



Universidad
Rey Juan Carlos

GRADO EN INGENIERÍA DEL SOFTWARE

Curso Académico 2012/2013

Trabajo de Fin de Grado

OMIT - Opinion Mining In Teaching

Un sistema para la evaluación y mejora de la docencia basado en
análisis de opinión

Autor: Jónatan Núñez Martín

Tutor: Soto Montalvo Herranz

Resumen

Este documento describe el proyecto llevado a cabo en la realización del Trabajo Fin de Grado para finalizar los estudios del Grado en Ingeniería del Software.

Una de las maneras de mejorar un servicio ofrecido a la sociedad es conociendo la opinión de los usuarios que hacen uso de él. Juntando los pequeños juicios individuales se pueden llegar a conocer carencias importantes en el servicio o peticiones constantes que los usuarios demandan. Para quien gestiona el servicio, esto representa un alto valor de cara al futuro, de cara a nuevos objetivos o enfoques.

En la Universidad Rey Juan Carlos los alumnos pueden valorar a los profesores y las asignaturas que estos imparten (también servicios como la limpieza o el comedor). La forma oficial con la cual los alumnos evalúan todo lo relacionado con las asignaturas es mediante un único cuestionario por asignatura, que se rellena al final de cada cuatrimestre.

Estos cuestionarios consisten en un conjunto de preguntas fijas e iguales para todo profesor y asignatura. Los alumnos valoran entre 5 grados de satisfacción distintos puntos: calidad de las clases del profesor, si se ajusta a los contenidos de la Guía Docente, su puntualidad, etc.

Esta forma de evaluar y, por tanto, enfocar tiene limitaciones importantes: los profesores no pueden personalizar las preguntas y por tanto enfocar los cuestionarios hacia temas concretos, los alumnos sólo pueden opinar una vez, el día que el cuestionario es entregado, y únicamente pueden opinar sobre las preguntas impuestas. Por otro lado, se gastan recursos como podría ser el papel de los cuestionarios.

El trabajo realizado en el TFG pretende solventar las limitaciones citadas además de llevar la gestión de las evaluaciones de los alumnos a un sistema informático fácilmente mantenible, flexible y útil.

Unos de los requisitos más importantes es disminuir los trabajos manuales e informatizar el sistema de evaluación actual de la URJC dejando a un lado el uso del papel. Con el nuevo sistema presentado en este trabajo, todas las etapas y usos de la aplicación se realizarían desde un navegador web conectado a Internet.

El sistema actual ofrece a los profesores estadísticas y gráficas procesando las respuestas de los cuestionarios. La nueva aplicación permite añadir, modificar y eliminar las preguntas fijas para que los profesores puedan orientar la evaluación hacia temas donde deseen. Sin embargo, la personalización de estas preguntas conlleva a unas conclusiones sesgadas: aunque las preguntas intenten abarcar un amplio número de temas, los alumnos no pueden opinar sobre otras cuestiones. Esta limitación se solventa introduciendo un área de texto donde el alumno puede comentar sobre cualquier tipo de tema.

Con el objetivo de mejorar las estadísticas y aportar más información relevante, la nueva aplicación permite realizar búsquedas sobre los comentarios de los alumnos, con palabras clave, rangos de fecha o polaridad de los comentarios.

Dado los diferentes perfiles que harán uso del nuevo sistema, y teniendo en cuenta los distintos conocimientos de informática que cada uno podría tener, la nueva aplicación intenta ser intuitiva, accesible, cómoda, segura y fácilmente usable.

Listado de acrónimos

- **AJAX**: del inglés, *Asynchronous JavaScript And XML*, JavaScript Asíncrono y XML.
- **API**: del inglés, *Application Programming Interface*, Interfaz de Programación de Aplicaciones.
- **CSV**: del inglés, *Comma-Separated Values*, Valores Separados por Coma.
- **CSS**: del inglés, *Cascading Style Sheets*, Hojas de Estilo en Cascada.
- **CU**: Caso de Uso.
- **DOM**: del inglés, *Document Object Model*, Modelo de Objetos del Documento.
- **GPL**: del inglés, *General Public License*, Licencia Pública General.
- **HTML**: del inglés, *HyperText Markup Language*, Lenguaje Etiquetado de Hipertexto.
- **HTTP**: del inglés, *HyperText Transfer Protocol*, Protocolo de Transferencia de Hipertexto.
- **JAR**: del inglés, *Java ARchive*, Archivo de Java.
- **JSON**: del inglés, *JavaScript Object Notation*, Objeto de Notación de JavaScript.
- **MD5**: del inglés, *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5.
- **MVC**: Modelo Vista Controlador.
- **PHP**: del inglés, *PHP Hypertext Pre-processor*, Procesador PHP de Hipertexto.
- **PLN**: Procesamiento del Lenguaje Natural.
- **RIA**: del inglés, *Rich Internet Application*, Aplicación Rica de Internet.
- **SGBD**: Sistema de Gestión de Bases de Datos.
- **TFG**: Trabajo de Fin de Grado.
- **XML**: del inglés, *eXtensible Markup Language*, Lenguaje de Etiquetado Extensible.
- **URL**: del inglés, *Uniform Resource Locator*, Localizador Uniforme de Recursos.

Índice general

Resumen	1
Listado de acrónimos	2
1. Introducción	7
1.1 Entorno	7
1.2 Caso	7
1.3 Problemas	7
1.4 Posibles soluciones	8
1.5 Solución propuesta	8
1.6 Procesamiento del lenguaje	9
2. Metodología y tecnologías	11
2.1 Metodología escogida	11
2.2 Tecnologías utilizadas	13
3. Propuesta	15
3.1 Sistema y servidor para la página web	15
3.2 Procesamiento del Lenguaje Natural	15
3.2.1 <i>Clustering</i>	15
3.2.2 Análisis de sentimientos	16
3.3 Bases de datos: Lucene y MySQL	17
3.4 Página web	17
3.4.1 Acceder	18
3.4.2 Alumno	18
3.4.3 Profesor	20
3.4.4 Administrador	23
4. Descripción informática	25
4.1 Modelos de casos de uso	25
4.1.1 Modelo de casos de uso para acceder	25
4.1.2 Modelo de casos de uso del alumno	26
4.1.3 Modelo de casos de uso del profesor	27
4.1.4 Modelo de casos de uso del administrador	30
4.2 Diseño	32
4.2.1 Diseño de la página web	32
4.2.2 Diseño de la base de datos MySQL	34
4.2.3 Diseño de la base de datos Lucene	35

4.2.4	Diseño del algoritmo de análisis de sentimientos	36
5.	Resultados experimentales	38
5.1	Resultados experimentales del algoritmo de análisis de sentimientos	38
5.1.1	Resultados experimentales: 32 <i>tuits</i>	38
5.1.2	Resultados experimentales: 3080 críticas de cine	40
5.2	Resultados experimentales del algoritmo de <i>clustering</i>	41
6.	Conclusiones y trabajos futuros	44
6.1	Conclusiones	44
6.2	Trabajos futuros	44
	Bibliografía	46

Índice de figuras

1.1: Diagrama de actividad del alumno del envío de comentarios.	9
1.2: Diagrama de actividad del profesor para pedir temas de los comentarios.	10
2.1: Prácticas de la Programación Extrema.	12
3.1: Comentarios agrupados aplicando sobre ellos el algoritmo de <i>clustering</i> STC.	16
3.2: Página de acceso para los usuarios.	18
3.3: El alumna selecciona asignatura y profesor a evaluar.	18
3.4: Página del alumno respondiendo a las preguntas del profesor.	19
3.5: Un alumno escribiendo un comentario para el profesor.	20
3.6: Un profesor editando las preguntas para la asignatura “Informática” de primer curso. ...	20
3.7: Gráfico con 3 valoraciones de los alumnos realizadas en julio	21
3.8: Listado de comentarios realizados por los alumnos a un profesor determinado.	22
3.9: Ejemplo de temas de los que pueden opinar los alumnos en sus comentarios.	22
3.10: Cuando accede el administrador, puede exportar información, importarla o borrar los datos del sistema.	23
4.1: Modelo de casos de uso de los usuarios para acceder	25
4.2: Modelo de casos de uso del alumno	26
4.3: Modelo de casos de uso del profesor	27
4.4: Modelo de casos de uso del administrador.	31
4.5: Diseño MVC de la página web.	33
4.6: Paquete de clases de ayuda para la aplicación web	33
4.7: Diseño de la base de datos MySQL.	35
4.8: Diseño de la base de datos Lucene.	36

Índice de tablas

3.1: Tabla de titulaciones de la universidad	23
3.2: Tabla de cursos correspondidos con una titulación.....	23
3.3: Tabla de las asignaturas, cada una correspondida con un curso.	23
3.4: Tabla con los datos de los profesores. El campo de contraseña se encuentra codificada con el algoritmo MD5.	23
3.5: Tabla de asignación entre un profesor y una asignatura	24
4.1: Flujo de eventos básico del caso de usos “Acceder”.	26
4.2: Flujo de eventos básico del caso de usos “Alumno”.	27
4.3: Flujo de eventos básico del caso de usos “Ver comentarios”.	28
4.4: Flujo de eventos básico del caso de usos “Buscar en los comentarios”.	28
4.5: Flujo de eventos básico del caso de usos “Buscar temas de los comentarios”.	29
4.6: Flujo de eventos básico del caso de usos “Ver estadísticas de las preguntas”.	29
4.7: Flujo de eventos básico del caso de usos “Editar preguntas”.	30
4.8: Flujo de eventos básico del caso de usos “Cargar nuevos alumnos”.	30
4.9: Flujo de eventos básico del caso de usos “Cargar nuevos datos para la universidad”.	31
4.10: Flujo de eventos básico del caso de usos “Borrar datos del sistema”.	32
5.1: <i>Tuits</i> analizados por el algoritmo de análisis de sentimientos.	40
5.2: Resultados experimentales de los comentarios de cine.	41
5.3: Distribución de los grupos de las noticias.	42
5.4: Resultados de las pruebas de <i>clustering</i>	43

1. Introducción

En este capítulo se describe el problema que se quiere solventar, su entorno, soluciones actuales, la solución que proponemos y los objetivos que habría que cumplir.

1.1 Entorno

En la actualidad, Internet forma parte del día a día de la sociedad. Incluso unas de las prioridades de los países emergentes es proporcionar a su población una infraestructura lo suficientemente robusta para permitir un acceso a Internet de alta calidad.

La globalización y la conexión constante entre todos los puntos del planeta provoca una interacción entre personas y sistemas informáticos que deja una cantidad de datos tan grande, que en ocasiones procesarla para obtener información es un reto muy exigente.

Sin embargo, procesar los datos es una de las únicas maneras de obtener información valiosa sobre, por ejemplo, el rendimiento de determinada página web, sistema informático, herramienta o servicio.

Una de las maneras en las que se generan datos en la Web durante las interacciones mencionadas es cuando un internauta responde a un cuestionario vía web. Estos cuestionarios suelen tener preguntas fijas cuyas respuestas son diferentes grados de satisfacción. En ocasiones también se permite al usuario opinar libremente, usando el lenguaje natural, aunque en este caso procesar el texto para obtener información útil es más complicado.

1.2 Caso

En la URJC, como en el resto de organizaciones, se necesitan opiniones de las personas que hacen uso de ella. Estas opiniones ayudan a conocer el grado de satisfacción de diferentes colectivos de la comunidad universitaria, así como conocer qué aspectos se pueden mejorar.

Los profesores son evaluados por sus alumnos cuando la asignatura que les imparten va a terminar. La forma de evaluarlos es mediante un cuestionario que dispone de preguntas fijas y comunes para todos los profesores y asignaturas. La respuesta a elegir por el alumno es un número entre 1 y 5 que representa su grado de conformidad.

Las preguntas evalúan aspectos como la calidad de las clases del profesor, si se ajusta a los contenidos de la Guía Docente, su puntualidad, etc. Una vez respondidas, como están en formato físico, una máquina debe tratarlas para pasar los datos a un sistema informático, donde después se generarán informes y estadísticas para los profesores. Los profesores pueden acceder a ellos desde el Campus Virtual, una herramienta dentro de la página web de la universidad.

1.3 Problemas

La forma de evaluación actual tiene importantes limitaciones. Por un lado, dado que las preguntas son las mismas para todo profesor y asignatura, los profesores no pueden personalizarlas y por tanto enfocar la evaluación hacia temas que ellos deseen. Por otro lado,

los alumnos sólo pueden opinar sobre las preguntas impuestas, no pueden comentar temas que no lleguen a abarcar las preguntas.

Como la evaluación es entregada a los alumnos una vez, en los últimos días de clase, estos sólo tienen una oportunidad para opinar, es decir, no pueden hacerlo durante el resto del curso.

El auge de los sistemas informáticos facilita la gestión de la información, permite ahorrar recursos y ayuda enormemente en tareas repetitivas. El uso de papel para los cuestionarios y el control de los mismos se pueden reducir o incluso eliminar.

1.4 Posibles soluciones

Existen algunas soluciones para los problemas anteriormente citados, algunas usadas actualmente.

Los profesores pueden pasar sus propios cuestionarios con las preguntas que ellos elijan, pero el procesamiento de las respuestas tendrán que hacerlo manualmente, y es algo muy costoso.

Para evitar el trabajo manual, pueden modificar las preguntas pero no la forma de respuesta: los alumnos siguen eligiendo la satisfacción del 1 al 5 y las máquinas leen los cuestionarios y pasan los datos al sistema informático actual. Esto es una solución parcial, ya que los alumnos siguen valorando sólo algunos temas, los que llegan a abarcar las preguntas de los profesores, pero no tienen una opción para opinar de lo que quieran.

Los profesores pueden añadir un campo de texto, en una hoja separada, donde los alumnos podrán opinar de lo que quieran. Sin embargo, el sistema informático actual no está preparado para leer comentarios, ni procesar lenguaje humano, por lo tanto necesitarían de nuevo volver al trabajo manual.

Se podría modificar el sistema informático actual para que procese el lenguaje natural, pero aún existiría el problema de que los alumnos sólo pueden opinar una vez, cuando el cuestionario es entregado.

1.5 Solución propuesta

El trabajo desarrollado consiste en analizar el entorno y posibilidades, preparar una solución y llevarla a cabo. La solución propuesta consiste en una aplicación web, que hemos denominado OMIT (*Opinion Mining In Teaching*) y que a grandes rasgos consta de las siguientes características y objetivos.

Toda la gestión debe realizarse desde una página web, conectado a Internet. Esto soluciona los problemas del papel y los trabajos manuales. Así podrá estar todo informatizado y los alumnos podrán opinar siempre que lo deseen.

Habrán unas preguntas por defecto, pero los profesores podrán personalizarlas añadiendo o quitando para adaptarlas a su asignatura y gustos.

En la página web donde el alumno realizará la evaluación habrá un campo de texto donde podrá opinar sobre cualquier tema. Estos comentarios deben ser procesados para obtener información relevante, para que se puedan generar gráficos y los profesores puedan hacer búsquedas personalizadas.

La página web dispondrá de un apartado para la gestión de los datos como pueda ser la información de los alumnos, profesores, asignaturas, titulaciones, cursos, etc.

1.6 Procesamiento del lenguaje

Los comentarios que envían los alumnos deben ser procesados para obtener de ellos información relevante y pueda ser mostrada a los profesores. Por un lado se aplicará un análisis de sentimientos con el objetivo de conocer la polaridad del comentario: si es positivo, negativo o neutral. Por otro lado, a los comentarios se les aplicará un algoritmo de *clustering* para poder conocer qué temas tratan los alumnos en sus opiniones. Ambos procesamientos se aplican en diferentes etapas y son explicados profundamente en capítulos posteriores.



Figura 1.1: Diagrama de actividad del alumno del envío de comentarios.

En la Figura 1.1 se ilustra el diagrama de actividad del alumno para el envío de comentarios. Utilizando un dispositivo con un navegador web se conectará a la página web donde se encuentre alojado el sistema. Le pedirá las credenciales de acceso, cuando el usuario las envíe, el servidor hará una petición a la base de datos MySQL (donde se encuentra la información de los alumnos) para comprobar si los datos enviados son correctos. En caso de que el proceso haya transcurrido correctamente, el alumno podrá responder a las preguntas del profesor y enviar su comentario. El servidor procesará el texto del comentario para realizar un análisis de sentimientos e insertará el resultado junto a otros datos necesarios en la base de datos Lucene.

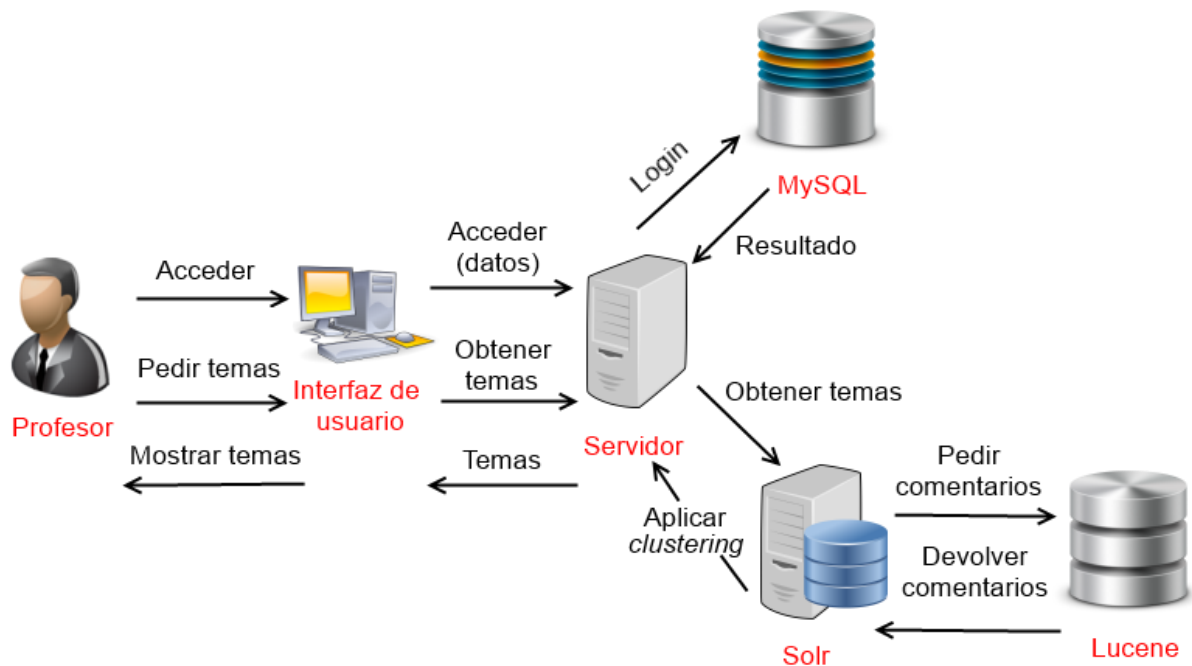


Figura 1.2: Diagrama de actividad del profesor para pedir temas de los comentarios.

El análisis de sentimientos, por lo tanto, se aplica sólo una vez por cada comentario, cuando el alumno lo envía. Sin embargo, el algoritmo de *clustering* es ejecutado cada vez que el profesor pide los temas de los comentarios de sus asignaturas (ver Figura 1.2). Al igual que el alumno, cuando el profesor accede a la página web se le piden sus credenciales. Después podrá seleccionar una asignatura y pedir los temas de los comentarios de los alumnos. El servidor pedirá a Solr los temas, quién los obtendrá pidiendo a Lucene los comentarios y aplicando un algoritmo de *clustering* sobre ellos.

El resto de la presente memoria se estructura en diferentes capítulos como sigue: en el Capítulo 2 se presentan la metodología y tecnologías utilizadas; en el Capítulo 3 se justifican todas las decisiones tomadas con respecto al sistema desarrollado; en el Capítulo 4 se detalla la descripción informática del sistema; en el Capítulo 6 se enumeran las conclusiones y se dictan algunas líneas que pueden seguirse en un futuro para continuar con este trabajo.

2. Metodología y tecnologías

En este capítulo se explica la metodología escogida para el desarrollo de Trabajo de Fin de Grado y las tecnologías usadas a lo largo de la implementación.

2.1 Metodología escogida

Una metodología puede definirse como un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a producir nuevo software. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Captura y presenta las mejores prácticas que el estado actual de la tecnología permite. En consecuencia, reduce el riesgo y hace el proyecto más predecible. El efecto global es el fomento de una visión y una cultura comunes [1].

Programación Extrema es la metodología utilizada en la realización del proyecto. Se trata de un enfoque de la Ingeniería del Software, formulado por Ken Beck, considerado como el más destacado de los procesos ágiles. Los cuatro valores en los que se basa esta metodología son la "realimentación continua" entre el cliente y el equipo, la "comunicación" entre todos los participantes, la "simplicidad" en las soluciones implementadas y la "valentía" a la hora de afrontar los cambios [2].

Al tratarse de un Trabajo de Fin de Grado, los tutores ejercerán el rol de cliente durante todo el proceso y el alumno representará el rol de desarrollador.

Algunas de las prácticas que propone esta metodología y que han sido utilizadas en este trabajo son:

- **El juego de la planificación** (*The Planning Game*): el cliente debe decidir acerca de algunos aspectos, como son el establecimiento de prioridades, el contenido de las distintas versiones o las fechas en que tendrán que ser entregadas las versiones más importantes. El equipo, en cambio, debería tomar decisiones en cuanto a la organización del trabajo y la planificación detallada de este.

El tutor detalló los requisitos más importantes que tenía que cumplir el sistema desarrollado: se debía implementar un sistema desde el cual los alumnos pudieran valorar a los profesores respondiendo a preguntas y enviando comentarios. Además los profesores podrían ver estadísticas y gráficas de las valoraciones. Para llevar a cabo esta función se debía procesar los comentarios de los alumnos aplicando técnicas de Procesamiento del Lenguaje Natural. El cliente (tutor) estableció los requisitos principales, pero ha sido el desarrollador (alumno) el que ha organizado y planificado el trabajo de la manera que ha creído más conveniente.

- **Entregas pequeñas** (*Small Releases*): el sistema se ha implementado de forma incremental, produciendo rápidamente versiones operativas del sistema aunque no contaran con todas las funcionalidades.
- **Metáfora** (*Metaphor*): el sistema y sus diferentes partes se definen mediante metáforas compartidas entre el cliente y los desarrolladores. En este caso, las metáforas utilizadas han sido "Página web", "Análisis de sentimientos" o "Algoritmo de *clustering*".

- **Diseño simple** (*Simple Design*): se ha intentado no complicar el diseño durante el desarrollo, con vistas a poder adaptar fácilmente los cambios.
- **Pruebas** (*Testing*): en cada reunión con el tutor se repasaban y detallaban los requisitos que tenía que cumplir la futura entrega, junto con un conjunto de pruebas para validar el correcto funcionamiento.
- **Refactorización** (*Refactoring*): el código ha ido reestructurándose a lo largo de todo el proceso con intención de simplificarlo y facilitar posteriores cambios.
- **Programación por Parejas** (*Pair Programming*): esta práctica no resulta aplicable en este caso puesto que sólo existe un desarrollador.
- **Propiedad colectiva del código** (*Collective Ownership*): no aplicable por existir un único desarrollador.
- **Integración Continua** (*Continuous Integration*): cada parte del código se ha integrado en el sistema una vez que estaba lista. De esta manera se facilita la evolución de los requisitos iniciales.
- **40 horas por semana** (*40-Hour Week*): esta práctica establece que no se deben trabajar horas extras durante periodos largos, ya que el agotamiento y la presión pueden ser contraproducentes para el proyecto en general. Aunque este trabajo es un Trabajo de Fin de Grado, y 40 horas semanales es una jornada excesiva para ello, se ha mantenido una regularidad en el tiempo empleado para llevarlo a cabo.
- **Cliente in-situ** (*On-Site Customer*): el tutor ha estado disponible todo el tiempo para proporcionar el feedback necesario y resolver dudas.
- **Estándares de programación** (*Coding Standards*): se ha intentado implementar la aplicación siguiendo siempre la misma línea e intentando producir un código fácilmente legible.



Figura 2.1: Prácticas de la Programación Extrema.

Aunque la mayoría de estas prácticas ya habían sido propuestas antes en Ingeniería del *Software*, XP las integra de una forma efectiva, de manera que se obtiene el mayor rendimiento cuando estas se aplican de manera conjunta y equilibrada, apoyándose unas en

otras. En la Figura 2.1, extraída de [2], se aprecian las relaciones de equilibrio existentes entre las diferentes prácticas de esta metodología.

2.2 Tecnologías utilizadas

Para el desarrollo de la solución propuesta se han utilizado diferentes tecnologías, que se presentan a continuación.

PHP (*PHP Hypertext Pre-processor*) [3]: es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente para la interpretación del lado del servidor, aunque actualmente se puede utilizar desde una interfaz de línea de comandos o en la creación de otros tipos de programas. Es un lenguaje multiplataforma y de libre distribución, y permite aplicar tanto técnicas de programación estructurada como de programación orientada a objetos. La parte del servidor para mostrar e interactuar con la página web ha sido desarrollada con la versión 5.4 de este lenguaje.

MySQL [4]: es un Sistema de Gestión de Bases de Datos (SGBD) relacional de código abierto. Los datos son almacenados en tablas, entre las cuales se establecen relaciones para manejar los datos de una manera eficiente y segura. Ofrece la posibilidad de aprovechar la potencia de los sistemas multiprocesador gracias a su implementación multihilo. En este proyecto se ha utilizado la versión 5.5 para la gestión de los datos de los alumnos, profesores, titulaciones, asignaturas, cursos, etc.

HTML (*HiperText Markup Language*) [5]: es el lenguaje de marcado que predomina para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. En este proyecto se ha usado para implementar la parte cliente del portal web multilingüe.

CSS (*Cascading Style Sheets*) [6]: es un lenguaje de hojas de estilos usado para describir la presentación de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL. La presentación y forma de la página web se ha hecho con CSS.

DOM (*Document Object Model*) [7]: es una API que permite a los programas y scripts acceder dinámicamente y modificar el contenido, la estructura y el estilo de un documento HTML o XML. Los documentos se representan en forma de árbol, de manera que los elementos del documento serán nodos interconectados. Existen versiones de esta API para la mayoría de lenguajes de programación. En este trabajo se ha utilizado junto a JavaScript y AJAX.

JavaScript [8]: es un lenguaje de programación interpretado. Se utiliza principalmente en el lado del cliente implementado como parte de un navegador web permitiendo la creación de páginas dinámicas y la mejora de la interfaz de usuario. Para poder interactuar con la página web, JavaScript dispone de una implementación de la API DOM. Es utilizado en la página web para permitir al usuario un mejor uso de la misma.

jQuery [9]: es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos o hacer uso de AJAX. Es software libre y de código abierto. Esta librería es utilizada en la página web.

jQuery UI [10]: es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plug-ins, widgets y efectos visuales para la creación de páginas web.

Highcharts [11]: es una librería JavaScript que ayuda a mostrar gráficos en una página web. Se utiliza para mostrar los gráficos de las estadísticas de las respuestas de los alumnos.

AJAX (*Asynchronous JavaScript And XML*) [12]: es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente (el navegador del usuario), mientras se mantiene una comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre las páginas sin la necesidad de recargarlas, lo que deriva en un aumento de la interactividad, la velocidad y la usabilidad en las aplicaciones. AJAX combina cuatro tecnologías ya existentes, a saber: HTML y CSS, para el diseño que acompaña a la información; DOM, para mostrar e interactuar dinámicamente con la información presentada; XMLHttpRequest, para intercambiar datos de forma asíncrona con el servidor; y XML, el formato usado generalmente para la transferencia de datos solicitados al servidor. La página web se ha implementado utilizando AJAX.

JSON (*JavaScript Object Notation*) [13]: es un formato ligero para el intercambio de datos. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Debido a la ubicuidad de JavaScript en casi cualquier navegador web, JSON ha sido aceptado por parte de la comunidad de desarrolladores AJAX. Se ha utilizado para intercambio de datos con el servidor en las peticiones AJAX.

Git [14]: es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Git es utilizado, como *plug-in* en el sistema de desarrollo Netbeans [15], para mantener el control de versiones del proyecto en local y en el servidor remoto gratuito Github [16].

CSV (*comma-separated values*) [17]: es un formato sencillo para representar datos en forma de tabla. Cada columna se separa por comas (u otro carácter escogido), y cada fila con un salto de línea. CSV es usado para exportar e importar información en el sistema desarrollado.

Lucene [18]: es una API de código abierto para la recuperación de información mediante la indexación de los datos que almacena. Dispone de herramientas para procesar el lenguaje humano. En este proyecto es usado para guardar los comentarios de los usuarios y después poder agruparlos por temática mediante *clustering*.

Solr [19]: es un motor de búsqueda basado en Lucene con APIs en HTTP, XML y JSON. Actúa como servidor para el intercambio de información entre el navegador web y la base de datos de Lucene.

3. Propuesta

En este capítulo se describe el sistema de implementado, justificando cada una de las decisiones tomadas para su desarrollo.

3.1 Sistema y servidor para la página web

El proyecto debe permitir que alumnos puedan opinar sobre sus profesores y asignaturas cuando lo deseen y utilizando los mínimos recursos necesarios. Por otro lado, los profesores han de ver los comentarios y evaluaciones que sus alumnos realizaron.

Dado el auge de Internet y teniendo en cuenta que existe software libre, gratuito y de calidad para implementar un servidor web, el sistema se ha implementado para poder ser utilizado con un navegador web, visitando una página web.

Una página web moderna necesita: un servidor web HTTP que controle las peticiones del cliente, una base de datos para guardar la información necesaria y un lenguaje de programación del lado del servidor que interaccione con la base de datos y genere dinámicamente la salida HTML para el usuario. Volviendo al software libre y gratuito, una de las mejores opciones es usar el servidor HTTP de Apache, el lenguaje de programación PHP para el lado del servidor, y la base de datos MySQL desarrollada por Oracle.

En el lado del cliente, la página web necesita interaccionar con el documento HTML y el árbol DOM y ejecutar peticiones AJAX. La mayoría de los navegadores interpretan código JavaScript para lograr estas funciones. Sin embargo, utilizar directamente JavaScript puede llegar a ser algo engorroso, por este motivo en el proyecto se ha utilizado la librería jQuery, que simplifica el lenguaje y dispone de un framework fácil de usar y completo.

3.2 Procesamiento del Lenguaje Natural

Para hacer la aplicación más útil, los comentarios de los alumnos se deben procesar para poder generar estadísticas para los profesores y ver si los comentarios son positivos o negativos. Aunque procesar el lenguaje humano no es una tarea sencilla, existen herramientas y algoritmos que ayudan en la tarea.

La aplicación desarrollada realiza dos tipos de procesamientos sobre los comentarios de los alumnos. Por un lado se aplica un algoritmo de agrupamiento (*clustering*) para relacionar unos comentarios con otros y saber qué temas tratan los alumnos. Por otro lado es necesario llevar a cabo un análisis de sentimientos para saber si los comentarios son positivos, negativos o neutrales. Uniendo ambos tipos de procesamiento en el sistema, el profesor es capaz de saber la opinión general de los alumnos sobre un determinado tema.

3.2.1 *Clustering*

El *clustering* es un proceso de aprendizaje no supervisado que consiste en organizar una colección de objetos en diferentes grupos o *clusters*, de tal manera que los objetos del mismo *cluster* sean lo más similares posible entre sí, manteniendo a su vez el menor grado de similitud posible con respecto a los objetos de otros *clusters* [20].

Para el sistema desarrollado, los alumnos van opinando sobre diferentes temas durante todo el cuatrimestre, y para ver sobre qué temas han opinado se utiliza *clustering*, agrupando las opiniones que sean similares.

Uno de los algoritmos más utilizados para agrupar textos es STC (*Suffix Tree Clustering*). Mediante este algoritmo la aplicación es capaz de agrupar comentarios de los alumnos para posteriormente deducir palabras claves entre ellos y por tanto conocer temas de los que hablan (ver Figura 3.1).

Los motivos de escoger este algoritmo de *clustering* son dos principalmente. En primer lugar es un algoritmo que obtiene buenos resultados cuando los textos a agrupar son textos cortos, que es el caso de las opiniones de los alumnos. Y en segundo lugar es un algoritmo que no necesita como dato de entrada el número de grupos final, y este dato no lo tenemos porque a priori no se sabe sobre qué temas van a opinar los alumnos.

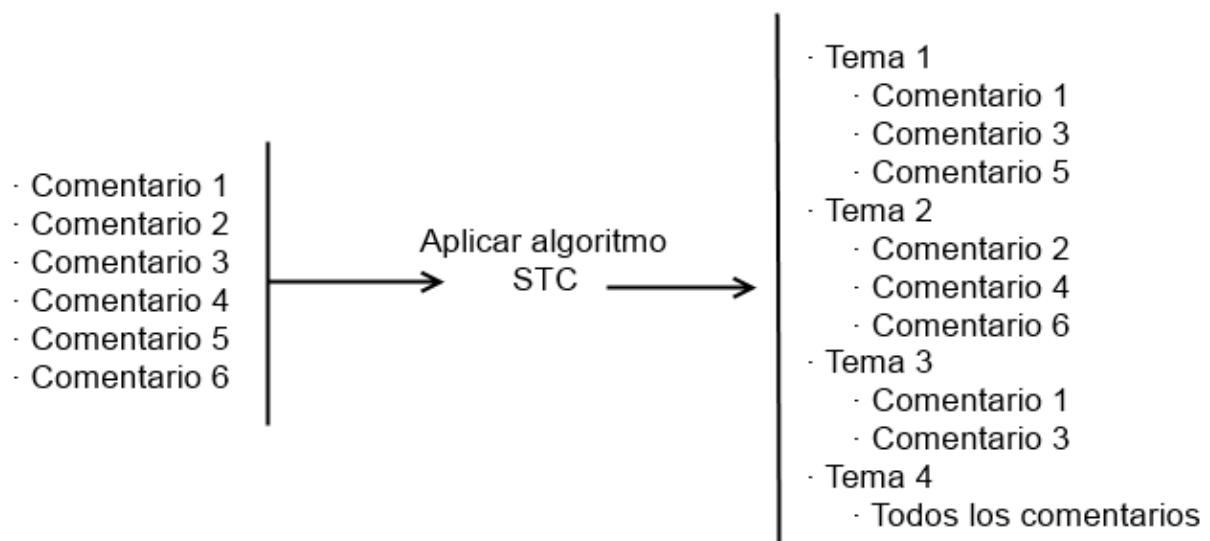


Figura 3.1: Comentarios agrupados aplicando sobre ellos el algoritmo de *clustering* STC.

Existe una implementación del algoritmo STC escrita en Java, como extensión dentro del motor de *clustering* Carrot2 [21]. En el lado del servidor de la página web se utiliza PHP, y es posible ejecutar archivos JAR (*Java ARchive*) con PHP, sin embargo existe la base de datos Lucene, implementada en Java con parte de Carrot2 integrado y en concreto el algoritmo STC que facilita el *clustering* mediante búsquedas a la base de datos. Es decir, se podría extraer el algoritmo STC de Carrot2, crear un JAR, con PHP ejecutar una consulta (*select*) a MySQL para obtener los comentarios y pasárselos al JAR. Sin embargo, esto es un proceso lento, con Lucene se puede ejecutar una consulta a su base de datos y directamente pedir que, sobre los resultados, aplique el algoritmo STC y devuelva los temas de los comentarios. Ya que Lucene está implementado en Java, en el proyecto se utiliza Solr, que añade una capa superior, se ejecuta como servidor HTTP y acepta un intercambio de datos e interacción mediante el formato JSON. Así pues, desde PHP se ejecuta la consulta a Lucene mediante Solr y este devuelve la respuesta en JSON.

3.2.2 Análisis de sentimientos

El análisis de sentimientos es una tarea del PLN que identifica opiniones relacionadas con un objeto [22].

En el sistema propuesto, además de conocer los temas de los que opinan los alumnos en sus comentarios, también es útil saber si los comentarios son positivos, negativos o neutrales. Existen algoritmos de análisis de sentimientos implementados en diferentes lenguajes de programación. La mayoría de ellos se apoyan sobre un diccionario de palabras. Cada palabra tiene un peso asociado, que describe su magnitud; y un tipo, que dependiendo cuál sea (positivo, negativo, modificador...) afecta de una manera u otra en el comentario.

Por ejemplo, la palabra “malo” es negativa y podría tener un peso de 0.6 sobre un total de 1. Por otro, la palabra “maravilloso” es positiva y podría tener un peso de 0.8. Si la suma de pesos de las palabras positivas es mayor que la suma de pesos de palabras negativas y además supera un determinado umbral, entonces se etiqueta el comentario como positivo (o viceversa). En caso de que no supere el umbral el comentario será neutral.

También existen palabras como “muy”, que aumentan el peso de la palabra que le sucede. Así, “malo” es menos negativo que “muy malo”. La palabra “no”, sin embargo, disminuye o incluso cambia el tipo de la palabra a las que le suceden: “es malo” es negativo y “no es malo” podría ser positivo o neutral.

Si se profundiza en el análisis, es necesario tener en cuenta otras características como controlar la ironía o el sarcasmo, modificar el valor de palabras alargadas (“es muy lentoooooooo”), tener un diccionario de frases hechas o refranes, ser capaz de evaluar textos con faltas de ortografía o palabras escritas sin todas las letras (“pra mi k no save esplicr”) e incluso dar un valor a los emoticonos (“:-D”).

El problema de los algoritmos de análisis de sentimientos existentes es que no proporcionan un diccionario español con pesos. Por este motivo, en el proyecto se ha desarrollado un algoritmo simple, y se ha usado un diccionario en español que define únicamente el tipo de palabra que es (positiva, negativa o modificador). El algoritmo es explicado en profundidad en la sección 4.2.4.

3.3 Bases de datos: Lucene y MySQL

La página web guardará los comentarios y las respuestas a las preguntas que los profesores impongan en la base de datos de Lucene, ya que mediante Solr se pueden ejecutar consultas. Además, que sobre los resultados que devuelva se aplicará el algoritmo de *clustering* STC para conocer los temas que se tratan en los comentarios.

Además de lo anterior es necesario guardar otros tipos de datos: las titulaciones y cursos que existen en la universidad, las asignaturas, los alumnos que están matriculados en ellas, los profesores que las imparten y las preguntas personalizadas que tiene cada profesor en cada una de sus asignaturas. Estos datos se guardarán en la base de datos MySQL, ya que se comunica gratamente con el lenguaje de programación PHP utilizado en el servidor.

La estructura de las tablas utilizadas tanto en MySQL como en Lucene se describe en las secciones 4.2.2 y 4.2.3 respectivamente.

3.4 Página web

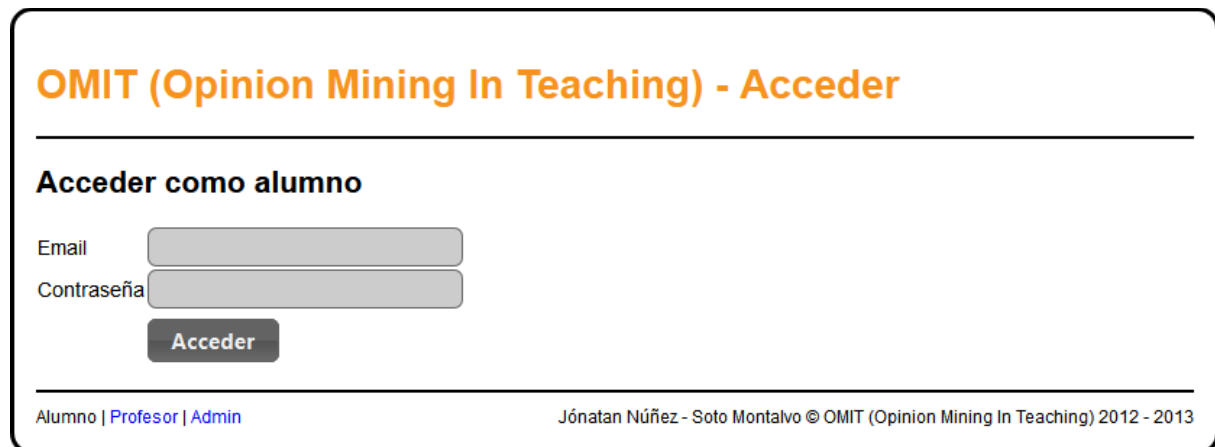
Habrà tres tipos de perfiles que usará la aplicación: el alumno, que podrá comentar y responder a las preguntas del profesor; el profesor, que podrá ver los comentarios y estadísticas de las evaluaciones de los alumnos; y el administrador, que podrá gestionar las

titulaciones, cursos y asignaturas existentes además de poder dar de alta a nuevos profesores en el sistema.

A continuación se describen las diferentes vistas de la página web.

3.4.1 Acceder

Todo el que visita la página web necesita un usuario y contraseña para poder acceder, como se puede ver en la Figura 3.2. El perfil de administrador, cuando da de alta a los profesores, elige la contraseña que les asignará. Los alumnos son dados de alta por los profesores subiendo a la página web un archivo Excel que pueden encontrar en el “portal servicios” de la URJC y que contiene los datos de los alumnos de sus asignaturas. Cuando los profesores suben este archivo, la contraseña de los alumnos es su número de DNI.



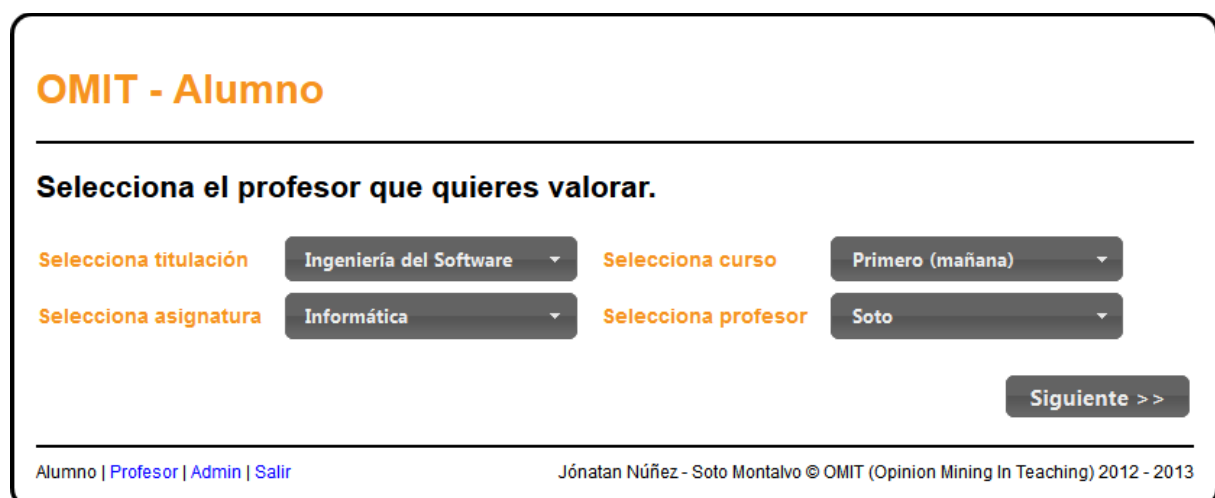
The screenshot shows the login interface for the OMIT system. At the top, the title "OMIT (Opinion Mining In Teaching) - Acceder" is displayed in orange. Below it, the section "Acceder como alumno" is highlighted. There are two input fields: "Email" and "Contraseña", both with gray borders. A dark gray button labeled "Acceder" is positioned below the password field. At the bottom left, there are links for "Alumno", "Profesor", and "Admin", with "Profesor" and "Admin" in blue. At the bottom right, the copyright notice "Jónatan Núñez - Soto Montalvo © OMIT (Opinion Mining In Teaching) 2012 - 2013" is visible.

Figura 3.2: Página de acceso para los usuarios.

Dependiendo del rol con el que se quiera acceder, el usuario puede hacer click en cada uno de ellos, situados en la parte inferior izquierda de la pantalla.

3.4.2 Alumno

Una vez acceda el alumno, podrá elegir la asignatura y profesor de los que quiere opinar (ver Figura 3.3).



The screenshot shows the selection interface for a student. The title "OMIT - Alumno" is at the top in orange. Below it, the instruction "Selecciona el profesor que quieres valorar." is displayed. There are four selection options, each with a label in orange and a dropdown menu: "Selecciona titulación" (Ingeniería del Software), "Selecciona curso" (Primero (mañana)), "Selecciona asignatura" (Informática), and "Selecciona profesor" (Soto). A dark gray button labeled "Siguiente >>" is located at the bottom right. At the bottom left, there are links for "Alumno", "Profesor", "Admin", and "Salir", with "Profesor", "Admin", and "Salir" in blue. At the bottom right, the copyright notice "Jónatan Núñez - Soto Montalvo © OMIT (Opinion Mining In Teaching) 2012 - 2013" is visible.

Figura 3.3: El alumna selecciona asignatura y profesor a evaluar.

En la figura Figura 3.4 puede observarse cómo el alumno tiene la opción de no responder a las preguntas del profesor. Esto es porque el alumno tiene acceso durante todo el año a la aplicación, puede no querer responder a las preguntas y únicamente escribir el comentario.

OMIT - Alumno

Preguntas personalizadas

¿Desea valorar diferentes temas acerca del profesor y la asignatura? **Sí** No

- 1 El profesor da a conocer a los alumnos la guía docente de la asignatura a principios de curso.
Satisfecho
- 2 El profesor ha informado claramente sobre los criterios de evaluación de la asignatura.
Satisfecho
- 3 El profesor ha establecido algún sistema de comunicación y tutoría.
Muy satisfecho
- 4 El profesor está disponible para atender a los alumnos.
Satisfecho
- 5 El profesor aclara adecuadamente las dudas de las distintas actividades propuestas en la asignatura.
Muy satisfecho
- 6 El profesor utiliza un material (texto, presentaciones, vídeos, videoconferencias, ...) que facilita el aprendizaje de la asignatura.
Satisfecho
- 7 Las actividades docentes se ajustan a los objetivos, contenidos y metodología especificada en la guía docente de la asignatura.
Satisfecho
- 8 El desarrollo de la asignatura me permite un seguimiento y aprendizaje adecuados.
Muy satisfecho
- 9 Teniendo en cuenta todos los aspectos mencionados, estoy satisfecho/a con la labor que desarrolla el profesor.
Satisfecho

<< Anterior

Siguiente >>

Alumno | [Profesor](#) | [Admin](#) | [Salir](#)

Jónatan Núñez - Soto Montalvo © OMIT (Opinion Mining In Teaching) 2012 - 2013

Figura 3.4: Página del alumno respondiendo a las preguntas del profesor.

Una vez haya acabado y pulse en el botón “Siguiente” podrá escribir el comentario para opinar sobre el tema que desee. En la Figura 3.5 se muestra un ejemplo de comentario que un alumno podría realizar.

OMIT - Alumno

Puedes incluir un comentario sobre cualquier aspecto relacionado con las asignatura o el profesor.

Comentario

Me encanta la profesora, ha sido de los mejores que he tenido. He aprendido un montón con ella y explica muy bien. Lo único que se puede echar en cara es que es un poco dura corrigiendo los exámenes, jejeje. Pero es mejor eso que sea blanda, porque así aprendemos más.

¡Gracias por todo!

Enviar

< < Anterior

Alumno | [Profesor](#) | [Admin](#) | [Salir](#)

Jónatan Núñez - Soto Montalvo © OMIT (Opinion Mining In Teaching) 2012 - 2013

Figura 3.5: Un alumno escribiendo un comentario para el profesor.

3.4.3 Profesor

Cuando accede el profesor, debe seleccionar una de las asignaturas que imparte.

OMIT - Profesor

Informática - Primero (mañana)

Preguntas

Comentarios

Cargar alumnos

Pregunta 1

El profesor da a conocer a los alumnos la guía docente de la asignatura a principios de curso.

Pregunta 2

El profesor ha informado claramente sobre los criterios de evaluación de la asignatura.

Pregunta 3

El profesor ha establecido algún sistema de comunicación y tutoría.

Pregunta 4

El profesor está disponible para atender a los alumnos.

Pregunta 5

El profesor aclara adecuadamente las dudas de las distintas actividades propuestas en la asignatura.

Pregunta 6

El profesor utiliza un material (texto, presentaciones, vídeos, videoconferencias, ...) que facilita el aprendizaje.

Pregunta 7

Las actividades docentes se ajustan a los objetivos, contenidos y metodología especificada en la asignatura.

Pregunta 8

El desarrollo de la asignatura me permite un seguimiento y aprendizaje adecuados.

Pregunta 9

Teniendo en cuenta todos los aspectos mencionados, estoy satisfecho/a con la labor que desarrolla el profesor.

Pregunta 10

Estadísticas

Guardar

Se perderán las valoraciones actuales si se modifican las preguntas

Alumno | [Profesor](#) | [Admin](#) | [Salir](#)

Jónatan Núñez - Soto Montalvo © OMIT (Opinion Mining In Teaching) 2012 - 2013

Figura 3.6: Un profesor editando las preguntas para la asignatura “Informática” de primer curso.

Además de editar las preguntas asociadas a cada una de sus asignaturas (ver Figura 3.6), también puede obtener estadísticas por meses de las respuestas de sus alumnos, como puede observarse en la Figura 3.7.

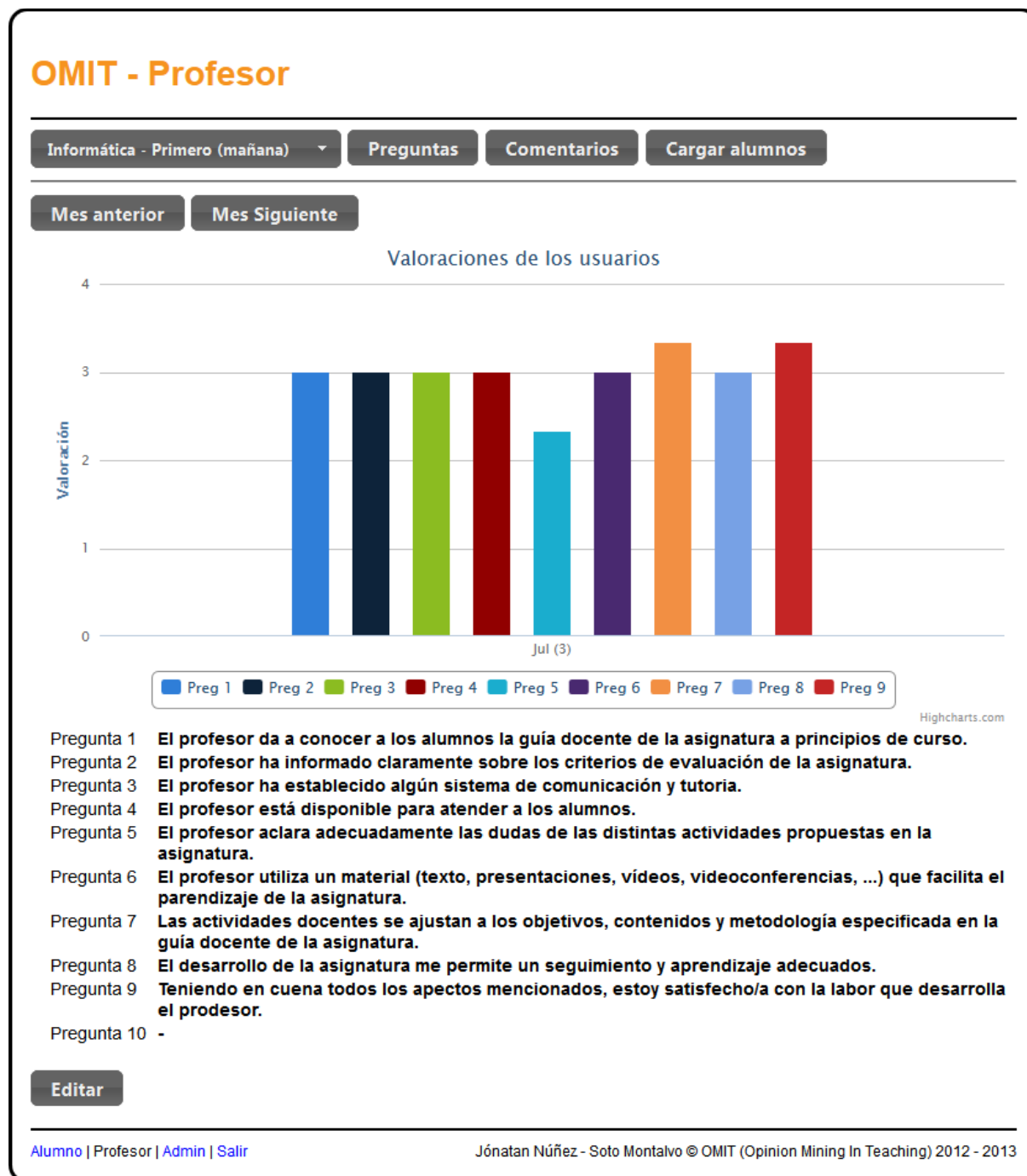


Figura 3.7: Gráfico con 3 valoraciones de los alumnos realizadas en julio

Los profesores, desde el “portal servicios” de la URJC pueden descargar ficheros en formato Excel con los datos de sus alumnos. La opción de “Cargar alumnos” que se observa como botón en las vistas del profesor, permitirá cargar el fichero Excel para dar de alta a los alumnos en la asignatura que el profesor desee.

Si pulsa sobre el botón “Comentarios” podrá ver los comentarios que ha recibido con el análisis de polaridad realizado sobre cada uno de ellos.

OMIT - Profesor

Informática - Primero (mañana) Preguntas Comentarios Cargar alumnos

Buscar... Fecha 01/09/2012 - 01/09/2013 Opinión Todas

Buscar comentarios

Mostrando página 1 de 1 - 3 comentarios totales

Buscar temas Estadísticas

Estadísticas
 Opinión Neutral: 0
 Opinión Positiva: 1
 Opinión Negativa: 2

Me encanta la profesora, ha sido de los mejores que he tenido. He aprendido un montón con ella y explica muy bien. Lo único que se puede echar en cara es que es un poco dura corrigiendo los exámenes, jejeje. Pero es mejor eso que sea blanda, porque así aprendemos más.
 ¡Gracias por todo!
 Opinión: positiva

Los exámenes han sido demasiado difíciles para el temario que hemos dado. Además casi no hemos hecho ejercicios, y los pocos que hemos hechos no son del mismo tipo que en el examen.
 Por otro lado, no se ha ajustado a la guía docente que está publicada en la página web de la URJC.
 Opinión: negativa

Eres horrible, llegas a clase y te pones a leer transparencias. Para eso lo hago yo en mi casa. Perdona que escriba así, pero me pregunto dónde te han dado el título.
 Opinión: negativa

Alumno | Profesor | Admin | Salir Jónatan Núñez - Soto Montalvo © OMIT (Opinion Mining In Teaching) 2012 - 2013

Figura 3.8: Listado de comentarios realizados por los alumnos a un profesor determinado.

El botón “Buscar temas” (ver Figura 3.8) aplicará el algoritmo STC sobre los comentarios para mostrar los temas de los que hablan los alumnos.

OMIT - Profesor

Informática - Primero (mañana) Preguntas Comentarios Cargar alumnos

Buscar... Fecha 01/09/2012 - 01/09/2013 Opinión Todas

Buscar comentarios

prácticas horario examen guía docente difícil

Alumno | Profesor | Admin | Salir Jónatan Núñez - Soto Montalvo © OMIT (Opinion Mining In Teaching) 2012 - 2013

Figura 3.9: Ejemplo de temas de los que pueden opinar los alumnos en sus comentarios.

La Figura 3.9 muestra los temas de los comentarios de los alumnos aplicando el algoritmo de *clustering* STC.

3.4.4 Administrador

El administrador es el gestor de la información de la universidad. Puede descargar los datos e importar nuevos y además borrar los datos del sistema para dejarlo limpio. Las tres opciones se pueden ver en la Figura 3.10.



Figura 3.10: Cuando accede el administrador, puede exportar información, importarla o borrar los datos del sistema.

El archivo que descarga en formato Excel está organizado en tablas, para que sean fácilmente editables. Las tablas son las siguientes:

titulaciones	
id	nombre
1	Ingeniería del Software
2	Periodismo

Tabla 3.1: Tabla de titulaciones de la universidad

cursos		
id	nombre	titulacion
1	Primero (mañana)	1
2	Segundo (tarde)	1
3	Tercero (tarde)	2
4	Cuarto	2

Tabla 3.2: Tabla de cursos correspondidos con una titulación.

asignaturas		
id	nombre	curso
1	Informática	1
2	Paradigmas de la programación	1
3	Java	2

Tabla 3.3: Tabla de las asignaturas, cada una correspondida con un curso.

profesores					
id	nombre	email	apellido1	apellido2	password
1	Soto	soto@urjc.es	Montalvo	Hernaz	09aba14b14c1ddb10346068577d21b6b
2	Pedro	pedro@urjc.es	González	Prieto	a8f5f167f44f4964e6c998dee827110c
3	Enrique	enrique@urjc.es	Bumbury		19d867e6ef679a0e9529f69b40fa456c

Tabla 3.4: Tabla con los datos de los profesores. El campo de contraseña se encuentra codificada con el algoritmo MD5.

profesores_asignaturas	
profesor	asignatura
1	1
2	1

Tabla 3.5: Tabla de asignación entre un profesor y una asignatura

En cualquier momento, el administrador puede añadir, modificar o eliminar registros y cargar los nuevos datos pulsando sobre el botón “Subir datos en CSV”.

La Tabla 3.1 guarda las titulaciones que existen en la universidad. Cada titulación tiene un nombre y tiene asignado un identificador único (id). Este identificador se utilizará para la Tabla 3.2 de cursos: cada curso se encuentra en una titulación. Además, cada curso dispone de un nombre y un id. En el caso de las asignaturas (ver Tabla 3.3), necesitan otra vez un nombre, un id, y una referencia a un curso. La Tabla 3.4 de profesores necesita un id, nombre, email y apellidos así como una contraseña codificada con el algoritmo MD5. Para asignar un profesor a una asignatura que imparta, se utilizará la Tabla 3.5 donde se colocará en cada fila un id de profesor y otro de asignatura por cada asignación.

4. Descripción informática

En este capítulo se detalla la descripción informática del sistema implementado. Además, se aportan algunos artefactos que se consideran necesarios, tanto para el entendimiento de sistema a nivel técnico, como para una posible utilización o extensión de este en un futuro.

4.1 Modelos de casos de uso

Un caso de uso (CU) es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada CU proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Los diagramas de CU sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas.

En esta sección se aportan los modelos de CU, que además serán detallados mediante un flujo de eventos.

4.1.1 Modelo de casos de uso para acceder

En la página web existen tres roles diferentes: alumnos, profesor y administrador. Todos ellos deben utilizar sus credenciales de acceso para entrar en la página web y poder navegar por ella. La forma de acceso es común, y se ilustra en la Figura 4.1.



Figura 4.1: Modelo de casos de uso de los usuarios para acceder

Caso de uso “Acceder”

- **Actor:** usuario (alumno, profesor o administrador)
- **Descripción:** el usuario introduce las credenciales en el formulario para acceder a su vista principal en la página web. El flujo de eventos básico se presenta en la Tabla 4.1.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Entra en la página web 2. Selecciona el rol con el que desea acceder (alumno, profesor, administrador). 3. Introduce las credenciales.	4. Recoge los datos enviados por el usuario.

	5. Ejecuta una consulta a la base de datos MySQL para comprobar que los datos son correctos. 6. Inicializa la sesión del usuario con los datos correspondientes. 7. Redirige la petición HTTP a la página principal del usuario dependiendo del rol escogido.
--	---

Tabla 4.1: Flujo de eventos básico del caso de usos “Acceder”.

- **Flujo de eventos alternativos:**

- En el evento 5, los datos que envía el usuario no se corresponden con ninguno en la base de datos (usuario o contraseña incorrectos).

4.1.2 Modelo de casos de uso del alumno

En la Figura 4.2 puede verse el CU del alumno, quién podrá responder a las preguntas del profesor y escribir un comentario para la asignatura.

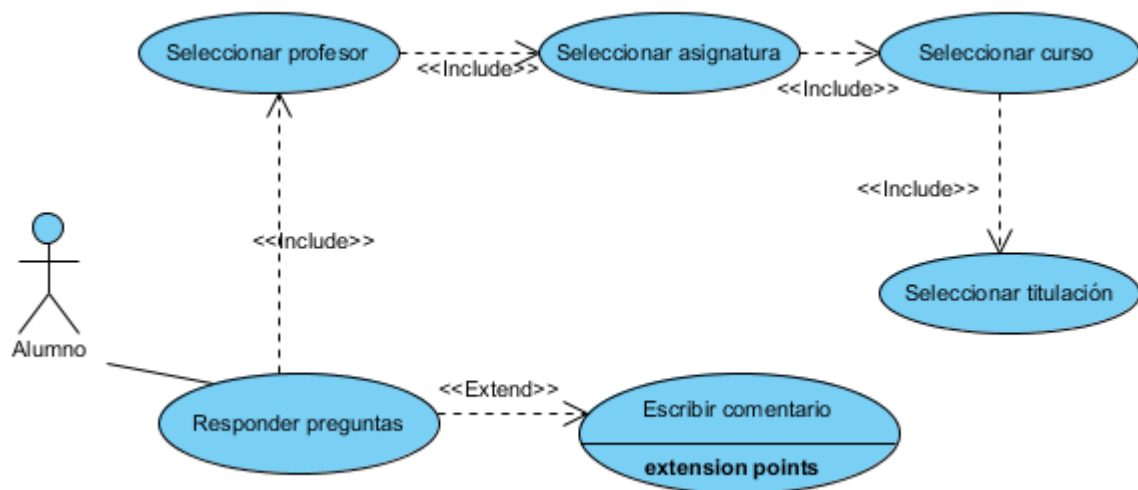


Figura 4.2: Modelo de casos de uso del alumno

Caso de uso “Alumno”

- **Actor:** alumno
- **Descripción:** el alumno, una vez ha accedido comienza a realizar la valoración a un profesor y asignatura deseados. El flujo de eventos básico se presenta en la Tabla 4.2.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Selecciona una titulación. 2. Selecciona un curso. 3. Selecciona una asignatura. 4. Selecciona un profesor. 5. Responde a las preguntas del profesor.	

6. Escribe un comentario. 7. Envía la valoración.	8. Recoge los datos enviados por el alumno. 9. Aplica el algoritmo de <i>clustering</i> STC sobre el comentario. 10. Guarda los datos en la base de datos Lucene.
--	---

Tabla 4.2: Flujo de eventos básico del caso de usos “Alumno”.

- **Flujo de eventos alternativos:**

- Desde el evento 4, puede pasar directamente al 6 sin responder las preguntas del profesor.
- Desde el evento 5, puede pasar directamente al 7 sin escribir un comentario.

4.1.3 Modelo de casos de uso del profesor

El profesor podrá realizar varias acciones en la página web. Estas acciones son ilustradas en el modelo de casos de usos del profesor (ver Figura 4.3).

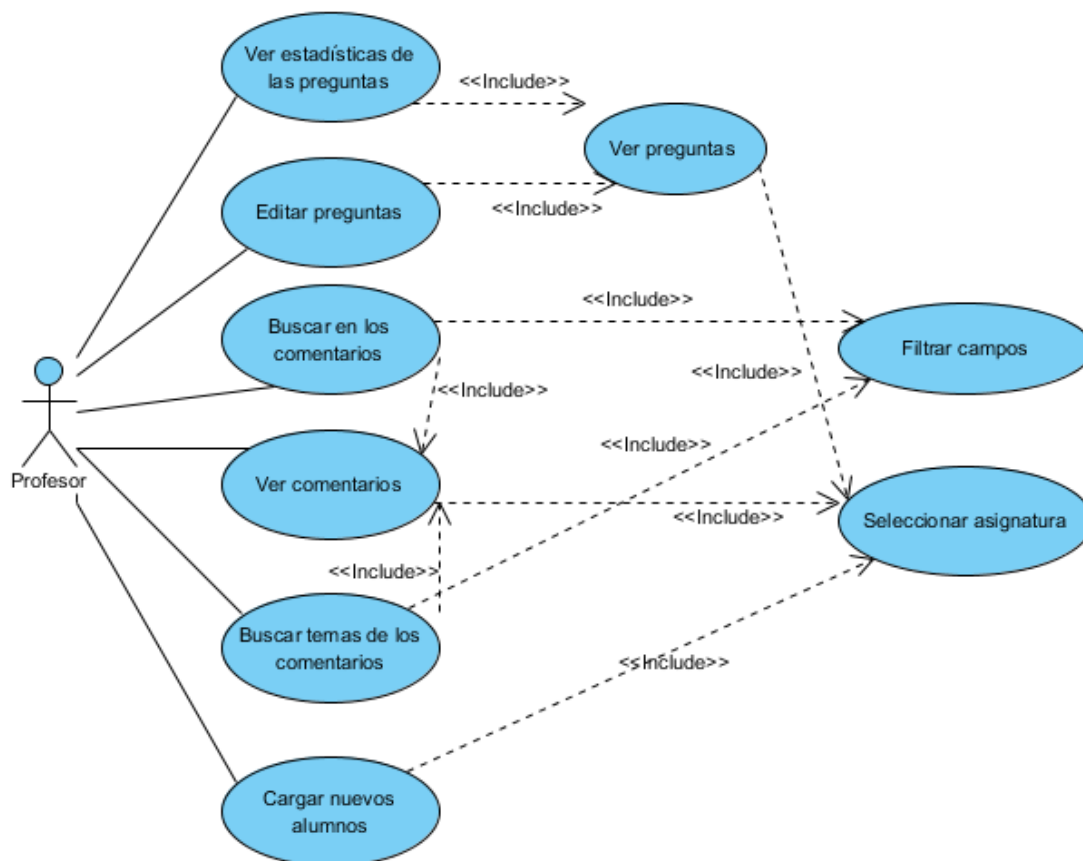


Figura 4.3: Modelo de casos de uso del profesor

A continuación se describe cada uno de los casos de uso que aparecen en la Figura 4.3.

Caso de uso “Ver comentarios”

- **Actor:** profesor

- **Descripción:** el profesor selecciona ver los comentarios de los alumnos realizar para una determinada asignatura. El flujo de eventos básico se presenta en la Tabla 4.3.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Selecciona una asignatura. 2. Pincha sobre el botón “Comentarios”.	3. Mediante Solr busca los comentarios del profesor realizados para la asignatura seleccionada. 4. Muestra los comentarios en la página web.

Tabla 4.3: Flujo de eventos básico del caso de usos “Ver comentarios”.

Caso de uso “Buscar en los comentarios”

- **Actor:** profesor
- **Descripción:** el profesor busca en los comentarios que los alumnos han hecho en su asignatura. El flujo de eventos básico se presenta en la Tabla 4.4.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Selecciona una asignatura. 2. Pincha sobre el botón “Comentarios”. 3. Filtra los campos para realizar la búsqueda deseada. 4. Pincha en el botón “Buscar”.	5. Recoge los parámetros de búsqueda del profesor establecidos. 6. Mediante Solr busca los comentarios del profesor realizados para la asignatura seleccionada introduciendo filtros. 7. Muestra los comentarios en la página web. 8. Resalta los parámetros de búsqueda seleccionados por el profesor en los nuevos comentarios.

Tabla 4.4: Flujo de eventos básico del caso de usos “Buscar en los comentarios”.

Caso de uso “Buscar temas de los comentarios”

- **Actor:** profesor
- **Descripción:** el profesor pide al sistema los temas que se tratan en los comentarios de los alumnos. El flujo de eventos básico se presenta en la Tabla 4.5.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Selecciona una asignatura. 2. Pincha sobre el botón “Comentarios”.	

<ol style="list-style-type: none"> 3. Filtra los campos para realizar la búsqueda deseada. 4. Pincha en el botón “Buscar temas”. 	<ol style="list-style-type: none"> 5. Recoge los parámetros de búsqueda del profesor establecidos. 6. Envía a Solr los filtros seleccionados para que haga una búsqueda de los comentarios y aplique el algoritmo de <i>clustering</i> STC sobre ellos para obtener los comentarios. 7. Muestra los temas que se tratan en los comentarios de los alumnos.
--	---

Tabla 4.5: Flujo de eventos básico del caso de usos “Buscar temas de los comentarios”.

Caso de uso “Ver estadísticas de las preguntas”

- **Actor:** profesor
- **Descripción:** el profesor pide al sistema las estadísticas de las respuestas del alumno así como posibles gráficas. El flujo de eventos básico se presenta en la Tabla 4.6.Tabla 4.5

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
<ol style="list-style-type: none"> 1. Selecciona una asignatura. 2. Pincha sobre el botón “Preguntas”. 5. Pincha en el botón “Estadísticas”. 	<ol style="list-style-type: none"> 3. Busca en MySQL las preguntas personalizadas del profesor. 4. Muestra las preguntas del profesor. 6. Busca en Solr las respuestas de los alumnos para la asignatura seleccionada y las procesa. 7. Envía los datos a la página web. 8. Una librería JavaScript recoge los datos y los muestra una gráfica a partir de ellos.

Tabla 4.6: Flujo de eventos básico del caso de usos “Ver estadísticas de las preguntas”.

Caso de uso “Editar preguntas”

- **Actor:** profesor
- **Descripción:** el profesor edita las preguntas asignadas a una de sus asignaturas. El flujo de eventos básico se presenta en la Tabla 4.7Tabla 4.6.Tabla 4.5

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
3. Selecciona una asignatura. 4. Pincha sobre el botón “Preguntas”.	

6. Pincha en el botón “Editar”. 7. Modifica, añade o elimina algunas de sus preguntas. 8. Pulsa sobre el botón “Guardar”.	5. Busca en MySQL las preguntas personalizadas del profesor. 6. Muestra las preguntas del profesor. 9. Recoge las nuevas preguntas establecidas por el profesor. 10. Sobrescribe las preguntas del profesor en la base de datos MySQL. 11. Elimina las respuestas de los alumnos realizadas al profesor y a la asignatura seleccionada (ya las respuestas no se corresponderán con las nuevas preguntas).
---	---

Tabla 4.7: Flujo de eventos básico del caso de usos “Editar preguntas”.

Caso de uso “Cargar nuevos alumnos”

- **Actor:** profesor
- **Descripción:** el profesor carga en el sistema los datos de nuevos alumnos matriculados en su asignatura. Los datos se encuentran en un archivo en formato CSV que pueden encontrar en el Campus Virtual. Los alumnos creados tendrán que usar su email y su DNI como contraseña para acceder al sistema. El flujo de eventos básico se presenta en la Tabla 4.8.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Selecciona una asignatura. 2. Pincha sobre el botón “Cargar alumnos”. 3. Selecciona el fichero CSV con los datos de los alumnos previamente descargados desde el Campus Virtual.	4. Procesa el archivo de entrada. 5. Inserta los alumnos no existen en la base de datos MySQL. 6. Crea las relaciones alumno-asignatura en la base de datos MySQL.

Tabla 4.8: Flujo de eventos básico del caso de usos “Cargar nuevos alumnos”.

4.1.4 Modelo de casos de uso del administrador

El rol de administrador será quien gestione los datos de la universidad: titulaciones, cursos, asignaturas y profesor. En la Figura 4.4 puede observarse el CU.

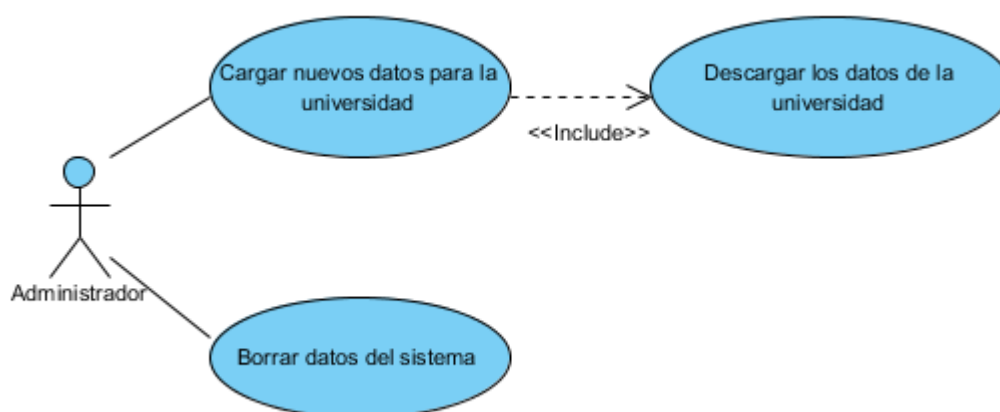


Figura 4.4: Modelo de casos de uso del administrador.

Caso de uso “Cargar nuevos datos para la universidad”

- **Actor:** administrador
- **Descripción:** el administrador carga en el sistema nuevos datos: titulaciones, cursos, asignaturas y profesores. El flujo de eventos básico se presenta en la Tabla 4.9.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Pincha sobre el botón “Descargar datos en CSV”. 5. Edita los datos utilizando un visor de archivos CSV. 6. Pincha en el botón “Subir datos en CSV”.	2. Realiza búsquedas a la base de datos de las titulaciones, cursos, asignaturas y profesores. 3. Crea un archivo en formato CSV para el usuario. 4. Envía el archivo CSV. 7. Procesa el archivo enviado por el administrador. 8. Vacía las tablas de la base de datos de las titulaciones, cursos, asignaturas y profesores. 9. Inserta en MySQL los nuevos datos encontrados en el fichero.

Tabla 4.9: Flujo de eventos básico del caso de usos “Cargar nuevos datos para la universidad”.

Caso de uso “Borrar datos del sistema”

- **Actor:** administrador
- **Descripción:** el administrador borra los datos del sistema: titulaciones, cursos, asignaturas, profesores, alumnos, comentarios y valoraciones. Sólo se conservarán los perfiles de los usuarios administradores y las preguntas por defecto de la tabla en MySQL “preguntas_default”. El flujo de eventos básico se presenta en Tabla 4.10.

Flujo de eventos básico	
Acción del Actor	Acción del Sistema
1. Pincha sobre el botón “Borrar datos de todo el sistema”. 2. Confirma la acción.	3. Vacía las tablas MySQL pertenecientes a las titulaciones, cursos, asignaturas, profesores y alumnos. 4. Vacía la base de datos Lucene con los comentarios y valoraciones de los alumnos.

Tabla 4.10: Flujo de eventos básico del caso de usos “Borrar datos del sistema”.

4.2 Diseño

En este TFG se ha intentado no complicar el diseño de cada uno de los componentes, con el fin de facilitar el trabajo a futuros desarrolladores que deseen extenderlo o utilizarlo. En esta sección se presenta el diseño de cada uno de los componentes, junto con el diseño de la base de datos utilizada para almacenar toda la información necesaria.

4.2.1 Diseño de la página web

La página web se ha diseñado haciendo uso del patrón "Modelo - Vista - Controlador" (MVC). El "Modelo" representa la información con la que trabaja la aplicación, es decir, la lógica de negocio. La "Vista" transforma el modelo en una página web para permitir al usuario interactuar con ella. El "Controlador" se encarga de procesar las interacciones del usuario con el servidor y realizar los cambios apropiados en la vista.

En la Figura 4.5 se muestran las clases dentro del patrón MVC. Obviando las relaciones entre clases con mismo nombre (login, alumno, preguntas y profesor), a continuación se explica el motivo de las asociaciones entre el resto de clases:

- **alumno (vista) – preguntas:** cuando el alumno entra en su página por defecto, necesita obtener las preguntas del profesor que quiere evaluar.
- **profesor (vista) – uni:** el modelo “uni” proporciona 2 funciones para obtener la fecha por defecto para los campos de filtro de los comentarios. En concreto, una de las funciones devuelve qué día se inicia el curso presente y otra cuándo acaba. Así, cuando el profesor vaya a buscar comentarios tendrá los campos del rango de fechas establecidos de forma predeterminada.
- **alumno (controlador) – solr:** cuando el alumno envía el comentario, se debe guardar en la base de datos de Lucene mediante Solr.
- **profesor (controlador) – solr:** el profesor necesita acceder a Solr para obtener los comentarios de los alumnos almacenados en la base de datos Lucene.
- **admin (controlador) – uni:** el modelo “uni” devuelve las titulaciones, cursos, asignaturas y usuarios de la universidad para poder ser exportados en formato CSV y que el administrador pueda editarlo.

Además de utilizar el patrón MVC, la aplicación web utiliza clases de “ayuda” durante su ejecución. Estas clases pueden verse en la Figura 4.6.

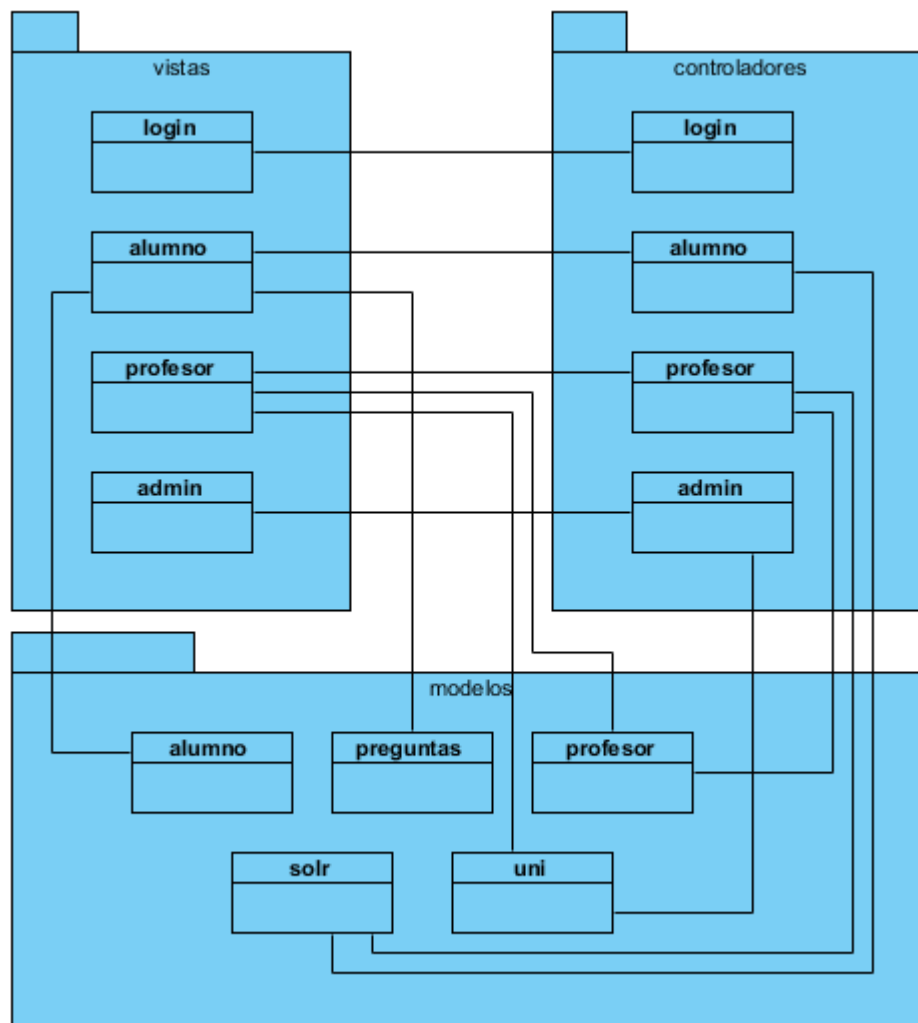


Figura 4.5: Diseño MVC de la página web.

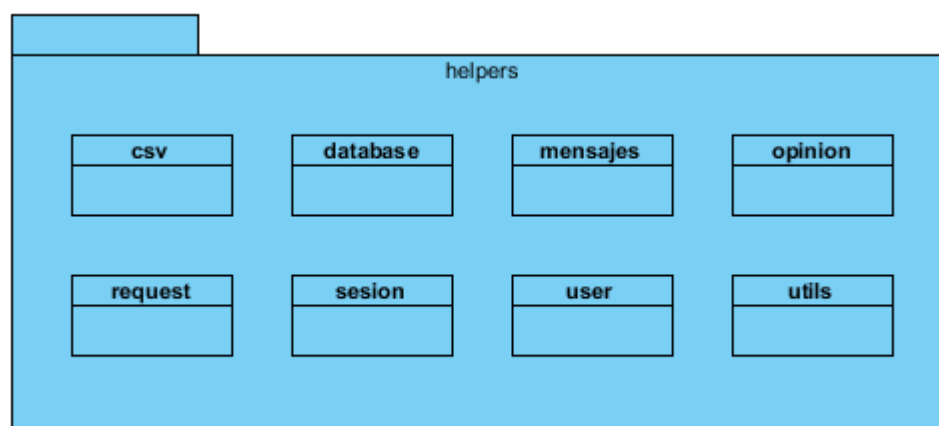


Figura 4.6: Paquete de clases de ayuda para la aplicación web

A continuación se explica brevemente cada una de las clases:

- **csv**: se utiliza para leer el fichero de los alumnos que suben los profesores para darles de alta (y que obtienen del Campus Virtual), y para leer y crear el archivo CSV que maneja el administrador para gestionar los datos de la universidad (titulaciones, cursos, asignaturas, profesores, etc).
- **database**: clase creada para simplificar la conexión a la base de datos MySQL y gestionar de manera fácil los datos.
- **mensajes**: se utiliza para mostrar mensajes de alerta e información al usuario, por ejemplo para avisar de que el usuario y contraseña que ha introducido en el formulario de acceso son incorrectos o para informar de que el comentario que ha enviado se ha guardado con éxito.
- **opinión**: contiene el algoritmo de análisis de sentimientos para los comentarios así como funciones para poder leer el archivo diccionario de palabras.
- **request**: clase que ayuda a leer los datos que el usuario envía al servidor.
- **sesion**: utilizada para controlar la sesión del usuario cuando accede a la página web.
- **user**: clase que almacena información del usuario que está actualmente navegado por la página web y proporciona funciones básicas para por ejemplo poder acceder o desconectarse.
- **utils**: funciones extras utilizadas en el sistema como por ejemplo convertir un texto a codificación UTF-8.

4.2.2 Diseño de la base de datos MySQL

Para almacenar la información relativa a la universidad se utiliza una base de datos relacional, en concreto, el SGBD utilizado en este trabajo es MySQL. El diseño de las tablas y sus relaciones se presenta en la Figura 4.7.

A continuación se describe lo que representa cada una de las tablas:

- **titulaciones**: en esta tabla se guardan las titulaciones existentes en la universidad. Únicamente necesita el nombre de la titulación y un identificador único.
- **cursos**: almacena los diferentes cursos que hay para cada titulación. Por ejemplo “Primero (turno mañana)” o “Segundo (grupo B)”. Sólo precisa de un nombre, un identificador único y una referencia a la titulación.
- **asignaturas**: en esta tabla se guardan las distintas asignaturas que hay para cada curso. Por ejemplo, se definirían las “asignaturas para el curso primero (grupo tarde) de la titulación Ingeniería del Software”. Únicamente guarda el nombre de la asignatura, un identificador y una referencia al curso.
- **usuarios**: almacena los datos de los usuarios. Cada usuario dispone de un nombre, email único y apellidos. Además, en la tabla se guardará la contraseña del usuario codificada con el algoritmo MD5, y un valor ‘type’ que se corresponde con el tipo de usuario (alumno, profesor o administrador).
- **usuarios_asignaturas**: en esta tabla se crean las relaciones entre las asignaturas y los usuarios. Por cada columna se establece un id de usuario y un id de asignatura. Así, si un

profesor está relacionado con una asignatura significará que la imparte, y si lo está un alumno querrá decir que este alumno está matriculado en dicha asignatura.

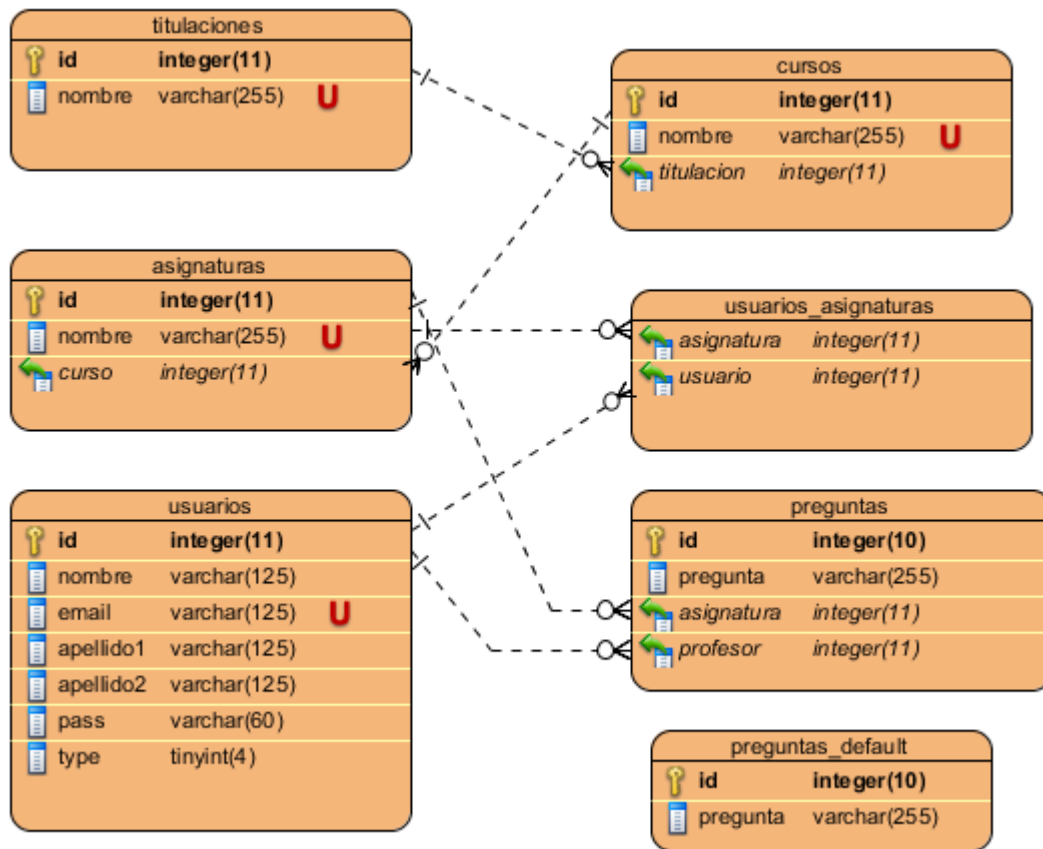


Figura 4.7: Diseño de la base de datos MySQL.

- **preguntas**: almacena las preguntas personalizadas de los profesores para cada una de sus asignaturas. La tabla tiene 4 columnas para: un identificador único, el texto de la pregunta, el id de profesor y el id de asignatura.
- **preguntas_default**: cuando un profesor no ha personalizado las preguntas para una de sus asignaturas, si un usuario va a realizar una valoración, se obtendrán las preguntas por defecto que se encuentran en esta tabla. Únicamente almacena un id de pregunta y el texto de la misma.

4.2.3 Diseño de la base de datos Lucene

Los comentarios de los alumnos y sus respuestas a las preguntas de los profesores no se almacenan en la base de datos MySQL, sino en Lucene. El motivo es explicado previamente en la sección 3.2.1. Únicamente se utiliza una colección (equivalente a las tablas en MySQL), se puede ver en la Figura 4.8.

Por cada valoración, se almacenará: un identificador único de valoración, el id del usuario que la realizó, el id de la asignatura y profesor que fueron valorados, el comentario, la opinión de la valoración una vez se ha aplicado el algoritmo de análisis de sentimientos, la fecha en la que fue realizada la valoración y las respuestas a las preguntas del profesor.



Figura 4.8: Diseño de la base de datos Lucene.

4.2.4 Diseño del algoritmo de análisis de sentimientos

En la sección 3.2.2 se explicaba por qué usar un algoritmo de análisis de sentimientos sobre los comentarios. Ya que no existe un algoritmo que aportara un diccionario en español completo (la mayoría sólo incluían un diccionario en inglés), en el proyecto se creó un algoritmo simple utilizando un diccionario con 3646 palabras en español encontrado en la página web de la Universidad del Norte de Texas [23].

El diccionario divide cada palabra y sus datos por líneas, que a la vez se dividen en tres columnas (separadas por espacios): nombre de la palabra (en español), id de la palabra en referencia a la base de datos WordNet 1.6 [24] y tipo de palabra (“neg” para negativa, “pos” para positiva).

En el diccionario faltaría el peso de cada palabra, como el diccionario es largo y revisar cada palabra una a una supone un trabajo costoso, se ha decidido que cada palabra tenga peso igual a 1. Uno de los puntos clave en cuanto a la mejora de la aplicación es utilizar un diccionario con pesos específicos para cada palabra, esto mejorará notablemente el algoritmo de análisis de sentimientos.

El formato de línea que proporcionaba la Universidad del Norte de Texas en su fichero se ha modificado para ahorrar espacio y proporcionar información más útil.

Por un lado se ha eliminado la columna del id a WordNet 1.6 ya que hace referencia a una base de datos léxica con palabras en inglés (en el proyecto se procesan únicamente comentarios en español), y aunque la base de datos aporte otro tipo de información, como la categoría gramatical de la palabra, en el algoritmo de análisis de sentimientos estos datos no son necesarios.

Además, se ha añadido el peso de la palabra, como no disponemos de los pesos de cada palabra, el valor de cada una de ellas será siempre 1. De esta manera, cuando se utilice un nuevo diccionario con los pesos establecidos, no hará falta modificar el algoritmo de análisis de sentimientos porque ya estará preparado para leer los pesos.

También se ha añadido un nuevo tipo de palabra: mod (modificador). Cuando una palabra es de tipo mod necesita el valor de una cuarta columna: sobre cuántas palabras se aplicará. Sobre las palabras que le afecte se multiplicará el peso de cada palabra por el del modificador. Por ejemplo, la línea:

muy 3 mod 1

dice que la palabra “muy” modificará la palabra que le suceda (valor 1 de la cuarta columna), y que multiplicará su peso por 3. Así, si se encuentra en un comentario “muy bonito” y “bonito” es una palabra positiva con peso 1, el modificador la cambiará a peso 3 ($1*3=3$). Por el contrario, la línea:

no -1 mod 3

multiplicará por -1 el valor de los pesos de las 3 siguientes palabras que le sucedan. Entonces, si se encontrará en un comentario “no es bonito” donde la palabra “es” no se encuentra en el diccionario y “bonito” es una palabra positiva con peso 1, el modificador “no” hará que la palabra “bonito” pase a valer -1 ($1*-1=-1$).

En el diccionario se han agregado otras palabras modificadoras como “profundamente”, “bastante”, “nunca” o “gran”.

La forma de decidir la polaridad del comentario (positivo, negativo o neutral) es la siguiente: se van sumando por un lado los pesos positivos y por otro los pesos negativos de las palabras. Después, si la diferencia entre la suma de los pesos positivos y negativos es mayor que un umbral seleccionado (20% por defecto) la polaridad del comentario será la del tipo de peso que gane (positivo o negativo), si no de tipo neutral.

Durante el análisis, las palabras del comentario son transformadas para prevenir errores ortográficos: todas las letras se pasan a minúsculas y además se eliminan los acentos.

La implementación del algoritmo del análisis de sentimientos se puede encontrar en el archivo */helpers/opinion.php*.

5. Resultados experimentales

En ese capítulo se presentan y describen los resultados obtenidos al aplicar los algoritmos de Procesamiento del Lenguaje Natural sobre un conjunto de datos.

5.1 Resultados experimentales del algoritmo de análisis de sentimientos

A continuación se presentan dos experimentos realizados para conocer la precisión del algoritmo de análisis de sentimientos. El primer experimento se ha realizado utilizando 32 *tuits* y el segundo con 3808 críticas de cine.

5.1.1 Resultados experimentales: 32 *tuits*

El Taller de Análisis de Sentimientos de la SEPLN (Sociedad Española para el Procesamiento del Lenguaje Natural) [25] organizó el año 2013 un evento donde los programadores competían por realizar el mejor algoritmo de análisis de sentimientos. El taller proporcionaba un archivo en formato XML con *tuits* y, entre otros datos, la valoración del *tuit* en cuanto a polaridad, establecida manualmente por personas. La forma de decidir la polaridad era la siguiente: un conjunto de personas leían los *tuits* y los etiquetaban manualmente, cuando una polaridad destacaba ampliamente sobre otras, se escogía para añadirla en el documento XML.

Para comprobar la eficiencia del algoritmo de análisis de sentimientos usado en este proyecto, se han utilizado algunos de los *tuits* de ejemplo que proporcionaba el taller [26]. En la Tabla 5.1 se muestran los *tuits*, con la polaridad anotada manualmente, la polaridad establecida por el algoritmo, y un comentario del autor de esta memoria. Además, las filas verdes denotan acierto del algoritmo, rojas fallo y naranjas cuando el autor no está de acuerdo con la polaridad establecida por SEPLN.

N	<i>Tuit</i>	Polaridad anotada	Polaridad obtenida	Comentario
1	Portada 'Público', viernes. Fabra al banquillo por 'orden' del Supremo; Wikileaks 'retrata' a 160 empresas espías. http://t.co/YtpRU0fd	Negativa	Neutral	¿La justicia es negativa? ¿Dar con 160 empresas espías es negativo? Podría no ser negativa.
2	Grande! RT @veronicacalderon "El periodista es alguien que quiere contar la realidad, pero no vive en ella" via @galtres	Neutral	Positiva	Podría ser un comentario positivo, por la palabra "Grande" del principio.
3	Gonzalo Altozano tras la presentación de su libro 101 españoles y Dios. Divertido, emocionante y brillante. http://t.co/4BdljMhB	Positivo	Positivo	
4	Mañana en Gaceta: TVE, la que pagamos tú y yo, culpa a una becaria de su falsa información sobre el cierre de @gaceta	Negativo	Negativo	
5	Qué envidia "@mfcastineiras: Pedro mañana x la mañana me voy a París, cuando esté por la almendra parisina recordaré #Elprimernaufragio."	Neutral	Neutral	
6	Más mañana en Gaceta. Amañur depende de Uxue Barkos para crear grupo propio. ERC no cumple el req. del 15% y el PNV no quiere competencia	Negativa	Negativa	
7	Muy buenas noches followercetes, mañana va a	Neutral	Neutral	

	ser un día bastante mítico para mi, ya os contare... http://t.co/U4obbEge			
8	Más de mañana en Gaceta. UPyD contará casi seguro con grupo gracias al Foro Asturias. Eso se dice en el Congreso	Neutral	Positiva	El algoritmo confunde la palabra “gracias”. Por ella cambia el comentario a positivo.
9	La felicidad no esta en los grandes anhelos , sino con pequeñas cosas que ocurren todos los días. Buenas noches , mañana mas ;-))	Positiva	Positiva	
10	#ff para @pperezf	Neutral	Neutral	
11	"Ya lo veremos, ya lo veremos..." les ha respondido Rajoy a Rudi y Bauzá cuando han reclamado la moratoria sobre la deuda. Mañana en @gaceta	Negativa	Neutral	Las palabras “moratoria” y “deuda” no están en el diccionario.
12	Definitivamente, creo que me he resfriado. Con este tiempo de locos que ha estado haciendo estos meses, ahora toca las consecuencias.	Neutral	Negativa	Podría ser negativa.
13	Habia prometido responder a todos, pero me ha sido imposible. Y hoy no doy para mas. MUCHAS GRACIAS A TODOS	Positiva	Neutral	"Imposible" se resta con "gracias". El algoritmo debería aumentar el peso de las palabras en mayúsculas.
14	Salgo de #VeoTV ,que día más largooooo...	Negativa	Neutral	El algoritmo debería interpretar las palabras alargadas. “Largo” no se encuentra en el diccionario.
15	Muchas gracias a todos por los comentarios durante el programa de hoy en #VeoTV @MundoaCuestas :-))	Positiva	Positiva	
16	@PauladeLasHeras No te libraras de ayudar me/nos. Besos y gracias	Positiva	Positiva	
17	@lazarodemundo Me ayudaras aunque no seas de la asociacion, y acabaras haciendote	Positiva	Neutral	Podría no ser positiva.
18	@carmenmorodo Gracias	Positiva	Positiva	
19	#Frailemoroso RT @JorgeNavasGarcia.. algun alcalde que se haya adelantado a si mismo la paga de Navidad en agosto. #Parla si	Neutral	Neutral	Podría no ser neutral.
20	@marodriguezb Gracias MAR	Positiva	Positiva	
21	Medir las palabras en 140 caracteres: http://t.co/s41kO7jt	Neutral	Neutral	
22	@periodistas21 @aprensamadrid @fgurbaneja Graci, Jun, y seguimos trabajando todos po lo mismo	Positiva	Neutra	Es complicado que el algoritmo interprete "trabajar todos por lo mismo" como algo positivo.
23	@anapastor_tve Gracias Ana, y a todo los que recibisteislos Ondas mientras recontabamos votos	Positiva	Positiva	
24	"Soy de ese tipo de personas que no acaban de comprender las cosas hasta que las ponen por escrito" Murakami	Neutral	Neutral	
25	Bueno, mañana a las seis. Duermo poco	Neutral	Positiva	El algoritmo debería saber diferenciar “bueno” como interjección y “bueno” como adjetivo.
26	Off pensando en el regalito Sinde, la que se va de la SGAE cuando se van sus corruptos. Intento no	Negativa	Positiva	“Sacar” está en el diccionario como palabra negativa, el

	sacar conclusiones (lo intento)			“no” la convierte en positiva. En el diccionario deberían estar palabras en plural para poder valorar la palabra “corruptos”.
27	Confirmada la firma en Madrid el día 6 de Diciembre a las 18:00 en el cine Callao de Madrid.	Neutral	Neutral	
28	RT @FabHddzC: Si amas a alguien, déjalo libre. Si grita ese hombre es mío era @paurubio...	Neutral	Neutral	
29	Holaa ya viernes. Con Pilar Urbano como invitada. Y el gran @JostoMaffeo. Antes @LeticiaIG y @t5rf. ¡ Que disfruteis de la buena información!	Positiva	Positiva	
30	Las importantes acciones de Málaga como ciudad inteligente han estado bien presentes en el Smart City Expo World Congress de Barcelona	Positiva	Positiva	
31	Nominado a Premios Lo Nuestro 2012! Que chevere! Gracias mi gente!	Positiva	Positiva	
32	Buenos días. Llegar al borde del abismo antes de resolver el problema, es un método. Con Merkel, hemos dado un paso mas.	Neutral	Negativa	En el diccionario no están las palabras "buenos", "borde" y "abismo". Sí “problema”, que hace al comentario negativo.

Tabla 5.1: *Tuits* analizados por el algoritmo de análisis de sentimientos.

De los 32 *tuits* analizados, 19 son acertados, 8 son fallados y 5 podrían contabilizarse como fallo o acierto, dependiendo cómo se considere la polaridad del *tuit*. El autor de esta memoria considera todos los *tuits* en naranja, salvo el número 12, acertados por el algoritmo. Se podría considerar, entonces, que se han acertado 23 *tuits* y fallado 9. Así pues, el algoritmo habría acertado el 72% de los *tuits*.

Dado que es un algoritmo simple, no se ha utilizado un diccionario con pesos y además que entre las personas se está de acuerdo en la clasificación de un texto en cuanto a su polaridad en un 79% [27] (es decir, que aunque el algoritmo tenga una precisión del 100%, los humanos estarían en desacuerdo el 20% de las veces) podría, entonces, considerarse el algoritmo moderadamente eficiente.

5.1.2 Resultados experimentales: 3080 críticas de cine

Sin embargo, 32 *tuits* no son suficientes para conocer la precisión del algoritmo. Se han analizado 3808 críticas de cine etiquetados en cuanto a su polaridad, nuevamente, de forma manual por personas [28]. La media de palabras por comentario es de 408. La cantidad será, muy probablemente, superior a la de los comentarios de los alumnos.

En la Tabla 5.2 se presenta los resultados tras analizar los 3808 comentarios:

- El número total de comentarios clasificados en cada categoría es la suma de filas: 1451 positivos, 1115 negativos, 1242 neutrales.
- Las columnas representan la predicción hecha por el algoritmo. En la primera columna, 550 comentarios positivos fueron clasificados como positivos, 321 comentarios negativos fueron clasificados incorrectamente como positivos y 455 comentarios neutrales fueron clasificados incorrectamente como neutrales. Los 776 comentarios clasificados de manera incorrecta se consideran falsos positivos.

Polaridad según humanos	Polaridad según algoritmo			
		Positivo	Negativo	Neutral
	Positivo	550	382	519
	Negativo	321	<u>479</u>	315
	Neutral	455	391	<u>396</u>

Tabla 5.2: Resultados experimentales de los comentarios de cine.

- Leyendo de izquierda a derecha desde la parte superior, del total de comentarios positivos, 550 fueron clasificados como positivos, 382 como negativos y 519 como neutrales. 901 comentarios positivos fueron perdidos y se concedieran falsos negativos.
- Los valores de la diagonal han sido correctamente clasificados y están subrayados. El resto han sido clasificados de manera incorrecta.

Para obtener una precisión rápida, simplemente hay que dividir los comentarios acertados entre el total. Así, la precisión del algoritmo en este caso es: $1425/3808=0.37$. Es un valor muy bajo. El algoritmo necesita mejoras para llegar a un porcentaje de acierto superior. Utilizar un diccionario con pesos, añadir en el diccionario las palabras en plural para que el algoritmo las interprete, ser capaz de analizar palabras en mayúsculas o alargadas o tener en cuenta las frases hechas o los refranes, podrían ser diferentes aspectos a tener en cuenta en un nuevo algoritmo. Además, es necesario destacar que la longitud de las críticas de cine utilizadas en estos experimentos será considerablemente mayor que la longitud que las opiniones que los alumnos introduzcan en el sistema OMIT.

5.2 Resultados experimentales del algoritmo de *clustering*

Existen muchos algoritmos de *clustering*, cada uno de ellos diferente. En este proyecto se necesitaba un algoritmo con unos requisitos mínimos: que no necesitara un parámetro de entrada con la cantidad de grupos que debía crear, que fuera de dominio genérico (capaz de crear los grupos tanto para críticas de cine, como para *tuits*, comentarios de alumnos...) para que no tuviera que ser entrenado, que estuviera enfocado en textos cortos (ya que se presupone que los comentarios de los alumnos no serán extensos) y que pudiera comunicarse de manera eficiente con el lenguaje PHP del servidor.

Tras descartar algoritmos que no cumplieran los requisitos mencionados, las propuestas se quedaron en el algoritmo STC y en el algoritmo Lingo. Ambos se encontraban en el motor de búsqueda Carrot2 e integrados en el servidor Solr, quién permitía una interacción fácil con PHP.

Para escoger uno de los dos algoritmos de *clustering* se ha utilizado una selección de noticias pertenecientes a diferentes colecciones de artículos comparables. Esta recopilación ha sido creada en el Grupo de Procesamiento de Lenguaje Natural de la URJC [29].

En concreto, hay 38 artículos con una media de 420 palabras por artículos, y 6 grupos de noticias como se muestran en la Tabla 5.3.

La prueba consistía en utilizar la recopilación de noticias sobre los algoritmos y comprobar cuál de ellos creaba mejor los grupos. En concreto se han realizado 4 pruebas, 2 de ellas con las noticias de la URJC sobre el algoritmo STC y el algoritmo Lingo, y otras 2 pruebas realizadas sobre los mismos algoritmos pero utilizando únicamente el primer párrafo de las noticias, y los comentarios de los alumnos no serán tan extensos y es necesario ver

cómo se comportan los dos algoritmos de *clustering* con textos cortos. Usando únicamente el primer párrafo, la media de palabras baja de 420 a 77 por noticia.

Nombre del grupo	Nombre del archivo de la noticia
Grupo 1	19950330-21584_C161.xml
	19950401-00196_C161.xml
	19951010-06115_C161.xml
	19951018-12166_C161.xml
Grupo 2	20000102_606_C33.source
	20000102_614_C33.source
	20000102_623_C33.source
	20000102_624_C33.source
	20000102_633_C33.source
	20000102_647_C33.source
	20000102_671_C33.source
Grupo 3	newC00002_001.ES
	newC00002_002.ES
	newC00002_003.ES
	newC00002_004.ES
	newC00002_005.ES
Grupo 4	newC00005_001.ES
	newC00005_002.ES
	newC00005_003.ES
	newC00005_004.ES
Grupo 5	newC00012_001.ES
	newC00012_002.ES
	newC00012_003.ES
	newC00012_004.ES
	newC00012_005.ES
	newC00012_006.ES
	newC00012_007.ES
	newC00012_008.ES
	newC00012_009.ES
	newC00012_010.ES
	newC00012_011.ES
	newC00012_012.ES
	newC00012_013.ES
	newC00012_014.ES
	newC00012_015.ES
Grupo 6	newC00018_001.ES
	newC00018_002.ES
	newC00018_003.ES

Tabla 5.3: Distribución de los grupos de las noticias.

Los resultados de las pruebas se presentan en la Tabla 5.4:

- La columna Grupos representa el número de grupos que se han creado.
- La columna TP representa los “verdaderos positivos” (*true positive*) o las noticias que están en un grupo y deben estarlo.

- La columna TN representa los “verdaderos negativos” (*true negative*) o las noticias que no están en un grupo y además no deben estarlo.

N. de Prueba	Noticias	Algoritmo	Grupos	TP	TN	FP	FN	RI (Rand Index)
1	Completas	STC	14	85	299	84	64	0.721805
2		Lingo	18	60	515	0	109	0.840643
3	Primer párrafo	STC	15	94	386	21	69	0.842105
4		Lingo	15	38	435	2	95	0.829825

Tabla 5.4: Resultados de las pruebas de *clustering*.

- La columna FP representa los “falsos positivos” (*false positive*) o las noticias que están en un grupo y no deberían estar.
- La columna FN representa, los “falsos negativos” (*false negative*) o las noticias que no están en un grupo y deberían.
- La columna Precisión representa cuánto de efectivo es el algoritmo para las condiciones impuestas. La precisión se calcula con la siguiente fórmula [30]:

$$Rand\ index = \frac{TP + TN}{TP + TN + FP + FN}$$

Aunque el algoritmo Lingo funciona mejor para textos largos, dado que los comentarios de los alumnos no serán extensos, se utilizará el algoritmo STC que ha ganado, aunque por poco, a Lingo.

6. Conclusiones y trabajos futuros

En este capítulo se presentan las conclusiones obtenidas en el desarrollo del proyecto y los posibles trabajos futuros que pueden servir para continuar este TFG.

6.1 Conclusiones

Los objetivos del proyecto, desde el principio, fueron mejorar el sistema actual que utiliza la URJC para que los alumnos valoren a sus profesores y otros aspectos relacionados con una asignatura. Con este proyecto se ha conseguido dicho objetivo, sin embargo el nuevo sistema necesita ser mejorado en algunos aspectos.

Para ayudar a los nuevos desarrolladores que intenten mejorar y ampliar el sistema, el proyecto se ha creado escribiendo un código fácil de leer, extensible y documentando. Además, se ha alojado en un repositorio Git público¹.

En las universidades se encuentran diferentes perfiles con distinto nivel de conocimiento en informática, por eso la página web se ha creado teniendo en cuenta aspectos de usabilidad. El objetivo es que la web pueda ser utilizada por cualquier tipo de alumno, profesor o asignatura, aunque la parte de accesibilidad es una de las partes a mejorar.

Dada la gran cantidad de tecnologías que se utilizan en este proyecto, se ha creado un manual técnico para ayudar a poner en marcha el sistema, en concreto se ha escrito como guía para la configuración del entorno en el Sistema Operativo GNU/Linux Ubuntu 12.04 32-bit. El manual se deja adjunto en un CD junto a la memoria y, además, puede encontrarse en el repositorio Git.

El Procesamiento del Lenguaje Natural se encuentra en la actualidad en auge, por el potencial que supone su uso sobre la gran cantidad de información que se genera en Internet. Entender cómo funcionan algunos algoritmos leyendo documentación variada y artículos científicos ha supuesto un reto importante para el desarrollo de este trabajo.

6.2 Trabajos futuros

Unos de los puntos débiles del proyecto es el algoritmo de análisis de sentimientos. Es altamente recomendable encontrar o crear un diccionario español con pesos de polaridad para cada una de sus palabras (ya que actualmente cada palabra tiene peso 1), además de añadir en el diccionario palabras en plural para que el algoritmo las tenga en cuenta. Incluso se podrían utilizar algoritmos ya implementados y con un alto recorrido con un diccionario español. Existen nuevos algoritmos que son capaces de dar valor a los emoticonos que se escriben en los mensajes, procesar textos con faltas de ortografía o por ejemplo dar una interpretación el significado de las palabras alargadas. Si se desea mejorar el algoritmo creado, debería ser capaz de aumentar el peso de las palabras alargadas o frases con exclamaciones, reconocer emoticonos o dar valor a las frases hechas, refranes.

Aunque se han realizado pruebas experimentales del algoritmo de *clustering* sobre ciertos tipos de textos, sería necesario probarlo con comentarios reales de alumnos, ya que el dominio del texto a procesar puede variar notablemente el tipo y número de temas que se obtendrían tras aplicar el algoritmo de *clustering*. Si el resultado no fuera el deseado, se

¹ Github – Omit. <https://github.com/jonijnm/omit/>

pueden crear nuevos mecanismos y funcionalidades para obtener los temas de los que hablen los alumnos en los comentarios. Por ejemplo, se podrían definir por defecto temas genéricos, y que cada profesor pudiera cambiarlos para cada una de sus asignaturas. Así podría haber temas específicos. Para evitar la prohibición de temas de los que podría hablar el alumno, se dejaría un campo donde el alumno podría crear un nuevo tema, donde no escogería uno de los sugeridos. Después, el profesor, si ve que varios alumnos han sugerido un nuevo tema del que opinar, podría añadirlo a la lista. O incluso un algoritmo podría realizar esta parte manualmente: cuando un número determinado de alumnos han creado un nuevo tema, éste, automáticamente pasa a la lista de temas sugeridos para los alumnos de esa asignatura.

Actualmente, con el nuevo sistema sólo se pueden valorar a los profesores y las asignaturas que estos imparten, sin embargo se podrían valorar otros aspectos como el servicio de limpieza o el comedor. Llevando esta idea algo más lejos, se podría crear un sistema con el que se pudiera valorar cualquier servicio, que no tuviera relación con la universidad.

El proyecto podría ampliarse en otros aspectos. Por ejemplo, el administrador, en vez de descargar un archivo Excel con los datos de las titulaciones, cursos, asignaturas y profesores, podría gestionar estos datos directamente desde la web, mediante tablas animadas y utilizando AJAX.

Por otro lado, sería una buena idea que la página web tuviera un botón de ayuda en la parte superior derecha, por ejemplo, donde explique cómo utilizar cada uno de los elementos de la página web y cómo interaccionar con ella.

La página web no ha sido diseñada para acceder desde un dispositivo móvil. Se puede aprovechar el auge que supondrá HTML5 a medio plazo, donde podrán visualizarse páginas webs desde dispositivos móviles, portátiles convencionales, o televisores, para realizar la página web utilizando el estándar HTML5 y crear diferentes diseños para pantallas de dimensiones reducidas. Además, sería necesario comprobar y mejorar la página web en cuanto a su accesibilidad.

Cuando los profesores envían el archivo CSV de los datos de los alumnos, los nuevos ingresos tendrán que utilizar, para acceder al sistema, el usuario de la universidad y su DNI como contraseña. Es necesario crear un apartado en la página web para que los usuarios puedan cambiar la contraseña.

Por último, otra carencia importante es que no se pueden modificar las preguntas por defecto de la tabla MySQL “preguntas_default” sin un gestor de base de datos como phpMyAdmin. El administrador necesitaría un apartado para este propósito.

Bibliografía

- [1] Jacobson I. Booch G. and Rumbaugh J. *El Proceso Unificado de Desarrollo de Software*. Pearson - Addison Wesley, 1999.
- [2] Beck K. *eXtreme Programming eXplained*. Addison Wesley, 2000.
- [3] PHP. <http://php.net/>.
- [4] MySQL. <http://www.mysql.com/>.
- [5] HTML. <http://www.w3.org/html/>.
- [6] CSS. <http://www.w3.org/Style/CSS/>.
- [7] DOM. <http://www.w3.org/DOM/>.
- [8] Wikipedia - JavaScript. <http://es.wikipedia.org/wiki/JavaScript>.
- [9] jQuery. <http://jquery.com/>.
- [10] jQuery UI. <http://jqueryui.com/>.
- [11] Highcharts. <http://www.highcharts.com>.
- [12] AJAX - Un nuevo acercamiento a Aplicaciones Web.
<http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.
- [13] JSON. <http://www.json.org/>.
- [14] Git. <http://git-scm.com/>.
- [15] Netbeans. <https://netbeans.org/>.
- [16] Github. <https://github.com/>.
- [17] Wikipedia - CSV. http://en.wikipedia.org/wiki/Comma-separated_values.
- [18] Lucece. <http://lucene.apache.org/>.
- [19] Solr. <http://lucene.apache.org/solr/>.
- [20] T. Hofmann. 1999. Probabilistic latent semantic indexing. En *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, páginas 50–57, New York, NY, USA.

- [21] Carrot2. <http://project.carrot2.org/>.
- [22] B. Liu. *Opinion Mining and Sentiment Analysis. Web Data Mining. Springer*, 2011.
- [23] Veronica Perez Rosas, Carmen Banea, Rada Mihalcea, Learning Sentiment Lexicons in Spanish. En *Proceedings of the International Conference on Language Resources and Evaluations (LREC 2012)*, Istanbul, Turkey, May 2012.
- [24] WordNet. <http://wordnet.princeton.edu/>.
- [25] Taller de Análisis de Sentimientos en la SEPLN.
<http://www.daedalus.es/TASS2013/corpus.php>.
- [26] Villena-Román, Julio, Lana-Serrano, Sara, Martínez-Cámara, Eugenio, González-Cristobal, José Carlos. 2013. Revista de Procesamiento del Lenguaje Natural, 50, pp 37-44. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/4657>.
- [27] How Companies Can Use Sentiment Analysis to Improve Their Business.
<http://mashable.com/2010/04/19/sentiment-analysis/>.
- [28] Cruz, F.L., J.A. Troyano, F. Enríquez, y J. Ortega. 2008. Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Sociedad Española para el Procesamiento de Lenguaje Natural*, nº 41.
- [29] News Corpus for document clustering.
http://www.etsii.urjc.es/~smontalvo/corpus_resources.html#corpus.
- [30] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval, Cambridge University Press*. 2008.
- [31] Servidor HTTP Apache. <http://httpd.apache.org/>.
- [32] Usefulness of Confusion Matrices. <http://khartig.wordpress.com/2012/03/26/usefulness-of-confusion-matrices-2/>.