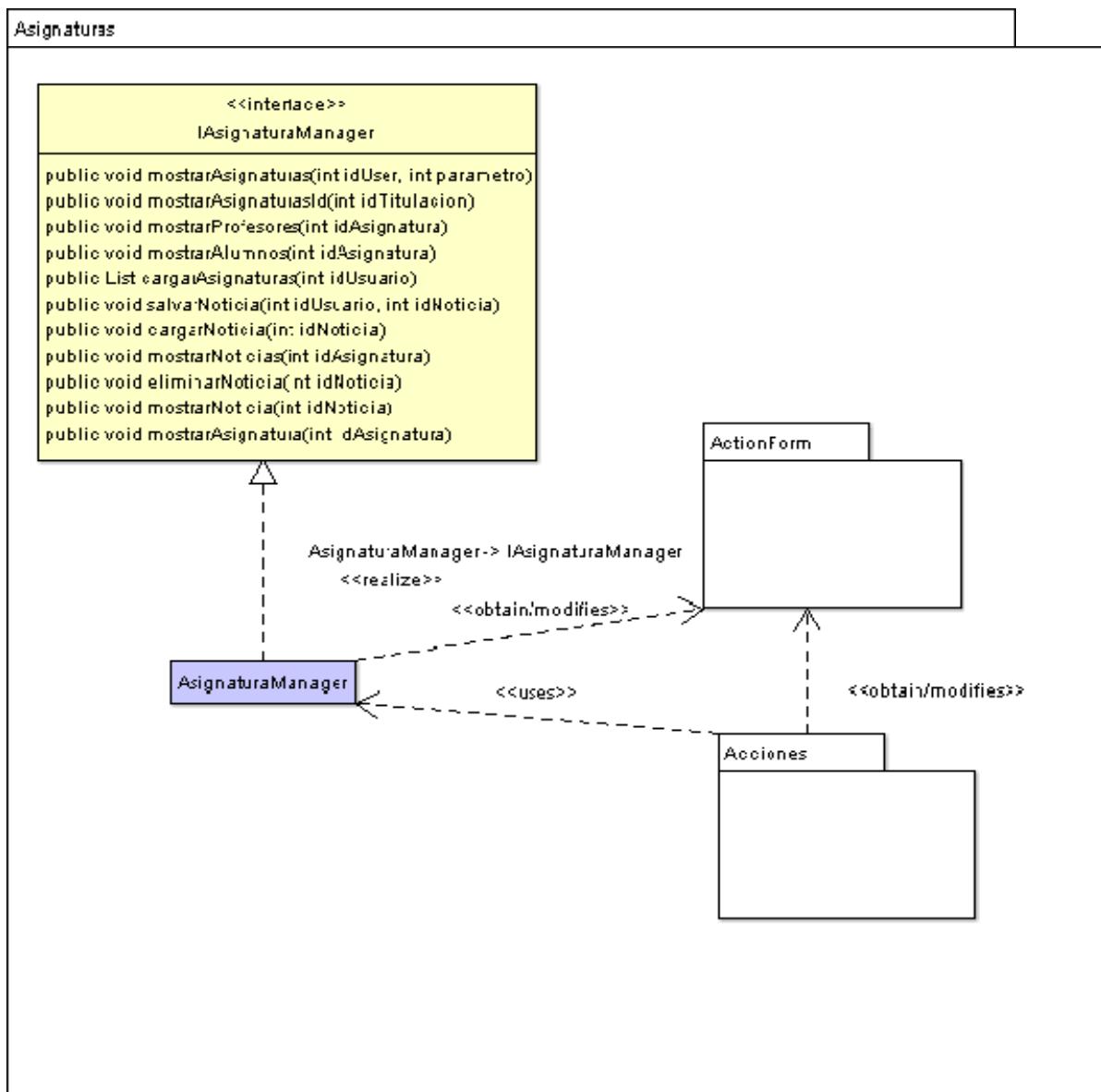
El segundo modulo se encarga de la funcionalidad relacionada con las tutorías de los profesores, como la activación, desactivación, eliminación, adición y modificación de tutorías.



Lógica de aplicación del modulo tutorías

Módulo Asignaturas

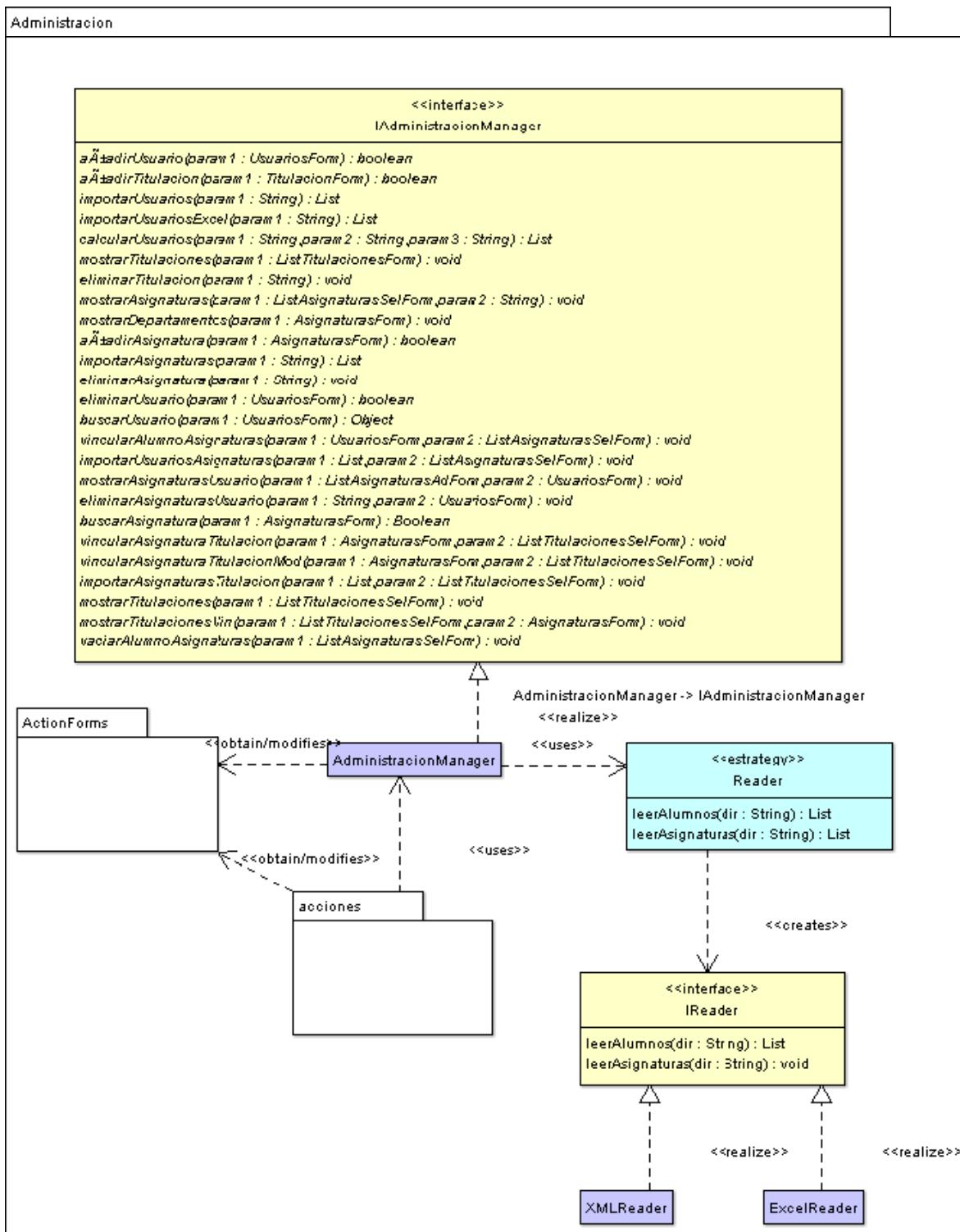
El siguiente módulo es el encargado de manejar la información referente a las asignaturas, y por consiguiente también las noticias asociadas a las asignaturas.



Lógica de aplicación del módulo asignaturas

Módulo administración

Es el módulo con más complejidad sin duda, es el encargado de manejar los usuarios y asignaturas del sistema. Además permite importar los usuarios y asignaturas en diferentes formatos, mediante el uso de la clase Reader que implementa el patrón estrategia.



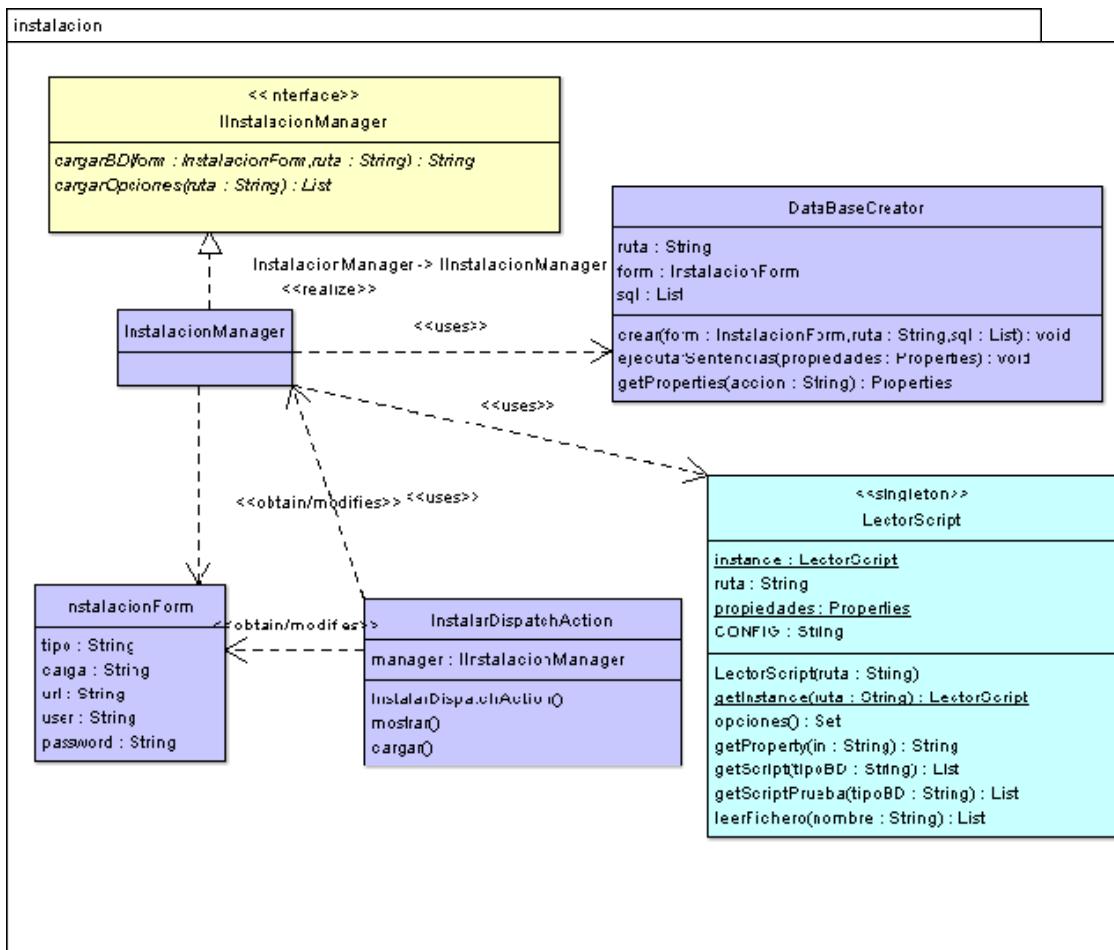
Lógica de aplicación del módulo administración

Módulo Instalación

Este módulo es el encargado de instalar la aplicación en una base de datos y los correspondientes datos de configuración de Hibernate. Para ello InstalacionManager se apoya en dos clases: DataBaseCreator y LectorScript.

DataBaseCreator es la encargada de generar el fichero de properties que usa hibernate para su configuración y también de ejecutar el script de la base de datos correspondiente.

LectorScript tiene la responsabilidad de leer el archivo de properties con los datos de los gestores de bases de datos disponibles, configurable por el desarrollador. También es el encargado de devolver una lista con las sentencias del fichero de script correspondiente.

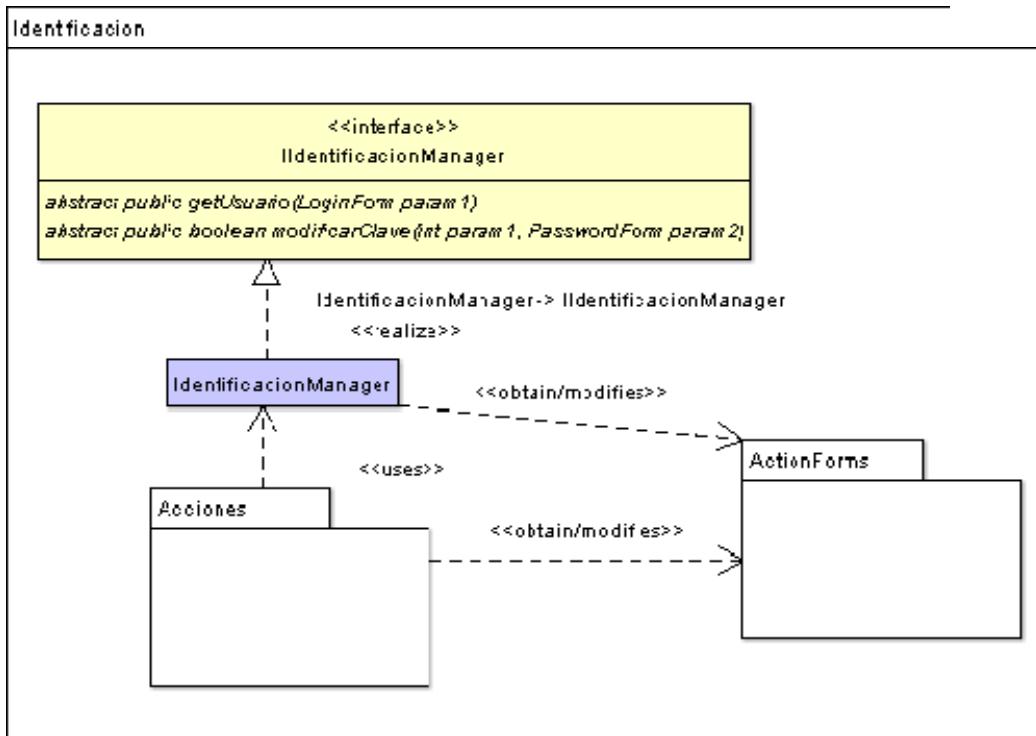


Lógica de aplicación del módulo Instalación

Hay que destacar que dado que en este módulo sólo hay un actionform y una acción se ha incluido en el diagrama.

Módulo Identificación

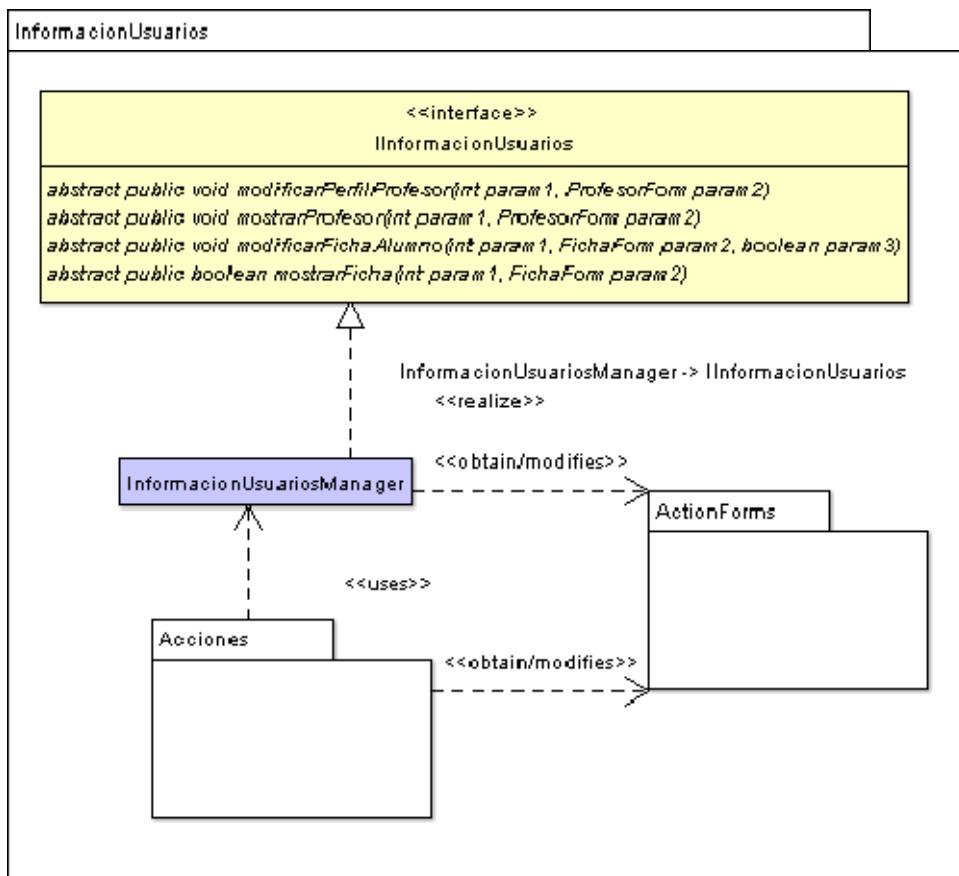
El siguiente módulo tiene como responsabilidad la identificación, modificación de la clave y salida del usuario.



Lógica de aplicación del módulo Identificación

Módulo InformacionUsuario

Módulo que gestiona todo lo relacionado con la información referente a los usuarios. En el caso del profesor la información referente a su perfil (despacho, nombre, etc...). En el caso del alumno una funcionalidad importante y por la que empezó el proyecto, la gestión de las fichas.

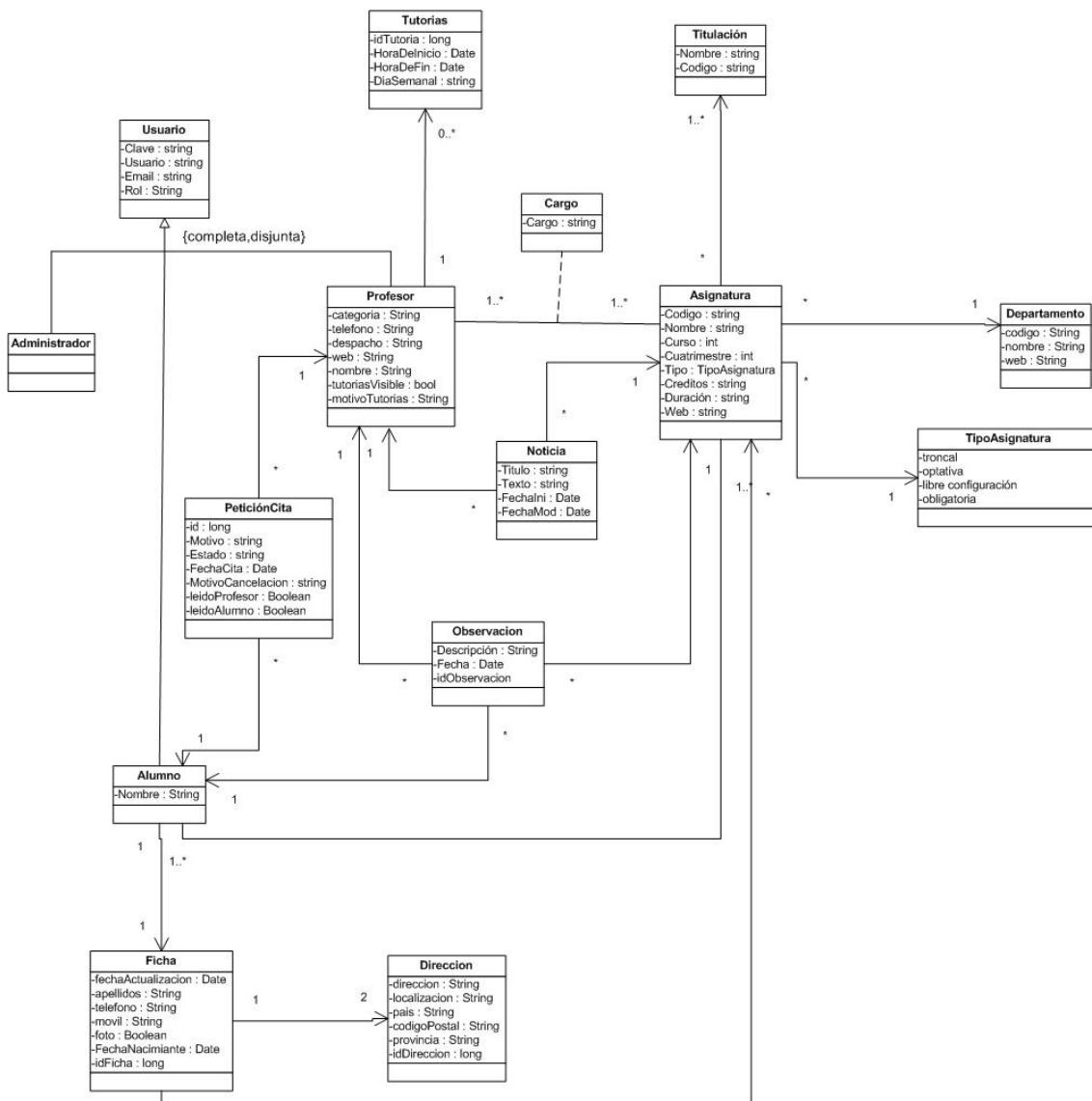


Lógica de aplicación del módulo InformacionUsuarios

8.3.2 Diseño de la base de datos

En este punto se diseñará el esquema de la base de datos del sistema para satisfacer los requisitos de la aplicación con respecto al almacenamiento de datos.

El proceso comienza con el paso del esquema de tipos representado, realizado en el documento de análisis del sistema, a un ERD (Diagrama entidad relación).

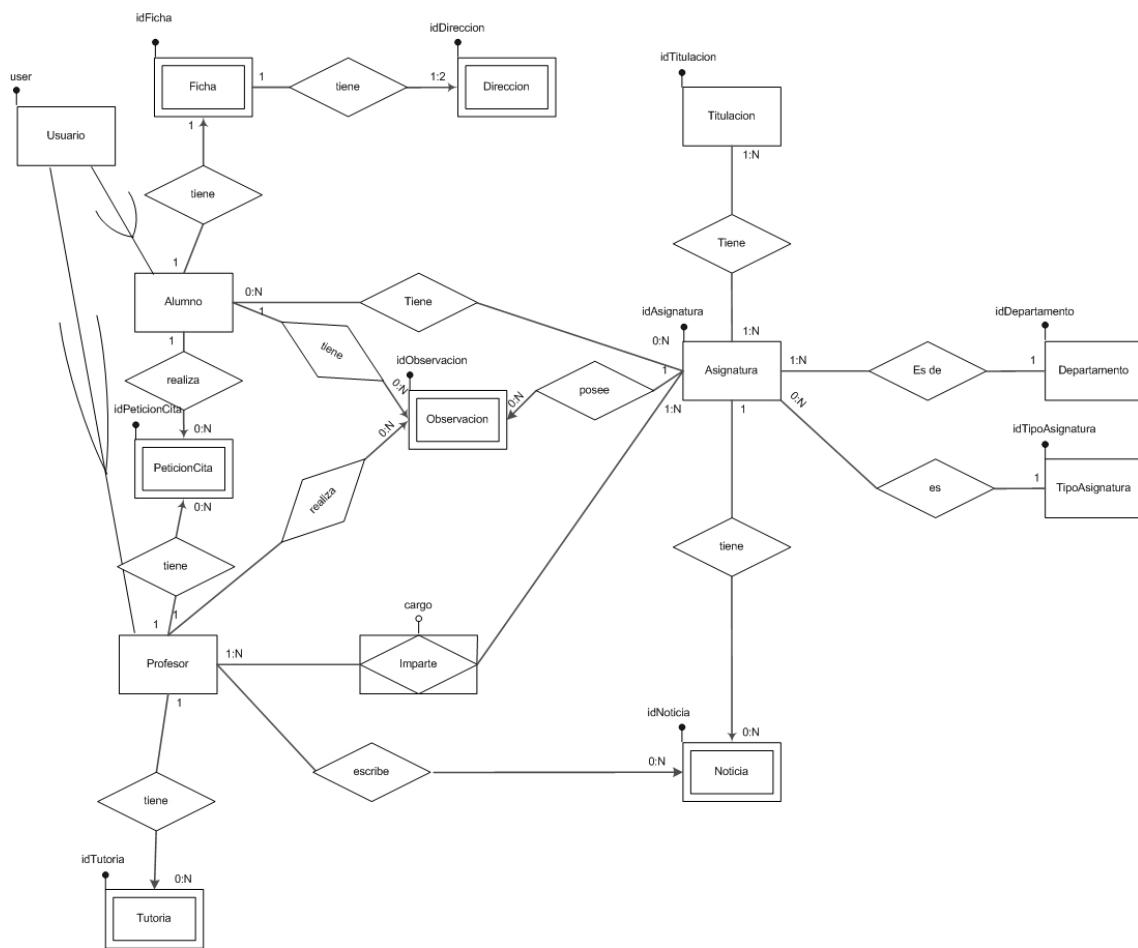


Esquema del diagrama de tipos

Las similitudes de los dos esquemas hacen que el paso no sea de extrema dificultad, se hace de forma casi automática siguiendo los siguientes criterios:

- Todo tipo de entidad se convierte en una relación/tabla de la Base de Datos.
- Todo tipo de interrelación N:M se transforma en una relación/tabla, es decir, se añadirá una tabla nueva que contendrá la clave primaria de la entidad perteneciente al extremo N y de la entidad perteneciente al M, de manera que la clave primaria de esta nueva tabla será la suma de las dos.
- Todo tipo de interrelación 1:N, se construirán dos tablas de manera que pueda haber dos opciones para la clave primaria: que la clave primaria de la entidad que posee 1 pase a la de N o viceversa.

Después de aplicar los criterios el ERD queda de la siguiente forma:



Esquema ERD

A continuación, detallaremos las tablas de la base de datos creadas a partir del ERD, mostrando cada uno de sus campos.

TABLA FICHA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idFicha	Integer	No	-1	X		Identificador de la ficha
fechaActualizacion	Date	Si				Fecha de actualización de la ficha
telefono	Integer	Si				Teléfono del contacto
Apellidos	Varchar(255)	Si				Apellidos del usuario
movil	Integer	Si				Número de teléfono móvil
Foto	Tynint	SI				Booleano que representa si el usuario tiene foto
fechaNacimiento	Date	Si				Fecha de nacimiento
idDireccionA	Integer	Si		X		Dirección de residencia
idDireccionB	Integer	Si		X		Dirección de residencia real

Tabla Ficha

TABLA DIRECCION

ANÁLISIS DEL SISTEMA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idDireccion	Integer	No	-1	X		Identificador de direccion
Direccion	Varchar(100)	SI				Direccion de la ficha
Localizacion	Varchar(100)	SI				Localización de la ficha
Pais	Varchar(50)	SI				Pais de la ficha
Provincia	Varchar(50)	SI				Provincia de la ficha
codigoPostal	Varchar(5)	SI				Codigo postal

Tabla Direccion

TABLA USUARIO

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idUsuario	Integer	No	-1	X		Identificador de direccion
Clave	Varchar(20)	No				Clave del usuario
Usuario	Varchar(20)	No				Nombre del usuario
Correo	Varchar(50)	Si				Correo del usuario
Rol	Enum (alumno, profesor, administrador)					

Tabla Usuario

TABLA ALUMNO

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idUsuario	Integer	No	-1	X	X	Identificador del usuario
Nombre	Varchar(255)	Si				Nombre del usuario
idFicha	Integer	Si			X	Identificador de la ficha

Tabla Alumno

TABLA PROFESOR

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idUsuario	Integer	No	-1	X	X	Identificador del usuario
Categoría	Varchar(100)	Si				Categoría de trabajo
Despacho	Varchar(100)	Si				Dirección del despacho
Web	Varchar(100)	Si				Página Web
Nombre	Varchar(100)	Si				Nombre del profesor
motivoTutorias	Varchar(100)	Si				Motivo de cancelación de las tutorías
tutoriaActivada	Tinyint(1)	Si				Booleano que indica si están activadas las tutorías

Tabla profesor

TABLA DEPARTAMENTO

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idDepartamento	Integer	No	-1	X		Identificador del departamento
Nombre	Varchar(255)	Si				Nombre del departamento
Web	Varchar(255)	Si				Página Web
Código	Integer(11)	Si				Código de departamento

Tabla departamento

TABLA ASIGNATURA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idAsignatura	Integer(9)	No	-1	X		Identificador de la asignatura
cuatrimestre	tinyint(1)	Si				Cuatrimestre

ANÁLISIS DEL SISTEMA

nombre	varchar(100)	Si				Nombre
curso	tinyint(1)	Si				Curso
codigo	Integer(11)	Si				Código de la asignatura
creditos	float(9,3)	Si				Créditos
Duracion	Enum (anual, cuatrimestral)	Si				Duración de la asignatura
Web	Varchar(150)	Si				Página Web
tipoAsignatura	Enum (optativa, troncal, obligatoria, libre configuracion)	Si				Tipo de asignatura
idDepartamento	Integer(9)	Si				Departamento

Tabla Asignatura

TABLA APROBADAASIGNATURAS

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idFicha	Integer(9)	No		X	X	Identificador de la ficha
idAsignatura	Integer(9)	No		X	X	Identificador de asignatura

Tabla aprobadaAsignaturas

TABLA TITULACION

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idTitulacion	Integer(9)	No	-1	X		Identificador de titulación
Codigo	Integer(11)	Si				Código de la titulación
Nombre	Varchar(255)	Si				Nombre de la titulación

Tabla Titulacion

TABLA ASIGNATURATITULACIONES

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idTitulacion	Integer(9)	No		X	X	Identificador de titulación
idAsignatura	Integer(9)	No		X	X	Identificador de asignatura

Tabla AsignaturaTitulaciones

TABLA IMPARTEASIGNATURA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
profesor	Integer(9)	No		X	X	Identificador de profesor
idAsignatura	Integer(9)	No		X	X	Identificador de asignatura
Cargo	Varchar(255)	Si				Cargo del profesor en la asignatura

Tabla ImparteAsignatura

TABLA NOTICIA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idNoticia	Integer(9)	No	-1	X		Identificador de noticia
Titulo	Varchar(100)	Si				Titulo de la noticia
Texto	Text	Si				Contenido
fechaIni	Date	Si				Fecha de creación
fechaMod	Date	Si				Fecha de modificación
Asignatura	Integer(9)	No		X		Identificador de noticia
Profesor	Integer(9)	No		X		Identificador de noticia

Tabla Noticia

TABLA PETICIONCITA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idCita	Integer(9)	No	-1	X		Identificador de la cita
Motivo	Varchar(255)	Si				Motivo de la

ANÁLISIS DEL SISTEMA

					cita
Estado	Enum(cancelada, en proceso, aceptada)	Si			Estado de la petición
fechaCita	Timestamp	Si			Fecha y hora de la cita
motivoCancelacion	Varchar(255)	Si			Motivo de cancelación
leidoProfesor	Tinyint(1)	Si			Booleano que indica si la cita ha sido leída por el profesor
leidoAlumno	Tinyint(1)	Si			Booleano que indica si la cita ha sido leída por el alumno
Alumno	Integer(9)	No		X	Identificador de alumno
Profesor	Integer(9)	No		X	Identificador de profesor

Tabla PeticionCita

TABLA RECIBEASIGNATURA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idAlumno	Integer(9)	No		X	X	Identificador de alumno
idAsignatura	Integer(9)	No		X	X	Identificador de asignatura

Tabla recibeAsignatura

TABLA TUTORÍA

Campo	Tipo	Nulo	Predeterminado	PK	FK	Comentario
idTutoria	Integer(9)	No	-1	X		Identificador de tutoría
idUsuario	Integer(9)	No			X	Identificador del profesor
horaInicio	Smallint(2)	Si				Hora de inicio de la tutoría
minInicio	Smallint(2)	Si				Minutos de inicio de la tutoría
horaFin	Smallint(2)	Si				Hora de fin de la tutoría
minFin	Smallint(2)	Si				Minutos de fin de la tutoría

Tabla Tutoria

8.4 Puntos de variación y evolución

8.4.1 Puntos de variación

En este apartado explicaremos los posibles cambios que pueden surgir y como afectaría esto al sistema, si nos hemos protegido ante el cambio realizando un análisis del coste y el impacto que tendría.

Cambio de idioma de la aplicación

Nos hemos protegido ante el cambio de idioma de la aplicación, usando una de las funcionalidades que nos proporciona struts que soporta internacionalización. Mediante ficheros de mensajes en los que se relaciona una clave con un texto, luego el texto se muestra dada una clave concreta en

la página Web. El fichero de mensajes se carga en la caché para que se realice de forma eficiente. Aunque no hemos realizado el fichero en otros idiomas, sería tan sencillo como crearlo con el nombre de Mensajes_xx.properties, donde xx son las siglas del país (Por ejemplo inglés es EN). Esto es muy interesante ya que no hay que modificar nada de código ni recompilar la aplicación. El idioma se recoge a través del idioma que tenga el navegador.

Cambio de gestor de base de datos

Gracias al ORM Hibernate, el cambio de base de datos es muy sencillo, no hay que modificar nada de código. Sólo habría que tocar algún que otro fichero de configuración, añadir el conector de la base de datos y crear el script de la base de datos, aunque existen programas que convierten el script de un tipo de base de datos a otro. Hibernate soporta una infinidad de gestores de bases de datos entre ellos los más utilizados.

Cambio de diseño de la página

Hemos usado CSS, sólo habría que modificar su configuración, por tanto también nos hemos protegido ante el cambio de diseño. Además con el uso de las taglibraries está más separada la lógica de negocio de la capa de presentación, y un cambio en la capa de presentación no supone un gran impacto ni coste en la aplicación.

Adición de un nuevo módulo

La adición de un nuevo módulo o funcionalidades no supone un gran impacto en la aplicación, ya que las capas están debidamente separadas y todo está estructurado y comentado.

8.4.2 Puntos de evolución

Este apartado pretende ser una guía sobre las posibles mejoras con las que podría evolucionar el proyecto hacia las futuras versiones.

Cambiar el esquema de la base de datos

Al diseño de la base de datos se le podría añadir que tuviera en cuenta el año en el que esta matriculado un alumno en la base de datos. También se podría tener en cuenta la duración de las citas, aunque como mínimo hemos puesto que haya una separación de 30m, esto no se refleja en la base de datos.

Ajuste de la configuración de hibernate

Una vez estuviera la aplicación en producción sería bueno revisar el registro para adaptar los valores de la configuración de hibernate y su pool de conexiones al del número de usuario normales para obtener un mayor rendimiento en la aplicación y un mejor uso del caché que implementa Hibernate. Para ello habría que editar el fichero de configuración de Hibernate, en concreto el fichero Hibernate.properties que son las propiedades que lee hibernate para configurarse.

Validación en el lado del cliente

Para descargar el servidor de la tarea de realizar las validaciones, sería convenientes realizarlas en el lado del cliente, es decir mediante JavaScript.

Soporte de un mayor numero de base de datos

Para dar soporte a un mayor número de clientes de la aplicación sería conveniente soportar más tipos de gestores de bases de datos. Es decir, adaptar los script de generación en SQL hacia otros lenguajes de bases de datos para que así pudiera funcionar en otros sistemas usados como pueden ser Oracle y PostgreSQL.

Soporte de código HTML

Sería conveniente incluir un editor de texto en Javascript que permitiera introducir links y formato al texto de las noticias, ya que es común que los profesores introduzcan enlaces.

Exportar contactos en csv

Una funcionalidad muy interesante para los profesores sería añadir los correos de los alumnos a un gestor de correo, exportar los contactos en el formato csv que es soportado por muchas aplicaciones que gestionan los contactos.

Permitir hace login con LDAP

LDAP (Lightweight Directory Access Protocol) es un protocolo de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse. Por ello sería interesante que la aplicación soportara dicho protocolo como medio adicional de acceso.

Envío de notificaciones

Sería interesante que se enviaran las notificaciones de las citas mediante correo electrónico para que así el usuario pudiera saber su estado sin tener que entrar en la aplicación. Una posibilidad más sería el envío de mensajes de textos a móvil (SMS) para el aviso de la aceptación de una petición de cita, o la cercanía de una cita.

Generar informes en PDF

Una nueva funcionalidad relacionada con el rol de profesor, sería la de generar en formato PDF un informe con todos los alumnos que tiene una asignatura. También estaría bien generar un resguardo en PDF que el alumno pudiera imprimir y entregar posteriormente al profesor como justificante.

Integración con gestores de eventos

Estaría bien que se pudieran exportar las citas que tiene un usuario en un formato que pudiera aceptar un gestor de eventos, como podría ser Google Calendar o iCalendar y así poder realizar sincronizaciones.

Mejoras en el instalador

Sería interesante que para instalar la base de datos no fuera necesario que existiera una base de datos ya creada, donde se insertan las tablas, si no que sólo se introdujera la dirección del servidor.

Soporte a más lenguajes

Sería bueno que se crearan más ficheros de lenguajes traduciendo el fichero ya existente de mensajes en español, lo cual sería interesante como funcionalidad para los alumnos extranjeros, o para que la aplicación se usara en otros países.

Personalización para distintas entidades

Ya que no supone un gran coste el cambio de diseño de la aplicación, sería bueno que se adaptará la cabecera de la aplicación para que aceptara otras entidades de enseñanza, es decir se pudiera modificar el logo de la aplicación según los requerimientos.

9 Arquitectura del sistema

La aplicación esta estructurada en distintos módulos dependiendo de su funcionalidad. Todos los módulos siguen una estructura similar compuesta por los siguientes elementos:

JSP

Son los elementos mas externos de la aplicación, pertenecen a la capa de presentación y son los encargados de mostrar al usuario la información de forma correcta y ordenada con una interfaz amistosa.

Los JSP se ayudan para la representación de los datos de una hoja de estilo general para toda la aplicación. Todos los JSP tienen una misma estructura que incluyen una cabecera en la parte superior, un menú en la parte de la izquierda y un pie de página en la parte inferior; estos son también JSP.

Los JSP reciben la información a mostrar mediante ActionForm que se describirán a continuación.

ActionForm

Son las clases encargadas de proporcionar la información a la capa de presentación. Todos los ActionForm heredan de la clase ActionForm de STRUTS y están compuestos por diversos números de atributos.

Los ActionForm tienen implementados los métodos Get y Set para todos los atributos que poseen. Además, tienen definidos el método Validate (para validar que sea correcto el valor de los atributos) y el método Reset (para reiniciar los atributos con sus valores originales).

Los actionForm se generan de forma automática cada vez que se lleva a cabo una acción.

Acciones

Son las clases encargadas de obtener los valores de los ActionForm y de responder a las peticiones HTTP, así como de tratar las sesiones. Estas clases inicializan o completan los ActionForm, analizan los resultados y redireccionan al JSP correspondiente. Todas las acciones heredan de la clase “acción”. Para obtener los datos necesarios, las clases se ayudan de las clasesManager.

La clase “acción” es la encargada de comprobar que los usuarios que navegan por la aplicación están correctamente identificados. La clase Action hereda a su vez de la clase Action de Struts.

ActionServlet

Es el eje de Struts. Se encarga a partir de una petición HTTP de generar los ActionForm y de pasárselos a una acción concreta. Posteriormente trasfiere el control al JSP determinado. Se apoya en los ficheros web.xml y struts-config.xml.

ClaseManager

Es la clase encargada de suministrar la información a las Acciones. Contienen todos los métodos necesarios para tratar la información. Esta clase se encarga de hacer interactuar las clases del dominio, con los ActionForm.

Para obtener la información necesaria, se apoyan en las clasesDAO.

La claseManager implementa la interfazManager.

InterfazManager

Interfaz que posee todos los métodos necesarios para el modulo.

ClasesDAO

Son las clases encargadas de acceder a la base de datos. Se ayudan de hibernate para acceder a la información y la retornan en forma de clases del dominio. Cada clase implementa una interfazDAO propia.

Clases del Dominio

Son las clases propias de la aplicación que siguen una estructura similar a la base de datos en cuanto a atributos se refiere.

Hibernate

Se encarga de ejecutar las sentencias SQL y de crear, modificar o eliminar los objetos de la aplicación (Objetos del Dominio). Para ello se utiliza un XML propio, que existe para cada clase de dominio en el sistema

XML del Dominio

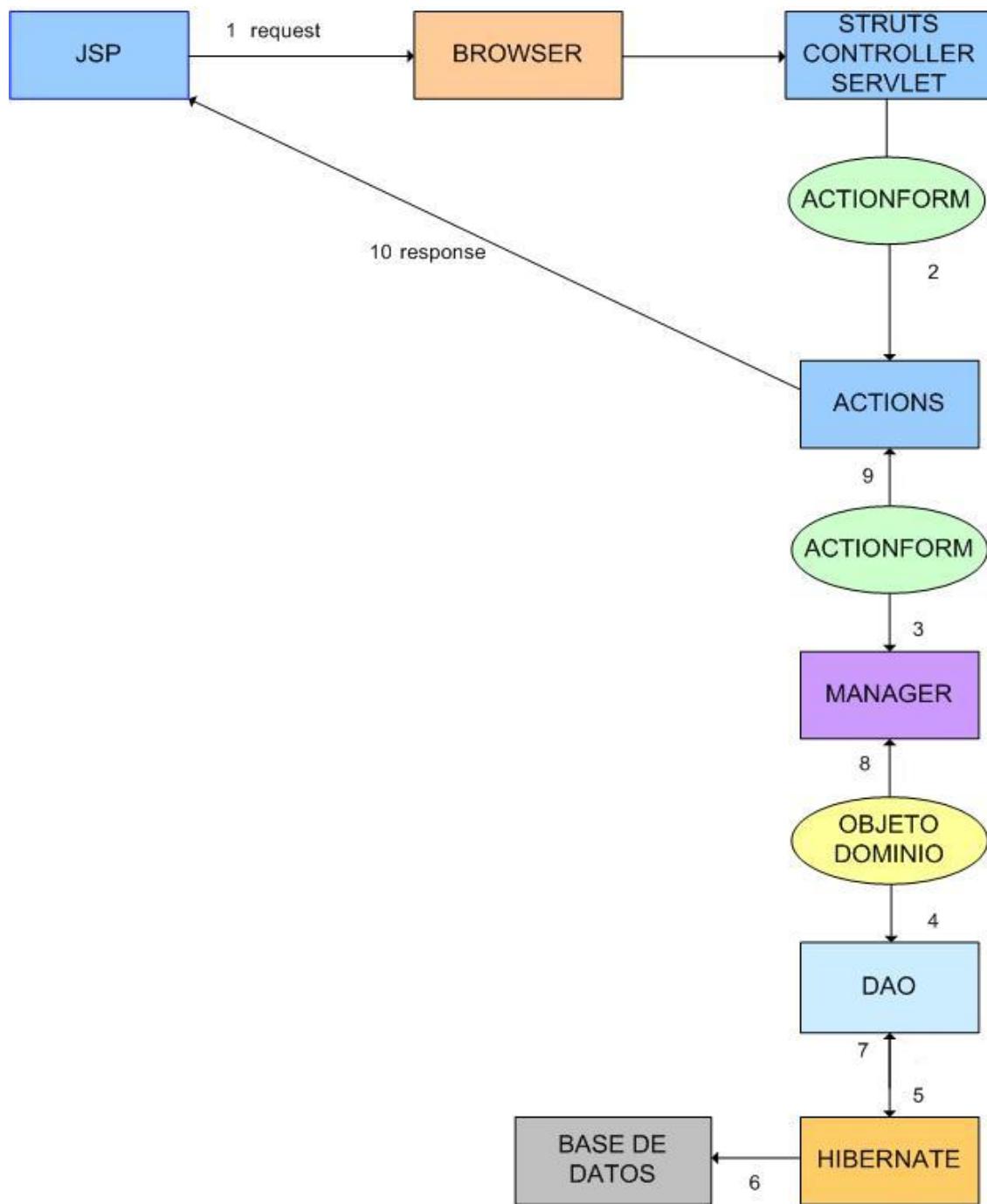
Se encarga de relacionar las clases del dominio con las tablas de la base de datos.

INTERACCION ENTRE LOS ELEMENTOS

A continuación recogeremos el proceso general que sigue nuestra aplicación cuando se realiza una petición Web por parte de un usuario.

1. El usuario realiza una petición a través de un JSP de nuestra aplicación.
2. Se genera una acción encargada de tratar la petición y un actionForm, que se encontrara vacío o completo dependiendo de si se trata de una petición para enviar o consultar información. En caso de que posea información, esta será validada por el ActionForm. Si se produce un error en la validación, se redirecciona al JSP de inicio y se indica el error.
3. Si la superclase acción valida correctamente la sesión del usuario, la subclase realizará una petición a la claseManager, en caso contrario, será redireccionado hacia un JSP indicando el error.
4. La clase manager solicitará o facilitará la información a la claseDAO.
5. La claseDAO elaborará la sentencia SQL necesaria y llamará a los procedimientos necesarios de hibernate.

6. Hibernate ejecutara la sentencia SQL y generara un objeto del dominio si por el tipo de consulta fuese necesario
7. Hibernate devolverá la información a la clase DAO.
8. La clase Dao devolverá el objeto del dominio a la clase Manager.
9. La clase Manager utilizara la información del Objeto del dominio para cumplimentar el ActionForm y realizará los procedimientos necesarios. Finalmente, devolverá la información a la acción mediante el ActionForm.
10. La acción dependiendo del resultado obtenido redireccionará la aplicación a un JSP concreto.



Esquema de la arquitectura del sistema

Este es el esquema general que realiza nuestra aplicación, pero existen pequeñas variaciones dependiendo el tipo de consulta. Cabe resaltar dos:

- a) Peticiones sin Actionform: Son peticiones que no requieren un Actionform ni para ejecutar la petición, ni después de haberla ejecutado. Siguen el esquema anterior pero no se generan los ActionForm.
- b) Peticiones que realizan nuevas peticiones: Siguen la estructura del esquema anterior, salvo en el punto 9, ya que dependiendo del resultado, pueden ser redirecccionados a una nueva acción.

10 Implementación del sistema

En este apartado se mostrarán las distintas elecciones que hemos tenido que realizar para llevar a cabo la implementación.

10.1 Entorno de desarrollo

La definición de un entorno de desarrollo influye decisivamente en el desarrollo del proyecto, un mal entorno puede disminuir drásticamente la productividad y provocar muchos quebraderos de cabeza. Por ello hemos probado muchas configuraciones y nos hemos tomado tiempo en decidirlo, además hemos tenido que tener en cuenta que las herramientas de desarrollo fueran gratuitas.

TIPO	DETALLE	VERSIÓN
Entorno de desarrollo	Eclipse Lomboz	3.2
Plugins Eclipse	Subversion	1.4.3
	Amateras StrutsIDE	2.0.2
	Amateras Eclipse HTML Editor	2.0.2
	Hibernate Tools	3.1
	MySQL GUI Tools	1.1.20
Administración BD	Internet Explorer	6.0
	Internet Explorer	7.0
	Mozilla Firefox	2.0.0.1
Gestor de Base de Datos	MySQL Server	5.0
Contenedor Web	Apache Tomcat Server	5.5

Especificación de las herramientas del entorno de desarrollo

A continuación expondremos las características principales de los ordenadores utilizados durante el desarrollo de la aplicación como orientación.

Procesador	Memoria RAM	Disco Duro
Intel Pentium Centrino M 1.8GHz	1 GB	75 GB
AMD Athlon XP 2600+	1GB	160 GB
Centrino Duo 2 1,66GHz	2GB	100 GB

Como observación hay que decir que el entorno utilizado requiere de un ordenador potente, ya que el entorno de programación Eclipse requiere de al menos 512Mb de memoria para funcionar con soltura.

10.1.1 Instalación del entorno

A continuación explicaremos brevemente como instalar el entorno Eclipse junto con los plugins importantes.

- **Instalación de Lomboz**

Entrar en la página web oficial de Lomboz y descargar la versión completa:
<http://lomboz.objectweb.org/downloads/download.php>

- **Instalación de los plugins de Amateras**

Entrar en la dirección <http://sourceforge.jp/projects/amateras/files/> y descargar las últimas versiones de *EclipseHTMLEditor* y *StrutsIDE*. Son archivos comprimidos en formato zip, descomprimir y copiar su contenido a la carpeta *plugins* de eclipse. Eclipse los cargará automáticamente al inicio.

- **Instalación Subclipse**

Instalar mediante el gestor de features¹ de Eclipse ingresando la dirección:
http://subclipse.tigris.org/update_1.0.x

10.1.2 Configuración del repositorio RedIris

- Modificar la vista de eclipse a vista SVN Repository Exploring.
- Luego introducir un nuevo repositorio mediante un clic con el botón derecho seleccionando la opción new Repository Location.
- Introducir la siguiente dirección perteneciente al repositorio del proyecto: <https://forja.rediris.es/svn/gestionalumnos>
- Una vez creado el repositorio, pulsar con el botón derecho del ratón y seleccionar la opción CheckOut para lo cual le pedirá una clave de usuario, introduzca su usuario y clave correspondiente.
- Luego le dará la opción de elegir el nombre que tendrá el proyecto en su workspace, seleccione la más conveniente. Pulsar siguiente y esperar a que termine la descarga del proyecto.

10.2 Componentes más relevantes

Como resumen del proceso de implementación se expondrán a continuación los componentes más relevantes y destacados que hemos tenido que resolver en el desarrollo de esta aplicación Web.

Implementación de funcionalidad de la capa de datos

La clase que lleva a cabo el manejo de las conexiones es *HibernatePool*, se encarga también de cargar una sesión con *Hibernate* usando los parámetros de configuración establecidos en el fichero de configuración *hibernate.cfg.xml*, de ahí recoge el nombre de usuario, su password y la url de la base de datos.

¹ Para mas información sobre como instalarlo entrar en la siguiente dirección:
<http://subclipse.tigris.org/install.html>

```

public class HibernatePool {

    private static SessionFactory session;
    private final Log log = LogFactory.getLog(HibernatePool.class);
    public static HibernatePool instance;

    private HibernatePool() throws ErrorHibernateException{
        try
        {
            /*Se crea una configuracion basada en el xml*/
            Configuration cfg=new Configuration().configure();
            /*se crea una sesion con la configuracion*/
            session=cfg.buildSessionFactory();
            /*Se muestra un mensaje de informacion*/
            log.info("Se ha cargado hibernate satisfactoriamente");
        }
        catch(Exception he)
        {
            /*Se muestra un mensaje de error*/
            log.debug("No se ha podido cargar hibernate");
            /*Se lanza una exception del tipo carga*/
            throw new
                ErrorHibernateException(he,ErrorHibernateException.LOAD);
        }
    }

    public static synchronized HibernatePool getInstance() throws
ErrorHibernateException{
        if(instance==null)
            instance=new HibernatePool();
        return instance;
    }

    /**
     * Metodo que devuelve la sesion
     * @return sesion de Hibernate
     */
    public Session getSession(){
        return session.openSession();
    }
}

```

Como se puede observar utiliza el patrón Singleton, para que la carga de la base de datos sólo se realice una vez, ya que es de un gran costo. Además cabe destacar el método getSession que es el encargado de devolver una conexión donde realizar transacciones.

El acceso a la información de la base de datos se realiza mediante los DAOs, lo mostraré mediante código con la clase NoticiaDAO que recoge las funciones más representativas, es decir, un select, un delete, un saveOrUpdate y una consulta en SQL.

```

public class NoticiaDAO implements INoticiaDAO {

    private final Log log = LogFactory.getLog(NoticiaDAO.class);
    private Session s;

    public Noticia select(int idNoticia) throws ErrorHibernateException
    {

```

```

s=HibernatePool.getInstance().getSession();
/*Valor por defecto en la noticia*/
Noticia n=null;
try
{
    /*Devuelve la Noticia dado el id*/
    n=(Noticia)s.get(Noticia.class,idNoticia);
    /*Cierre de conexion*/
    s.close();
}
catch(HibernateException he)
{
    /*Cierre de conexion*/
    s.close();
    throw new
        ErrorHibernateException(he,ErrorHibernateException.DATA);
}
return n;
}

public void delete(int idNoticia) throws ErrorHibernateException
{
    /*Se recoge la sesion*/
    s=HibernatePool.getInstance().getSession();
    /*Comienza la transacción*/
    Transaction tx = s.beginTransaction();
    try
    {
        s.delete(new Noticia(idNoticia));
        /*Se realiza la transaccion*/
        tx.commit();
        /*Se cierra la conexion*/
        s.close();
    }
    catch(HibernateException he)
    {
        /*Como ha ocurrido un error se vuelve atras*/
        tx.rollback();
        /*Se cierra la conexion*/
        s.close();
        log.debug("Error endelete de noticia con id:"
                  +idNoticia);
        throw new
            ErrorHibernateException(he,ErrorHibernateException.DATA);
    }
}

public void saveOrUpdate(Noticia noticia) throws ErrorHibernateException
{
    s=HibernatePool.getInstance().getSession();
    Transaction tx = s.beginTransaction();
    try
    {
        /*Salva o actualiza la noticia*/
        s.saveOrUpdate(noticia);
        /*Se realiza la transacción*/
        tx.commit();
        /*Se cierra la conexion*/
        s.close();
    }
    catch(HibernateException he)
    {
        /*Como ha ocurrido un error se vuelve atras*/
        tx.rollback();
        /*Se cierra la conexion*/
        s.close();
    }
}

```

```

        log.debug("Error en saveOrUpdate de noticia con id:"
                  +noticia.getIdNoticia());
        throw new
        ErrorHibernateException(he,ErrorHibernateException.DATA);
    }
}

public List<Noticia> noticiasByAsignatura(int id) throws ErrorHibernateException
{
    s=HibernatePool.getInstance().getSession();
    List<Noticia> list=new ArrayList<Noticia>();
    try
    {
        /*Consulta en hql*/
        String hql="from Noticia N " +
                   "Where N.asignatura.idAsignatura=:idAsignatura " +
                   "order by fechaMod desc";
        Query q=s.createQuery(hql);
        /*Establece el parametro*/
        q.setInteger("idAsignatura",id);
        /*Devuelve la lista de noticias*/
        list=q.list();
        /*Cierra la conexión*/
        s.close();
    }
    catch(HibernateException he)
    {
        s.close();
        log.debug("Error en saveOrUpdate de noticia con id:"
                  +id);
        throw new
        ErrorHibernateException(he,ErrorHibernateException.DATA);
    }
    return list;
}

```

Como se puede observar en el código, para realizar una modificación en la base de datos se recurre a las transacciones para mantener la concurrencia. Además es como mejor se observa la reducción importante de código debida al uso de Hibernate, se pueden realizar acciones de DAOs con pocas líneas de código. También hemos incluido en los DAOs un logger para facilitar el mantenimiento.

Implementación de las clases de la lógica de la aplicación

Las clases que implementan la lógica de la aplicación se encuentran en cada módulo, estas implementan el patrón Fachada. Tienen la responsabilidad de pasar de ActionForm que son manejados en la vista, a clases del dominio que son usadas por Hibernate mediante los DAOs para insertar o modificar información de la base de datos. También en el otro sentido para mostrar datos referentes de la BD. Estas clases son llamadas desde los actions. Un ejemplo claro y simple para ver la implementación de la lógica es el siguiente:

Ejemplo del modulo InformacionUsuarios que modifica la información de un profesor con los datos pasados por un ActionForm.

```

/**
 * Modifica los datos del profesor con los introducidos en el form
 *
 * @param idProfesor identificador del profesor

```

```

* @param profesor actionForm con los parametros del formulario
* @throws ErrorHibernateException
*/
public void modificarPerfilProfesor(int idProfesor, ProfesorForm
profesor) throws ErrorHibernateException
{
    IProfesorDAO profesorDAO=new ProfesorDAO();
    Profesor p=profesorDAO.select(idProfesor);
    profesor.changeProfesor(p);
    profesorDAO.update(p);
}

```

Ejemplo de la carga en un ActionForm de información, en este caso referente a la información de una asignatura.

```

/**
 * Función que carga en el form la información de las asignatura
*/
public void mostrarAsignatura(int idAsignatura, InformacionAsignaturaForm form)
throws ErrorHibernateException
{
    IAsignaturaDAO asignaturaDAO=new AsignaturaDAO();
    INoticiaDAO noticiaDAO=new NoticiaDAO();
    /*Datos de la asignatura*/
    form.setAsignatura(asignaturaDAO.select(idAsignatura));
    /*Profesores que imparten la asignatura*/
    form.setProfesores(asignaturaDAO.selectProfesores(idAsignatura));
    /*Noticias que tiene la asignatura*/
    form.setNoticias(noticiaDAO.noticiasByAsignatura(idAsignatura));
}

```

Implementación de la capa de presentación

El diseño de la página se ha separado en varios componentes modularizando para favorecer el mantenimiento y disminuir el acoplamiento. Hemos separado la pagina en 4 partes principales:

Menu.jsp:

Contiene los enlaces del menú que muestran las distintas opciones posibles según el tipo de usuario. Se puede observar el uso de las taglibraries para comprobar el objeto introducido en la sesión UsuarioForm, del que se le compara el atributo rol, mediante la etiqueta logia>equals, y dependiendo del cual sea el valor, se carga unas opciones u otras.

```

<logic:equal scope="session" name="UsuarioForm" property="rol" value="ALUMNO">
    <li class="titulo" ><bean:message key="menu.ficha" /></li>
    <ul>
        <li><html:link action="verFicha">
            <bean:message key="menu.verFicha" />
        </html:link></li>

        <li><html:link action="verFicha?r=mod">
            <bean:message key="menu.editarFicha" />
        </html:link></li>
    </ul>
    ...

```

Cabecera.jsp:

Como su nombre indica contiene los elementos que son mostrados en la cabecera, en este caso tiene algo de funcionalidad ya que muestra distintas capas según el tipo de usuario identificado por el sistema.

```
<logic:equal scope="session" name="UsuarioForm" property="rol" value="ALUMNO">
<div id="cabeceraAl"></div>
</logic:equal>

<logic:equal scope="session" name="UsuarioForm" property="rol" value="PROFESOR">
    <div id="cabeceraPr"></div>
</logic:equal>

<logic:equal scope="session" name="UsuarioForm" property="rol" value="ADMINISTRADOR">
    <div id="cabeceraAd"></div>
</logic:equal>
```

Pie.jsp:

Pie de la página que muestra la información de la aplicación.

```
<div id="pie">
<p>Versi&oacute;n 1.0 Alpha - Noel Guillen Montilla, Fernando L&oacute;pez Iglesias</p>
</div>
```

Head.jsp:

Es la parte del head del html, como es una parte común se ha modularizado.

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Gesti&oacute;n de Fichas</title>
<link rel="stylesheet" type="text/css" href="estilos.css" title="default" />
</head>
```

Para las vistas hemos usado una página por defecto donde hemos introducido el contenido.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<%@ include file="../include/head.jsp"%>

<body>
    <div id="cuerpo">
        <%@ include file="../include/cabecera.jsp"%>
        <%@ include file="../include/menu.jsp"%>
        <div id="contenido">

            CONTENIDO DE LA PÁGINA

        </div>
        <%@ include file="../include/pie.jsp"%>
    </div>
</body>
</html>
```

De esta forma hemos simplificado la creación de las vistas.

Implementación de la seguridad de acceso de los action

Dado que Struts no proveía por defecto de una solución para la seguridad, impidiendo el acceso de un usuario no identificado a la aplicación, decidimos implementarla de un modo simple pero a la vez valido, extendiendo la clase Action de Struts mediante una clase abstracta llamada Accion, usando el patrón Plantilla. A continuación muestro el código para un mejor entendimiento:

```

public abstract class Accion extends Action {

    private UsuarioForm usuario;

    public ActionForward execute(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response) throws Exception {
        checkSession(request);
        return ejecutar(mapping,form,request,response);
    }

    /**
     * Método de enganche
     */
    public abstract ActionForward ejecutar(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response) throws Exception;

    /**
     * Chequea que el usuario tiene sesión y si no lo manda a una página de error
     */
    private void checkSession(HttpServletRequest request) throws
        AccesoNoPermitidoException{
        HttpSession session=request.getSession();
        /*Inicializa el atributo usuario*/
        this.usuario=(UsuarioForm)session.getAttribute("UsuarioForm");
        if(usuario==null)//Si no esta logueado
        {
            /*Lanza la excepción que es capturada por struts*/
            throw new AccesoNoPermitidoException();
        }
    }
}

```

Como se puede observar se comprueba que existe un usuario en la sesión y si no es así, se lanza una excepción que será capturada por struts para mostrar una pantalla de aviso. Todos los actions que usamos heredan de esta clase, por tanto cada vez que se muestra una página se comprueba si el usuario tiene sesión.

Uso de AJAX en la aplicación

En la aplicación hemos usado las AjaxTags, que nos proveen de unas taglibraries para la vista y unas clases que incluso se integran con struts en el lado del servidor.

Código referente al uso de AjaxTags en la vista nueva petición para el problema tipico de inputs del tipo select anidados.

```

...
<ajax:select
    baseUrl="${pageContext.request.contextPath}/CargarProfesores.do"
    parameters="asignatura={asignatura}"
```

```

        source="asignatura"
        target="profesor">
</ajax:select>
..

```

Este código es insertado después del form y genera el código Javascript necesario para el uso de AJAX en la parte del cliente.

Por otro lado en el lado del servidor:

```

public class CargarProfesores extends BaseAjaxAction {

    public String getXmlContent(ActionMapping mapping, ActionForm
form,HttpServletRequest request, HttpServletResponse response) throws
Exception {
        IPeticionCitaManager manager=new PeticionCitaManager();
        int idAsignatura =Integer.parseInt(request.getParameter("asignatura"));
        List<Profesor> profesores=manager.cargarProfesores(idAsignatura);
        return new
AjaxXmlBuilder().addItems(profesores, "nombre", "idUsuario").toString();
    }
}

```

Se puede ver que la clase hereda de BaseAjaxAction que implementa la funcionalidad necesaria que a su vez hereda de la clase de struts Actions, también se observa el uso de AjaxXmlBuilder que dada una lista genera el xml.

También hemos usado AJAX para mostrar las citas que tiene un profesor en forma de calendario sin tener que recargar la página. En este caso no se han podido usar las AjaxTags ya que es demasiado específico, así que el código javascript referente a la parte del cliente lo hemos tenido que crear nosotros.

Internacionalización

Aunque la internacionalización la implementa Struts explicamos brevemente su uso en la aplicación. Tenemos un archivo Menssages_es.properties dónde hay conjunto de claves y cada una tiene asociada un texto.

```

...
---Grupos del menu---
menu.tutorias=Tutorías
menu.citas=Citas
menu.ficha=Ficha
menu.fichas=Fichas
menu.usuario=Datos de usuario
menu.asignaturas=Asignaturas
...

```

En la vista sólo para usarlo solo hay que usar la taglib bean con su función message como se muestra a continuación.

```

...
<bean:message key="menu.verFicha" />
...

```

Como se ve, su uso es simple, además es eficiente ya que el fichero de mensajes se carga en caché al inicializar la aplicación. Struts escoge el idioma a usar dependiendo del navegador del usuario, y si no existe escoge el definido por defecto.

Validación de datos

La validación de datos se realiza en el lado del servidor en los ActionForms mediante el método validate que provee Struts para que se implemente. Un ejemplo de una validación en el caso de la clase NoticiaForm, que es el formulario usado para introducir una nueva noticia, es el siguiente:

```
public ActionErrors validate(ActionMapping mapping, HttpServletRequest
request)
{
    ActionErrors errors=new ActionErrors();

    /*Si no se ha introducido ningún texto*/
    if(texto==null||texto.equals("")){
        errors.add("noticia",new ActionMessage("error.noticia.texto"));
    }

    /*Si no se ha introducido ningún título*/
    if(titulo==null||titulo.equals("")){
        errors.add("noticia",new ActionMessage("error.noticia.titulo"));
    }

    /*Si no se ha introducido ninguna asignatura*/
    if(asignatura.equals("-1")){
        errors.add("noticia",new ActionMessage("error.noticia.asignatura"));
    }

    return errors;
}
```

En la vista esta sólo hay que introducir lo siguiente:

```
...
<html:errors/>
...
```

Se puede ver que se usan claves para la internacionalización de los mensajes de error, y que su complejidad no es alta ya que Struts simplifica mucho el trabajo.

11 Pruebas del sistema

La fase de pruebas del software es una de las partes más importantes del proceso de desarrollo de software. Probar una aplicación involucra la generación de casos de pruebas, la ejecución de la aplicación contra estos casos de prueba y la observación del comportamiento de la aplicación para determinar su corrección.

11.1 Pruebas de Interfaces y Contenidos

Las actividades de esta etapa consisten en hacer revisiones precisas de la forma en que se despliegan las páginas del sitio y ver si cumplen con los *Términos de Referencia* en estos temas y, además, si cumplen con los estándares mínimos que se hayan definido como meta a ser cumplida.

Las acciones de prueba sugeridas para realizar en esta etapa son las siguientes:

Verificación de Contenidos:

Es una prueba básica para revisar si el Sitio Web desarrollado incluye todos los contenidos que se han especificado en los *Términos de Referencia* o los que se hayan definido en el marco del plan de desarrollo. Se puede hacer en forma manual o automática, de acuerdo a las siguientes orientaciones:

Sistema Manual:

Se refiere a hacer una revisión manual de los contenidos del Sitio Web a través de la navegación de sus páginas. Los elementos que deben probarse obligatoriamente son: ortografía y redacción, enlaces principales, imágenes en páginas.

Sistema Automático:

Especialmente orientado a la verificación de enlaces rotos, lo cual se puede hacer utilizando sistemas basados en Internet o, bien, software especializado.

Sitio en Construcción:

Se debe verificar que el Sitio Web no contenga espacios vacíos o que tenga el título de *en construcción*. No es adecuado, bajo ningún sentido, usar espacios con dicha leyenda; en tal caso es preferible eliminar esa zona y volver a incluirla cuando exista el contenido correspondiente en el sitio.

Verificación de Estándares:

Aunque los sitios Web pueden ser construidos a partir de diferentes lenguajes, todos deben cumplir ciertas normas de organización de su código fuente (sintaxis), que permitan su visualización por software

equivalente en diferentes plataformas. Dicha sintaxis está estandarizada y puede ser probada a través de herramientas públicas que están disponibles en Internet. Las dos más importantes son:

Validación de HTML:

la realiza el World Wide Web Consortium² e indica si el código usado en la página es correcto. Como resultado entrega un reporte con los eventuales errores para ayudar a su reparación.

Validación de CSS:

La realiza el W3C³ e indica si la Hoja de Cascada de Estilo (Cascading Style Sheet) cumple con la sintaxis estándar y por lo tanto podrá ser visualizada correctamente en todos los sistemas que cumplan dicho estándar.

Verificaciones de Interfaces:

Mediante esta prueba se revisan aspectos gráficos del Sitio Web, para determinar si su despliegue en las páginas es correcto. Dentro de los elementos más importantes a ser verificados, se incluyen los siguientes:

Plug-ins necesarios:

Cuando se utilicen elementos audiovisuales o interactivos que requieran de algún software incrustado para funcionar (plug-ins), se debe ofrecer un enlace para que los usuarios que no lo tengan instalado, puedan bajarlo y hacer el proceso de instalación.

Consistencia de la Diagramación:

Cada una de las páginas del sitio debe tener elementos consistentes, con el fin de ofrecer al usuario una experiencia similar en cualquier área del Sitio Web; por nombrar sólo tres aspectos, lo anterior implica que los menús deben aparecer siempre en el mismo lugar; que los listados deben estar diseñados de similar manera en todo el sitio y que los colores y formas de uso de las interfaces deben ser similares a lo largo de las páginas.

Ancho de la Diagramación:

Si la Web se ha diseñado para un ancho determinado (por ejemplo, 800 píxeles de ancho), en esta etapa se debe probar si ello se cumple. Asimismo, se debe probar en una pantalla configurada con una menor resolución (por ejemplo 640 x 480 píxeles), cuál es el área visible del sitio y cómo afecta eso a la navegación por el mismo.

Diagramación vs. Browsers:

² <http://validator.w3.org>

³ <http://jigsaw.w3.org/cssvalidator>

Aunque la codificación en los lenguajes soportados por los programas visualizadores puede apegarse a los estándares, no todos muestran de la misma manera los sitios Web. Las pruebas deberían realizarse al menos en los navegadores Microsoft Internet Explorer, Opera y Mozilla, ya que con ellos se cubrirá un amplio espectro. Lo que se debe revisar en este caso es el despliegue de todos los elementos que se muestran en la pantalla, para asegurar de que aparecen en las posiciones que se les han asignado en el diseño.

Imágenes Escaladas:

Se debe verificar que las imágenes que aparezcan en el sitio no estén siendo mostradas en tamaño reducido artificialmente; es decir, que se tome una imagen de grandes dimensiones y por programación se muestre en un tamaño menor. El efecto de eso es que las páginas con ese tipo de imágenes serán muy pesadas y harán que el acceso a ellas sea lento.

Imágenes Sin Atributo ALT:

Para cumplir con las normas de accesibilidad es necesario que todas las imágenes que se usen en un Sitio Web, tengan una descripción utilizando el atributo ALT (para texto alterno) del lenguaje HTML.

En relación con las siguientes pruebas se ha realizado el siguiente informe

Prueba	Resultados
Verificación de contenidos	Se han verificado de forma manual
Sitio en construcción	No existen páginas en blanco
Validación HTML	No es posible pasarla debido al código generado en los input por las etiquetas de struts.
Validación CSS	Debido al uso de hacks en la hoja de estilos para la correcta visualización en las versiones de Explorer y mozilla no ha cumplido el estándar de la W3C
Plug-ins necesarios	No se requieren plug-ins para visualizar la página
Consistencia de la diagramación	La página Web tiene un diseño similar en todas las acciones.
Ancho de diagramación	Al reducir la resolución a 800x600, se visualiza correctamente, aunque hay que usar las barras de desplazamiento.
Diagramación vs. Browsers	Las pruebas con los navegadores Internet Explorer (versión 6 y 7), Mozilla Firefox y Opera han sido resueltas satisfactoriamente, solo hay leves diferencias de visualización entre estos navegadores.
Imágenes escaladas	Todas las imágenes de la Web han sido escaladas y en ninguna página se superan los 100KB, e incluso en la mayoría no se superan los 50KB, esto es así debido a que

Imagenes sin atributo alt	no usamos gran cantidad de imágenes. Todas las imágenes tienen el atributo alt
----------------------------------	---

Informe de las pruebas de interfaces y contenidos

10.2 Pruebas de Funcionalidades y Operación

Las actividades de este apartado se refieren a hacer chequeos completos respecto de las funcionalidades y aplicaciones que ofrece el sitio, ya sean de aplicaciones simples como formularios hasta más complejas, como consultas y modificaciones de registros en base de datos.

En este sentido, las pruebas se deben hacer sobre diferentes elementos, siendo algunos de los más importantes los siguientes:

Validación de Formularios:

Si el Sitio Web tiene formularios para el envío o ingreso de datos, se debe utilizar sistemas de validación del ingreso de datos para asegurar que éstos sean bien ingresados. En este aspecto, algunas de las validaciones más importantes deben ser las siguientes:

Campos Obligatorios:

Se debe validar que en los formularios sean ingresados todos aquellos campos que sean necesarios; éstos deben ser marcados de alguna manera (usualmente con un asterisco) que permita a los usuarios entender la obligatoriedad de ingresar información en ellos; adicionalmente, debe indicarse tal condición en forma explícita.

Validaciones Locales:

Para reducir la carga de validaciones en el servidor, se recomienda incorporar la mayor cantidad de éstas en el computador del cliente, utilizando en forma estándar el lenguaje Javascript para hacerlas.

Sintaxis de Ingreso:

Se debe validar que, en algunos casos, los campos sean ingresados con datos válidos; el mejor ejemplo es el caso del ingreso del número de RUT, cuyos números tienen un algoritmo conocido para ser validado.

Ingreso de Datos:

Si se cuenta con un sistema que permite el ingreso de información hacia una base de datos, se debe revisar en la tabla de destino que efectivamente se estén enviando los datos de la manera que se ha previsto.

Reingreso y Corrección de Datos:

Para mejorar la interacción del Sitio Web, cuando tras el ingreso y envío de los datos de un formulario (después de la validación local del formulario) el usuario presiona el botón Back de su programa visualizador para volver atrás y modificar algún campo, se le deben presentar todos los datos que hayan sido ingresados. De esta manera se aprovecha la información ingresada previamente, evitando la

frustración del usuario por tener que escribir nuevamente el contenido completo del formulario.

Elementos de Interfaz:

Al usar elementos del lenguaje HTML para la creación de las pantallas, se recomienda no modificar radicalmente sus atributos de despliegue (colores, formas) y comportamientos tradicionales, para lograr que el usuario sepa intuitivamente cómo usarlo y no deba aprender de nuevo su operación.

Multiplataforma:

Se debe comprobar que los formularios funcionan en diferentes versiones de programas visualizadores, de sistemas operativos y de tipos de conexión a Internet.

Administración del Error 404:

Cuando se ingresa una dirección equivocada, el software del servidor Web muestra una pantalla de error anunciando el número de código del problema (Error 404). No obstante, dicho software puede ser configurado para que muestre una página diferente, en la que se explique a los usuarios las probables razones del error. Es importante incluir, en dicha página, un enlace al Mapa del Sitio y un Buscador, de tal manera que el usuario tenga más herramientas para resolver la inexistencia del contenido que buscaba. Se recomienda, además, que el Administrador de Sistemas de la institución entregue un reporte semanal basado en los logs del servidor, que permita ver qué es lo que más buscan los usuarios y de qué manera el Sitio Web les está respondiendo sus consultas.

Pruebas	Resultados
Campos obligatorios	La validación se ha realizado sólo en el lado del servidor
Validaciones locales	Las validaciones se han realizado en el servidor
Ingresos de datos	Las comprobaciones sobre el ingreso de datos se han realizado satisfactoriamente
Reingreso y corrección de datos	Ante un error de validación se vuelve a la página donde está el formulario junto con los datos ingresados.
Elementos de interfaz	Los botones y formularios tienen un estilo estándar.
Multiplataforma	La aplicación Web ha pasado satisfactoriamente las pruebas con los tres navegadores, Internet Explorer(versión 6 y 7), Mozilla Firefox y Opera
Administrador del error 404	La aplicación captura los errores y muestra una página Web que indica el tipo de error junto con un enlace a la página principal. Además se puede configurar que se guarden los errores en un log

Informe de las pruebas de funcionalidad y operación

10.3 Pruebas de carga

La carga de trabajo se refiere a la capacidad máxima que tiene un servidor Web, para atender a un conjunto de usuarios de manera simultánea. Por ello, las actividades de esta etapa tienen relación con comprobar, de manera anticipada, el funcionamiento que tendrá el servidor del Sitio Web cuando esté en plena operación.

Las pruebas en este caso consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el software instalado cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware es capaz de soportar la cantidad de visitas esperadas.

Programa usado

Para realizar la medición usaremos el JMeter un programa que permite realizar pruebas de rendimiento y pruebas funcionales.

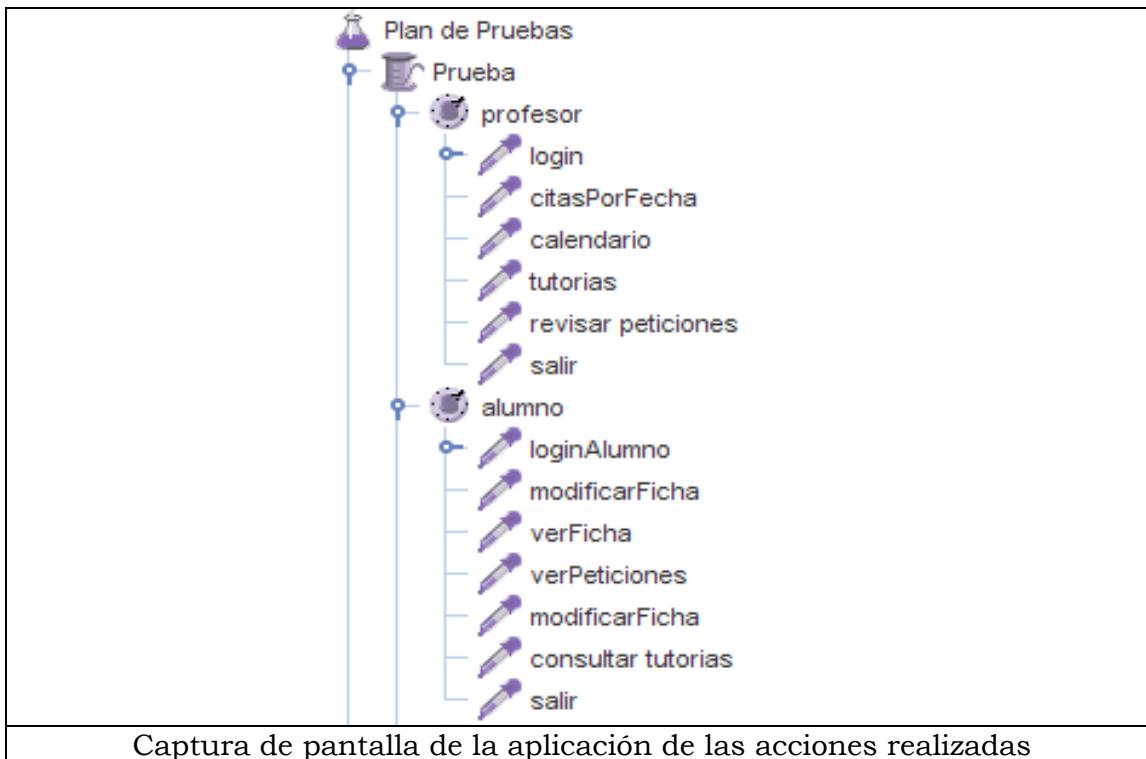
Inicialmente diseñada para aplicaciones web, hoy en día, su arquitectura ha evolucionado no sólo para llevar a cabo pruebas en componentes habilitados en Internet (HTTP), sino además en Bases de Datos, programas en Perl, FTP y prácticamente cualquier otro medio.

Además, posee la capacidad de realizar desde una solicitud sencilla hasta secuencias de peticiones que permiten diagnosticar el comportamiento de una aplicación en condiciones de producción.

En este sentido, simula todas las funcionalidades de un Navegador, o de cualquier otro cliente, siendo capaz de manipular resultados en determinada requisición y reutilizarlos para ser empleados en una nueva secuencia.

Descripción de la prueba

Mediante el programa JMeter simularemos la entrada de usuarios de tipo profesor y alumno en la aplicación, mediante hilos de ejecución, que además cada uno realizará las acciones más usuales asociadas a su perfil.



Como se puede observar en la imagen anterior, por cada tipo de usuario se han realizado una serie de acciones comunes para simular el acceso de un usuario a la aplicación Web realizada.

Las pruebas se han realizado para 10 usuarios del tipo alumno con 7 acciones, 10 usuarios del tipo profesor con 6 acciones y 10 iteraciones, lo cual nos proporciona un total de 1300 pruebas para realizar las estadísticas. La base de datos tiene una cierta carga de datos, 3 titulaciones y 2 asignaturas adscritas, y un número de peticiones y noticias introducidas.

Ficha técnica del ordenador utilizado en las pruebas de servidor:

Procesador	AMD Athlon XP 2600+
Memoria RAM	1GB
Sistema Operativo	Windows XP SP2
Disco duro	160GB a 7200rpm

Consideraciones

Hay que tener en cuenta que la figura del administrador se ha excluido dado que comparado con el uso que podría tener cualquiera de los otros dos tipos de usuarios es inapreciable en el sistema.

Se ha realizado una prueba previa debido a que en la primera carga de un jsp, el contenedor Tomcat lo compila y tarda un tiempo mayor del que tardaría en una segunda carga. Esto lo hemos hecho así porque queríamos probarlo con el entorno en producción.

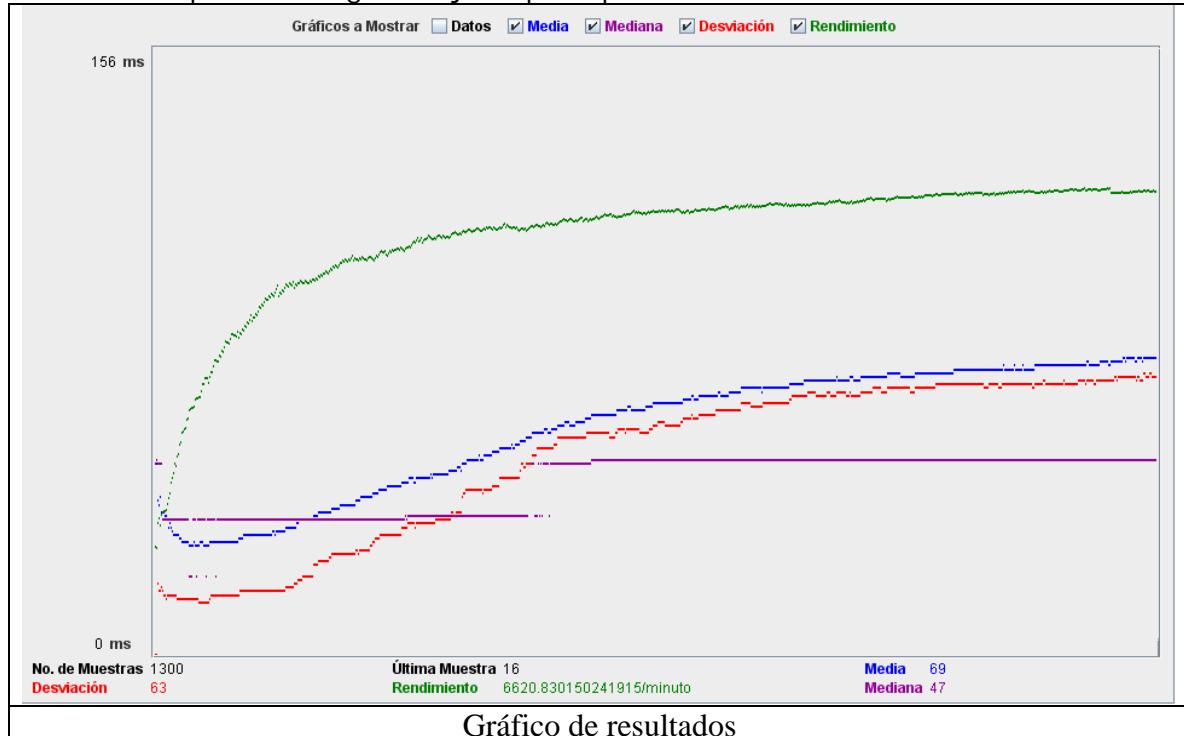
Resultados

Como se puede observar en la tabla la media de acceso total ha sido de 69ms, el percentil 90 de 141ms, con valores máximos de 437ms y mínimos de 15ms, valores bastante aceptables con tal número de muestras y usuarios. También hay que tener en otro atributo como el rendimiento que equivale al número de peticiones que puede servir en un segundo, en total ha sido de 110,3 peticiones por segundo.

Label	# Muestras	Media	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
login	100	89	94	141	15	203	0,00%	8,7/sec	19,95
citasPorFecha	100	68	47	125	15	344	0,00%	8,7/sec	22,01
calendario	100	21	16	32	15	78	0,00%	8,8/sec	21,91
tutorias	100	100	94	172	31	266	0,00%	8,7/sec	29,68
revisar peticion...	100	66	47	110	15	313	0,00%	8,8/sec	21,67
salir	200	20	16	32	15	63	0,00%	17,3/sec	21,11
loginAlumno	100	63	47	110	15	235	0,00%	8,8/sec	15,58
modificarFicha	200	66	62	125	15	266	0,00%	17,4/sec	82,25
verFicha	100	67	47	125	16	312	0,00%	8,8/sec	18,58
verPeticiones	100	68	47	110	15	437	0,00%	8,8/sec	18,68
consultar tutori...	100	177	187	313	15	406	0,00%	8,8/sec	18,08
TOTAL	1300	69	47	141	15	437	0,00%	110,3/sec	281,77

Tabla de resultados

JMeter también genera un grafico como el que se puede ver a continuación en el cual se puede ver el comportamiento del rendimiento, la media, mediana, y desviación típica a lo largo del eje X que representa al número de muestras.



Como se puede observar en él, a mayor número de muestras va tiendiendo más a lo que debería ser el comportamiento de la aplicación.

12 Instalación del sistema

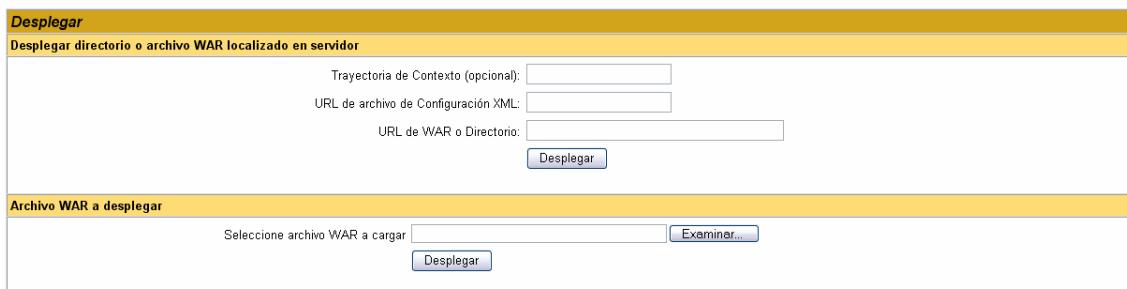
12.1 Requisitos

- Apache Tomcat versión 5.0 o superior (<http://tomcat.apache.org>) 
- MySQL Server 5.0⁴ (<http://www.mysql.com>) 

12.2 Pasos de la instalación

Despliegue de la aplicación

- Inicie el navegador
- Introduzca la siguiente dirección <http://127.0.0.1:8080/manager/html> y luego al entrar el nombre de usuario y la clave
- A continuación tiene le saldrá la pagina de administración de aplicaciones de Tomcat, vaya debajo de la página donde pone desplegar e inserte en examinar el .war de la aplicación, luego pulse desplegar y espere a que se recargue la página.



Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Desplegar

Archivo WAR a desplegar

Seleccione archivo WAR a cargar

Captura de pantalla de la zona desplegar

- Ya está la aplicación desplegada, para acceder a ella introducir la siguiente dirección <http://127.0.0.1:8080/Proyecto/> aunque el nombre de acceso se puede cambiar, si antes de desplegar renombramos el fichero en .war de la aplicación

Instalación de la Base de datos

- Abra el navegador e inserte la siguiente dirección : <http://127.0.0.1:8080/Proyecto/instalar.do?method=mostrar>
- A continuación se le mostrará una pantalla como se le muestra abajo, en la que se le pide que rellene datos referentes a la base de datos:

⁴ Versión con la cual ha sido desarrollada y probada, posiblemente funcione con versiones inferiores

- Dirección del servidor: Introduzca la dirección del servidor junto con el nombre de la base de datos que se usará (Ej : localhost/Test), ésta debe de existir⁵.
- Usuario y contraseña: usuario y contraseña con los permisos para modificar la base de datos.
- Tipo de BD: Aunque sólo se puede elegir MySQL se prevé que soporte mayor número de base de datos.
- Opciones de instalación: Se puede elegir que sólo se cree el esquema, en cual se crea sólo el usuario tipo Administrador con nombre de usuario admin y contraseña practica. También se puede elegir que se inserten los datos de prueba o conjuntamente

Instalación

Recuerde que es recomendable borrar el archivo de instalación despues de realizarla. También, tenga en cuenta que la dirección del servidor debe de incluir el nombre de la base de datos (Ej: localhost/gestion), y que ésta debe de estar disponible en la BD

Dirección del servidor	<input type="text"/>
usuario	<input type="text"/>
Contraseña	<input type="text"/>
Tipo de BD	MySQL <input type="button" value="▼"/>
Opciones de instalación	Esquema <input type="button" value="▼"/>
<input type="button" value="Aceptar"/>	

Captura de pantalla de la instalación

- Una vez que han introducido todas los campos deseados pulsar aceptar, si todo se ha realizado correctamente se mostrará un mensaje de información diciéndolo, si no se mostrará uno diciendo el tipo de error.

Eliminación del fichero de configuración

Como se puede leer en las recomendaciones de la pagina de instalación cuando la aplicación este en producción es recomendable que se elimine el archivo de instalación. Para ello vaya al directorio de tomcat, normalmente se encuentra en “archivos de programas/Apache Software Foundation/Tomcat”, entre posteriormente en el webapps que es el directorio donde se guardan las aplicaciones Web, luego en la que corresponde a la aplicación instalada, y borre el archivo *instalar.jsp*

⁵ Tenga en cuenta que en la instalación de MySQL se crea por defecto la base de datos llamada Test que se puede usar para probar el funcionamiento de la aplicación

13 Manual de uso del sistema

Este manual pretende ser una guía para el uso de la aplicación, aunque en su diseño se ha tenido en cuenta que sea lo mas intuitiva posible.

13.1 Manual de funcionamiento

La aplicación Web "Gestión de Fichas" soporta tres tipos de usuarios: alumno, profesor y administrador. Todos los usuarios accederán al portal mediante su página principal, facilitando su clave de identificación, que será su DNI, y su contraseña personal.



Observación: Si es la primera vez que se accede al portal, por norma general la contraseña personal coincidirá con su clave de identificación (DNI), salvo que se haya especificado lo contrario.

Si todos los datos son correctos, se accederá a la página de inicio.
La página de inicio será distinta de acorde al tipo de usuario que se trate, al igual que las acciones que se puedan realizar.
Todos los usuarios podrán realizar la acción: Modificar contraseña.

Modificar contraseña.

Para consultar su ficha personal basta con seleccionar la opción “Modificar clave” en el menú izquierdo de la aplicación. A continuación se introducirá la contraseña actual y la nueva contraseña (dos veces para evitar errores). Se finalizará el proceso mediante el botón “OK”.

Contraseña actual	<input type="text"/>
Nueva contraseña	<input type="text"/>
	<input type="text"/>
	<input type="text"/>
ok	

Nota: La contraseña, debe poseer al menos 6 caracteres.

Las distintas operaciones a realizar por los distintos tipos de usuarios se especificarán en los siguientes submanuales:

13.2 Manual de alumno

Las acciones que el usuario alumno podrá realizar son:

- Consultar su ficha personal.
- Modificar su ficha personal.
- Consultar el horario de tutoría de un profesor.
- Solicitar una cita.
- Consultar el estado de sus citas.
- Consultar su calendario de citas.
- Ver asignaturas.

13.2.1 Consultar su ficha personal

Para consultar su ficha personal basta con seleccionar la opción “ver ficha” en el menú izquierdo de la aplicación. A continuación, se mostrará su ficha personal con su foto en la parte superior derecha. En caso de que su ficha no exista o desee modificarla, ver “Modificar Ficha” (1.2).

13.2.2 Modificar su ficha personal

Para editar su ficha personal basta con seleccionar la opción “editar ficha” en el menú izquierdo de la aplicación. A continuación se indica la dirección de la foto personal y se modifican o se introducen los nuevos datos. Se finaliza el proceso mediante el botón “OK”.

Introducir foto

 Examinar...

Ficha personal

Nombre:	Fernando
Correo:	dreameam_3@hotmail.com
Dni:	11111111
Apellidos:	Guillén Montilla
FechaNacimiento:	2006-12-14
Teléfono:	954819432
Movil:	645645678

Direccion de residencia

Dirección:	calle santo cristo 106
Localización:	pedrera
CodigoPostal:	12121
Provincia:	Sevilla
país:	España

Direccion mientras estudia

Dirección:	avd. reina mercedes
Localización:	Sevilla
CodigoPostal:	11111
Provincia:	Sevilla
país:	España

Nota: Todos los campos son necesarios, excepto la fotografía personal. El único formato válido para la fotografía es el jpg.

13.2.3 Consultar el horario de tutoría de un profesor

Para consultar el horario de tutoría de un profesor basta con seleccionar la opción “Consultar tutorías” en el menú izquierdo de la aplicación. A continuación, se seleccionará la asignatura pertinente y uno de los profesores de dicha asignatura. Se mostrará el horario y el sistema mostrará la opción de

solicitar una cita (ver 13.2.4).

Elija una asignatura

- [ip1](#)
- [ip2](#)
- [Etc](#)
- [CI](#)

Elija el profesor del que desea ver las tutorías

- [David Benavides](#)

Datos personales

Nombre	David Benavides
Categoría	Dr

Datos de contacto

Teléfono	0
Despacho	
Web	

Tutorías

Lunes	Martes	Miércoles	Jueves	Viernes
	De 11:12 a 14:15			

Pedir cita

13.2.4 Solicitar una cita

Para solicitar una cita basta con seleccionar la opción “Pedir Cita” en el menú izquierdo de la aplicación. Se indicará la asignatura, el profesor, la fecha y el motivo de la cita. Se finalizara el proceso mediante el botón “OK”.

Nota: Todos los campos son obligatorios. No es posible realizar peticiones con más de dos meses de antelación.

Nueva petición

Asignatura	<input type="text" value="ip1"/>
Profesor	<input type="text" value="David Benavides"/>
Fecha	<input type="text" value="1 / 2 / 2007"/> (Formato:dd/MM/yyyy)
Hora	<input type="text" value="17 : 0"/>
Motivo	<input type="text" value="Dudas Tema 3"/>
<input type="button" value="ok"/>	

13.2.5 Consultar el estado de sus citas

Para consultar el estado de la cita basta con seleccionar la opción “Ver Peticiones” en el menú izquierdo de la aplicación. A continuación se mostrarán todas las peticiones organizadas por su estado actual.

EN PROCESO	ACEPTADAS	CANCELADAS
17:00 1/2/2007	08:00 1/2/2007	
Profesor: David Benavides Motivo: Dudas Tema 3	Profesor: David Benavides Motivo: popo	

13.2.6 Ver asignaturas

Para consultar las asignaturas que el alumno posee, basta con seleccionar la opción “Ver Asignaturas” en el menú izquierdo de la aplicación. Al seleccionar una asignatura concreta, aparecerá la información básica de la asignatura (Descripción, Profesores y Noticias).

Etc
Descripción
Web: www.lsi.ip1.vocm
Cuatrimestre: 2º
Tipo: troncal
Curso: 1º
Duración: cuatrimestral
Créditos: 7.5
Profesores
• David Benavides
Noticias

13.3 Manual del profesor

Las acciones que el usuario profesor podrá realizar son:

- Consultar su horario de tutorías.
- Modificar su horario de tutorías.
- Desactivar las tutorías.
- Consultar las fichas de los alumnos.
- Cancelar citas.
- Responder peticiones de citas.
- Consultar citas por fecha.
- Consultar su calendario de citas.
- Modificar su información personal.
- Introducir noticias.
- Modificar noticias.
- Consultar noticias.

13.3.1 Consultar su horario de tutorías.

Para consultar el horario de tutoría basta con seleccionar la opción “Ver tutorías” en el menú izquierdo de la aplicación.

Datos personales

Nombre	David Benavides
Categoría	Dr

Datos de contacto

Teléfono	0
Despacho	
Web	

Tutorías

Lunes	Martes	Miércoles	Jueves	Viernes
	De 11:12 a 14:15			

Pedir cita

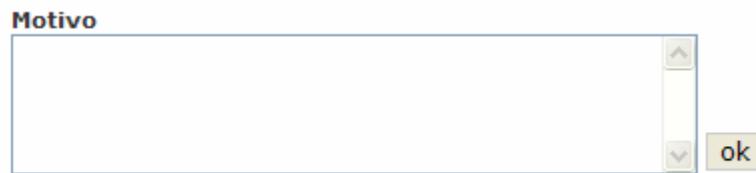
13.3.2 Modificar su horario de tutorías.

Para modificar el horario de tutoría basta con seleccionar la opción “Editar tutorías” en el menú izquierdo de la aplicación. A continuación se modificarán o se eliminarán los datos actuales de las tutorías. También, se podrán añadir nuevos horarios de tutorías mediante el botón “Insertar”;

Día:	Martes
Hora de inicio:	11 h 12 min
Hora de fin:	14 h 15 min
Eliminar Modificar	
Día:	Miércoles
Hora de inicio:	14 h 00 min
Hora de fin:	15 h 00 min
Eliminar Modificar	
Insertar	

13.3.3 Desactivar las tutorías

Para desactivar las tutorías, basta con seleccionar la opción “Desactivar tutorías” en el menú izquierdo de la aplicación. Se indicará el motivo por el cual se desactivan las tutorías, dicha explicación será visible para los alumnos. Se finalizará el proceso mediante el botón “OK”.



13.3.4 Consultar las fichas de los alumnos

Para consultar las fichas personales de los alumnos, basta con seleccionar la opción “Ver Fichas” en el menú izquierdo de la aplicación. Se seleccionara la asignatura y a continuación el alumno deseado, a partir de la lista de usuarios obtenida.

13.3.5 Cancelar citas

Para cancelar las citas ya establecidas con los alumnos, basta con seleccionar la opción “Ver Fichas” en el menú izquierdo de la aplicación. A continuación se indicará la fecha y las citas que se desean acordar, y se introducirá una breve justificación para el alumno

Nota: Las citas que se cancelan con este procedimiento son citas que fueron anteriormente aceptadas por el profesor. No se trata de peticiones de cita.

Citas:

Motivo:
Congreso

1/2/2007
Nombre: Fernando
Motivo:
popo

1/2/2007
Nombre: Fernando
Motivo:
Dudas Tema 3

OK

The screenshot shows a window titled "Citas:". It displays two rows of information for cancellation. Each row includes a "Motivo:" field (containing "Congreso" and "popo" respectively), a date field (showing "1/2/2007"), a "Nombre:" field (showing "Fernando" and "Fernando" respectively), and a "Motivo:" field (showing "Dudas Tema 3"). Below each row is a checkbox. At the bottom of the window is an "OK" button.

13.3.6 Responder peticiones de citas

Para responder las citas realizadas por el alumno, basta con seleccionar la opción “Peticiones de Cita” en el menú izquierdo de la aplicación. A continuación, se resolverán de forma individual las distintas peticiones mediante las opciones “Aceptar” y “Cancelar”.

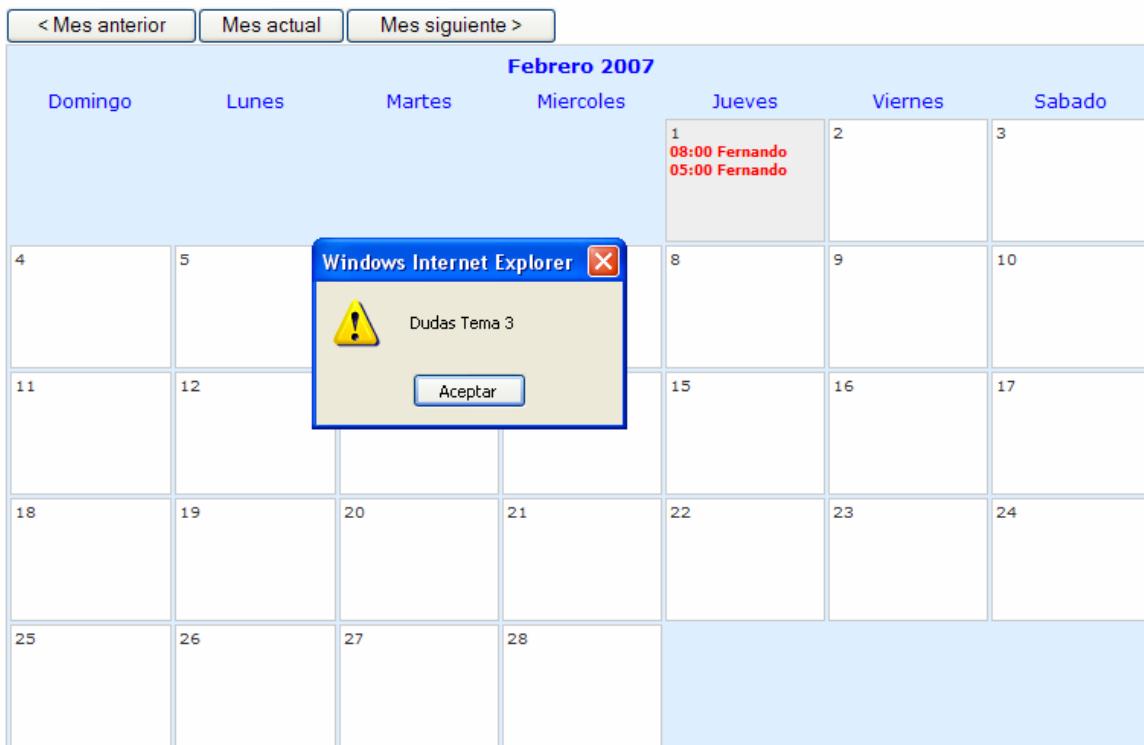
Peticiones de citas

- Profesor:David Benavides
- Fecha1/2/2007
- Hora05:00 PM
- MotivoDudas Tema 3
- Estado:EN PROCESO

[\[Aceptar\]](#) [\[Cancelar\]](#)

13.3.7 Consultar su calendario de citas.

Para consultar las citas acordadas con los alumnos, basta con seleccionar la opción “Calendario” en el menú izquierdo de la aplicación. A continuación se seleccionara el mes de la consulta mediante los botones “mes actual”, “mes anterior” y “mes siguiente”. Para consultar el motivo de una cita pendiente, basta con seleccionar la cita en concreto y la información aparecerá en una ventana emergente.



Nota: Las citas que aquí se muestran, son sólo las que han sido aceptadas previamente por el profesor. Para más información ver: 2.6.

13.3.9 Modificar su información personal

Para la información personal basta con seleccionar la opción “modificar perfil” en el menú izquierdo de la aplicación. Se introducirán los nuevos datos y se

modificarán los antiguos en caso de existir. El proceso finalizará mediante el botón “OK”.

Datos personales

Nombre	David Benavides
Categoría	Dr

Datos de contacto

Teléfono	954323435
Despacho	
Web	

ok

Nota: Todos los campos son obligatorios.

13.3.10 Introducir noticias

Para introducir una noticia basta con seleccionar la opción “Introducir Noticia” en el menú izquierdo de la aplicación. Se seleccionará la asignatura a la que pertenece la noticia, la cual contendrá un título y una breve descripción. Se finalizará el proceso mediante el botón “OK”.

INSERTAR NOTICIA

Asignatura	Elija una asignatura
Título	
Texto	

aceptar

13.3.11 Modificar noticias

Para modificar una noticia basta con seleccionar la opción “Modificar Noticia” en el menú izquierdo de la aplicación. A continuación se seleccionará una asignatura y una noticia de la lista de asignaturas que se han obtenido. Se introducirán o se modificarán los datos o se eliminará completamente mediante el enlace “eliminar”.

Elija una asignatura

- [aaaa](#)
- [ip1](#)
- [ip2](#)
- [Etc](#)
- [CI](#)

Elija una noticia

- [Examen](#)

MODIFICAR NOTICIA

Asignatura	<input type="text" value="ip1"/> <input type="button" value="▼"/>
Titulo	<input type="text" value="Examen"/>
Texto	<div style="border: 1px solid #ccc; padding: 5px;"><p>El examen corregido se ha subido a la web</p></div>

[Eliminar](#)

13.3.12 Consultar noticias

Para consultar una noticia basta con seleccionar la opción “Consultar Noticia” en el menú izquierdo de la aplicación. A continuación se seleccionará una asignatura y una noticia de la lista de asignaturas que se han obtenido.

Elija una asignatura

- [aaaa](#)
- [ip1](#)
- [ip2](#)
- [Etc](#)
- [CI](#)

Noticias:

- **Examen(01/02/2007 00:00)**

El examen corregido se ha subido a la web

13.4 Manual del administrador

Las acciones que el usuario administrador podrá realizar son:

- Añadir titulaciones.
- Eliminar titulaciones.
- Añadir Asignaturas.
- Eliminar Asignaturas.
- Añadir usuarios.
- Eliminar usuarios.
- Vincular una asignatura a una titulación.
- Desvincular una asignatura a una titulación.
- Vincular un usuario a una asignatura.
- Desvincular un usuario de una asignatura.
- Importar una lista de usuarios.
- Importar una lista de asignaturas y titulaciones.
- Vaciar lista de asignaturas.
- Añadir departamentos
- Modificar Departamentos

¿Cómo empezar a organizar el portal?

Para facilitar el trabajo del administrador se recomienda que se proceda como se describe a continuación:

- 1) Añadir las distintas titulaciones
- 2) Añadir los departamentos sino se van a importar las asignaturas.
- 3) Añadir las asignaturas al sistema de forma manual o mediante un archivo de importación (Véase el apartado 13.4.3 y 13.4.12).
Nota: Se recomienda generar distintos archivos de importación, con las asignaturas comunes a las titulaciones y con las asignaturas propias de cada titulación.
- 4) Seleccionar las distintas titulaciones para ser vinculadas con las asignaturas.
- 5) Añadir los usuarios, bien de forma manual o mediante un fichero de importación (Véase el apartado 13.4.5 y 13.4.11).
Nota: Se recomienda generar distintos archivos de importación con la lista de usuarios de las asignaturas.
- 6) Seleccionar las distintas asignaturas de los usuarios.

Nota: Se recomienda trabajar con las opciones de importación (xml y Excel) para facilitar el trabajo del administrador y usar el resto de las herramientas para posibles incidencias o excepciones.

13.4.1 Añadir Titulaciones.

Para añadir una titulación basta con seleccionar la opción “Añadir Titulación” en el menú izquierdo de la aplicación. A continuación se mostrará un formulario, donde se incluirá el nombre de la titulación y su código correspondiente.

Añadir Titulación	
Nombre	<input type="text"/>
codigo	<input type="text"/>
	<input type="button" value="ok"/>

Observación: Para que la acción se pueda realizar correctamente deberá indicar un código único en la aplicación. De lo contrario se indicará el error y se solicitará uno nuevo.

13.4.2 Eliminar Titulaciones

Para eliminar una titulación se seleccionará la opción “Eliminar Titulación” en el menú izquierdo de la aplicación. A continuación se mostrará una lista con las titulaciones actuales y se seleccionará la titulación que se deseé eliminar. Se procederá de forma iterativa mientras el usuario desee eliminar titulaciones. Cuando no se deseé eliminar más titulaciones se seleccionará la opción “terminar”.

Eliminar Titulación	
• 1 Gestión	Eliminar
• 2 Sistemas	Eliminar
• 3 Biología	Eliminar
• 4 Medicina	Eliminar

Observación: Las titulaciones se eliminarán de forma individual en cada iteración y no mediante la opción “terminar”.

Nota: No es necesario desvincular previamente las asignaturas que pueda poseer una titulación. Este proceso es automático.

13.4.3 Añadir Asignaturas

Para añadir una asignatura se seleccionará la opción “Añadir Asignatura” en el menú izquierdo de la aplicación. A continuación se complementará un formulario con los datos básicos de la asignatura y se seleccionará el departamento al que pertenece. Para finalizar el proceso se pulsará en el botón “OK”. Posteriormente se pasará al proceso “vincular una asignatura a una titulación” (ver apartado 13.4.7 del manual)

Añadir Asignatura

Nombre	ip1
Curso	1
codigo	32
Cuatrimestre	1
Créditos	7.5
Web	www.lsi.ip.com
Duración	Anual
Tipo	Optativa
Departamento	<input checked="" type="radio"/> LSI <input type="radio"/> DTE
ok	

Observación: Todos los campos son obligatorios y debe existir departamento en el sistema. Para importar departamentos a partir de un fichero ver el apartado 3.12

13.4.4 Eliminar Asignaturas.

Para eliminar una asignatura se seleccionará la opción “Eliminar Asignatura” en el menú izquierdo de la aplicación. A continuación se mostrará una lista con las titulaciones del sistema. El administrador podrá seleccionar una titulación si conoce las asociaciones de la asignatura o seleccionará la opción “ver todas las asignaturas”. En ambos casos, se mostrará una lista con las asignaturas y se seleccionará la asignatura a eliminar.

Esta acción no se puede realizar de forma iterativa. Para eliminar más de una asignatura se deberá repetir el procedimiento.

Eliminar asignatura

Elija una asignatura

- ip1 www.lsi.ip1.vocm [Eliminar](#)
- ip2 www.lsi.ip1.vocm [Eliminar](#)
- Etc www.lsi.ip1.vocm [Eliminar](#)
- CI www.lsi.ip1.vocm [Eliminar](#)

Observación: Al eliminar una asignatura a partir de una titulación, ésta desaparecerá completamente del sistema (no se está desvinculando dicha asignatura, para ello véase el apartado 3.8).

Nota: Al eliminar la asignatura, se eliminarán de forma automática todas las relaciones que existan entre dicha asignatura y los usuarios y titulaciones del sistema, no es necesario desvincular la asignatura ni previa ni posteriormente.

13.4.5 Añadir Usuario

Para añadir un nuevo usuario se seleccionará la opción “Añadir usuario” en el menú izquierdo de la aplicación. A continuación se mostrará un formulario en el que se indicará el código de identificación del usuario (el DNI del usuario) y su clave personal (originalmente será su DNI).

Añadir Usuario

usuario	11111111	ok
Contraseña		
<input checked="" type="radio"/> Alumno	<input type="radio"/> Profesor	

Nota: Si el DNI ya existe en el sistema, se indicará el error y se solicitará uno nuevo.

13.4.6 Eliminar Usuario

Para eliminar un nuevo usuario se seleccionará la opción “Eliminar Usuario” en el menú izquierdo de la aplicación. A continuación se mostrará un formulario en el que se indicará el DNI del alumno a eliminar.

Eliminar Usuario

usuario	22222222	ok
---------	----------	----

13.4.7 Vincular una asignatura a una titulación

Este procedimiento se puede realizar seleccionando la opción “vincular asignatura” en el menú izquierdo de la aplicación o bien de forma automática después de haber añadido o importado una asignatura. Para vincular una asignatura en el sistema, primero se selecciona una asignatura de la lista, en el caso de que se haya accedido desde el menú, y posteriormente se seleccionará la lista de titulaciones a vincular. Para finalizar el procedimiento se pulsará “OK”.

Buscar Asignatura

Elija una asignatura

ok

Seleccionar Titulacion

Gestion
 Sistemas
 Biologia
 Medicina

ok

13.4.8 Desvincular una asignatura a una titulación

Para desvincular una asignatura de una titulación se seleccionará la opción “modificar asignatura” en el menú izquierdo de la aplicación. A continuación, se seleccionará la asignatura de la lista y se mostrará una lista con todas las titulaciones del sistema, marcando a las que pertenece dicha asignatura. El administrador podrá habilitar o deshabilitar las distintas titulaciones para que sean vinculadas o desvinculadas respectivamente. Se finalizará el proceso mediante el botón “OK”.

Observación: Este proceso permite tanto desvincular como vincular asignaturas.

Añadir Usuario

usuario: 11111111

Contraseña:

Alumno Profesor

13.4.9 Vincular un usuario a una asignatura

Este procedimiento se puede realizar seleccionando la opción “vincular usuario” en el menú izquierdo de la aplicación o bien de forma automática

después de haber añadido un nuevo usuario.

Para vincular un usuario en el sistema, primero se selecciona un usuario mediante una búsqueda del DNI, en el caso de que se haya accedido desde el menú, y posteriormente se seleccionará la titulación concreta o la opción “ver todas las asignaturas”. El sistema mostrará una lista de asignaturas dependiendo de la opción seleccionada. Se seleccionarán las asignaturas y se finalizará el proceso mediante el botón “OK”.



13.4.10 Desvincular un usuario de una asignatura

Para desvincular una asignatura de una titulación se seleccionará la opción “Desvincular usuario” en el menú izquierdo de la aplicación. A continuación se realizará una búsqueda de un usuario mediante su DNI. Posteriormente se seleccionará la titulación a eliminar y finalizará el proceso.



13.4.11 Importar una lista de usuarios

Para importar una lista de usuarios, se pueden llevar a cabo dos métodos:

- Importar usuario a partir de un fichero XML

Para importar una lista de usuarios, basta con seleccionar la opción “Importar usuarios” en el menú izquierdo de la aplicación.

A continuación se seleccionará la titulación si se quiere vincular los usuarios a asignaturas concretas de una titulación, o la opción “todas las asignaturas”, para ver todas las asignaturas del sistema.

Seleccionar Titulacion

- [1 Gestion](#)
- [2 Sistemas](#)
- [3 Biología](#)
- [4 Medicina](#)

[Ver Todas las Asignaturas](#)

Posteriormente, se indicará el archivo xml que contiene la lista de usuarios a importar y se seleccionará el tipo correspondiente.

Usuarios

 Examinar...
 XML EXCEL

Seleccionar Asignaturas

<input type="checkbox"/>	aaaa
<input type="checkbox"/>	ip1
<input type="checkbox"/>	ip2
<input type="checkbox"/>	Etc
<input type="checkbox"/>	CI
subir	

Un único archivo xml podrá contener tanto a usuarios profesores, como a usuarios alumnos. El archivo xml tendrá la siguiente estructura:

```
<usuarios>
    <alumnos>
        ...
        <alumno>
            <dni></dni>
            <contraseña></contraseña>
        </alumno>
        ...
    </alumnos>

    <profesores>
        ...
        <profesor>
```

```
<dni></dni>
<contraseña></contraseña>
</profesor>
...
</profesores>
</usuarios>
```

A continuación se seleccionará la lista de asignaturas a la que los usuarios estarán vinculados.

Los usuarios cuyo DNI ya se encuentre registrado en el sistema, serán descartados y no serán añadidos al sistema, pero sí entrarán en el proceso de vinculación con las asignaturas.

Observación: Se recomienda que las asignaturas ya se encuentren almacenadas en el sistema antes de importar la lista de usuarios, para facilitar la tarea del administrador.

b) Importar usuario a partir de un fichero Excel

Para importar una lista de usuarios, basta con seleccionar la opción “Importar usuarios” en el menú izquierdo de la aplicación.

A continuación se seleccionará la titulación si se quiere vincular los usuarios a asignaturas concretas de una titulación o la opción “todas las asignaturas”, para ver todas las asignaturas del sistema.

Posteriormente, se indicará el archivo Excel que contiene la lista de usuarios a importar y se seleccionará el tipo correspondiente.

El archivo Excel sólo podrá contener un tipo concreto de usuario. La estructura del documento Excel será la siguiente:

- Columna 0, Fila 0: En esta celda se indicará el tipo de usuario del que se trata (alumno o profesor)
- Columna 0: En esta columna se indicará los DNI (sin letra) de los usuarios.
- Columna 1: En esta columna se indicará los apellidos y los nombres de los usuarios. Apellidos y nombre estarán delimitados mediante una coma.

A continuación se seleccionará la lista de asignaturas a la que los usuarios estarán vinculados.

Los usuarios cuyo DNI ya se encuentre registrado en el sistema, serán descartados y no serán añadidos al sistema, pero sí entrarán en el proceso de vinculación con las asignaturas.

Las filas del documento Excel que no estén formadas correctamente serán descartadas.

Observación: Se recomienda que las asignaturas ya se encuentren almacenadas en el sistema antes de importar la lista de usuarios, para facilitar la tarea del administrador.

13.4.12 Importar una lista de asignaturas

Para importar una lista de asignaturas, se pueden llevar a cabo los siguientes procedimientos.



a) Importar a partir de un archivo XML

Para importar una lista de asignaturas a partir de un archivo xml basta con seleccionar en el menú izquierdo la opción “importar asignaturas”. A continuación se mostrará un campo solicitando el archivo a importar. Posteriormente, se seleccionará la opción XML.

El archivo xml debe de seguir la siguiente estructura:

```
<asignaturas>
...
<asignatura>
    <codigo></codigo>
    <nombre></nombre>
    <curso></curso>
    <cuatrimestre></cuatrimestre>
    <departamento></departamento>
    <web></web>
    <creditos></creditos>
    <duración></duración>
    <tipo></tipo>
</asignatura>
...
</asignaturas>
```

Para un correcto funcionamiento de la aplicación, el administrador debe comprobar que el departamento existe en el sistema. En caso contrario la asignatura será descartada y no será añadida al sistema.

En caso de que el departamento no exista en el sistema, se podrá importar en el mismo fichero de la asignatura de la siguiente forma:

```
<departamentos>
...
<departamento>
    <nombre></nombre>
    <codigo></codigo>
    <web></web>
</departamento>
...
</departamentos>
```

A continuación, se seleccionarán las titulaciones de la lista para que la lista de asignaturas quede vinculada a estas.

Observación: No es necesario garantizar por parte del administrador que la asignatura y el departamento no se encuentren en el sistema, si estas ya han sido insertadas se descartarán automáticamente. Las asignaturas descartadas sí entrarán en el proceso de vinculación.

Si la asignatura no existe en el sistema pero su código sí, será descartada y no será añadida al sistema.

Se recomienda importar las asignaturas cuando las titulaciones ya se encuentren en el sistema (ver punto 13.4.1).

b) Importar a partir de un fichero Excel

Para importar una lista de asignaturas a partir de un archivo Excel basta con seleccionar en el menú izquierdo la opción “importar asignaturas”. A continuación se mostrará un campo solicitando el archivo a importar. Posteriormente, se seleccionará la opción Excel.

El archivo Excel deberá seguir la siguiente estructura:

En el caso de no existir los departamentos de las asignaturas en el sistema, se colocaran al principio del documento de la siguiente forma:

- En la columna 0, fila 0, se escribirá “departamento”.
- En las posteriores columnas 0 se escribirá el código del departamento.
- En las posteriores columnas 1 se escribirá el nombre del departamento.
- En las posteriores columnas 2 se escribirá la Web del departamento.

Cuando finalicen los departamentos, se escribirá “asignatura” en la columna cero y en la fila que corresponda. Posteriormente los datos se colocaran de la siguiente forma:

- En la columna 0 se escribirá el nombre de la asignatura.
- En la columna 1 se escribirá el curso de la asignatura.
- En la columna 2 se escribirá el código de la asignatura.
- En la columna 3 se escribirá el cuatrimestre de la asignatura.
- En la columna 4 se escribirán los créditos de la asignatura.
- En la columna 5 se escribirá la Web de la asignatura.

En la columna 6 se escribirá la duración de la asignatura.

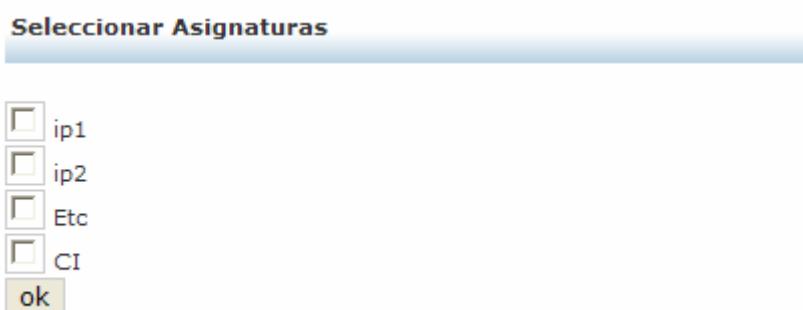
En la columna 7 se escribirá el tipo de la asignatura.

En la columna 8 se escribirá el código del departamento al que pertenece la asignatura.

A continuación, se seleccionarán las titulaciones de la lista para que la lista de asignaturas queden vinculadas a estas.

13.4.13 Vaciar Lista asignatura

Para eliminar todos los usuarios de una asignatura, se seleccionará la opción “vaciar lista asignatura” en el menú izquierdo de la aplicación. A continuación se seleccionará la titulación si se conoce la localización de las asignaturas, o la opción “ver todas las asignaturas” en caso contrario. Posteriormente se seleccionarán las asignaturas de la lista y se finalizará el proceso mediante el botón “OK”.



13.4.13 Añadir Departamento

Para añadir un departamento, se seleccionará la opción añadir departamento en el menú izquierdo de la aplicación. A continuación se indicara el nombre, el código y la Web del departamento.

Añadir Departamento

Nombre	<input type="text" value="lsi"/>	<input type="button" value="ok"/>
codigo	<input type="text" value="23"/>	
Web	<input type="text" value="www.lsi.com"/>	

Nota: El código del departamento deberá ser único en el sistema.

13.4.13 Modificar Departamento

Para modificar un departamento, se seleccionará la opción modificar departamento en el menú izquierdo de la aplicación. A continuación se seleccionara el departamento a modificar y se introducirán los nuevos datos.

Buscar Departamento

<input type="button" value="Elija un departamento"/>	<input type="button" value="ok"/>
<input type="text" value="Elija un departamento"/>	
<input type="text" value="dte"/>	
<input style="background-color: #0070C0; color: white;" type="text" value="lsi"/>	
<input type="text" value="ee"/>	
<input type="text" value="sdc"/>	
<input type="text" value="asp"/>	
<input type="text" value="iso"/>	

PARTE IV: CONCLUSIONES

14 Cumplimiento de los objetivos del proyecto

En el ámbito funcional, el cumplimiento de los objetivos propuestos se ha realizado de manera satisfactoria. Así, si en un primer momento decidimos centrarnos en la Gestión de las fichas personales de los alumnos, posteriormente, los objetivos se ampliaron considerablemente abarcando la Gestión de peticiones de tutorías. Esta ampliación nos supuso un mayor esfuerzo en el desarrollo del proyecto pero finalmente hemos podido cumplir todos los objetivos que establecimos.

Además, a medida que hemos ido desarrollando nuestra aplicación, hemos ido añadiendo funcionalidades que en un principio no pensamos desarrollar. Algunos ejemplos son:

- Importación mediante Excel: originalmente en los objetivos nos propusimos la importación mediante xml, pero no mediante archivos del tipo Excel.
- Instalador: que permite instalar en una maquina, nuestra aplicación para su uso de manera muy sencilla. Originalmente sólo contemplamos la opción de hacerlo manualmente.

En cuanto al aprendizaje de nuevas tecnologías, los objetivos se han cumplido también de manera satisfactoria. Así, hemos podido aprender e integrar en nuestra aplicación las dos tecnologías que nos propusimos originalmente: Struts e Hibernate.

Destacar también que a lo largo del desarrollo del proyecto hemos ampliado nuestros objetivos de aprendizaje con una nueva tecnología como es AJAX. El uso de AJAX en nuestra aplicación, no fue contemplado originalmente.

Como objetivo de aprendizaje no cumplido podríamos mencionar el intento de migración desde Struts a Struts 2.0 de nuestra aplicación. El proceso resultaba muy costoso ya que nos suponía grandes cambios, así que fue descartado. Este objetivo surgió a lo largo del desarrollo de nuestro proyecto, no se trataba de un objetivo original.

En conclusión, podemos considerar que nuestra aplicación ha cumplido todos los objetivos iniciales marcados, incorporando nuevos objetivos que surgieron durante el desarrollo del proyecto y que consideramos muy beneficiosos para nuestra aplicación.

15 Esfuerzo empleado en el proyecto

En este apartado se explicará la planificación que se ha seguido y en qué modo se ha cumplido, y también se estimará el coste del proyecto y se explicarán las técnicas usadas para ello.

15.1 Planificación

La planificación es una estimación del tiempo necesario para llevar a cabo las distintas tareas que se realizan durante el desarrollo de un proyecto.

Así, una estimación es “*una predicción que tiene la misma probabilidad de estar por encima o por debajo del valor actual (Tom DeMarco)*”. El principal objetivo de la estimación es reducir los costes.

Para nuestro estudio, realizaremos una estimación inicial al comienzo del proyecto y anotaremos el tiempo real empleado al finalizar el proyecto. A partir de esos datos calcularemos el error relativo de estimación de la siguiente forma:

$$ER = (\text{Valor Real} - \text{Valor Estimado}) / \text{Valor Real}$$

También calcularemos el error relativo medio:

$$\text{Error Relativo Medio} = (1/n) \sum_{i=1}^n ER_i$$

A continuación se muestran los resultados obtenidos:

Tarea	Estimación Inicial	Estimación Final	ER
Búsqueda de la información	9 días	12 días	25%
Análisis de tareas	4 días	2 días	-50%
Elicitación de requisitos	14 días	17 días	17.64%
Análisis de requisitos	12 días	15 días	20%
Instalación de la plataforma	3 días	5 días	40%
Diseño del sistema	16 días	13 días	-23%
Prototipo del sistema	4 días	4 días	0%
Implementación	65 días	78 días	16.6%
Pruebas	8 días	6 días	-33%
Rediseño	5 días	5 días	0%
Presentación	5 días	3 días	66%
TOTAL	145 días	160 días	9.37%
Error relativo Medio			7.17%

Teniendo en cuenta que el proceso de estimación, es un proceso muy difícil de realizar, podemos considerar que la estimación inicial realizada fue buena. También podemos ver que la estimación inicial ha sido inferior al tiempo real necesario para realizar el proyecto, echo que suele ocurrir en los procesos de planificación.

15.1 Estimación del coste del proyecto

Los costes de realización de un proyecto suelen ir asociados tanto al software como al hardware. En esta apartado nos centraremos en los costes asociados al software, y más concretamente, en los costes dedicados al desarrollo y prueba de nuestra aplicación.

Existen varios tipos de métricas para la estimación del coste: las métricas que utilizan el “tamaño” y las que utilizan los “puntos de función”.

MÉTRICAS BASADAS EN EL TAMAÑO

El tamaño de nuestro proyecto será indicado por las líneas de código existentes en nuestra aplicación (LOC). Cabe destacar, que no siempre se corresponde el número de líneas de código, con el esfuerzo realizado.

Como nuestro estudio no tiene en cuenta los puntos de función, utilizaremos el modelo de COCOMO clásico.

E1 COCOMO Clásico.

Existen diferentes subtipos de cocomo clásico que tienen en cuenta distintos grados de complejidad. Estos son:

- Cocomo básico: obtiene una estimación del esfuerzo, basándose en el tamaño al inicio del ciclo de vida de la aplicación.
- Cocomo intermedio: que añade unos atributos que influyen en el coste (CDA) y que pretenden tener en cuenta la aplicación, el entorno de desarrollo
- Cocomo Adelantado: semejante al cocomo intermedio, pero con la diferencia de que los CDA no son constantes para todo el ciclo de vida.

En nuestro caso, utilizaremos el cocomo clásico.

La ecuación del esfuerzo del cocomo clásico es:

$$E = a^*(LC)^b$$
$$T = c^*E^d$$

Donde E representa el esfuerzo en personas-mes y LC las líneas de código, en KLOC (miles de líneas de código).

Atendiendo al tipo de proyecto, se pueden obtener los siguientes modelos, que nos ayudaran a determinar los coeficientes anteriores:

- Orgánico: Proyectos relativamente pequeños y sencillos, con poca innovación tecnológica, donde trabaja un pequeño equipo. Son menores de 50000 líneas.
- Semiacoplado: Proyectos de nivel intermedio en tamaño, complejidad y sofisticación técnica, donde varios equipos diferentes colaboran para construir una aplicación.
- Encajado: Proyectos de un tamaño y complejidad francamente elevados, donde intervienen varios equipos diferentes. Son proyectos muy innovadores.

ESFUERZO EMPLEADO EN EL PROYECTO

En nuestro estudio, tomaremos el modelo Orgánico. Así los coeficientes quedaran determinados de la siguiente manera:

$$A = 2.4, b = 1.05, c = 2.5, d = 0.38$$

Aplicando Metrics hemos obtenido que el número de líneas de código de nuestra aplicación es de 7220.

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method
Number of Overridden Methods (avg/max per type)	36	0,298	0,555	2	/Proyecto/src/action/identificacion/LoginForm.java	
Number of Attributes (avg/max per type)	244	2,017	3,053	16	/Proyecto/src/dominio/Asignatura.java	
Number of Children (avg/max per type)	44	0,364	3,23	35	/Proyecto/src/action/Accion.java	
Number of Classes (avg/max per packageFragment)	121	11	6,715	25	/Proyecto/src/action/administracion	
Method Lines of Code (avg/max per method)	3755	5,095	7,32	47	/Proyecto/src/action/administracion/FileUploadXMLUsAction...	execute
Number of Methods (avg/max per type)	736	6,083	7,178	37	/Proyecto/src/dominio/Asignatura.java	
Nested Block Depth (avg/max per method)		1,288	0,609	6	/Proyecto/src/action/administracion/VerTitulacionesAction...	ejecutar
Depth of Inheritance Tree (avg/max per type)		2,306	1,043	5	/Proyecto/src/action/asignatura/AsignaturasAction.java	
Number of Packages	11					
Afferent Coupling (avg/max per packageFragment)		14,364	20,764	50	/Proyecto/src/action	
Number of Interfaces (avg/max per packageFragment)	17	1,545	3,34	12	/Proyecto/src/data	
McCabe Cyclomatic Complexity (avg/max per method)		1,49	1,542	26	/Proyecto/src/action/informacionUsuarios/FichaForm.java	validate
Total Lines of Code	7220					
Instability (avg/max per packageFragment)		0,633	0,4	1	/Proyecto/src/action/administracion	
Number of Parameters (avg/max per method)		1,243	1,664	15	/Proyecto/src/dominio/Profesor.java	Profesor
Lack of Cohesion of Methods (avg/max per type)		0,224	0,34	0,939	/Proyecto/src/dominio/Asignatura.java	
Efferent Coupling (avg/max per packageFragment)		11,273	8,791	26	/Proyecto/src/action/administracion	
Number of Static Methods (avg/max per type)	1	0,008	0,091	1	/Proyecto/src/data/HibernatePool.java	
Normalized Distance (avg/max per packageFragment)		0,377	0,369	1	/Proyecto/src/dominio	
Abstractness (avg/max per packageFragment)		0,09	0,137	0,5	/Proyecto/src/data	
Specialization Index (avg/max per type)		0,199	0,511	2	/Proyecto/src/action/FileUploadAction.java	
Weighted methods per Class (avg/max per type)	1098	9,074	10,852	76	/Proyecto/src/action/administracion/AdministracionManage...	
Number of Static Atributes (avg/max per type)	49	0,405	0,675	5	/Proyecto/src/action/tutorias/PeticionCitaMan...	

Sustituyendo en la ecuación de Cocomo obtenemos:

$$E = a * (LC)^b = 2.4 * (7'220)^{1.05} = 19'13 \text{ personas - mes}$$

$$T = c * E^d = 2.5 * (19.13)^{0.38} = 7'67 \text{ mes}$$

$$\text{Número medio de personas} = 19'13 / 7'67 = 2-3 \text{ personas}$$

ESFUERZO EMPLEADO EN EL PROYECTO

Respecto a los resultados hay que tener en cuenta que aplicando metrics, solo obtenemos las líneas de código en java, no se están recogiendo las líneas existentes en los jsp. Por tanto el número de líneas del proyecto seria muy superior a 7220.

También hay que destacar que el esfuerzo ha sido mayor ya que aquí no se recoge el esfuerzo de aprendizaje de las nuevas tecnologías aplicadas en el proyecto.

PARTE V: BIBLIOGRAFÍA CONSULTADA

16 Bibliografía y referencias

- <http://www.laliluna.de/tutorials.html>
- <http://www.javahispano.com>
- <http://www.monografias.com/>
- <http://es.wikipedia.org/>
- <http://www.guiaweb.gob.cl/guia/capitulos/cuarto/queprobar.htm>
- *Hibernate in Action*, Christian Bauer, Gavin King
- *Struts in Action*, Ted Usted, Cedric Dumoulin, George Franciscus, David Winterfeldt
- *Sistema R.E.M., versión 1.2.2.* Amador Durán Toro. Departamento de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Septiembre 2004.
- *Metodología para el Análisis de Requisitos de Sistemas Software. Versión 2.3.* Amador Durán Toro, Beatriz Bernárdez Jiménez. Departamento de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Diciembre 2001.