

Departamento de Ingeniería del Software e Inteligencia Artificial  
Facultad de Informática  
UNIVERSIDAD COMPLUTENSE DE MADRID

# Proyecto DALGS

Degrees and Learning Goals System



TRABAJO FIN DE GRADO EN INGENIERÍA DEL SOFTWARE

Curso 2014-2015

Madrid, junio de 2015

**Autores:**

Almudena Fernández Romo, Roberto García Sánchez

**Director:**

Iván Martínez Ortiz



# Autorización de difusión y utilización

SE AUTORIZA A LA UNIVERSIDAD COMPLUTENSE A DIFUNDIR Y UTILIZAR CON FINES ACADÉMICOS, NO COMERCIALES Y MENCIONANDO EXPRESAMENTE A SUS AUTORES, TANTO LA PROPIA MEMORIA, COMO EL CÓDIGO, LOS CONTENIDOS AUDIOVISUALES, INCLUSO SI INCLUYEN IMÁGENES DE LOS AUTORES, LA DOCUMENTACIÓN Y/O EL PROTOTIPO DESARROLLADO.

Autores: Almudena Fernández Romo y Roberto García Sánchez

Firma:

En Madrid, a 20 de junio de 2015



*“La ciencia puede divertirnos y fascinarnos,  
pero es la Ingeniería la que cambia el mundo”*

*- Isaac Asimov -*



# Agradecimientos

Gracias a todos los profesores que nos han ido acompañando en este viaje, sin ellos no habríamos llegados hasta aquí. A la Facultad de Informática que durante estos años ha sido una parte importante de nuestra vida y a la que podemos considerar también nuestra familia.

También darles las gracias a nuestros padres por darnos esta oportunidad, confiar en nosotros y apoyarnos siempre, aunque las cosas no siempre hayan salido como esperábamos.

Porque como suele decirse: *"Todo esfuerzo tiene su recompensa"*

Y un agradecimiento especial a Iván Martínez Ortiz y al departamento de Ingeniería del Software e Inteligencia Artificial por haber sido tan paciente con nosotros y guiarnos en todo momento durante los últimos años. Gracias por asesorarnos y ayudarnos cuando lo hemos necesitado.



# Tabla de contenido

Índice de Ilustraciones .....	XIII
Índice de Cuadros .....	XV
Índice de Abreviaturas .....	XVII
Resumen .....	XIX
Abstract.....	XXI
Capítulo 1. Estado del arte.....	1
1.1    Introducción.....	1
1.2    Problema.....	2
1.3    Solución: DALGS .....	2
1.4 Estructura del trabajo .....	4
Capítulo 2. Definición de los Casos de Uso.....	5
2.1 Diagrama General.....	6
2.2 Casos de Uso .....	7
2.2.1 Casos de Uso: Usuario.....	7
2.2.2 Casos de Uso: Estudiante.....	8
2.2.3 Casos de Uso: Profesor.....	9
2.2.4 Casos de Uso: Coordinador .....	10
2.2.5 Casos de Uso: Administrador.....	11
Capítulo 3. Descripción de la Herramienta.....	13
3.1 Modelo de datos .....	14
3.2 Módulos Gestionados.....	15
3.2.1 Módulo de Gestión de Usuarios.....	15
3.2.2 Módulo de Gestión de Cursos Académicos .....	16
3.2.3 Módulo de Gestión de Actividades.....	17
3.2.4 Módulo de Gestión de Actividades Externas.....	18
3.2.5 Módulo de Gestión de Competencias.....	19
3.2.6 Módulo de Gestión de Cursos.....	20

3.2.7 Módulo de Gestión de Titulaciones .....	21
3.2.8 Módulo de Gestión de Grupos .....	22
3.2.9 Módulo de Gestión de Resultados de Aprendizaje .....	23
3.2.10 Módulo de Gestión de Módulos de una titulación .....	24
3.2.11 Módulo de Gestión de Asignaturas .....	25
3.2.12 Módulo de Gestión de Materias.....	25
3.3 Importación y exportación de datos .....	26
3.4 Gestión de los correos de coordinación.....	28
Capítulo 4. Diseño.....	31
4.1 Arquitectura de la aplicación .....	31
4.1.1 ResultClass<E> .....	35
4.1.2 Patrones de diseño: MVC .....	36
4.1.3 Modelo.....	37
4.1.4 Vista.....	37
4.1.5 Controlador .....	38
4.2 API REST .....	38
4.3 Persistencia .....	39
4.3.1 JPA .....	40
4.3.2 MySQL .....	42
4.4 Securización: Spring Security.....	44
4.4.1 Autenticación .....	45
4.4.2 Autorización.....	46
4.4.3 Gestión de ACLs .....	47
4.4.4 Spring Security en DALGS.....	48
4.4.5 Spring Security OAuth.....	52
Capítulo 5. Desarrollo de la aplicación.....	55
5.1 Esquema general de la aplicación.....	55
5.2 Contexto tecnológico.....	56
5.2.1 Tecnologías utilizadas.....	56

5.2.2 Frameworks .....	58
5.2.3 Contenedor Java EE.....	59
5.2.4 Control de versiones .....	61
5.3 Herramientas de diseño .....	62
5.3.1 LucidChart.....	62
5.4 Herramientas de desarrollo .....	63
5.4.1 Spring Tool Suite .....	63
5.4.2 Sequel Pro.....	63
Capítulo 6. Diario de Trabajo .....	65
6.1 Metodología de trabajo .....	65
6.2 Organización y planificación .....	66
6.1 Diario mensual de desarrollo.....	67
Capítulo 7. Conclusiones .....	69
7.1 Resumen de contribuciones.....	69
Chapter 8. Conclusions.....	71
8.1 Summary of contributions.....	71
Capítulo 9. Trabajo futuro .....	73
9.1 Arquitectura multi-tenant.....	73
9.2 Experience API .....	74
9.3 Mozilla Open Badges .....	75
9.5 Plugin para Moodle.....	76
9.6 Login con Google.....	76
9.7 Otras mejoras .....	77
Apéndice A: Historias de Usuario.....	79
Historias de usuario.....	79
1. Epic e historias de usuario: Usuario .....	80
2. Epic e historias de usuario: Alumno .....	83
3. Epic e historias de usuario: Profesor .....	86
4. Epic e historias de usuario: Coordinador .....	92

5. Epic e historias de usuario: Administrador .....	98
Apéndice B: API DALGS .....	115
Referencias.....	119
Bibliografía .....	123
Anexo A: Manual de Instalación.....	125
1. Generación del archivo ejecutable .war .....	125
2. Importación del proyecto .....	127
3. Importación del Proyecto (BIS).....	129
4. Configuración del entorno .....	130
4.1 pom.xml.....	130
4.2 applicationContext.xml .....	131
4.3 settings.xml .....	132
5. Despliegue.....	133
Glosario.....	135

# Índice de Ilustraciones

Ilustración 1: Cabecera DALGS .....	3
Ilustración 2: Diagrama General .....	6
Ilustración 3: Casos de uso de Usuario .....	7
Ilustración 4: Casos de uso de Estudiante .....	8
Ilustración 5: Casos de uso de Profesor .....	9
Ilustración 6: Casos de uso de Coordinador.....	10
Ilustración 7: Casos de uso de Administrador .....	11
Ilustración 8: Modelo de datos utilizado en DALGS .....	13
Ilustración 9: Listado de profesores.....	15
Ilustración 10: Vista de un Curso Académico .....	16
Ilustración 11: Listado de Cursos Académicos.....	16
Ilustración 12: Vista de una Actividad .....	17
Ilustración 13: Actividades y Actividades Externas.....	18
Ilustración 14: Listado de las actividades externas .....	18
Ilustración 15: Vista de competencias.....	19
Ilustración 16: Vista de un Curso.....	20
Ilustración 17: Vista de una titulación .....	21
Ilustración 18: Vista de un Grupo .....	22
Ilustración 19: Vista de un Resultado de Aprendizaje .....	23
Ilustración 20: Relación entre competencias y los resultados de aprendizaje .....	23
Ilustración 21: Vista de un Módulo de una titulación .....	24
Ilustración 22: Vista de una Asignatura .....	25
Ilustración 23: Vista de un Materia.....	25
Ilustración 24: Vista de Subida de archivos CSV.....	27
Ilustración 25: Vista del correo .....	29
Ilustración 26: Descarga del correo.....	29
Ilustración 27: Arquitectura de los módulos de gestión de titulaciones .....	33
Ilustración 28: Arquitectura de los módulos de gestión de cursos académicos .....	34
Ilustración 29: Uso de ResultClass.....	35
Ilustración 30: Arquitectura de la aplicación.....	36
Ilustración 31: JPA.....	41
Ilustración 32: Uso de borrados lógicos.....	43

Ilustración 33: Login de acceso.....	45
Ilustración 34: Acceso denegado.....	46
Ilustración 35: Diagrama de clases de Spring Security ACL.....	47
Ilustración 36: Niveles de Usuarios.....	51
Ilustración 37: Interacción cliente-servidor .....	53
Ilustración 38: Esquema general de la aplicación .....	55
Ilustración 39: Menú deslizante con Bootstrap.....	58
Ilustración 40: Comparación de servidores [ 26 ] .....	60
Ilustración 41: Apache Tomcat [ 27 ] .....	60
Ilustración 42: MySQL [ 16 ] .....	61
Ilustración 43: Git [ 10 ] .....	61
Ilustración 44: Lucidchart [ 34 ].....	62
Ilustración 45: Spring Tool Suite [ 36 ] .....	63
Ilustración 46: Sequel Pro [ 36 ].....	63
Ilustración 47: Acceso multi-tenant.....	73
Ilustración 48: Tin Can API [ 29 ].....	74
Ilustración 49: Mozilla Open Badges [ 30 ].....	75
Ilustración 50: Moodle [ 19 ].....	76
Ilustración 51: Paginación.....	77

# Índice de Cuadros

Tabla 1: Permisos del administrador.....	49
Tabla 2: Permisos del Coordinador .....	49
Tabla 3: Permisos del Profesor.....	49
Tabla 4: Permisos del alumno.....	50
Tabla 5: Permisos del usuario .....	50
Tabla 6: Diario de desarrollo Septiembre - Noviembre .....	67
Tabla 7: Diario de desarrollo Noviembre - Enero .....	67
Tabla 8: Diario de desarrollo Enero - Marzo .....	68
Tabla 9: Diario de desarrollo Marzo - Mayo .....	68
Tabla 10: Diario de desarrollo Mayo - Junio .....	68
Tabla 11: Diario de desarrollo Junio .....	68



# Índice de Abreviaturas

<i>ACL</i>	<i>Access Control List</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>DB</i>	<i>Database</i>
<i>CSS</i>	<i>Cascading Style Sheets</i>
<i>CSV</i>	<i>Comma-Separated Values</i>
<i>ECTS</i>	<i>European Credit Transfer and Accumulation System</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>J2EE</i>	<i>Java Enterprise Edition</i>
<i>JPA</i>	<i>Java Persistence API</i>
<i>JSP</i>	<i>JavaServer Pages</i>
<i>MVC</i>	<i>Model–View–Controller</i>
<i>POJO</i>	<i>Plain Old Java Object</i>
<i>POM</i>	<i>Project Object Model</i>
<i>REST</i>	<i>Representational State Transfer</i>
<i>SaaS</i>	<i>Software as a Service</i>
<i>STS</i>	<i>Spring Tool Suite</i>
<i>TI</i>	<i>Information technology</i>
<i>UCM</i>	<i>Universidad Complutense de Madrid</i>
<i>URI</i>	<i>Uniform Resource Identifier</i>
<i>URL</i>	<i>Uniform Resource Locator</i>
<i>US</i>	<i>User Story</i>
<i>VM</i>	<i>Virtual Machine</i>
<i>JPQL</i>	<i>Java Persistence Query Language</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>DBMS</i>	<i>Database Management System</i>
<i>MySQL</i>	<i>My Structured Query Language</i>



# Resumen

Con la entrada en vigor del Proceso Bolonia, el sistema de enseñanza superior ha sufrido un profundo cambio en los planes de estudio, pasando a un modelo de habilidades o competencias que los alumnos deben adquirir durante su formación universitaria.

Asimismo, este cambio en las titulaciones también ha introducido un proceso de evaluación periódico de las mismas, de manera que es necesario recopilar evidencias a lo largo de los diferentes cursos académicos en los que se imparte la docencia de los títulos. Dentro de estas evidencias destaca la coordinación entre profesores llevada a cabo dentro de las asignaturas de la titulación y la recopilación de las actividades docentes del día a día y su relación con los resultados de aprendizaje y competencias que tienen asignadas las titulaciones.

Por este motivo nace DALGS como un sistema gestor de planes académicos que contempla los resultados de aprendizaje y las actividades diarias llevadas a cabo por los de los alumnos y que permite recopilar las evidencias de coordinación entre profesores.

En esta memoria se describe el diseño y la funcionalidad implementada en el sistema DALGS, destacando la ayuda que proporciona la herramienta en la recolección de las evidencias necesarias para demostrar la calidad de la enseñanza en las titulaciones ante procesos de evaluación externa.

**Palabras clave:** Proceso Bolonia, Calidad de Experiencia, Grados, Resultados de aprendizaje, Competencias académicas, Gestor académico.



# Abstract

The implementation of the Bologna Process has drastically changed the higher educational system, changing it to a model based on skills or competences that students must acquire during their university education.

Moreover, this change has also introduced a periodical evaluation process of new degrees. This evaluation process requires the gathering of evidence during the different academic terms where the degree is offered.

Among all the evidence that needs to be collected, the different day-to-day activities related to the learning goals and competences, as well as the coordination between professors who teach the degrees are highlighted.

The DALGS system has been created as a support tool to design degree curricula by taking into account the learning goals and the different activities implemented by the students that help to gather the coordination activities conducted by professors.

This report describes the design and functionality of the DALGS system, highlighting the assistance provided by this tool to gather evidence needed to demonstrate the quality of education during an external evaluation process.

**Keywords:** Bologna Process, Quality of Experience, Degrees, Learning Goals, Academic competences, Academic System.



# Capítulo 1. Estado del arte

## 1.1 Introducción

El proceso de reforma Universitaria a nivel europeo que comenzó en 1999 con la Declaración de Bolonia [ 14 ] ha supuesto un cambio sustancial en la enseñanza Universitaria Española en general y más en particular en las titulaciones impartidas en la Facultad de Informática (FDI) de la Universidad Complutense de Madrid (UCM).

Con la puesta en marcha en 2010 del denominado *Proceso de Bolonia* en la FDI, se han reestructurado las titulaciones orientándose a las necesidades de competencias y conocimientos requeridos para tres perfiles de la informática, creándose los grados de: Ingeniero Informático, Ingeniero del Software e Ingeniero de Computadores.

Como parte del proceso de creación de estas nuevas titulaciones, ha sido necesario someter a evaluación y autorización el diseño de planes de estudio por parte de la Agencia Nacional de la Evaluación de la Calidad y la Acreditación (ANECA). Estos planes de estudios incluyen una distribución de asignaturas en módulos y materias y la definición de las competencias que deben adquirir los estudiantes para obtener la titulación. Cabe destacar que los planes de estudio incluyen competencias profesionalizantes que vienen prefijadas [ 1 ] y otras son competencias específicas de los títulos impartidos en la FDI.

Según la guía de apoyo para la elaboración de la memoria de verificación de títulos oficiales, el concepto de **competencia** se define como: *“el conjunto de conocimientos, habilidades, actitudes que se adquieren o desarrollan mediante experiencias formativas coordinadas, las cuales tienen el propósito de lograr conocimientos funcionales que den respuesta de modo eficiente a una tarea o problema de la vida cotidiana y profesional que requiera un proceso de enseñanza y aprendizaje”* [ 13 ].

Según esta definición, las competencias de un título pueden considerarse como la elicitación de las habilidades necesarias para obtener el título, por ejemplo, una de las competencias necesarias para la titulación de Ingeniero en Informática es *“Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería informática.”* [ 1 ].

No obstante, es necesario concretar las competencias en **resultados de aprendizaje** que serán las herramientas de medida que permitirán evaluar si los alumnos adquieren las competencias a las que están asociados estos resultados de aprendizaje. Por ejemplo, uno de los resultados de aprendizaje de la asignatura Bases de Datos es “Diseño de una base de datos según el modelo entidad-relación” que está asociado a las competencias **“CT2-Capacidad de análisis y síntesis en la resolución de problemas.”** y **“CG17-Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos”.**

## 1.2 Problema

En la actualidad, en la FDI existe una herramienta que permite gestionar las titulaciones. Esta aplicación permite definir la estructura de una titulación (módulos, materias, asignaturas), la definición de las competencias asociadas al título y la asociación de dichas competencias a las materias y las asignaturas de la titulación. No obstante, esta herramienta no permite definir los resultados de aprendizaje para las titulaciones ni su relación con las competencias de las materias del título.

Por otro lado, pese a que competencias y resultados de aprendizaje perduran durante los diferentes cursos académicos, las actividades específicas que los diferentes profesores pueden llevar a cabo en una asignatura y que están relacionadas con los resultados de aprendizaje pueden cambiar entre los distintos cursos académicos (e.g. diferentes estilos de enseñanza, nuevas herramientas, etc.)

## 1.3 Solución: DALGS

DALGS (*“Degree and Learning Goals System”*) es una herramienta que permite crear y gestionar titulaciones académicas que siguen la estructura de títulos propuesta por la ANECA y gestionar los planes académicos, es decir, la instanciación y particularización de los títulos para un curso académico.

Los elementos principales de un plan de estudios son su estructura (i.e. módulos, materias, asignaturas), sus competencias (y la relación con las asignaturas y materias del plan de estudios) y los resultados de aprendizaje (relacionadas con las competencias y las asignaturas).

DALGS permite crear nuevos planes de estudio a través de una aplicación web, e incluso la importación masiva (mediante archivos CSV) de la información necesaria para crear el plan de estudios.

Asimismo, un plan de estudios es una plantilla que se instancia cada curso académico. En este sentido, para cada una de las asignaturas del plan de estudios se crean uno o varios grupos de la asignatura en la que se agrupan un conjunto de alumnos y uno o varios profesores. Esta división en grupos de una asignatura requiere de la figura de un profesor coordinador que se encarga de velar por la coordinación entre las actividades que realizan los diferentes profesores de la misma asignatura en los diferentes grupos coordinados.

Otra particularización importante es que, aunque exista coordinación a nivel de asignatura, el profesor de un grupo tiene la libertad de proponer actividades educativas que tienen una relación directa con los resultados de aprendizaje, y que no tienen por qué ser las mismas en todos los grupos.

Para facilitar la labor del profesor DALGS permite la integración y colaboración con otras aplicaciones para que, por ejemplo, desde el Campus Virtual se pueda notificar a DALGS cuándo un profesor ha creado una tarea para los alumnos, de modo que el profesor solo tenga que interactuar con DALGS para asignar la actividad al resultado de aprendizaje específico de la asignatura (sin tener que volver a introducir la información de la actividad).

Finalmente DALGS es una solución multilingüe por lo que se puede adaptar a cualquier idioma que se desee.



ILUSTRACIÓN 1: CABECERA DALGS

Se ha implementado en inglés y español, pero con un simple fichero de idioma se puede adaptar al idioma que se desee.

## 1.4 Estructura del trabajo

El presente documento se encuentra dividido en varios capítulos, con el objetivo de ilustrar e informar sobre los diferentes módulos que componen el sistema.

En el **capítulo 1** se habla sobre el estado del arte de proyecto, incluyendo una pequeña introducción y la **razón** de DALGS. En los **capítulos 2 y 3** se definen los **objetivos** y requisitos del sistema donde se incluyen los **casos de uso** más característicos del mismo. También se definen los módulos que conforman el sistema DALGS.

A lo largo del **capítulo 4** se detalla la arquitectura, persistencia y securización de la aplicación, es decir, del **diseño** de la misma, mientras que en el **capítulo 5** se habla sobre su **desarrollo**. Esto incluye, tecnologías y herramientas utilizadas para su implementación, así como el control de versiones y el servidor utilizado.

El diario de trabajo se detalla en el **capítulo 6**. En este capítulo se comenta la metodología de trabajo y la organización y planificación que se ha tenido a lo largo del presente curso académico.

Por último, los **capítulos 7, 8 y 9** hablarán sobre las **conclusiones** que hemos sacado tras finalizar este proyecto y **futuras mejoras** o adaptaciones que consideramos interesantes para la aplicación.

También se incluyen dos apéndices. El primero de ellos, **Apéndice A**, relata todas las **historias de usuarios** que por la extensibilidad de las mismas se ha decidido incluirlas aparte. El segundo apéndice, **Apéndice B**, trata de definir la **herramienta** que se ha implementado para mostrar el funcionamiento de la API que proporcionamos.

Al final de este documento se incluye un **manual de instalación** para el despliegue de la aplicación.

## Capítulo 2. Definición de los Casos de Uso

Antes de sumergirse en el desarrollo de un proyecto Software hay que tener una idea clara de qué se quiere conseguir con este o qué problema quiere resolverse. Esta idea inicial servirá como base, pero irá evolucionando conforme avance el desarrollo del mismo.

Por esta razón, nace la necesidad de definir los requisitos específicos del proyecto, los cuales servirán como mecanismo de comunicación entre todas las partes involucradas en él. No solo entre los propios desarrolladores, sino entre todos los participantes, más conocidos como usuarios finales o *stakeholders*.

Un proyecto no solo se basa en esa idea inicial, sino también en los conocimientos e ideas de todas las personas involucradas en dicho proyecto, y que generalmente son muy diferentes. Por tanto, se debe encontrar un punto intermedio que sea aceptado por todos los implicados en el proyecto y que, a su vez, sea entendible por todos ellos independientemente del nivel técnico que posean.

De igual modo, es imposible predecir la evolución de un proyecto a lo largo de su desarrollo sin conocer de antemano los problemas que se pueden encontrar. Por ello, ante proyectos de esta naturaleza, con fechas de entrega preestablecida, ha aumentado el uso de metodologías ágiles para su desarrollo.

Las metodologías ágiles se caracterizan por un desarrollo iterativo e incremental, y con un diseño centrado en el usuario final. Este modelo de desarrollo requiere que la toma de decisiones se realice conforme va avanzando el desarrollo del proyecto. Por tanto, las decisiones se toman basándose en cada momento en el estado actual del proyecto, lo que implica que se tomen decisiones realistas.

En cada iteración, también se definen las tareas que se deben llevar a cabo durante este proceso. Estas tareas son las que van definiendo las funcionalidades del proyecto y por tanto las historias de usuario [ 43 ].

Al tratarse de un proyecto muy extenso se ha optado por ilustrar los distintos casos de usos de forma generalizada. Por otro lado, las historias de usuario que se han comentado se detallan en el [Apéndice A](#)

## 2.1 Diagrama General

Tal y como se ha detallado en las secciones anteriores, DALGS se encuentra representado por dos grandes subsistemas, relacionados entre sí. Por lo que la coherencia entre ambas partes es de carácter obligatorio.

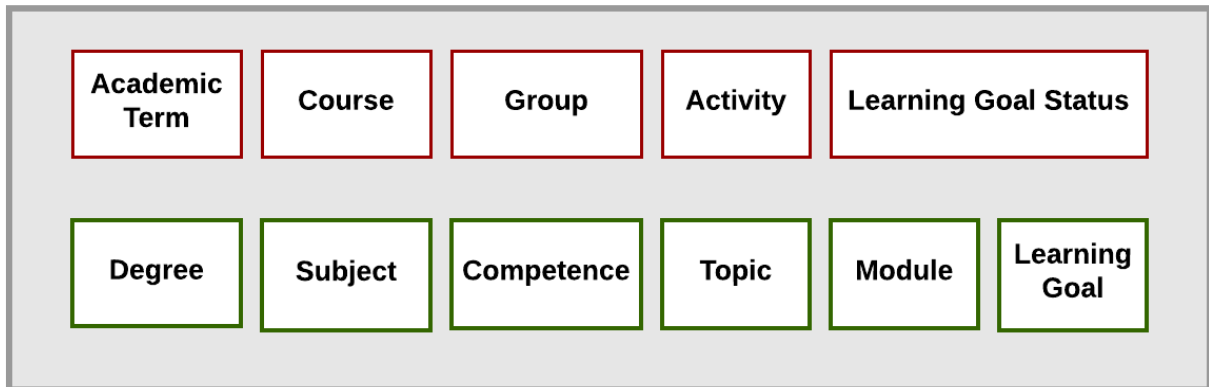


ILUSTRACIÓN 2: DIAGRAMA GENERAL

## 2.2 Casos de Uso

Como se especifica en la sección [4.4 Securización: Spring Security](#) el acceso a los diferentes recursos del sistema se restringe a través de roles de usuario y personas. En las siguientes subsecciones, se muestran de forma generalizada los diferentes casos de uso en función de los distintos roles existentes en el sistema.

### 2.2.1 Casos de Uso: Usuario

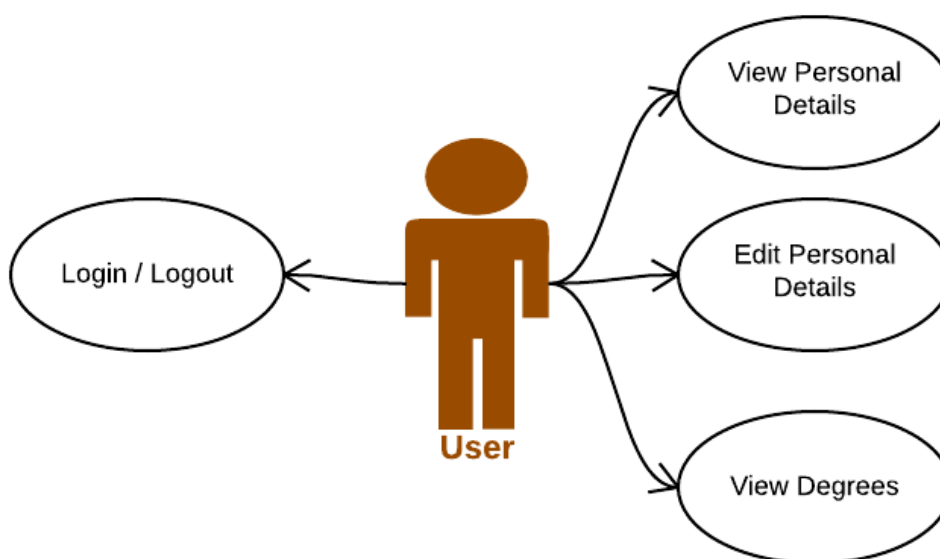


ILUSTRACIÓN 3: CASOS DE USO DE USUARIO

Todo usuario dado de alta en la aplicación puede llevar a cabo varias acciones en el sistema como conectarse a la aplicación y modificar sus datos personales, además de poder consultar toda la información relativa a las distintas titulaciones ofertadas.

Es decir, un usuario puede ver los módulos, materias, asignaturas, competencias y resultados de aprendizaje asociados a la titulación.

## 2.2.2 Casos de Uso: Estudiante

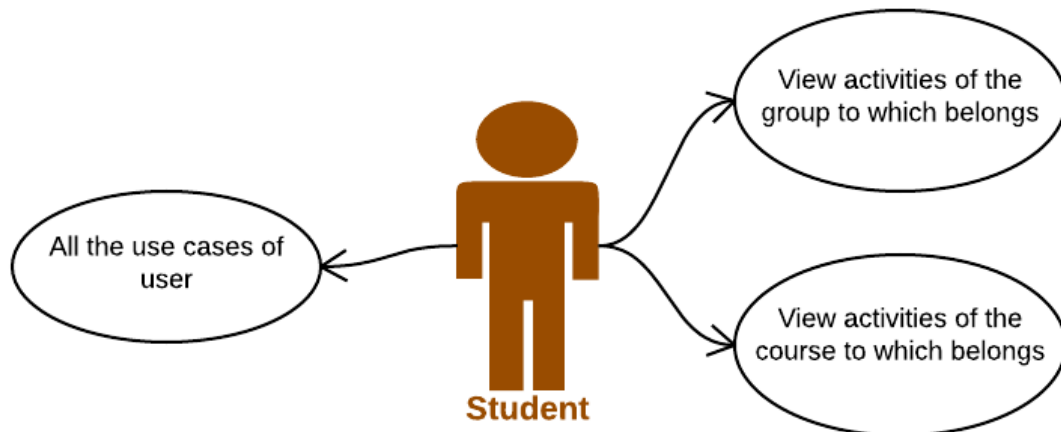


ILUSTRACIÓN 4: CASOS DE USO DE ESTUDIANTE

Todo estudiante dado de alta en la aplicación tiene acceso a todos aquellos grupos en los que participe, es decir, aquellos grupos en los que se encuentre matriculado durante el curso académico. Cabe destacar que, como se verá más adelante, los alumnos son añadidos por el profesor.

Este tipo de usuario solo tiene permisos de lectura. No puede realizar ninguna modificación en el sistema, únicamente la relativa a su perfil de usuario.

### 2.2.3 Casos de Uso: Profesor

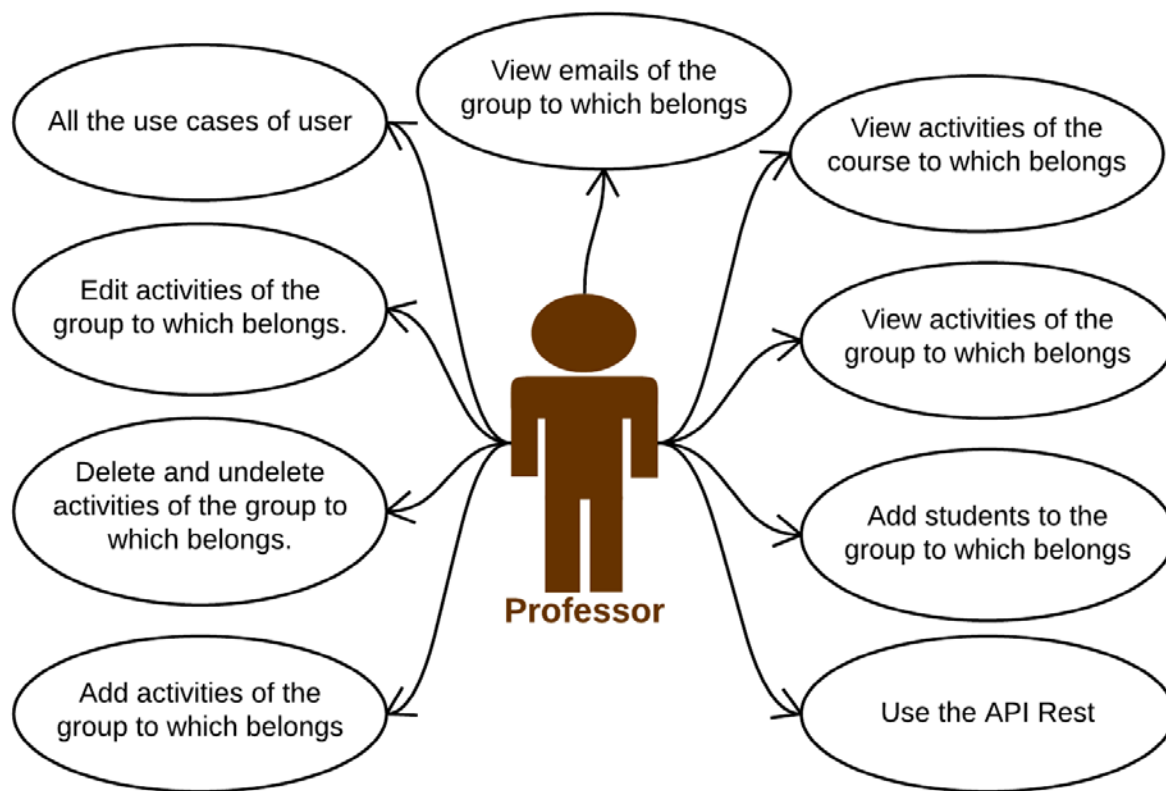


ILUSTRACIÓN 5: CASOS DE USO DE PROFESOR

Todo profesor puede estar dado de alta en uno o más grupos en los que imparte clase y en un mismo grupo pueden dar clase uno o más profesores. Estos, pueden dar de alta a los estudiantes en su grupo.

Además de consultar la información de esos grupos, pueden gestionar las actividades propias de los mismos. Es decir, todo profesor puede dar de alta, consultar, modificar, borrar y restaurar toda actividad que pertenezca a ese grupo. Igualmente, pueden consultar las actividades comunes del curso al que pertenece ese grupo.

Asimismo, pueden agregar actividades a través de la API REST que se ofrece o hacer uso del buzón de mensajes para comunicarse con otros profesores del grupo.

## 2.2.4 Casos de Uso: Coordinador

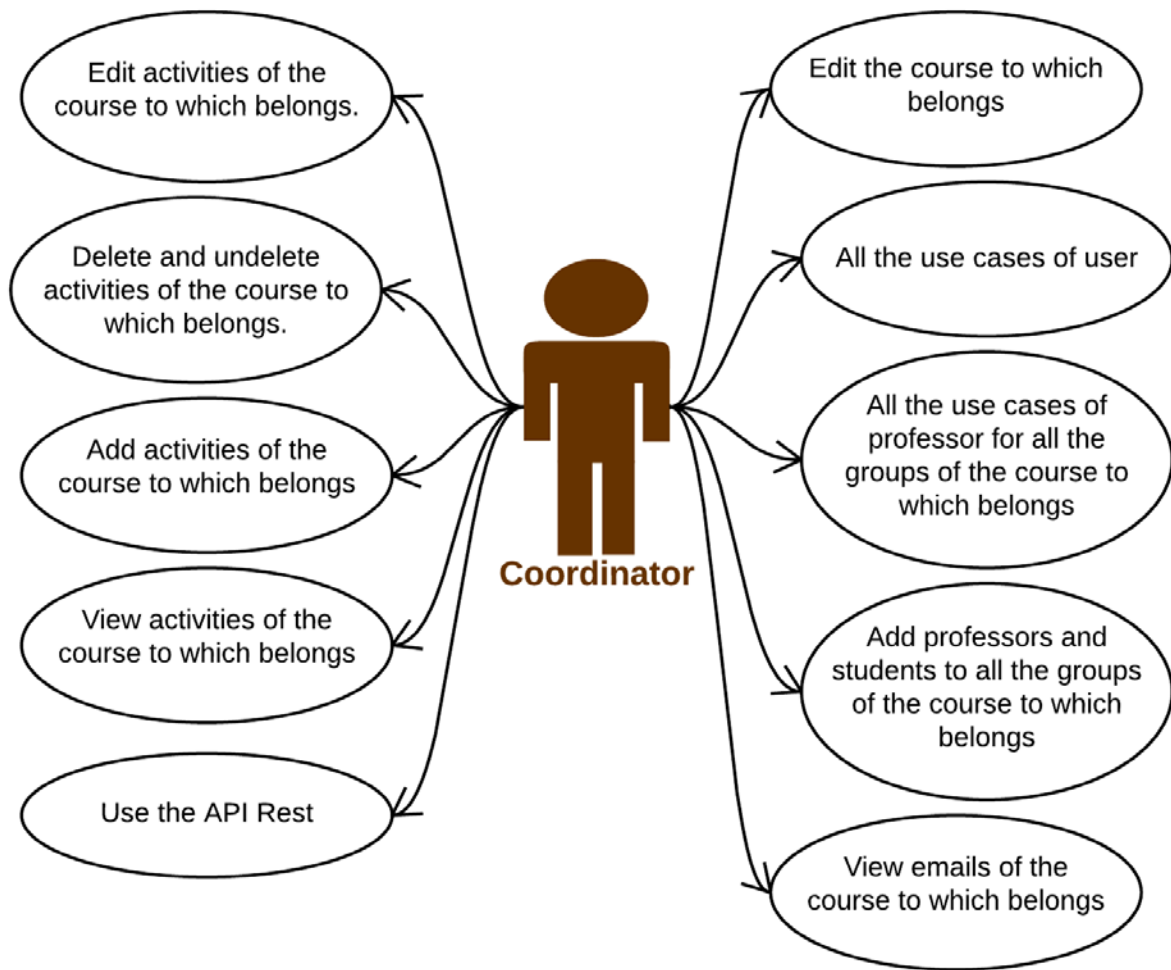


ILUSTRACIÓN 6: CASOS DE USO DE COORDINADOR

El coordinador de un curso puede gestionar las actividades comunes y propias de todos los grupos abiertos del mismo, manteniendo la coherencia entre ellos.

Al igual que el profesor, este puede hacer uso de la API REST que se ofrece para agregar actividades, y usar el buzón de mensajes para comunicarse con otros profesores.

Como máximo responsable del curso, es el encargado de dar de alta a los profesores en los grupos de este y tiene la posibilidad de añadir a los estudiantes a cada uno de ellos.

## 2.2.5 Casos de Uso: Administrador

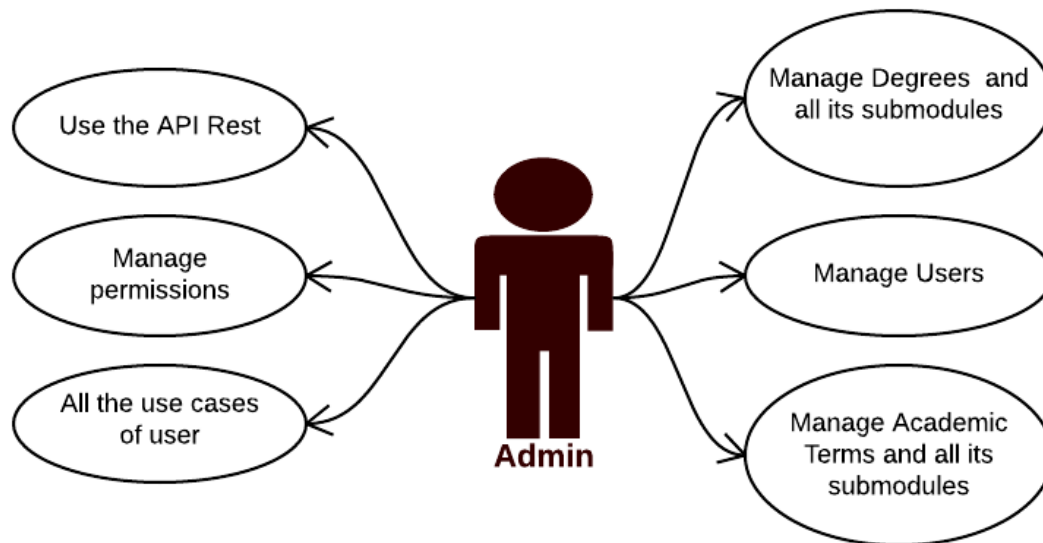


ILUSTRACIÓN 7: CASOS DE USO DE ADMINISTRADOR

El administrador del sistema es el encargado de gestionar toda la aplicación.

Entre sus tareas están:

- Dar de alta a los usuarios del sistema.
- Dar, eliminar permisos a los usuarios para acceder a la parte privada de la aplicación.
- Gestionar todos los módulos del sistema. (Véase la sección 3.2 Módulos Gestionados)
- Añadir actividades a Cursos y Grupos a través de la API REST que se ofrece



# Capítulo 3. Descripción de la Herramienta

En este capítulo se describe detalladamente el modelo de datos que se ha definido para la implementación de la herramienta, así como la descripción de cada uno de los módulos que se han creado.

Tras analizar el díptico de estudios del Grado en Ingeniería del Software ofrecido por la Universidad Complutense de Madrid se han identificado dos subsistemas esenciales para la correcta implementación del gestor académico. Por un lado está la definición de los estudios superiores ofertados, que incluyen las distintas asignaturas que conforman dicho plan de estudios y sus dependencias modulares, y por otro lado, la gestión de los grados a lo largo de los años en los que se detallan las actividades y metas de aprendizaje, y se lleva a cabo la apertura de los diferentes grupos en los que los alumnos se matriculan cada año.

Desde el punto de vista de un usuario podría definirse como una parte pública del sistema que se refiere a los estudios ofertados por la Facultad y que es de acceso general; y la parte privada en la que los usuarios solo pueden acceder a aquellas secciones en las que toman partido, ya sea como profesor dentro del equipo de docencia o como alumno matriculado en dicha asignatura.

En la siguiente figura, se puede observar dicha división de subsistemas y las relaciones que hay entre ambos. Cabe decir, que a pesar de esta división, ambas partes deben estar cohesionadas.

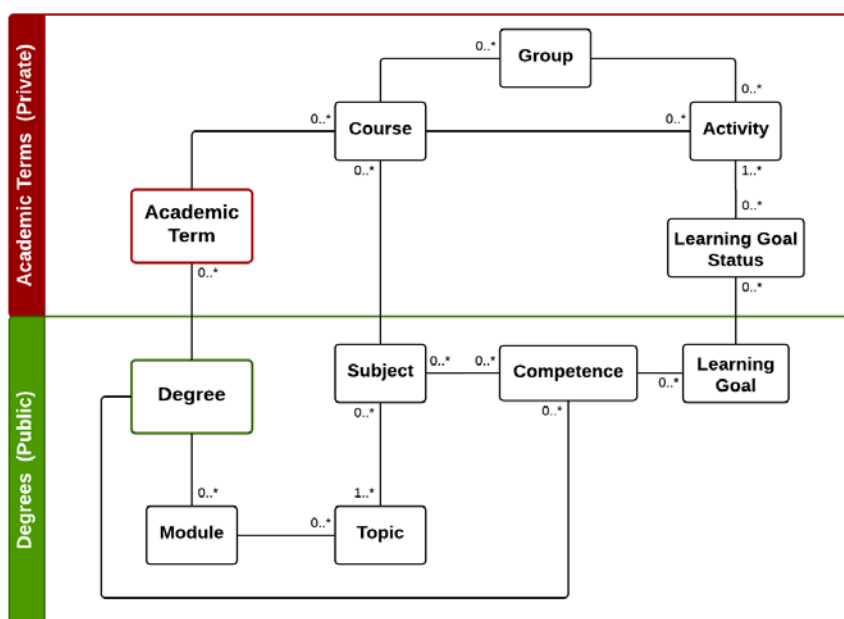


ILUSTRACIÓN 8: MODELO DE DATOS UTILIZADO EN DALGS

### 3.1 Modelo de datos

El modelo de datos se ha dividido según los subsistemas definidos en la sección anterior. La parte privada corresponde a un modelo dinámico que varía cada año y se adapta a los de los profesores, mientras que la parte pública es un modelo estático.

La lista de ítems que conforman dichos modelos se detalla a continuación.

#### Modelo estático o parte pública:

- ❑ **Titulación.** Se define como una titulación de estudios superiores cuya duración varía entre 3 y 6 años dependiendo de la titulación, universidad y país.
- ❑ **Módulo.** Divisiones en las que se organiza una carrera.
- ❑ **Materia.** Unidad académica formada por diferentes asignaturas.
- ❑ **Asignatura.** Cada una de las materias que se enseñan en un curso y que conforman un plan de estudios.
- ❑ **Competencia.** Conocimientos, habilidades, y destrezas que una persona adquiere al durante el estudio de una titulación.
- ❑ **Meta de aprendizaje.** Objetivos concretos de una competencia que los estudiantes pueden adquirir durante el desarrollo de una asignatura.

#### Modelo dinámico o parte privada:

- ❑ **Curso Académico.** Periodo del año en el que una institución educacional imparte estudios pudiendo ser trimestral, cuatrimestral, anual...
- ❑ **Curso.** Unidad mínima de un curso académico. Corresponde directamente a una asignatura concreta de una titulación.
- ❑ **Grupo.** Conjunto de alumnos y profesores en los que se imparte un curso.
- ❑ **Actividad.** Conjunto de tareas que el alumno debe de realizar para la consecución de ECTS.
- ❑ **Resultado de aprendizaje.** Peso o valor de una actividad para adquirir una competencia

## 3.2 Módulos Gestionados


Cómo ha podido verse, DALGS está formado por un total de 12 módulos independientes. En esta sección se detallan cada uno de los módulos citados anteriormente.

### 3.2.1 Módulo de Gestión de Usuarios

Este módulo permite gestionar a los actores que interactúan con el sistema. Se usa para para realizar diversas tareas como por ejemplo dar de alta a nuevos usuarios, modificar sus datos personales, restringir el acceso a los usuarios o asociar contenido a ciertos usuarios.

Viene representado por la clase *User* que es pieza principal del sistema de autenticación. Los principales atributos que identifican a los usuarios son el *username* y el *email*. Además cada usuario tiene una colección de roles que les permitirán acceder a ciertos recursos.

En líneas generales, el único usuario al que se le permite actuar sobre este módulo es el administrador. Sin embargo los profesores adquieren ciertos privilegios sobre sobre los cursos o grupos en los que imparten clase o forman parte del equipo de docencia.



The screenshot shows the DALGS TFG 2014/2015 user management interface. The header includes the DALGS logo and the Universidad Complutense de Madrid logo. The navigation bar contains links for Home, Admin, External, MailBox, and Logout, along with a CSV export button and a user status indicator (Signed in as admin).

The main content area displays a table titled "User List" with a CSV export button and an "ADD" button. The table has four columns: LastName, FirstName, Email, and a column for actions. The table contains four rows of user data, each with "View" and "Disabled" buttons.

LastName	FirstName	Email	View	Disabled
professorLast	professorFirst	professor@ucm.es	View	Disabled
professor2Last	professor2First	professor2@ucm.es	View	Disabled
coordinatorLast	coordinatorFirst	coordinator@ucm.es	View	Disabled
professor3Last	professor3First	professor3@ucm.es	View	Disabled

ILUSTRACIÓN 9: LISTADO DE PROFESORES

La agregación de participantes en la herramienta puede realizarse o bien de forma manual o bien mediante la importación de un fichero CSV.

### 3.2.2 Módulo de Gestión de Cursos Académicos

Este módulo, se encarga de gestionar los distintos cursos académicos que se abren cada año. Permite crearlos, modificarlos, deshabilitados o restaurarlos. Un curso académico está formado por una titulación y un periodo lectivo, por ejemplo 'GIS 2014/2015'. A su vez, un curso académico contiene una colección de cursos que corresponden a las distintas asignaturas de la titulación en cuestión.



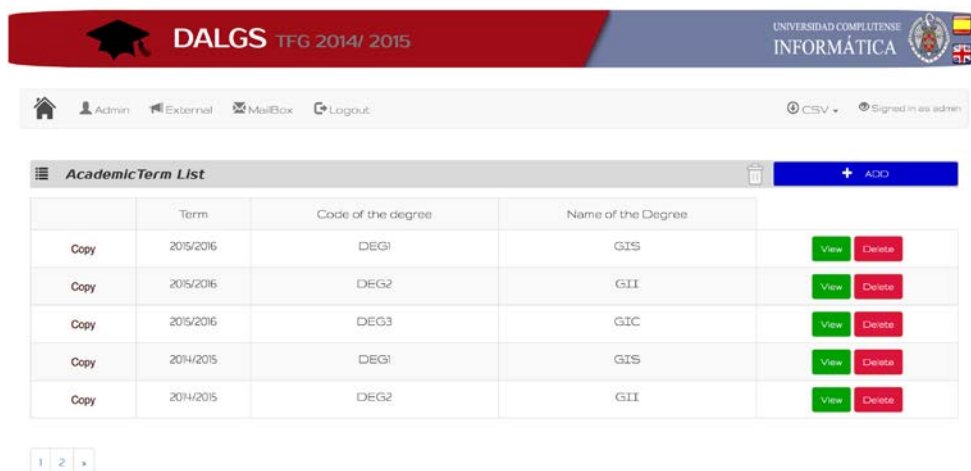
The screenshot shows the 'AcademicTerm Details' interface. At the top, there is a header with 'AcademicTerm Details' and an 'Edit' button. Below the header, the 'Term' is set to '2014/2015' and the 'Name of the Degree' is 'GIS'. The main section is titled 'Course List' and contains a table with three columns: 'Subject Code', 'Subject Name', and a set of 'View' and 'Delete' buttons for each row.

Subject Code	Subject Name	View	Delete
SUB1	IS	View	Delete
SUB2	BD	View	Delete
SUB3	SO	View	Delete

ILUSTRACIÓN 10: VISTA DE UN CURSO ACADÉMICO

El acceso a los cursos académicos está filtrado a nivel de usuario, de este modo sólo los usuarios que participan o se encuentran matriculados en curso académico determinado tienen acceso a sus recursos.

Para facilitar la gestión de los cursos académicos, se ha incluido un método de copia en cascada por lo que el administrador evita el crear, cada año, desde cero los cursos académicos.



The screenshot shows the 'AcademicTerm List' interface. At the top, there is a header with 'DALGS TFG 2014/ 2015' and 'UNIVERSIDAD COMPLUTENSE INFORMÁTICA'. Below the header, there is a navigation bar with 'Admin', 'External', 'MailBox', and 'Logout' buttons. The main section is titled 'AcademicTerm List' and contains a table with five columns: 'Term', 'Code of the degree', 'Name of the Degree', and a set of 'View' and 'Delete' buttons for each row. The table shows five rows, each with a 'Copy' label in the first column.

	Term	Code of the degree	Name of the Degree	View	Delete
Copy	2015/2016	DEG1	GIS	View	Delete
Copy	2015/2016	DEG2	GII	View	Delete
Copy	2015/2016	DEG3	GIC	View	Delete
Copy	2014/2015	DEG1	GIS	View	Delete
Copy	2014/2015	DEG2	GII	View	Delete

ILUSTRACIÓN 11: LISTADO DE CURSOS ACADÉMICOS

### 3.2.3 Módulo de Gestión de Actividades

Uno de los módulos más característicos del sistema al incorporar resultados de aprendizaje. Este módulo se encarga de gestionar las actividades que pertenecen en tanto a cursos como grupos, lo que incluye la creación, modificación, eliminación y restauración de las mismas. Además, cada actividad cuenta con una colección de archivos que formarán el enunciado de la misma, materiales adicionales y otros recursos que los profesores vean convenientes para la correcta realización de la actividad.

The screenshot displays the 'Activity Details' page in the DALGS TFG 2014/2015 system. The page header includes the DALGS logo and the text 'DALGS TFG 2014/ 2015' on the left, and 'UNIVERSIDAD COMPLUTENSE INFORMÁTICA' with a logo on the right. The navigation bar contains links for Home, Admin, External, MailBox, Logout, CSV, and Sign in as admin. The main content area is divided into three sections:

- Activity Details:** Shows the activity code (ACT1), name (Casos de Uso), and description (Desarrollar diagramas de casos de uso). An 'Edit' button is visible in the top right corner.
- Attachments:** A table listing files attached to the activity.
- Learning Goal Status List:** A table showing the learning goals associated with the activity.

File	Description	
Enunciado	Enunciado de la actividad 1	
Material	Material para la actividad 1	

Learning Goal	Weight
learning 1	10

ILUSTRACIÓN 12: VISTA DE UNA ACTIVIDAD

Como se puede observar, se incluyen resultados de aprendizaje que definen la relación con una o varias competencias que conforman la titulación, permitiendo así reflejar la calidad de experiencia tan demandante por ANECA.

### 3.2.4 Módulo de Gestión de Actividades Externas

Este módulo es el encargado de gestionar todas las actividades que son enviadas al sistema mediante llamadas a un servicio REST desde aplicaciones externas, permitiendo la añadir o eliminar una actividad en un grupo o curso concreto.

The image shows two screenshots of a web interface. The top screenshot, titled 'Activity List', features a blue header with a '+ ADD' button. Below it is a table with two columns: 'Name' and 'Description'. It contains two rows: 'Casos de Uso' with description 'Desarrollar diagramas de casos de uso' and 'Entidad-Relacion' with description 'Elaborar un diagrama Entidad-Relacion'. Each row has 'View' and 'Delete' buttons. The bottom screenshot, titled 'External activities list', has a similar layout but includes a third column for 'Group'. It shows one row: 'Relizar un diagrama de Secuencia' with description 'Utilizar Lucidchart', associated with '2014/2015 - BD' and 'group 1'. It includes 'View', 'Move', and 'Delete' buttons.

Name	Description	
Casos de Uso	Desarrollar diagramas de casos de uso	View Delete
Entidad-Relacion	Elaborar un diagrama Entidad-Relacion	View Delete

Name	Description	
Relizar un diagrama de Secuencia	Utilizar Lucidchart	View Move Delete

ILUSTRACIÓN 13: ACTIVIDADES Y ACTIVIDADES EXTERNAS

También se proporciona un listado de todas las actividades externas que aún no han sido tratadas para facilitar dicha tarea.

The image shows a screenshot of the 'External activities list' interface. It has a blue header with a menu icon and the title 'External activities list'. Below the header is a table with three columns: 'Name', 'Course', and 'Group'. It contains two rows: 'Relizar un diagrama de Secuencia' with '2014/2015 - BD' and 'group 1', and 'Diagrama de clases' with '2014/2015 - BD' and 'group 1'. Each row has 'View', 'Move', and 'Delete' buttons.

Name	Course	Group	
Relizar un diagrama de Secuencia	2014/2015 - BD		View Move Delete
Diagrama de clases	2014/2015 - BD	group 1	View Move Delete

ILUSTRACIÓN 14: LISTADO DE LAS ACTIVIDADES EXTERNAS

Los profesores, tendrán que revisar las actividades externas y aceptarlas para que formen parte del conjunto de actividades del grupo o curso correspondiente.

### 3.2.5 Módulo de Gestión de Competencias

Se encarga de establecer y enlazar las distintas competencias definidas en las fichas docentes de las titulaciones actuales. A cada titulación se le asigna una lista de competencias.

Este módulo es el responsable de añadir competencias a las titulaciones, eliminarlas, modificarlas o restaurarlas. Una vez que las competencias se han añadido a una titulación, es posible asociar una competencia a una o varias asignaturas de la titulación.



The screenshot shows a web interface titled "Competence List" with a CSV icon and an "ADD" button. Below the header is a table with two columns: "Name" and "Description". The table contains two rows of data. Each row has a "View" button (green) and a "Delete" button (red) to its right.

Name	Description	
UML	adquirir conocimientos en UML	<a href="#">View</a> <a href="#">Delete</a>
Patrones	aprender diversos patrones de Ingeniería Software	<a href="#">View</a> <a href="#">Delete</a>

ILUSTRACIÓN 15: VISTA DE COMPETENCIAS

### 3.2.6 Módulo de Gestión de Cursos

Un curso, no es más que la consistencia de una asignatura concreta dentro de un curso académico. Un curso es una instanciación concreta de una asignatura para un curso académico concreto y requiere de la asignación de un coordinador que se encargará de supervisar cada uno de los grupos asociados al curso en cuestión.

**Detalles del Curso** Buzón Editar

Curso Académico: 2014/2015  
Coordinador del Curso: .....

**Asignatura**

Asignatura: SUBI-15

**Actividades** + NUEVO

Nombre	Descripción	
Casos de Uso	Desarrollar diagramas de casos de uso	<span>Ver</span> <span>Eliminar</span>
Entidad-Relación	Elaborar un diagrama Entidad-Relación	<span>Ver</span> <span>Eliminar</span>

**Lista de Actividades Externas**

Nombre	Descripción	
Diagrama de clases	Realizar un diagrama de clases	<span>Ver</span> <span>Mover</span> <span>Eliminar</span>

**Grupos** + NUEVO

	Nombre	
Copiar	group 1	<span>Ver</span> <span>Eliminar</span>
Copiar	group 2	<span>Ver</span> <span>Eliminar</span>

ILUSTRACIÓN 16: VISTA DE UN CURSO

Las actividades que se definen en un curso son comunes para todos sus grupos, y solo pueden ser gestionadas o bien por el coordinador del curso, o bien por el administrador del sistema.

Además, cada curso tiene asociado un buzón de mensajes, el cual permite la comunicación entre los profesores que pertenecen a él.

Éste se detalla en el punto [3.4 Gestión de los correos de coordinación](#) de este documento.

### 3.2.7 Módulo de Gestión de Titulaciones

Este módulo se encarga de gestionar las titulaciones ofertadas por la facultad, sean del carácter que sean. Permite añadir, modificar, eliminar o restaurar titulaciones.

The screenshot displays the 'Degree Details' section with the following information:

- Code: DEGI
- Name: CIS
- Description: Grado en Ingeniería del Software

Below this, there are two sections for 'Module LMT' and 'Competence LMT', each with an '+ ADD' button.

**Module LMT**

Name	Description		
Materias Básicas	description1	View	Delete
Materias comunes a la Informática	description2	View	Delete
Tecnología específica	description3	View	Delete
Complementario	description4	View	Delete
Optativo	description5	View	Delete
Trabajo fin de grado	description6	View	Delete
Materias comunes a la Informática	description7	View	Delete

**Competence LMT**

Name	Description		
LML	adquirir conocimientos en LML	View	Delete
Patrones	aprender diversos patrones de Ingeniería Software	View	Delete

ILUSTRACIÓN 17: VISTA DE UNA TITULACIÓN

La visualización de las titulaciones está permitida a cualquier usuario que se encuentre autenticado en el sistema.

### 3.2.8 Módulo de Gestión de Grupos

Sirve para gestionar los distintos grupos que se abren cada año sobre un curso. Cada grupo está caracterizado por una colección de profesores, que conforman su equipo de docencia y una colección de estudiantes que se encuentran matriculados en dicho grupo.

Además, cada grupo tiene su propia lista de actividades que son específicas a dicho grupo y es gestionada por los profesores del grupo. Los profesores del grupo también pueden modificar la lista de estudiantes.

The screenshot displays the 'Detalles del Grupo' (Group Details) page. At the top, there is a navigation bar with icons for Home, Admin, Externas, Buzón, and Desconectar, along with user information 'CSV' and 'Logueado como admin'. The main content area is divided into several sections:

- Detalles del Grupo:** Shows the group name 'group1 (15-2014/2015)' and the course coordinator. A red 'Editar' button is visible.
- Actividades - Curso:** A table listing course activities with columns for 'Nombre' and 'Descripción'.

Nombre	Descripción	
Casos de Uso	Desarrollar diagramas de casos de uso	Ver
Entidad-Relacion	Elaborar un diagrama Entidad-Relacion	Ver
- Actividades - Grupo:** A table for group activities with columns for 'Nombre' and 'Descripción'. A blue '+ NUEVO' button is present.
- Lista de Actividades Externas - Grupo:** A table for external activities with columns for 'Nombre', 'Descripción', and action buttons.

Nombre	Descripción	
Diagrama de secuencia	Realizar un diagrama de secuencia	Ver Mover Eliminar
- Bottom Navigation:** Two buttons labeled 'Lista de Profesores' and 'Lista de Alumnos'.

ILUSTRACIÓN 18: VISTA DE UN GRUPO

Para facilitar las tareas de gestión de coordinadores y administradores, se ha incluido la copia de grupos, de modo que si dos grupos van a ser similares en cuanto a actividades, es posible crear un grupo duplicado en base a otro.

Además, cada grupo tiene asociado un buzón de mensajes, el cual permite la comunicación entre los profesores que pertenecen a él. Este se detalla en la sección [3.4 Gestión de los correos](#) de este documento.

### 3.2.9 Módulo de Gestión de Resultados de Aprendizaje

Las habilidades y competencias profesionales son los conocimientos y destrezas que todo alumno conseguirá durante el período en el que se estudia la titulación. Habitualmente las competencias son descripciones abstractas que se concretan mediante los resultados de aprendizaje, es decir, son las actividades educativas que se llevan a cabo para obtener la competencia.



ILUSTRACIÓN 19: VISTA DE UN RESULTADO DE APRENDIZAJE

Este módulo se encarga de gestionar los resultados de aprendizaje que son específicos para las competencias de una asignatura. Este módulo, permite crear, modificar, eliminar y restaurar los resultados de aprendizaje que forman una competencia.



ILUSTRACIÓN 20: RELACIÓN ENTRE COMPETENCIAS Y LOS RESULTADOS DE APRENDIZAJE

### 3.2.10 Módulo de Gestión de Módulos de una titulación

Este módulo gestiona cada uno de los módulos en los que se divide una titulación. Cada uno de ellos está formado por un conjunto de materias, que pueden ser básicas u opcionales. Los módulos permiten estructurar las titulaciones.

A través de este módulo se pueden crear, modificar, eliminar o restaurar los diferentes módulos que conforman las titulaciones.



ILUSTRACIÓN 21: VISTA DE UN MÓDULO DE UNA TITULACIÓN

### 3.2.11 Módulo de Gestión de Asignaturas

Las asignaturas, son las unidades elementales que se imparten cada curso. Una asignatura tiene una colección de competencias que el alumno debe obtener al cursar y aprobar dicha asignatura. Su identificador único en el sistema viene dado por el código interno.



ILUSTRACIÓN 22: VISTA DE UNA ASIGNATURA

### 3.2.12 Módulo de Gestión de Materias

Este módulo se encarga de gestionar cada una de las materias que componen los módulos de las titulaciones. Una materia es un conjunto de asignaturas que comparten una finalidad y competencias comunes. Vienen caracterizadas por un identificador, nombre y descripción a fin de detallar los aspectos generales de dicha materia. Permitiendo la creación, eliminación, modificación y restauración a nivel de módulo.



ILUSTRACIÓN 23: VISTA DE UN MATERIA

### 3.3 Importación y exportación de datos

Los módulos anteriormente descritos permiten gestionar los distintos componentes que forman una titulación de manera individualizada. No obstante, al examinar detenidamente la información docente de cada titulación, se ha observado la gran cantidad de información que está asociada a cada una. Por ejemplo, la titulación del Grado de Ingeniería Informática está compuesta por más de treinta competencias, más de sesenta asignaturas y un alto número de materias.

Igualmente, se ha considerado imprescindible facilitar la labor del administrador y los profesores permitiendo que estos puedan dar de alta listas de usuarios en la aplicación de manera masiva. De este modo es posible que un profesor de un grupo pueda cargar la lista de sus alumnos en el sistema.

Por tanto, ha sido necesario buscar una solución para que el usuario no pierda tiempo en la carga de datos en la aplicación. Esta, se ha encontrado en el uso de ficheros “CSV”, los cuales, *“son un tipo de documento sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas o punto y coma y las filas por saltos de línea”* [ 21 ]. Además, estos ficheros se pueden generar de forma rápida y sencilla, por ejemplo, en una hoja de cálculo de *Microsoft Excel*. Con lo cual, ha sido necesario encontrar una librería de código abierto en *Java*, que aporte la funcionalidad necesaria para el tipo de gestión que se quería llevar a cabo con estos archivos.

Una vez, consultadas varias librerías, se ha decidido usar *Super-CSV*, ya que aporta lectura y escritura de archivos “CSV” de forma sencilla en Java, siendo compatible con el Framework usado (*Spring*).

Además, gracias a la flexibilidad de los *'Cell Processor'*, se consigue automatizar conversiones de tipos de datos (análisis sintáctico y formato fechas, números enteros, booleanos, etc.) haciendo cumplir restricciones como columnas obligatorias o expresiones regulares usadas por ejemplo, para el campo de direcciones de correo electrónico, ya que tienen que tener una estructura particular.

Concretamente, en DALGS se ha utilizado para la importación masiva de datos desde los siguientes módulos: Competencias, Asignaturas, Materias, Módulos y Usuarios, teniendo en cuenta el contexto en el que se encuentra, es decir, las subidas de dichos datos se realizan sobre una titulación, una materia, un módulo concreto, etc.

Ya que se ha permitido la subida de datos a la aplicación, se ha considerado conveniente permitir la bajada ficheros *CSV* con información relevante que hay en DALGS. Es decir, se ha dado cobertura para que el usuario, en este caso el administrador, pueda descargar listas con usuarios de un grupo, módulos, materias o asignaturas que tiene una titulación o un listado de las titulaciones disponibles, y así, poder facilitar la información en el caso de ser requerida.

En la siguiente figura, se muestra la vista que permite gestionar la subida de archivos en la que se incluye una tabla con lo delimitadores más comunes para el manejo de archivo *CSV*, y que también permite seleccionar la combinación que más se adapte al fichero de manera cómoda y eficaz. En caso de que haya habido algún tipo de error durante la operación el usuario será debidamente informado y se le permitirá volver a intentarlo de nuevo.

The screenshot shows a web interface titled "Upload : Subject". It includes a dropdown menu for "Encoding" set to "ISO-8859-1". Under "File Syntax", there are input fields for "Quote char" (containing a double quote), "Delimiter char" (containing a semicolon), and "End of line symbols" (containing "\n"). A "CSV File" section has a "Choose a file" button. A green "Upload" button is located below these fields. At the bottom, a table titled "Most Common delimiters" lists four options with their respective settings.

Constant	Quote char	Delimiter char	End of line symbols
STANDARD-PREFERENCE	"	;	\n
EXCEL-PREFERENCE	"	,	\n
EXCEL-NORTH-EUROPE-PREFERENCE	"	;	\n
TAB-PREFERENCE	"	\t	\n

ILUSTRACIÓN 24: VISTA DE SUBIDA DE ARCHIVOS CSV

### 3.4 Gestión de los correos de coordinación

Uno de los mecanismos principales en la coordinación de los profesores de un curso / asignatura es el correo electrónico. Se ha creído conveniente la inclusión de un buzón de correo electrónico perdurable para que quede constancia de las tareas de coordinación llevadas a cabo por los profesores.

Para su implementación, se ha creado la cuenta de correo electrónico [dalgs.tfg15@gmail.com](mailto:dalgs.tfg15@gmail.com) con el fin de que se use como medio para poder acceder a todos los correos que se envían los profesores. El profesor tiene que poner en copia a este email cuando realice envíos a profesores del grupo o del curso y poner en el asunto del correo el grupo o curso relativo al mensaje que se manda.

Concretamente éste será de las siguientes formas:

- ❑ [group:id]
- ❑ [course:id]

Siendo el id, identificador único del grupo o curso al que se hace referencia, existiendo la posibilidad de tener "Re:" delante, indicando que se trata de una respuesta a un mensaje enviado.

La aplicación consulta todos los correos existentes en la bandeja de entrada de DALGS de manera que, estos son tratados y almacenados en la base de datos para poder mostrarlos posteriormente en el buzón de la aplicación.

Para ello, se guarda la información del mensaje y se procesa el asunto de cada correo para posteriormente asociar el mensaje a cada grupo o curso en función del identificador leído.

Por último, se guarda el texto del mensaje en un fichero que posteriormente podrá ser descargado por el profesor. Una vez realizado este proceso, se borran del buzón de [dalgs.tfg15@gmail.com](mailto:dalgs.tfg15@gmail.com) los correos leídos.

Esta tarea se ha programado para que se realice cada cierto tiempo en segundo plano y así no interferir sobre las distintas funcionalidades de la aplicación.

Cuando un profesor accede al buzón de un grupo o curso puede ver la información de la siguiente forma:

- ❑ De (From): usuario del que proviene el mensaje.
- ❑ A (To): destinatario/s del mensaje.
- ❑ Asunto (Subject): asunto del mensaje.
- ❑ Respuestas (Replies): muestra el número de respuestas del mensaje o "Reply" si se trata de una respuesta.
- ❑ Mensaje (Message): se trata de un fichero adjunto con el contenido del mensaje.

A continuación se muestra una captura del buzón de mensajes de la aplicación, donde se han ocultado las direcciones ajenas para mantener su privacidad.



ILUSTRACIÓN 25: VISTA DEL CORREO

Al igual que los recursos de las actividades, los mensajes pueden ser descargados mediante un simple clic.

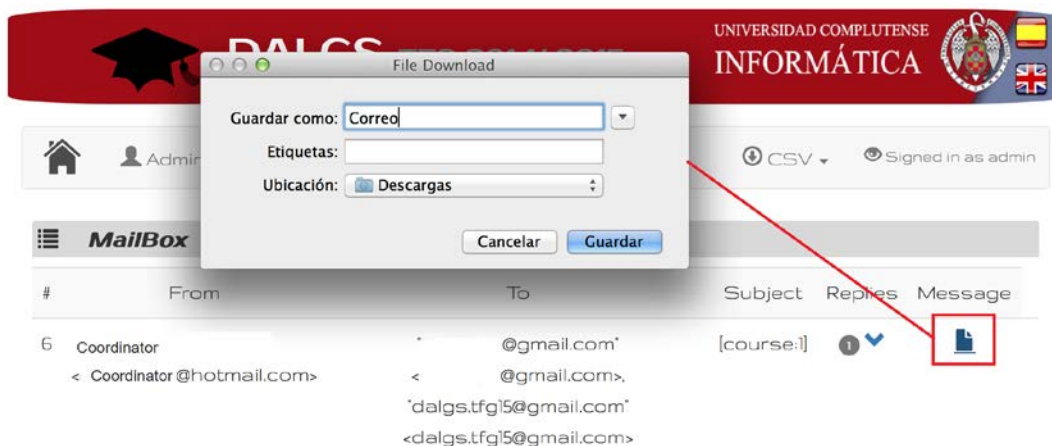


ILUSTRACIÓN 26: DESCARGA DEL CORREO



# Capítulo 4. Diseño

A lo largo del presente curso académico se ha ido desarrollando este sistema y cambiando los requisitos del proyecto, y por ende su arquitectura.

En este capítulo se especifica la arquitectura de la aplicación y su diseño, mediante la utilización del patrón de diseño Modelo-Vista-Controlador

## 4.1 Arquitectura de la aplicación

A continuación se detalla la arquitectura que se ha utilizado para el desarrollo del sistema. Este, concretamente se ha llevado a cabo mediante una programación por capas o arquitectura multicapa, ya que permite separar la lógica de negocio de la presentación de usuario y de la capa de datos. Es decir, cada capa es independiente, permitiendo un alto nivel de abstracción sobre el resto de capas.

Con esto se ha conseguido una aplicación modular, cohesiva y poco acoplada, permitiendo que todos los módulos estén estrechamente relacionados entre sí y al mismo tiempo garantizando que cualquier cambio en una capa, no tenga consecuencias relevantes sobre el resto de la aplicación.

Se ha optado por esta arquitectura ya que ofrece varias ventajas:

- ❑ Es independiente de la interfaz de usuario.
- ❑ Cualquier cambio en el almacén de datos no implica realizar grandes modificaciones en el resto de la aplicación.
- ❑ Lógica de negocio separada del resto de la aplicación.
- ❑ Cualquier modificación en el sistema afecta a la capa en la que se ha realizado dicho cambio.

Exactamente en DALGS, se pueden encontrar tres capas bien diferenciadas:

- ❑ **Capa de Presentación (Web):** Se compone de todas las vistas y controladores de la aplicación. Estos, tienen como objetivo mostrar al usuario la información que solicita en cada momento. El controlador, una vez recibe la petición del usuario, la envía a la capa de negocio y espera a recibir una respuesta.
  
- ❑ **Capa de Servicio (Servicio):** Es la capa que abarca toda la lógica de negocio de la aplicación y está compuesta por todos los servicios de la misma. Una vez que reciben la petición del controlador realizan todas las acciones necesarias para satisfacerla. Si es necesario, el servicio hace uso de la capa de integración o de datos para cumplir su objetivo.
  
- ❑ **Capa de Integración (Repositorio):** Esta capa tiene como función realizar el acceso a la unidad de persistencia de la aplicación. En caso de recibir una petición del servicio, esta realiza el almacenamiento o la recuperación contra la base de datos.

Cada módulo descrito en la sección [3.2 Módulos Gestionados](#) del presente documento consta de un controlador que contiene todos los métodos, los cuales, recogen todas las acciones que el usuario puede desempeñar en la interfaz. También dispone de un servicio, con la lógica de negocio, y de un repositorio que tiene acceso a los datos persistentes asociados.

Cada servicio tiene todos los métodos necesarios para llevar a cabo las tareas que se exigen en cada módulo. No obstante, se dan casos en los que un servicio requiere información de otro módulo, para ello requiere interactuar con su servicio.

Los siguientes diagramas presentan las capas y los distintos módulos existentes en la aplicación, y cómo estos interactúan entre sí a través de los servicios definidos en cada uno.

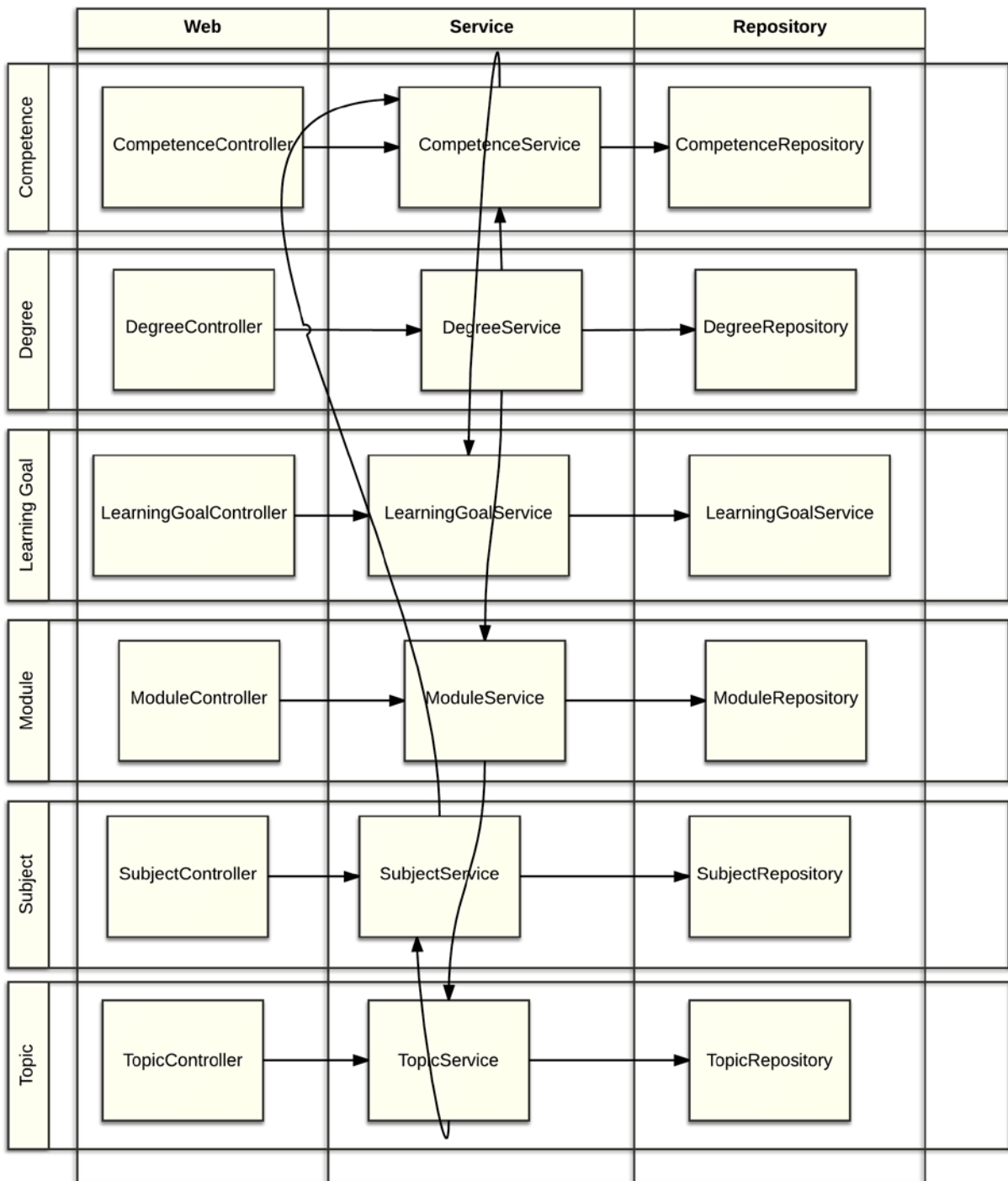


ILUSTRACIÓN 27: ARQUITECTURA DE LOS MÓDULOS DE GESTIÓN DE TITULACIONES

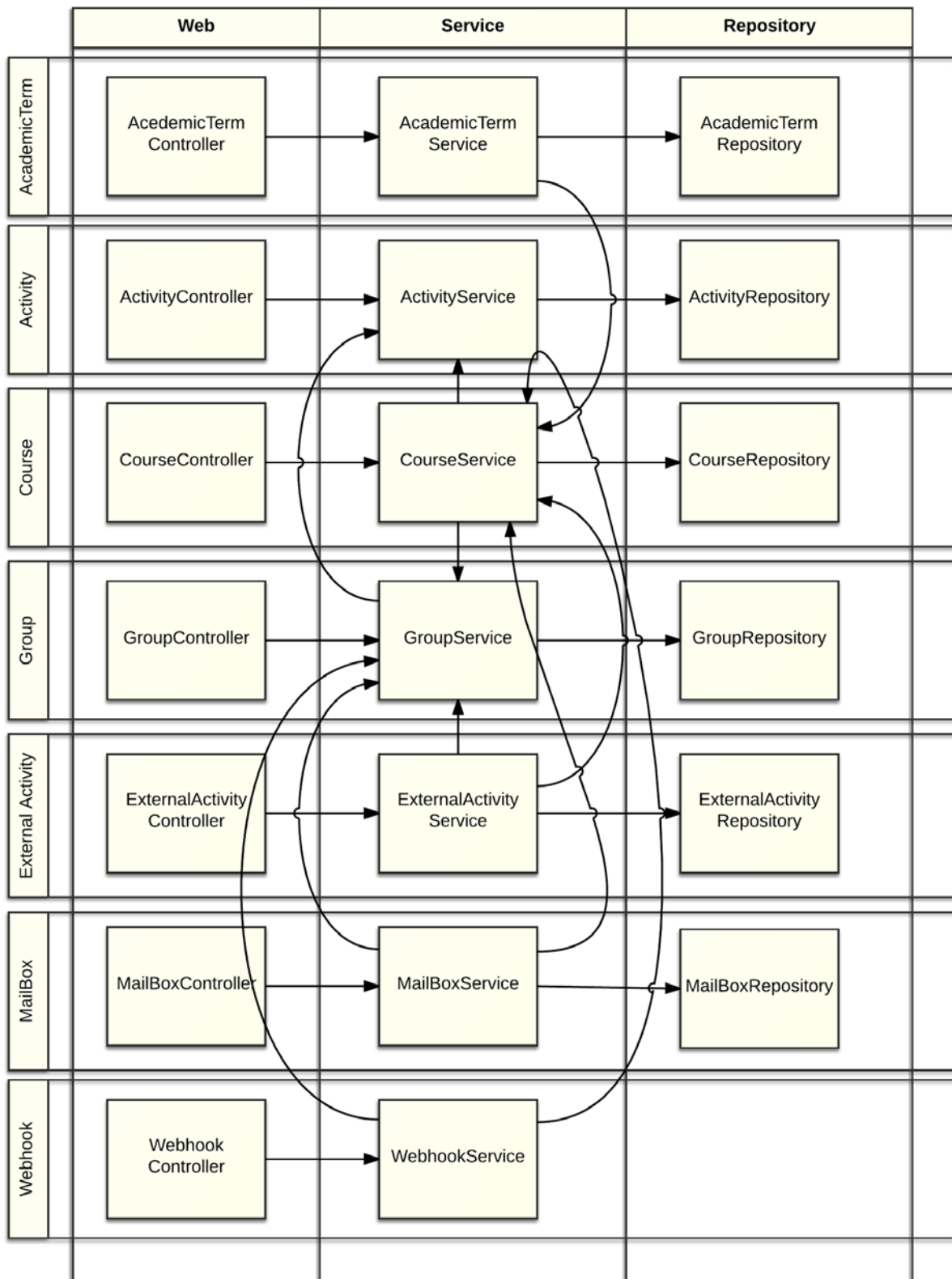


ILUSTRACIÓN 28: ARQUITECTURA DE LOS MÓDULOS DE GESTIÓN DE CURSOS ACADÉMICOS

Para comunicarse la capa de servicio con la capa web, se ha optado por el uso de un objeto contenedor de la información y de los errores obtenidos de la capa de servicio.

#### 4.4.1 ResultClass<E>

Una vez desarrollados los módulos, se ha observado que era de vital importancia mostrar al usuario determinados errores que se producían durante el proceso. Por ello se ha decidido realizar una clase que se ha denominado *ResultClass* y que es usada para el tratamiento de errores que se producen al realizar una determinada acción en el sistema. En caso de no producirse, guarda el resultado de dicha operación.

Esta se compone de los siguientes elementos:

- ❑ Una colección que almacena el resultado obtenido en el caso de que no haya habido errores. El tipo retornado es del tipo <E>, establecido al crear una instancia de esta clase.
- ❑ Un atributo booleano “*has errors*” que indica si se ha producido algún error.
- ❑ Una colección de errores “*errorsList*” que almacena todos los errores que se han encontrado al llevar a cabo la operación.
- ❑ Un atributo booleano “*isDeleted*” para indicar si un elemento ha sido borrado correctamente.

Esta clase, solo se usa en la capa de servicios, cada método del servicio devuelve un objeto *ResultClass* a quién le haya invocado, ya sea un método del controlador o desde otro servicio.

Cuando el controlador recibe el objeto, carga la vista en función del resultado obtenido, redirigiendo a una página de error, mostrando el formulario de vuelta con los campos conflictivos remarcados en rojo, o bien mostrando la vista oportuna en caso de que no haya habido errores.



The screenshot shows a web application interface. At the top, there is a navigation bar with a home icon, a user profile icon labeled 'Admin', and several menu items: 'Externas', 'Buzón', and 'Desconectar'. On the right side of the navigation bar, there is a 'CSV' icon and a 'Logueado como admin' indicator. Below the navigation bar, there is a header for a form titled '+ Añadir Grado'. The form contains three input fields: 'Código:' with the value 'DECI', 'Nombre:' with the value 'Ingeniería Informática', and 'Descripción:' with the value 'Descripción de la titulación'. A green 'Añadir' button is located at the bottom left of the form. Below the form, there is a red error message that reads: 'Errores: El código ya existe'.

ILUSTRACIÓN 29: USO DE RESULTCLASS

#### 4.1.2 Patrones de diseño: MVC

El *framework* de Spring MVC está diseñado alrededor de una pieza central, el *DispatcherServlet* que se encarga de enviar solicitudes a los *handlers*. Este *framework* es totalmente configurable; permite configurar tanto el mapeo de los controladores, como el idioma y la zona horaria de la aplicación, e incluso dar soporte para la carga de archivos. (V. 4.1.4.RELEASE) [4] [7]

El controlador se basa en anotaciones del tipo *@Controller* y *@RequestMapping*, con lo que ofrece la definición de métodos flexibles.

El siguiente diagrama detalla el flujo de funcionamiento de Spring MVC:

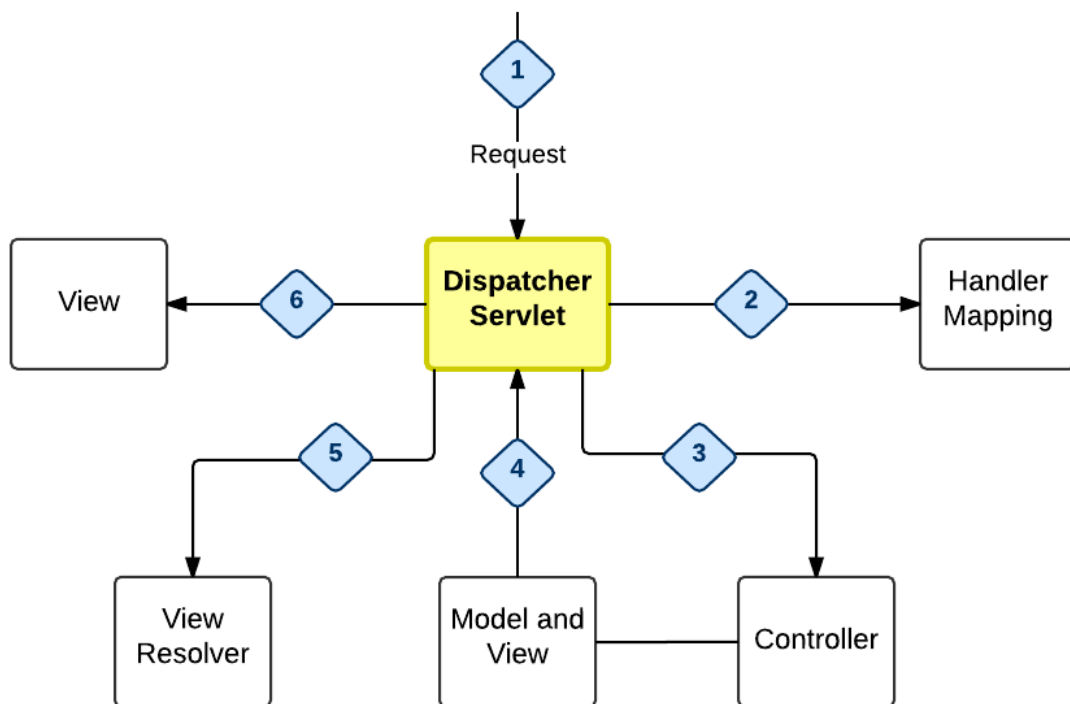


ILUSTRACIÓN 30: ARQUITECTURA DE LA APLICACIÓN

- 1) Una petición llega al *Dispatcher Servlet*.
- 2) El *Dispatcher Servlet* analiza dicha petición y busca el manejador asociado a la url de la petición (*Handler Mapping*). Es decir, busca entre todos los *@Controller* definidos el *@RequestMapping* que mapea la url.
- 3) Una vez localizado el *Controller*, éste gestionará la petición. Llamará a la capa de servicios y que aplicará la lógica de negocio de la aplicación.
- 4) Tras finalizar el tratamiento de la petición, el controlador devolverá un objeto de tipo *ModelAndView* al *Dispatcher Servlet*. Este objeto contiene tanto el nombre de la vista que debe cargar como todos los objetos necesarios para su correcta visualización. En otras palabras, el *ModelAndView* contiene toda la información necesaria para la vista.
- 5) El *Dispatcher Servlet*, a través del *View Resolver*, deberá asociar el nombre de la vista a la vista concreta (en este caso una *.jsp*) y pasarle el *Model*.
- 6) Finalmente, el *View* (la vista) se rellena con los objetos del modelo, y con ello termina el tratamiento de la petición.

#### 4.1.3 Modelo

El modelo es la parte de la aplicación que representa la lógica de negocio y administra los datos de la aplicación. No obstante, la lógica de persistencia debe permanecer en una capa distinta a la lógica de negocio para lograr una mayor modularidad.

Además, el modelo es independiente de la vista, ya que solo recibe la petición y la procesa, garantizando la encapsulación del estado de la aplicación.

#### 4.1.4 Vista

Representa la parte de la aplicación con la que el usuario interactúa. A través de la vista, el usuario notifica la acción que quiere que se lleve a cabo. Una vez esta se haya realizado, la vista se actualiza dependiendo del objeto resultante de dicha petición.

### 4.1.5 Controlador

El controlador tiene como única función, recibir la petición del usuario y enviarla al modelo para que ésta se satisfaga. Cuando el modelo termina, informa al controlador que actualiza la vista en función de la información recibida.

## 4.2 API REST

Una de las principales características de la aplicación es permitir que los profesores y coordinadores puedan generar las actividades para cada uno de sus grupos o cursos. No obstante, dichas tareas se puede realizar actualmente en el campus virtual de la Universidad Complutense de Madrid, con lo cual obligaría a los profesores a dar de alta dos veces las mismas actividades, contraponiéndose a una de las ideas por las que se concibió DALGS, la cual, es facilitar el trabajo diario de los profesores.

La solución que se ha encontrado, ha sido permitir que la aplicación pueda recibir eventos de aplicaciones externas, y de esta manera, generar las actividades que ya se han creado previamente desde otra aplicación, como es el caso de Moodle.

Para ello se ha creado una *API REST*, que recibe un objeto *JSON* de tipo Actividad y que el controlador envía al modelo para llevar a cabo dicha petición. Si se ha realizado correctamente, la aplicación retorna la actividad generada, en caso contrario devuelve un *JSON* con el mensaje de error asociado.

Una vez se ha generado correctamente, en la vista de cada curso o grupo se muestran estas actividades externas en un listado distinto al de las actividades generadas desde la misma aplicación. Es decir, se ha implementado un buzón para recoger estas actividades que provienen del exterior.

Una vez que se almacenan en el buzón de actividades externas del curso o grupo, estas pueden ser movidas al listado del resto de actividades. Igualmente se ha implementado la posibilidad de consultar dicha actividad externa o eliminarla en el caso de no ser necesaria.

Uno de los requisitos que se ha tenido en cuenta a la hora crear esta API es que cualquier aplicación que haga uso de esta, esté autorizada. Para ello se ha implementado un sistema de autorización OAuth2 que se explica en la sección [4.4.5 Spring Security OAuth](#) del presente documento.

## 4.3 Persistencia

Como todo sistema, los datos que el usuario gestiona, tienen que almacenarse en un medio persistente, es decir, se ha tenido que almacenar información relevante para el correcto funcionamiento de la aplicación. DALGS usa dos medios persistentes para almacenar la información: ficheros y base de datos.

En primer lugar, se han usado los ficheros para almacenar todos los mensajes que se muestran en las vistas. Concretamente han sido necesarios dos ficheros, uno para los mensajes en *español\_España (es\_ES)* y otro, para los mensajes en *inglés\_UK (en\_UK)*. Estos mensajes son recogidos de los ficheros y cargados en cada una de las vistas en el momento en el que el usuario acceda a ellas.

En segundo lugar se ha usado una base de datos relacional para almacenar toda la información relativa al modelo descrito en el [Capítulo 3. Descripción de la Herramienta](#) de este documento.

De entre los distintos tipos de bases de datos existentes en la actualidad, se ha escogido llevarla a cabo usando el modelo relacional puesto que facilita la implementación de bases de datos ya planificadas como ha sido nuestro caso.

Además, este modelo aporta una serie de ventajas tales como:

- ❑ Garantizar la integridad referencial haciendo que todos los datos sean correctos y sin repeticiones innecesarias y, por tanto, asegurando que la información existente sea coherente.
- ❑ Garantiza la integridad de los datos.
- ❑ Es sencilla de manipular y comprender por el usuario final.

Posiblemente, la gran desventaja que presenta este modelo es el alto tiempo que se requiere para acceder a los datos, y que en este caso se ha decidido asumir en favor de su integridad. Aun así, este inconveniente se ha intentado paliar realizando el menor número de accesos posible al almacén de datos.

El siguiente paso ha sido escoger una herramienta para desarrollar el código en un lenguaje orientado a objetos como Java y que a su vez, permitiera la persistencia de los datos usando el modelo relacional. La solución a este problema ha sido la selección de un ORM (Mapeo objeto-relacional).

Este no es más que “*una técnica de programación para convertir datos en tipos usados por un lenguaje orientado a objetos, utilizando una base de datos relacional como motor de persistencia*” [ 23 ]. Una de sus principales ventajas es la abstracción del modelo de datos permitiendo centrarse en el código a desarrollar.

De todos los *ORMs*, *JPA (Java Persistence API)* satisfacía las necesidades de comunicación entre el modelo orientado a objetos y la base de datos relacional, con lo cual, daba la opción de permitir la gestión de los datos usando código Java.

#### 4.3.1 JPA

“*JPA es un framework de programación que usa datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE)*” [ 24 ].

Existen varias implementaciones de JPA, tales como Hibernate, EclipseLink, OpenJPA o Toplink todas ellas Open Source. De entre todas se ha elegido la primera de ellas para llevar a cabo la persistencia de los datos exactamente por dos razones:

- ❑ Es compatible con Spring MVC.
- ❑ Ofrece seguridad en rendimiento puesto que es una de las implementaciones más usadas y probadas en la actualidad.

JPA está compuesta una serie de elementos que garantizan la comunicación entre el código en Java y el modelo relacional y se explican a continuación:

Cada Entidad, Entity (POJO) definida no es más que la representación de cada uno de los objetos que se almacena en la base de datos, es decir, cada una corresponde a una tabla de la base de datos. Todas ellas componen una unidad de persistencia que son gestionadas, creando o eliminando instancias, por un Gestor de Entidades o Entity Manager con métodos definidos para interactuar con ellas. Una vez que es invocado un método del EntityManager, el objeto pasa automáticamente a formar parte del contexto de persistencia.

JPA también provee una Factoría de Gestor de Entidades o Entity Manager Factory, que no es más que un singleton (patrón que garantiza una única instancia de una clase) y que genera los objetos Entity Manager. Cada Entity Manager Factory está ligado a una unidad de persistencia.

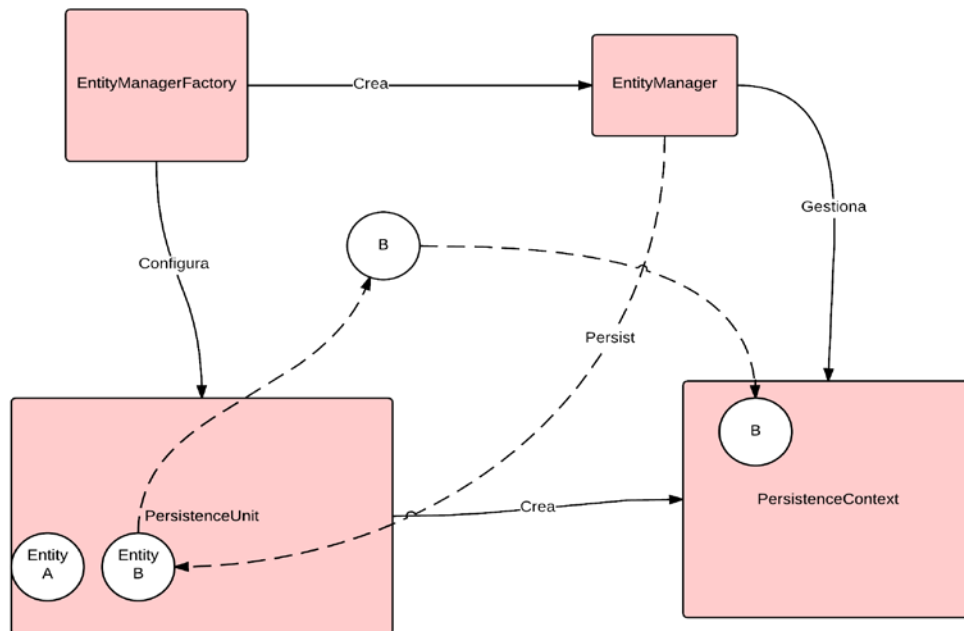


ILUSTRACIÓN 31: JPA

Otro de los elementos fundamentales de JPA son las transacciones, que representan un objeto de ejecución mediante el cual se pueden ejecutar varias operaciones como si fueran una sola. En caso de producirse algún error el proceso se revierte volviendo a su estado anterior. Estas son generadas por una factoría de transacciones y resultan básicas para garantizar la consistencia de los datos.

Por último, cabe destacar JPQL, el cual se trata de un lenguaje de consultas orientado a objetos usado para hacer consultas contra las entidades almacenadas en la base de datos. Además de estar inspirado en SQL, aporta todas las ventajas de un lenguaje orientado a objetos.

Igualmente JPA permite el uso de consultas nativas, es decir consultas en lenguaje SQL, cuyo uso se ha desestimado, ya que se ha pretendido en todo momento hacer un código portable, olvidando las particularidades de cada sistema gestor de bases de datos. De esta forma, se consigue que la aplicación pueda funcionar con un almacén de datos en MySQL, Oracle, etc...

### 4.3.2 MySQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

Para el desarrollo de la aplicación se ha usado como sistema gestor de base de datos MySQL (My Structured Query Language) por diversos motivos.

- ❑ Velocidad al realizar las operaciones.
- ❑ Permite escoger entre varios motores de almacenamiento.
- ❑ Es Open Source, pudiendo hacer ajustes libremente en el código.
- ❑ Se ha usado en otras ocasiones por los integrantes del grupo de trabajo.

Concretamente la base de datos de DALGS se ha almacenado usando MySQL Workbench, que no es más que MySQL con una interfaz de usuario muy intuitiva, que permite la generación de modelos Entidad-Relación y la creación de tablas a partir de ellos.

#### Borrados lógicos

Una de las características que se ha querido implementar en DALGS, es la posibilidad de restaurar información en caso de ser borrada de la aplicación, por ese motivo se ha optado por el uso de borrados lógicos. Es decir, se ha creado un atributo en cada entidad denominado "isDeleted" que guarda información sobre si el objeto ha sido borrado por el usuario o por el contrario, este sigue activo. Este campo únicamente puede tomar dos valores:

- ❑ isDeleted → true: en este caso la aplicación interpreta que el objeto ha sido eliminado por el usuario.
- ❑ isDeleted → false: el objeto permanece activo.

De esta forma se ha conseguido no perder los datos necesarios de la base de datos, para posteriormente poder restaurarlos y al mismo tiempo mostrar información activa en todo momento al usuario.

Únicamente, se ha optado por el borrado físico para las actividades externas, ya que estas no requería restaurarlas.



The screenshot shows a web application interface for managing degrees. At the top, there is a navigation bar with icons for Home, Admin, Externas, Buzón, and Desconectar. On the right, it shows 'CSV' and 'Logueado como admin'. Below this is a header for 'Titulaciones' with a 'CSV' icon and a '+ NUEVO' button. The main content is a table with three columns: 'Código', 'Nombre', and 'Descripción'. The table contains three rows of data. The first two rows have a green 'Restaurar' button, while the third row has 'Ver' and 'Eliminar' buttons.

Código	Nombre	Descripción	
DEG3	GIC	Grado en Ingeniería de Computadores	Restaurar
DEG2	GII	Grado en Ingeniería Informática	Restaurar
DEG1	GIS	Grado en Ingeniería del Software	Ver Eliminar

ILUSTRACIÓN 32: USO DE BORRADOS LÓGICOS

## 4.4 Securitización: Spring Security

La seguridad es un concepto muy sobrecargado en el mundo de las TI que puede definirse de muchas formas diferentes dependiendo del contexto en el que se encuentre. Sin embargo, todas las definiciones coinciden en que se trata de un mecanismo que intenta proteger los recursos sensibles y valiosos contra el uso malintencionado. No hay que olvidar, que dado el crecimiento de Internet y la búsqueda de llegar a más personas abren más las puertas a los ciberdelincuentes que intentan acceder a las aplicaciones de manera ilegítima.

En TI, existen muchas capas de infraestructura y código que pueden ser objeto de ataques maliciosos. Como buenos programadores, debemos asegurarnos que cada una de las capas tiene unos niveles adecuados de protección. Los tres grandes capas de seguridad que conforman una infraestructura TI son la red (firewalls y direcciones IP), el sistema operativo (puertas traseras del SO y virus) y la aplicación en sí. Dada la naturaleza de este proyecto vamos a centrarnos sobre la última capa.

La capa de seguridad de las aplicaciones se refiere a todas las limitaciones que se establecen dentro de una aplicación para asegurarse de que sólo “las personas adecuadas” puedan trabajar a través de la aplicación. La inmensidad de ataques que una aplicación puede sufrir podrían resumirse en: robo de información, suplantación de usuarios, ejecución de actividades restringidas o ilegales, corrupción de datos y acceso al SO, entre otras.

Dado que DALGS se ha desarrollado con Spring MVC, parece razonable cubrir la seguridad de la aplicación mediante Spring Security. Fundado en 2003 por Ben Alex, *“Spring Security se ha convertido en la librería de referencia dentro del mundo Java, para dar soporte a los servicios de autenticación y autorización de una aplicación”* [ 18 ]

Al igual que existen diferentes tipos de ataques maliciosos, la seguridad en una aplicación se logra mediante la aplicación de diversas técnicas. En las siguientes subsecciones se explican algunos de estos procesos y su relevancia e implementación dentro de DALGS.

#### 4.4.1 Autenticación

El proceso de autenticación en una aplicación se encarga de validar y verificar que un usuario particular es quien dice ser. Este proceso se resume en:

- Un usuario (actor del sistema) presenta la solicitud de autenticación con sus datos de acceso (normalmente, nombre de usuario y contraseña).
- La aplicación recibe esta información y comprueba si los credenciales coinciden con algún registro que se encuentra almacenado en la base de datos.
  - ❖ Si la información introducida por el usuario coincide con un registro en el servidor de autenticación, se dice que el usuario se ha autenticado con éxito en el sistema.
  - ❖ Si la información introducida por el usuario no coincide, redirigirá nuevamente al usuario a la página de autenticación, indicando que los datos introducidos no son correctos.

Cabe mencionar, que las contraseñas almacenadas en la base de datos se encuentran codificadas mediante el algoritmo SHA-256 para proporcionar mayor seguridad.

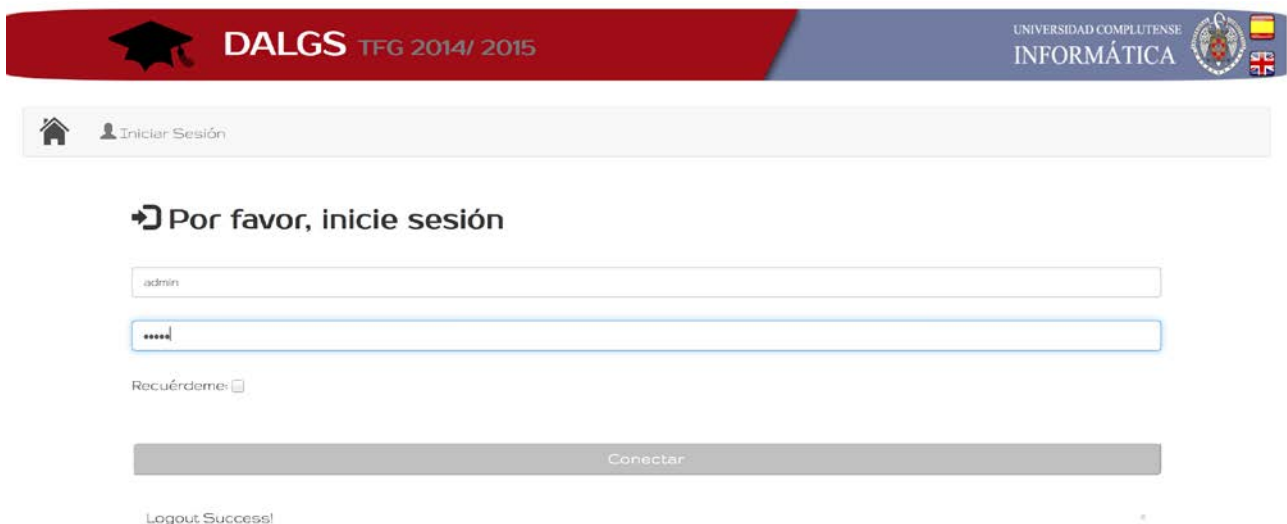


ILUSTRACIÓN 33: LOGIN DE ACCESO

## 4.4.2 Autorización

Resumiendo el punto anterior, cuando un usuario se autentica, significa que el usuario es conocido en el sistema y que ha sido reconocido por el mismo. Sin embargo, esto no quiere decir que el usuario es libre de hacer lo que quiera en dicho sistema.

Por ello, hay que determinar qué acciones se le permiten realizar a un determinado usuario, a que recursos tiene acceso, y por ende asegurarse de que si el usuario no tiene los permisos adecuados sobre una acción en particular no puede llevarla a cabo.

El proceso de autorización compara el conjunto de permisos de un usuario contra el conjunto de permisos necesarios para ejecutar una acción particular en la aplicación, y si se encuentra una coincidencia, el acceso es autorizado. Por otro lado, si no se encuentra una coincidencia, se le niega el acceso redirigiendo al usuario a la página de no autorizado.

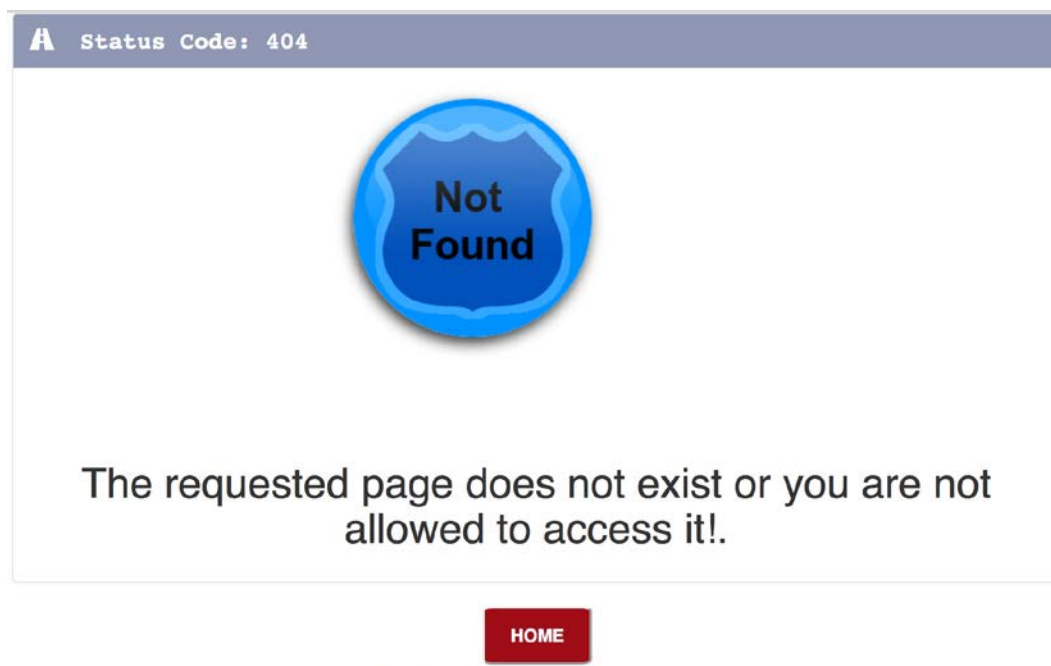


ILUSTRACIÓN 34: ACCESO DENEGADO

### 4.4.3 Gestión de ACLs

Las listas de control de acceso (ACL) son parte del proceso de autorización, el cual se ha explicado en la sección anterior. La diferencia clave es que las ACL normalmente funcionan en un nivel de grano más fino en la aplicación, a nivel de objeto.

ACL puede definirse simplemente como una colección de asignaciones entre los recursos, usuarios y permisos; que permiten establecer reglas como:

- El Profesor-A tiene permiso administrativo en el Grupo-A.
- El Alumno-B tiene permiso de lectura en el Grupo-A

Dichas asignaciones se recuperan de una base de datos externa MySQL que es la que se encarga de mapear dichas relaciones y que está formada por las siguientes *entities*:

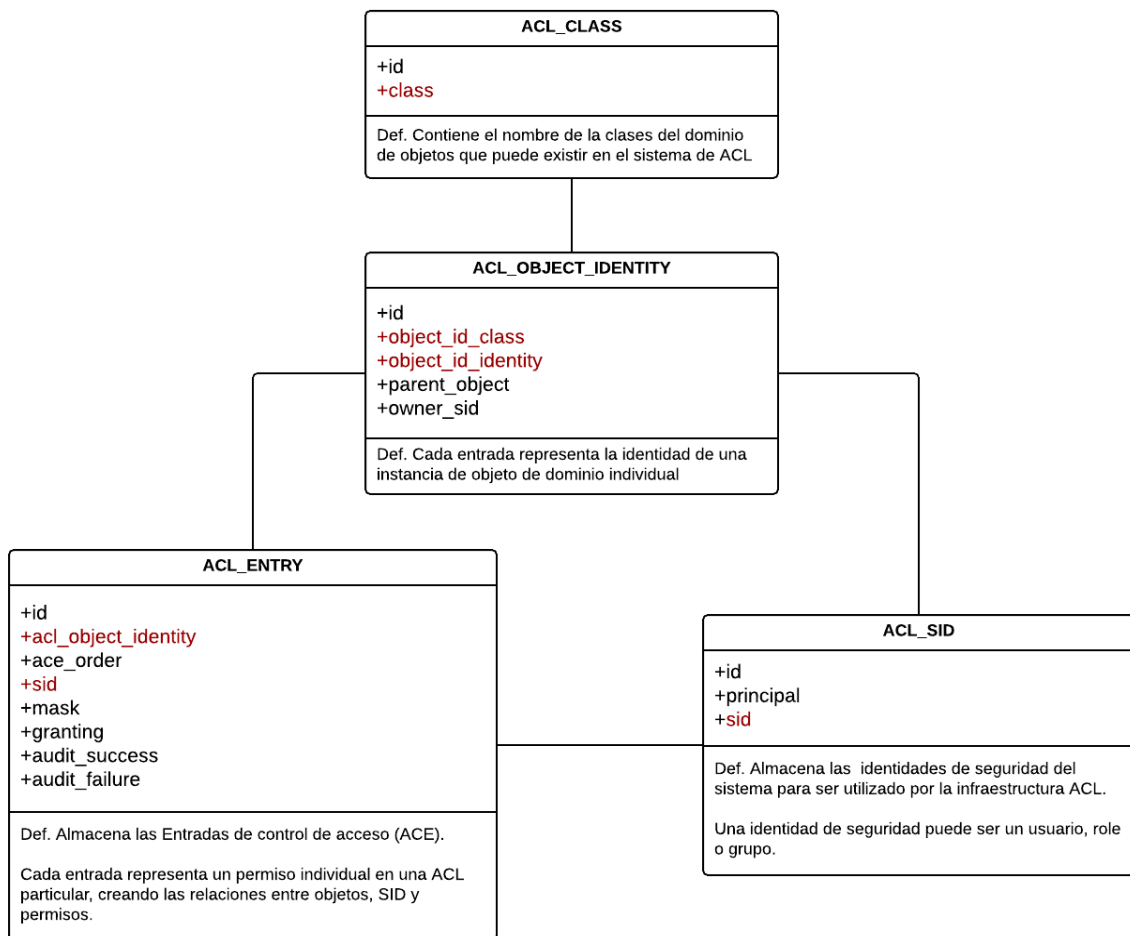


ILUSTRACIÓN 35: DIAGRAMA DE CLASES DE SPRING SECURITY ACL

Cada objeto mapeado en las tablas anteriores pertenecen a un único dueño (SID) que inicialmente es el único autorizado a dar permisos sobre dicho objeto al resto de usuarios del sistema. [ 3 ]

En la siguiente sección se detalla minuciosamente **el conjunto de reglas** que se han definido para cada tipo de usuario.

#### 4.4.4 Spring Security en DALGS

Como se ha mencionado antes, Spring Security proporciona diferentes de niveles de autorización. Para la securización del sistema se han utilizado los siguientes mecanismos:

- Por grupo de usuarios. El acceso a la parte del sistema de titulaciones, al tratarse de una parte pública, se realiza al nivel de autenticación. Es decir, para visualizar dicha sección el usuario solo debe estar autenticado en el sistema.
- Por objeto. El acceso al resto del sistema, cursos académicos y listados de usuarios se realiza o bien a nivel de rol, o bien a nivel de usuario.

Antes de definir las reglas de acceso que se han aplicado, hay que tener en cuenta los distintos permisos con los que las ACL van a trabajar:

1. *Permission.ADMINISTRATION*: Los usuarios que tengan este permiso sobre un objeto están habilitados para dar permisos a otros usuarios sobre dicho objeto.
2. *Permission.WRITE*: Los usuarios que tengan este permiso sobre un objeto están habilitados para modificar dicho objeto, pero no pueden dar permisos a otros usuarios, sea cual sea la naturaleza de dicho permiso.
3. *Permission.READ*: Los usuarios que tengan este permiso sobre un objeto pueden acceder a dicho objeto pero, en ningún caso, podrán realizar modificaciones ya sea sobre el objeto o sobre sus permisos.

Los permisos anteriormente citados son acumulativos. Spring Security proporciona más permisos pero no se han estimado necesarios para definir las reglas que regulan al sistema DALGS. En los siguientes cuadros se muestra cómo se han aplicado dichos permisos para autorizar las acciones básicas de acceso o lectura, escritura y administración sobre los objetos por tipo de usuario.

➤ Administrador del sistema

Post Type	Lectura	Escritura	Administración
Todos los módulos	X	X	X

TABLA 1: PERMISOS DEL ADMINISTRADOR

➤ Coordinadores de Cursos (Aplicable sólo en su/sus Curso/s y grupos que los conforman y el acceso a éstos)

Post Type	Lectura	Escritura	Administración
Curso	X	X	X
Grupo	X	X	X
Resto de módulos	X		

TABLA 2: PERMISOS DEL COORDINADOR

➤ Profesores de Grupos (Aplicable sólo en su/sus Grupo/s y el acceso a éstos)

Post Type	Lectura	Escritura	Administración
Curso	X		
Grupo	X	X	X
Resto de módulos	X		

TABLA 3: PERMISOS DEL PROFESOR

- Alumno(Aplicable sólo en su/sus Grupo/s y el acceso a éstos)

Post Type	Lectura	Escritura	Administración
Grupo	X		
Resto de módulos	X		

TABLA 4: PERMISOS DEL ALUMNO

- Usuario del Sistema

Post Type	Lectura	Escritura	Administración
Grupo	X		
Resto de módulos	X		
Datos personales	X	X	

TABLA 5: PERMISOS DEL USUARIO

Asimismo, el conjunto de reglas según el perfil del usuario son acumulativas siguiendo la siguiente jerarquía:

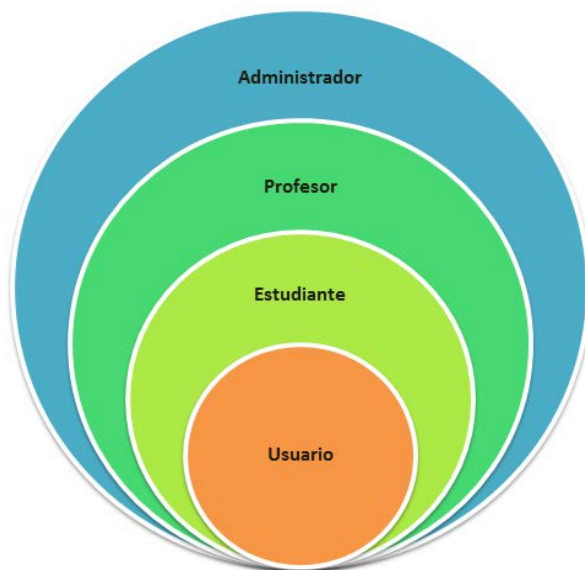


ILUSTRACIÓN 36: NIVELES DE USUARIOS

La definición de los niveles de usuarios sería la siguiente:

- ❖ Nivel Administrador: Tiene todos los accesos y permisos.
- ❖ Nivel Profesor/Coordinador: Tiene acceso general y puede gestionar sus cursos o grupos, a excepción de la inserción de usuarios en los mismos. También tiene acceso a la API Rest.
- ❖ Nivel Estudiante: Tiene el acceso general y acceso a los grupos en los que se encuentra matriculado.
- ❖ Nivel Usuario: Debe estar identificado para poder acceder a la información sobre los distintos grados que se ofertan.

Los conjuntos de reglas definidas en los cuadros anteriores se controlan mediante anotaciones Spring, tanto en la capa de Servicio como en las propias vistas (*.jsp*). Para ello, Spring Security soporta, mediante anotaciones, el uso de expresiones.

- ❑ Expresiones para el control de acceso: **@PreAuthorize** y **@PostAuthorize**. El control de acceso puede ser a nivel de rol (`hasRole`), a nivel de permiso sobre el objeto (`hasPermission`) o a través de la propia autenticación almacenado en el contexto de seguridad. Expresiones de filtrado: **@PreFilter** y **@PostFilter**. Son utilizadas para el filtrado de las colecciones y matrices, usualmente en el valor de retorno de un método.

La administración de los permisos de DALGS se gestiona a través de los métodos que pueden encontrarse en la clase `AcIObjectService.java`, mientras que el uso de anotaciones y expresiones se encuentran en la capa de servicios de la aplicación.

#### 4.4.5 Spring Security OAuth

DALGS, aparte de ser una aplicación web completa, independiente y funcional, también proporciona un servicio web para que aplicaciones de terceros puedan conectarse e interactuar con el sistema a través de una API REST.

El objetivo de esta API reside en la automatización de tareas provenientes de otros sistemas a los que, en un futuro, se encontraría conectados.

La API REST, al igual que el resto de la aplicación, requiere acceso a los distintos recursos del sistema, y aunque el acceso se realiza a través de otras vías, éstas también se encuentran protegidas mediante Spring Security.

Estos recursos, al estar protegidos, requieren autenticación mediante los credenciales del propietario del recurso (típicamente un nombre de usuario y contraseña).

Sin embargo, los propietarios de los recursos no deben ser obligados a compartir sus credenciales a la hora de conceder las solicitudes de acceso, a terceros, a sus recursos protegidos.

Por ello se ha utilizado como mecanismo de autenticación: Spring Security OAuth. [ 5 ]

*“OAuth (Open Authorization) es un protocolo abierto, que permite la autorización por parte de las aplicaciones web o proveedores de servicios web a usuarios (a través de la web, aplicaciones de escritorio o móviles), para poder compartir su información o parte de ella de manera segura.”*

[ 20 ]

OAuth proporciona un método para hacer peticiones HTTP autenticados mediante un *token*, un identificador utilizado para denotar un permiso de acceso con un alcance y duración específico.

[ 5 ]

De este modo, se permite el acceso a dichos recursos mediante la autenticación directamente con el servidor de autorización que a su vez emite un *token* para el cliente con el que puede acceder a los *webhook* definidos en el servidor. Este funcionamiento se detalla en la siguiente ilustración.

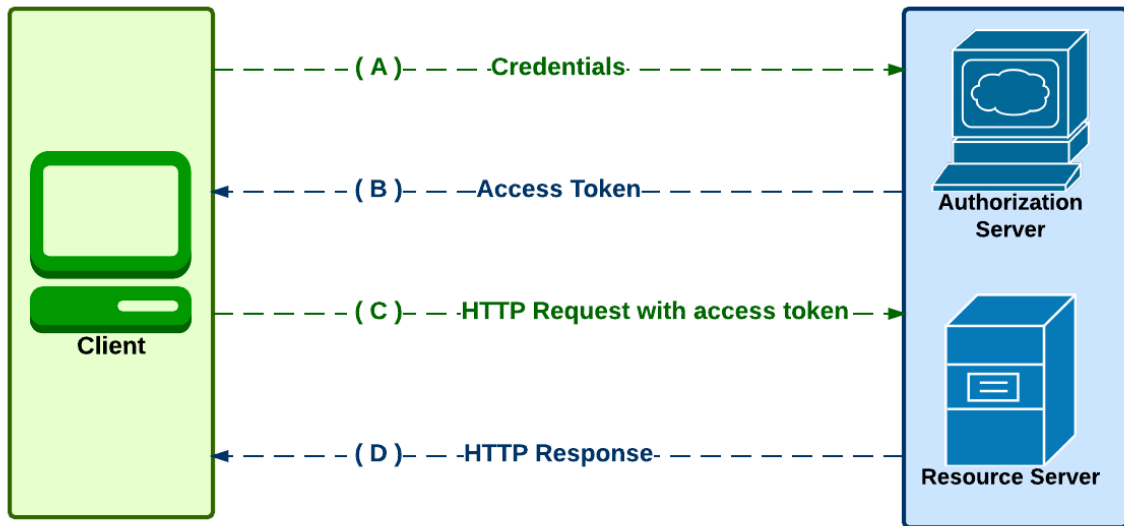


ILUSTRACIÓN 37: INTERACCIÓN CLIENTE-SERVIDOR

En la entrega, se proporciona una herramienta para mostrar la obtención del *token* de acceso y poder probar la API REST.



# Capítulo 5. Desarrollo de la aplicación

Este capítulo describe la metodología de desarrollo utilizada para la implementación del sistema DALGS así como las tecnologías y lenguajes de programación utilizados.

## 5.1 Esquema general de la aplicación

En la siguiente ilustración se muestran las conexiones de DALGS con el resto de sistemas involucrados en la gestión académica.

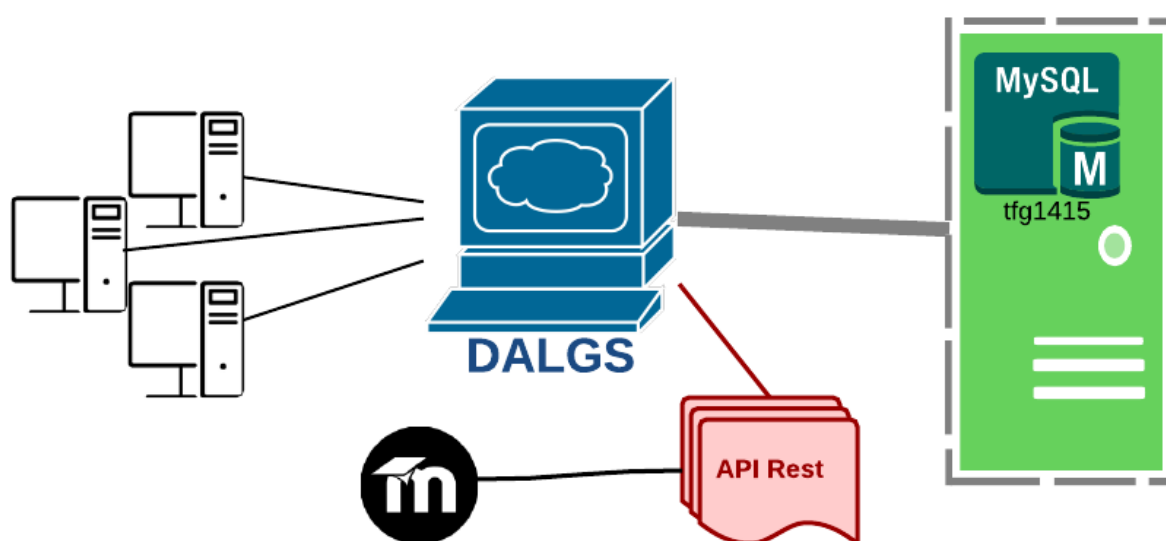


ILUSTRACIÓN 38: ESQUEMA GENERAL DE LA APLICACIÓN

El acceso al sistema DAGLS puede realizarse o bien desde aplicaciones externas a través de la API Rest, o bien desde cualquier cliente conectado directamente a DALGS de forma simultánea. DALGS es el único que se relaciona directamente con la base de datos, la cual se encuentra aislada en una máquina virtual.

## 5.2 Contexto tecnológico

En esta sección se detallan las tecnologías utilizadas para la implementación de DALGS. Especificando sobre los lenguajes de programación utilizados, los *frameworks*, el servidor y el control de versiones

### 5.2.1 Tecnologías utilizadas

En este apartado se muestran los lenguajes de programación y herramientas software que se han utilizado durante la codificación del sistema.

#### Java

DALGS es un proyecto desarrollado principalmente en Java. Se ha utilizado Java frente a otros lenguajes de programación ya que se trata de un lenguaje orientado a objetos lo que simplifica mucho el código.

Otras razones por las que se ha elegido Java es que cuenta con varias comunidades dedicadas a otorgar soporte e infinidad de librerías que multiplican las posibilidades de desarrollo. Sin olvidar que se trata de un lenguaje independiente de la plataforma, es decir, puede ser ejecutado en cualquier tipo de hardware.

#### Maven

Se trata de una herramienta software utilizada en proyectos Java. Nace de la necesidad de modelar el concepto de "proyecto" independientemente del entorno de desarrollo. Puede definirse como una herramienta SCM (Software Configuration Management) que permite controlar las dependencias del proyecto y librerías, los versionados de los proyectos y automatiza la compilación y despliegue de las aplicaciones. [ 37 ]

#### JSON

El web service utiliza formato JSON (JavaScript Object Notation) para el envío y recepción de información.

Se ha utilizado este formato debido a la facilidad de interpretación del mismo, así como la fácil transformación a objetos gracias a librerías como *Jackson*, *gson*.

## JavaEE

JavaEE es una plataforma que permite desarrollar, construir y desplegar aplicaciones empresariales. La plataforma JEE consiste en un conjunto de servicios, *APIs* y protocolos que proporcionan la funcionalidad para desarrollar aplicaciones de varios niveles, en particular, aplicaciones web.

No obstante, debido a la complejidad y rigidez de las *APIs JavaEE* han surgido *frameworks* alternativos que, basándose en un conjunto mínimo de *APIs de JavaEE*, permiten crear aplicaciones empresariales de manera más ágil. Uno de estos *frameworks* es *Spring* que ha sido utilizado para implementar este proyecto y que se describirá más adelante.

## JSP

*JavaServer Pages (JSP)* es una tecnología que se utiliza para crear páginas web dinámicas basadas en HTML y XML. Para su despliegue se requiere un contenedor web JavaEE, en este proyecto se ha utilizado *Apache Tomcat*. [ 40 ]

También pueden considerarse como una alternativa a los servlets, ya que el código JSP se traduce a código de servlet y es quien se encarga de generar las llamadas y respuesta de la aplicación web.

## CSS3

CSS (Cascading Style Sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación [ 41 ]. Permitiendo así la posibilidad de cambiar el estilo de una página web con tan solo cambiar o modificar este archivo. Se ha elegido la última versión ya incluye soporte de *media queries*, es decir, permite crear estilos que se adapten a las dimensiones de pantalla, sea cual sea el tipo de dispositivo.

## Python

Python se trata de un lenguaje de programación caracterizado por la sencillez y velocidad con la que se crean los programas, es un lenguaje muy cercano al lenguaje natural. Además existe una gran cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero. [ 42 ]

Las principales razones de esta elección es que Python es un lenguaje simplificado y rápido, que con pocas líneas de código se consiguen buenos resultados. Además es multiplataforma y puede ser ejecutado en cualquier sistema que tenga instalado un intérprete de Python.

## 5.2.2 Frameworks

En esta sección hablaremos de los *frameworks* utilizados para la implementación del sistema. Algunos de estos *frameworks* ya se han detallado a lo largo de este documento, por lo que se hace más hincapié en aquellos que aún no se han mencionado.

### Bootstrap

En la actualidad, existen muchos *frameworks* para el desarrollo web o *front-end*, entre los que destacan '*Bootstrap*' y '*Foundation*'.

Este tipo de frameworks facilitan el diseño web a los programadores de *back-end*, ya que programador y diseñador web son concepto distintos.

Su uso permite diseñar páginas web de forma más rápida y cómoda al proporcionar estructuras y componentes predefinidos, y de este modo se consigue que las aplicaciones queden con un aspecto más profesional.

Se ha decidido utilizar *Bootstrap*, ya que estamos más familiarizados con él. La característica que más se explota es el diseño *responsive* ya que permite que la aplicación se adapte a las características de la pantalla. También se ha utilizado un *plugin* adicional para *Bootstrap* para generar menús deslizantes como puede verse en la siguiente figura:



ILUSTRACIÓN 39: MENÚ DESLIZANTE CON BOOTSTRAP

## Spring

Spring Framework se trata de una plataforma que actúa como soporte para las aplicaciones Java. Spring es el encargado de garantizar que las dependencias entre módulos y objetos se creen correctamente.

Para ello, basándose en anotaciones Java o en ficheros XML se encarga de crear todos los objetos y de inyectar las dependencias entre ellos, manejando a su interacción y ciclo de vida.

Además, cuenta con una gran lista de módulos que cubren casi todas las necesidades de un proyecto, sea cual sea el carácter de éste. Lo que hace que Spring sea un *framework* muy potente y competitivo en el mundo de la programación.

En la sección [4.4 Securización: Spring Security](#) de este documento se ha hablado sobre Spring Security uno de los módulos más famosos que ofrece Spring.

## Hibernate JPA

Para llevar a cabo la persistencia, se ha escogido la implementación de Hibernate de JPA. Este framework ofrece abstracción sobre el modelo de datos permitiendo centrarse en la persistencia de la aplicación, y olvidando las particularidades de la base de datos.

Se detalla en la [sección 4.3 Persistencia](#)

### 5.2.3 Contenedor Java EE

Ya que la idea de esta aplicación es que brinde experiencia multiusuario, es necesario disponer de un servidor que permita almacenar los datos referentes a los distintos módulos del sistema. Al tratarse de un proyecto beta, nos hemos decantado por utilizar un servidor propio instalado en el mismo ordenador.

En el siguiente gráfico, Simon Maple refleja las preferencias de los desarrolladores a la hora de elegir un servidor basándose en la facilidad de descarga e instalación, las métricas de rendimiento real, las herramientas de apoyo y la facilidad de la configuración del mismo servidor [ 26 ]

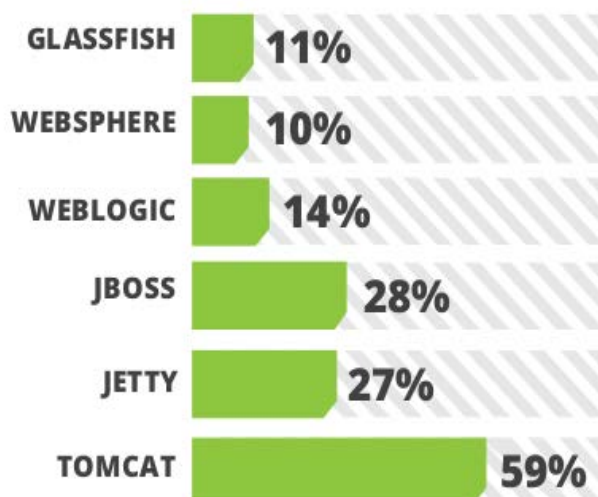


ILUSTRACIÓN 40: COMPARACIÓN DE SERVIDORES [ 26 ]

En este caso, nos hemos decantado por la utilización de Apache Tomcat, debido a su facilidad de uso y a la experiencia obtenida con proyectos anteriores. Además está escrito Java lo que hace que sea multiusuario, tan solo es necesario que el sistema operativo disponga de un máquina virtual Java.



ILUSTRACIÓN 41: APACHE TOMCAT [ 27 ]

“Apache Tomcat (también llamado Jakarta Tomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation (aunque creado por Sun Microsystems).” [ 27 ]

Tomcat incluye el compilador Jasper que es el encargado en compilar las JSPs y las convierte en servlets. DALGS utilizada JSPs para levantar la interfaz gráfica, la vista.

### Base de datos (MySQL)



ILUSTRACIÓN 42: MYSQL [ 16 ]

Como ya se ha especificado anteriormente (Véase la sección [4.3.2 MySQL](#), la base de datos utilizada es MySQL. Se ha elegido este servidor de bases de datos relacional por su versatilidad y por su disponibilidad en múltiples plataformas.

### 5.2.4 Control de versiones

¿Qué es un Sistema de Control de Versiones (VCS)? - Según Scott Chacon y Ben Straub: *“El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante”*. [ 10 ]



ILUSTRACIÓN 43: GIT [ 10 ]

Para el control de versiones de DALGS se ha optado por la utilización de Git ya que ofrece un modelo de ramificación. Dicho modelo nos permite tener múltiples *branches* que pueden ser totalmente independientes entre sí.

Permite también la colaboración, lo que hace que varios desarrolladores trabajando sobre un mismo proyecto en copias totalmente independientes sin intervenir en el desarrollo del otro.

La creación, fusión, eliminación o reversión de cambios en cuestión de segundos hace que el desarrollo sea flexible, cómodo y eficaz.

## 5.3 Herramientas de diseño

En esta sección se van a mostrar brevemente las herramientas utilizadas durante la fase de diseño. Se ha decidido utilizar Lucidchart debido a su facilidad de uso y a la calidad de los diagramas que pueden conseguirse con este software, sobre todo al utilizar una metodología ágil para el diseño del sistema.

### 5.3.1 LucidChart



ILUSTRACIÓN 44: LUCIDCHART [ 34 ]

Lucidchart es una aplicación en línea integrada con Google Drive que se caracteriza por ser una herramienta intuitiva que permite la colaboración.

Ofrece plantillas e infinidad de componentes con las que comenzar a diseñar los diagramas, e incluso te permite cargar tus propias imágenes arrastrándolas sobre el lienzo. Sin duda es una herramienta muy potente que ofrece resultados profesionales. [ 34 ]

## 5.4 Herramientas de desarrollo

La herramienta por excelencia que hemos utilizado para la implementación de DALGS ha sido Spring Tool Suite, puesto que se trata de un proyecto basado en Spring Java. Por otro lado, para controlar y gestionar la base de datos se ha utilizado Sequel Pro.

### 5.4.1 Spring Tool Suite



ILUSTRACIÓN 45: SPRING TOOL SUITE [ 36 ]

Spring Tool Suite (STS), es un entorno de desarrollo basado en Eclipse que incluye todas las herramientas necesarias para desarrollar proyectos Spring preinstalados, es decir, proporciona un entorno listo para implementar, depurar ejecutar y desplegar nuestras aplicaciones.

Las suites de herramientas proporcionan combinaciones listas para el uso de soporte de idiomas, marco de apoyo y soportes de ejecución, y los combinan con el Java existente, y las herramientas Web y Java EE de Eclipse. Además, el apoyo a estas distribuciones siempre está disponible a través de *StackOverflow* [ 36 ].

### 5.4.2 Sequel Pro



ILUSTRACIÓN 46: SEQUEL PRO [ 36 ]

Sequel pro se trata de una herramienta gratuita para Mac OS X que permite gestionar bases de datos MySQL a través de un entorno rápido, amigable y fácil de usar. [ 36 ]



# Capítulo 6. Diario de Trabajo

A lo largo del capítulo se detalla la forma en que, a lo largo del presente curso académico, se ha ido desarrollando este proyecto.

## 6.1 Metodología de trabajo

Para la realización de este proyecto se ha utilizado una metodología ágil, con reuniones semanales y un desarrollo iterativo e incremental.

La metodología ágil, enfatiza las comunicaciones cara a cara en vez de la documentación y se caracteriza por desarrollar software en lapsos de tiempo cortos durante los cuales, los requisitos y soluciones iban evolucionando.

El desarrollo ágil de software se desarrolla en iteraciones que pueden durar entre una y cuatro semanas, en este caso las iteraciones eran semanales. Cada iteración incluye las fases de planificación, análisis de requisitos, diseño, codificación, revisión y documentación.

El objetivo de las iteraciones, no es agregar mucha funcionalidad, sino tener una demo que pueda ser ejecutada sin errores. Durante las reuniones semanales se comentaban los objetivos conseguidos y los problemas encontrados, y se intentaban solucionar. Al final de cada reunión se volvían a evaluar las prioridades del proyecto, y se establecían nuevas metas a desarrollar durante la siguiente iteración. [ 32 ]

## 6.2 Organización y planificación

En la sección anterior se han mencionado las distintas fases de desarrollo de cada iteración que a continuación se van a detallar.

### **1. Fase de Planificación**

Durante esta etapa se define el problema a resolver y las metas del proyecto. También se identifican las restricciones del proyecto.

### **2. Fase de Análisis de requisitos**

En esta etapa se especifican los requisitos, a alto nivel, a desarrollar durante la iteración. También se planifican algunos requisitos para una segunda iteración. Se describen los componentes y módulos que van a formar el sistema.

### **3. Fase de Diseño**

La fase de diseño describe cómo el sistema va a satisfacer los requisitos.

### **4. Fase de Codificación**

Durante esta fase, el software se distribuye entre los programadores y se codifica. Durante la codificación se ha de comprobar que el sistema sigue funcionando correctamente, es decir, verificar que las nuevas modificaciones no corrompen el sistema para seguir manteniendo una versión estable del producto.

### **5. Fase de Revisión**

Una vez que ha concluido la fase de codificación, se asegura que el software ha sido implementado correctamente y de forma óptima, sin redundancia en los métodos y clases.

### **6. Fase de Documentación.**

En esta fase se detalla, los objetivos que han sido desarrollados y su forma. De éste modo, al iniciar este documento no se parte de cero, pudiendo recuperar la información de los informes semanales.

Durante los primeros meses, tuvimos la primera toma de contacto con las herramientas que íbamos a utilizar. Inicialmente este proyecto iba a integrarse con *Open Badges*, pero los requisitos del proyecto y junto a la poca documentación de *Open Badges*, el proyecto tornó hacia un gestor académico completo que incluía el estudio e implementación de los distintos módulos en que se componen las titulaciones, la securización de la aplicación y la creación de una API.

Siendo dos integrantes en el equipo, uno se ha especializado en *Hibernate JPA* mientras que el otro en *Spring Security*, pero en ambos casos se ha trabajado en todas las partes que conforman el proyecto.

## 6.1 Diario mensual de desarrollo

A continuación se detallan las actividades que se ha ido realizando a lo largo del presente curso académico para la implementación de este trabajo.

Fecha: Septiembre - Noviembre
<ul style="list-style-type: none"> <li>- Análisis e instalación de <i>Open Badges</i>.</li> <li>- Primera toma de contacto con Spring Framework generando la aplicación base para la posterior implementación de DALGS</li> <li>- Cambio de requisitos en la propuesta inicial del proyecto</li> </ul>

TABLA 6: DIARIO DE DESARROLLO SEPTIEMBRE - NOVIEMBRE

Fecha: Noviembre - Enero
<ul style="list-style-type: none"> <li>- Estructuración del sistema en titulaciones y cursos académicos</li> <li>- Desarrollo de las relaciones y acciones básicas entre los módulos: ver, añadir, eliminar y modificar</li> <li>- Primera versión estable del proyecto, desde el punto de vista de administrador</li> </ul>

TABLA 7: DIARIO DE DESARROLLO NOVIEMBRE - ENERO

Fecha: Enero - Marzo
<ul style="list-style-type: none"> <li>- Depuración de los módulos creados durante la iteración anterior y cambio en la interfaz del sistema para que se adapte a cada tipo de usuario</li> <li>- Integración de la gestión los resultados de aprendizaje</li> <li>- Securización básica del sistema</li> </ul>

TABLA 8: DIARIO DE DESARROLLO ENERO - MARZO

Fecha: Marzo - Mayo
<ul style="list-style-type: none"> <li>- Securización de la aplicación mediante acl</li> <li>- Gestión de archivos CSV</li> <li>- Métodos de copia de las titulaciones</li> <li>- Unificación de las vistas para añadir y modificar los módulos</li> </ul>

TABLA 9: DIARIO DE DESARROLLO MARZO - MAYO

Fecha: Mayo - Junio
<ul style="list-style-type: none"> <li>- Implementación del buzón de emails</li> <li>- Web Service</li> <li>- Gestión de actividades externas provenientes de aplicaciones de terceros</li> <li>- Depuración del sistema</li> <li>- Primera versión de la memoria</li> <li>- Generación de un almacén de datos para la subida de recursos al sistema</li> </ul>

TABLA 10: DIARIO DE DESARROLLO MAYO - JUNIO

Fecha: Junio
<ul style="list-style-type: none"> <li>- Depuración y cierre de la aplicación</li> <li>- Generación de la memoria final</li> <li>- Preparación de la presentación del trabajo de fin de grado</li> </ul>

TABLA 11: DIARIO DE DESARROLLO JUNIO

# Capítulo 7. Conclusiones

La adaptación a la nueva estructura de las titulaciones del Espacio Europeo de Educación Superior ha supuesto una serie de retos en la estructura de la Universidad española en general y en la Universidad Complutense y la Facultad Informática en particular.

Como parte de estos retos, ha surgido la necesidad de generar una traza de auditoría, en particular, las actividades educativas llevadas a cabo durante los diferentes cursos académicos y su relación con los resultados de aprendizaje y competencias que tienen las diferentes asignaturas que forman parte de la estructura de planes de estudio, y las actividades de coordinación entre los diferentes profesores.

La herramienta DALGS desarrollada durante este proyecto afronta y permite automatizar parte de estas tareas de recolección de evidencias ofreciendo una alternativa a la gestión académica que actualmente se utilizan en la Facultad de Informática.

## 7.1 Resumen de contribuciones

Desde el inicio, el primer propósito que se ha perseguido ha sido conocer en profundidad el nuevo sistema educacional establecido por ANECA y que supone un gran cambio respecto a los anteriores planes de estudios. Para ello se ha realizado un análisis en profundidad de su estructura interna, es decir, de los distintos elementos que constituyen una titulación y de los nuevos componentes que se han de incluir para cumplir con dicha normativa.

Debido a que el sistema implementado debe adaptarse a las necesidades que surgen en cada curso académico ha sido necesario desarrollar una infraestructura que permita llevar a cabo dichos cambios y todas sus tareas asociadas facilitando el trabajo al futuro administrador del sistema y proporcionando una forma rápida, segura y eficaz de gestionar los cursos académicos.

Otro objetivo relevante ha sido la búsqueda por encontrar la forma de evaluar y reflejar los resultados de aprendizaje que se espera que los alumnos desarrollen a lo largo de sus estudios.

Por tanto, para hacer un sistema más completo se ha agregado la funcionalidad de seguimiento de los emails que los profesores de un curso o grupo se intercambian entre ellos para discutir sobre la gestión de los mismos, la asignatura o las actividades que los componen, y de este modo, mantener las conversiones de forma organizada y disponible en todo momento.

Otra funcionalidad destacada, está relacionada con la gestión de los datos del sistema, permitiendo la importación y exportación de los mismos para poder realizar copias de seguridad de la información gestionada por DALGS. Permitiendo la posibilidad de realizar tareas con otras herramientas y posteriormente integrar los cambios dentro de DALGS.

También han de destacarse las técnicas de ingeniería del software utilizadas durante las fases de análisis, diseño e implementación, las cuales han ayudado a la fijación y definición de los objetivos y requisitos que debe cubrir el sistema. Esto nos ha permitido abordar la fase de desarrollo de manera rápida, ordenada y eficaz.

Aunque actualmente esta herramienta ha sido enfocada para los grados universitarios, permite gestionar fácilmente otro tipo de estudios, como es el caso de los másteres, programas de doctorado, títulos propios, etc. Igualmente, aunque DALGS ha sido desarrollada para la Facultad de Informática, podría ser utilizada en otras facultades de la Universidad Complutense o incluso de otras Universidades.

Por último, la internacionalización de la aplicación hace que DALGS sea una herramienta muy útil para los estudiantes de intercambio que se reciben en la Facultad de Informática, como es el caso de los Erasmus. Todas estas cosas hacen que DALGS sea una poderosa herramienta para gestionar el sistema académico.

# Chapter 8. Conclusions

The adaptation of the new studies structure of the European Space of Higher Education has required facing several structural challenges within Spanish universities in general and, in particular, at Complutense University's School of Computer Science and Engineering.

Some of these challenges have required the generation of an audit trail in order to link the learning activities implemented during the different academic terms and their relation to the learning goals and competences defined for the different courses that are part of the degree curriculum and, finally, the coordination activities conducted by the professors who teach these courses.

The DALGS system, the tool developed during this project, addresses and partially supports the automation of these tasks, helping the instructors to gather evidence and offering a powerful alternative to the current tools that are used at the School of Computer Science and Engineering of the Complutense University of Madrid (UCM).

## 8.1 Summary of contributions

This study began with the main purpose of finding out in depth the new education system set by ANECA, which has greatly changed the previous educational system. To achieve this, we have made a thorough analysis of the internal structure of ANECA, that is to say, the different elements that constitute a degree and the new resources that need to be included to fulfill the new rules.

Because the implemented system should be adapted to the needs arising in each academic plan, it has been necessary to provide an infrastructure to carry out such changes and all the tasks associated to make the work of the system administrator easier and to provide him with a fast, safe and effective way to manage the academic plans.

Another important goal was finding the way to assess and show the learning goals that students are expected to develop along their studies.

Therefore, to make a more complete system we added the functionality to monitor the emails that professors and lecturers of a course or a group exchanged among themselves to discuss the management of a subject including its activities, thus keeping the discussions in an organized way and making them available at all times.

Another important feature, is related to the data management system which allows to import and export data to make backups of the system information.

Also we have to highlight the software engineering techniques that we have used during the software development lifecycle, which have helped us to fix and define the objectives and requirements that the system had to cover. This has allowed us to tackle the development phase on a quick, order and efficient way.

Although this tool has been oriented toward university degrees, it can be easily adapted to other studies such as master, doctoral and university-specific degrees. Likewise, DALGS has been developed for the Faculty of Computer Science, but it could also be applied to other university faculties as well.

Lastly, the internationalization of the application makes it a very useful tool for students from other countries such as students on the Erasmus program. All these things make DALGS a powerful tool to manage the academic system.

# Capítulo 9. Trabajo futuro

En este capítulo se comentan algunas de las propuestas que se podrían implementar para continuar y mejorar el proyecto actual, dotándolo de mayor funcionalidad, para finalmente ser implantado en la facultad e integrado con el resto sistema actualmente vigentes en la misma.

## 9.1 Arquitectura multi-tenant

Primero definamos qué es una arquitectura multi-tenant.

*“Las arquitecturas multi-tenant (multi-propietario) son cada vez más utilizadas entre los proveedores de SaaS (Software as a Service). En un entorno multi-tenant, todos los clientes y sus usuarios consumen el servicio desde la misma plataforma tecnológica, el intercambio de todos los componentes de la tecnología incluyendo el modelo de datos, servidores y las capas de base de datos.”*[28]

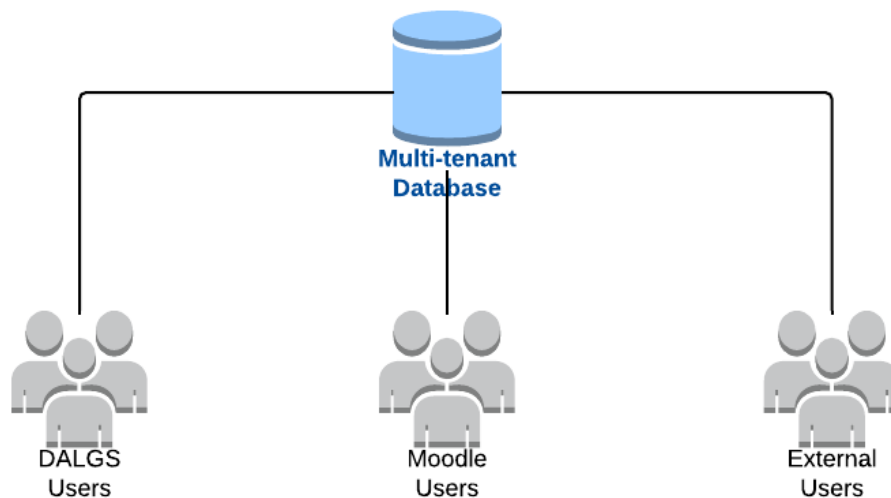


ILUSTRACIÓN 47: ACCESO MULTI-TENANT

Es decir, sobre un recurso operan múltiples usuarios y sistemas integrados. Esto supone un menor coste en tiempo de desarrollo, menor costo en hardware al usar servidores dedicados y mejor rendimiento de acceso a los recursos ya que las consultas se realizan a un mismo servidor.

## 9.2 Experience API

Los términos **TinCan API**, **Experience API** o **xAPI** se refieren a una nueva interfaz de programación de aplicaciones para el estudio del aprendizaje como tal. Permite recopilar datos sobre la amplia gama de experiencias que una persona tiene, basándose en acciones y actividades [ 29 ].



ILUSTRACIÓN 48: TIN CAN API [ 29 ]

En nuestro caso, resulta vital para realizar el seguimiento sobre los resultados de aprendizaje en que se componen las competencias cuando un alumno realiza una actividad. Es una forma de mostrar el progreso de académico de los alumnos.

### 9.3 Mozilla Open Badges

Se trata de un software libre y abierto a cualquier organización que quiera utilizarlo para crear, emitir y verificar insignias digitales [ 30 ].



ILUSTRACIÓN 49: MOZILLA OPEN BADGES [ 30 ]

De igual modo que *xAPI*, va ligado a los resultados de aprendizaje. En este caso, se trata de una forma de representación o reconocimiento de la adquisición de dichos conocimientos o competencias por medio de insignias digitales, y de este modo dar un aspecto más vistoso a la adquisición de competencias y que la información de las mismas pueda ser almacenada en un medallero de insignias, y compartidas a través de las distintas redes sociales profesionales.

Al comienzo del proyecto, se ha estudiado y trabajado con el sistema de *Open Badges y Mozilla Backpack*, pero desgraciadamente no ha podido implantarse en el sistema actual debido al cambio de los requisitos del proyecto, entre otros factores.

## 9.5 Plugin para Moodle

Para la integración de DALGS en el ámbito académico y la realización de las primeras pruebas reales, es necesario desarrollar un **plugin** para Moodle el cual, ante la subida de actividades en los distintos grupos, notifique la creación o cambios a este nuevo sistema con el fin de mantener la correlación entre ambos.



ILUSTRACIÓN 50: MOODLE [ 19 ]

Al finalizar estas pruebas se recomienda realizar una encuesta a los usuarios involucrados, para estudiar el manejo de la aplicación, su efectividad y otros aspectos, así como las mejoras que realizarían en la plataforma o qué cambios harían.

## 9.6 Login con Google

Una de las principales incomodidades en el uso de una nueva plataforma es la necesidad de tener una contraseña única por cada servicio. En este sentido y ya que los alumnos y personal de la Universidad Complutense disponen de una cuenta de Google, sería deseable integrar DALGS con el sistema de autenticación de Google [ 44 ] de modo que los usuarios se autenticarían en DALGS utilizando su cuenta institucional. De este modo DALGS sólo se encargaría de controlar la autorización de usuario para delimitar las acciones que pueden realizar los usuarios.

## 9.7 Otras mejoras

En este apartado comentamos pequeñas modificaciones que sugerimos realizar sobre la versión actual del sistema.

### ❖ Mejoras en la paginación

La paginación actual de la aplicación se lleva a cabo mediante dos consultas, la primera obtiene el número de páginas que se formarían dado el máximo número de registros a mostrar en cada página. La segunda, dado el número de la página a mostrar devuelve los registros que resultan de la multiplicación del número de la página por el número de registros a mostrar.

El problema radica en que la consulta, para realizar dicha operación se trae todas las entradas de la base de datos. Este cálculo, con el tiempo y el creciente número de recursos, puede resultar muy costoso.

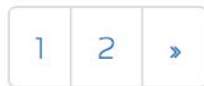


ILUSTRACIÓN 51: PAGINACIÓN

La idea que tenemos en mente es, que dado el número de la página actual, solo muestre la posibilidad de acceder a las páginas anterior y posterior actuales, tal y como funciona el sistema de paginación de Gmail entre otros.

## ❖ Mejoras en el cifrado de las contraseñas

Actualmente, las contraseñas de los usuarios se encuentran codificadas mediante la función hash SHA-256 y para otro tipo de aplicaciones podría resultar suficiente, pero ante la sensibilidad de los datos con los que se traban y el creciente número de hackers hacen que ésta sea insuficiente.

Se plantea utilizar un algoritmo de encriptación más avanzado y complejo, como es el caso de PBKDF2 el cual aplica una función pseudoaleatoria, tales como un hash criptográfico, una contraseña HMAC o una frase de paso de entrada (*'token'*) junto con un valor *'salt'* y repite el proceso muchas veces para producir una clave derivada, que luego puede ser utilizada como una clave criptográfica en posteriores operaciones.

Todo este trabajo computacional añadido hace el descifrado de contraseñas mucho más difícil, y es conocido *'key stretching'*. [ 31 ]

## ❖ Mejoras en la gestión de usuarios

DALGS es un sistema más orientado a la gestión de la información con carácter facultativo por lo que el módulo de usuarios que se ofrece es bastante limitado. Nos hemos centrado más en los roles y permisos de los usuarios dentro de la aplicación por lo que se ve necesario permitir otro tipo de acciones, tales como: disponer de un sistema de recordatorio de la contraseña, verificar que los alumnos solo estén matriculados en un grupo dentro de cada curso, un buzón de sugerencias para los alumnos, blog personal... Es decir, motivar la difusión de noticias y comentarios, y hacer un sistema académico más social.

Cómo se puede ver, son muchas las ampliaciones y modificaciones que hemos definido y que esperamos que, en un futuro cercano, se lleven a cabo para finalmente verla instalada como un programa más por el que se rija nuestra universidad.

Nuestro cometido era proporcionar una base sólida sobre la que puedan basarse y que esperamos haber conseguido con la realización de este trabajo.

# Apéndice A: Historias de Usuario

Las "Historias de usuario" (US) [ 8 ][ 9 ], que no son más que descripciones de funcionalidad enfocadas en los distintos usuarios que harán uso del sistema. Se define, únicamente, lo que el usuario necesita saber basado en sus necesidades pero sin detallar el cómo se va a realizar.

Las "Historias de usuario" sustituyen a los casos de uso de tradicionales, y según se van definiendo se van implementado lo que conlleva a un desarrollo incremental con iteraciones muy cortas (habitualmente semanas).

A veces, las US son demasiado amplias, por lo que a su vez hay que desglosarlas en otras historias de usuario más pequeñas. Aquellas que describen un requisito funcional de gran amplitud se las conoce como *epics*. A su vez cada historia de usuario se compone de *tasks* (tareas), que son cada uno de los elementos que conforman una historia.

A continuación se describe una versión simplificada de las distintas historias de usuario que se han tenido en cuenta a la hora de desarrollar el sistema. Las historias de usuarios que se relatan en la siguiente sección tienen el objetivo de dar una visión más detallada de la herramienta y de clarificar el papel que juegan cada uno de los roles de los usuarios que participan en la misma.

## Historias de usuario

En esta sección se detallan las distintas historias de usuario en las que se han detallado los epics del sistema con el objetivo de dar una visión más detallada de la herramienta y ver el papel que juegan cada uno de los actores que participan en la misma. Los epics, al basarse en roles de usuario son acciones jerárquicas por lo que se heredan siguiendo el siguiente orden, de menor a mayor:

1. Usuario
2. Alumno
3. Profesor
4. Coordinador
5. Administrador

Siguiendo el orden anterior se van a detallar los distintos epics e historias de usuarios del sistema, que como se ha mencionado antes son acumulativos entre sí, es decir, los alumnos heredan las mismas historias de los usuarios, los profesores incluyen las historias de usuario de los alumnos... y así sucesivamente.

## 1. Epic e historias de usuario: Usuario

El siguiente epic corresponde a la super historia de usuario de los coordinadores, a muy alto nivel.

<b>Título</b>	<b>EPIC 1</b>
Como usuario quiero poder gestionar mi perfil privado. Además quiero acceder a la información de las titulaciones que se imparten en mi facultad.	
<b>Descripción / Criterios de aceptación</b>	
Debe permitir el análisis de los diferentes módulos y materias que conforman una titulación para informar de las competencias que se adquieren al cursar	

Epic del usuario

A continuación se desglosa el epic citado anteriormente en las diferentes historias de usuario que se han contemplado.

<b>Título</b>	<b>US 1.1</b>
Como usuario quiero visualizar las titulaciones del sistema	
<b>Descripción / Criterios de aceptación</b>	
Debe estar dividida por módulos y reflejar las competencias profesionales que los alumnos pueden adquirir	

Ver titulaciones

**Título****US 1.2**

Como usuario quiero visualizar los módulos de las titulaciones detalladamente

**Descripción / Criterios de aceptación**

Debe estar dividido por materias

Ver módulo

**Título****US 1.3**

Como usuario quiero visualizar las materias de los módulos detalladamente

**Descripción / Criterios de aceptación**

Debe estar formado por asignaturas

Ver materia

**Título****US 1.4**

Como usuario quiero visualizar las asignaturas detalladamente

**Descripción / Criterios de aceptación**

Debe mostrar las competencias que los alumnos pueden adquirir

Ver asignatura

**Título****US 1.5**

Como usuario quiero visualizar los competencias de las titulaciones y asignaturas detalladamente

**Descripción / Criterios de aceptación**

Debe mostrar todas los resultados de aprendizaje asociadas a la misma

Ver competencia

**Título**

**US 1.6**

Como usuario quiero visualizar los resultados de aprendizaje de las asignaturas

**Descripción / Criterios de aceptación**

Deben ser mostrar información de la competencia a la que van asociadas

Ver resultados de aprendizaje

**Título**

**US 1.7**

Como usuario quiero poder editar mi perfil

**Descripción / Criterios de aceptación**

La edición debe realizarse a través de una interfaz cómoda y que realice comprobaciones sobre los campos modificados

Editar perfil

## 2. Epic e historias de usuario: Alumno

El siguiente epic corresponde a la super historia de usuario de los estudiantes, a muy alto nivel.

<b>Título</b>	<b>EPIC 2</b>
Como alumno quiero hacer un seguimiento de las actividades que mi curso y grupo para obtener los créditos de la asignatura y alcanzar las metas y objetivos. Además el acceso a mis cursos matriculados debe ser fácil.	
<b>Descripción / Criterios de aceptación</b>	
Debe tener acceso a los recursos necesarios para la realización de las actividades, así como, informarse sobre los resultados de aprendizaje de las mismas.	

Epic del alumno

A continuación se desglosa el epic citado anteriormente en las diferentes historias de usuario que se han contemplado.

<b>Título</b>	<b>US 2.1</b>
Como alumno quiero visualizar los cursos académicos en los que me encuentro matriculado	
<b>Descripción / Criterios de aceptación</b>	
Lista ordenada por término	

Ver curso académico

**Título****US 2.2**

Como alumno quiero ver todos los cursos en los que participo

**Descripción / Criterios de aceptación**

Debe existir una interfaz que muestre la información detallada relativa a cada curso

Ver curso

**Título****US 2.3**

Como alumno quiero ver la información completa de un grupo de los cursos en los que estoy matriculado

**Descripción / Criterios de aceptación**

Debe existir una interfaz que muestre la información detallada relativa a cada grupo, así como sus participantes

Ver grupo

**Título****US 2.4**

Como alumno quiero ver la información completa de una actividad, incluyendo sus recursos y resultados de aprendizaje pertenecientes a los grupos en los que me encuentro matriculado

**Descripción / Criterios de aceptación**

Debe existir una interfaz que muestre la información detallada de cada actividad

Ver actividad

**Título**

**US 2.5**

Como alumno quiero ver los perfiles de los usuarios del sistema

**Descripción / Criterios de aceptación**

El campo de contraseña debe permanecer oculto en todo momento

Ver usuario

### 3. Epic e historias de usuario: Profesor

El siguiente epic corresponde a la super historia de usuario de los coordinadores, a muy alto nivel.

<b>Título</b>	<b>EPIC 3</b>
Como docente de la universidad quiero una herramienta capaz de gestionar los distintos grupos, en los que formo parte del equipo docente, conforme a los planes de estudios prefijados en el BOE-A-2009-12977 con el fin de realizar el seguimiento de la calidad de la docencia.	
<b>Descripción / Criterios de aceptación</b>	
Es muy importante la definición de las competencias que deben adquirir los estudiantes para obtener las titulaciones, además de detallar los resultados de aprendizaje de cada una de ellas.	

Epic del profesor

A continuación se desglosa el epic citado anteriormente en las diferentes historias de usuario que se han contemplado.

<b>Título</b>	<b>US 3.1</b>
Como profesor quiero editar grupos en los cursos que imparto clase	
<b>Descripción / Criterios de aceptación</b>	
Cada vez que modifico a los participantes del grupo debe cambiar sus permisos de acceso. La modificación de los participantes se realiza a través de un fichero CSV	

Editar grupo

**Título**

**US 3.2**

Como profesor quiero crear actividades en los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Las actividades tienen un identificador único en sistema independientemente del curso o grupo al que pertenecen. Esta historia de usuario está relacionada con la evaluación de las competencias profesionales que los alumnos deben desarrollar a lo largo de su titulación por lo que la definición de las actividades debe ser clara

Crear actividad en grupo

**Título**

**US 3.3**

Como profesor quiero poder eliminar actividades de los grupo en los que imparto clase

**Descripción / Criterios de aceptación**

El borrado ha de ser lógico

Eliminar actividad de grupo

**Título**

**US 3.4**

Como profesor quiero poder restaurar las actividades de los grupo en los que imparto clase

**Descripción / Criterios de aceptación**

El borrado ha de ser lógico

Restaurar actividad de grupo

**Título****US 3.5**

Como coordinador quiero poder modificar una actividad, incluyendo sus recursos y resultados de aprendizaje pertenecientes a los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Debe existir una interfaz que me permita editar las actividades por completo

Editar actividad de grupo

**Título****US 3.6**

Como profesor quiero crear actividades en los grupos en los que imparto clase desde aplicaciones de terceros

**Descripción / Criterios de aceptación**

Este tipo de actividades deben estar identificadas en el sistema como actividades externas

Nueva actividad externa de grupo

**Título****US 3.7**

Como profesor quiero eliminar actividades externas de los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Las actividades externas deben poderse eliminarse sin afectar a las actividades internas

Eliminar actividad externa de grupo

**Título****US 3.8**

Como profesor quiero transformar las actividades externas a internas en los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

El traspaso de una actividad de una colección a otra debe tener un costo mínimo

Mover actividad externa en grupo

**Título****US 3.9**

Como profesor quiero añadir resultados de aprendizaje a las actividades de los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Los resultados de aprendizaje dependen de los resultados de aprendizaje predefinidos

Nuevo resultado de aprendizaje de actividad de grupo

**Título****US 3.10**

Como profesor quiero eliminar resultados de aprendizaje de las actividades de los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Los resultados de aprendizaje eliminados pueden volverse a dar crear

Eliminar resultado de aprendizaje de actividad de grupo

**Título**

**US 3.11**

Como profesor quiero añadir recursos a las actividades de los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Debe generar un archivo descargable

Nuevo recurso de actividad de grupo

**Título**

**US 3.12**

Como profesor quiero eliminar recursos de las actividades de los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Dichos recursos deben estar inaccesibles una vez eliminados

Eliminar recurso de actividad de grupo

**Título**

**US 3.13**

Como profesor quiero descargar los recursos de las actividades de los grupos en los que imparto clase

**Descripción / Criterios de aceptación**

Deben mantener el formato original

Descargar recurso de actividad de grupo

**Título**

**US 3.14**

Como profesor quiero ver los email relacionados con los grupos en los que imparto clase para poder realizar informes posteriores

**Descripción / Criterios de aceptación**

Debe mostrar un histórico de las conversiones

Ver emails de grupo

**Título**

**US 3.15**

Como profesor quiero descargar los email relacionados con los grupos en los que imparto clase para tener constancia de las conversaciones

**Descripción / Criterios de aceptación**

Debe descargar una copia del email original que se encontrará almacenada en el servidor del sistema

Descargar emails de grupo

#### 4. Epic e historias de usuario: Coordinador

El siguiente epic corresponde a la super historia de usuario de los coordinadores, a muy alto nivel.

<b>Título</b>	<b>EPIC 4</b>
Como docente de la universidad quiero una herramienta capaz de gestionar los distintos cursos, en los que formo parte del equipo de docencia, conforme al modelo de planes de estudios prefijados en el BOE-A-2009-12977 con el fin de realizar el seguimiento de la calidad de la docencia.	
<b>Descripción / Criterios de aceptación</b>	
Es muy importante la definición de las competencias que deben adquirir los estudiantes para obtener las titulaciones, además de detallar los resultados de aprendizaje de cada una de ellas. Además debe permitir administrar los grupos que conforman mis cursos	

Epic del Coordinador

A continuación se desglosa el epic citado anteriormente en las diferentes historias de usuario que se han contemplado.

<b>Título</b>	<b>US 4.1</b>
Como coordinador quiero poder editar los cursos que coordino	
<b>Descripción / Criterios de aceptación</b>	
Debe permitir cambiar las actividades de dicho curso, pero no la apertura de nuevos grupos	

Editar curso

**Título**

**US 4.2**

Como coordinador quiero editar grupos en los cursos que coordino

**Descripción / Criterios de aceptación**

Cada vez que modifico a los participantes del grupo debe cambiar sus permisos de acceso.  
La modificación de los participantes se realiza a través de un fichero CSV

Editar grupo

**Título**

**US 4.3**

Como coordinador quiero crear actividades en los cursos que coordino

**Descripción / Criterios de aceptación**

Las actividades tienen un identificador único en sistema independientemente del curso o grupo al que pertenecen. Esta historia de usuario está relacionada con la evaluación de las competencias profesionales que los alumnos deben desarrollar a lo largo de su titulación por lo que la definición de las actividades debe ser clara

Crear actividad de curso

**Título**

**US 4.4**

Como coordinador quiero poder eliminar actividades de los cursos que coordino

**Descripción / Criterios de aceptación**

El borrado ha de ser lógico

Eliminar actividad de curso

**Título**

**US 4.5**

Como coordinador quiero poder eliminar actividades de los cursos que coordino

**Descripción / Criterios de aceptación**

Debe permitir restaurar la actividad junto a sus recursos asociados

Restaurar actividad de curso

**Título**

**US 4.6**

Como como coordinador quiero poder modificar una actividad, incluyendo sus recursos y resultados de aprendizaje pertenecientes a los cursos que coordino

**Descripción / Criterios de aceptación**

Debe existir una interfaz que me permita editar las actividades por completo

Editar actividad de curso

**Título**

**US 4.7**

Como coordinador quiero crear actividades en los cursos que coordino desde aplicaciones de terceros

**Descripción / Criterios de aceptación**

Este tipo de actividades deben estar identificadas en el sistema como actividades externas

Nueva actividad externa de curso

**Título**

**US 4.8**

Como coordinador quiero eliminar actividades externas de los cursos que coordino

**Descripción / Criterios de aceptación**

Las actividades externas deben poderse eliminarse sin afectar a las actividades internas

Eliminar actividad externa de curso

**Título**

**US 4.9**

Como coordinador quiero transformar las actividades externas a internas en los cursos que coordino

**Descripción / Criterios de aceptación**

El traspaso de una actividad de una colección a otra debe tener un costo mínimo

Mover actividad externa de curso

**Título**

**US 4.10**

Como coordinador quiero añadir resultados de aprendizaje a las actividades de los cursos que coordino

**Descripción / Criterios de aceptación**

Los resultados de aprendizaje dependen de los resultados de aprendizaje predefinidas

Nuevo resultado de aprendizaje de actividad de curso

**Título****US 4.11**

Como coordinador quiero eliminar resultados de aprendizaje de las actividades de los cursos que coordino

**Descripción / Criterios de aceptación**

Los resultados de aprendizaje eliminados pueden volverse a dar crear

Eliminar resultado de aprendizaje de actividad de curso

**Título****US 4.12**

Como coordinador quiero añadir recursos a las actividades de los cursos que coordino

**Descripción / Criterios de aceptación**

Debe permitir la descarga de dicho recurso

Nuevo recurso de actividad de curso

**Título****US 4.13**

Como coordinador quiero eliminar recursos de las actividades de los cursos y grupos que coordino

**Descripción / Criterios de aceptación**

Dichos recursos deben estar inaccesibles una vez eliminados

Eliminar recurso de actividad de curso

**Título****US 4.14**

Como coordinador quiero descargar los recursos de las actividades de los cursos que coordino

**Descripción / Criterios de aceptación**

La descarga debe realizarse manteniendo el formato del recurso

Descargar recurso de actividad de curso

**Título****US 4.15**

Como coordinador quiero ver los email relacionados con los cursos que coordino para poder realizar informes posteriores

**Descripción / Criterios de aceptación**

La visualización detallada de un email se hará mediante la descarga del email a través de un fichero

Ver emails de cursos

**Título****US 4.16**

Como profesor quiero descargar los email relacionados con los cursos que coordino para tener constancia de las conversaciones

**Descripción / Criterios de aceptación**

Debe descargar una copia del email original que se encontrará almacenada en el servidor del sistema

Descargar emails de curso

## 5. Epic e historias de usuario: Administrador

El siguiente epic corresponde a la super historia de usuario del administrador, a muy alto nivel.

<b>Título</b>	<b>EPIC 5</b>
Como administrador de la universidad quiero una herramienta capaz de gestionar los distintos cursos académicos y titulaciones conforme a los planes de estudios prefijados en el BOE-A-2009-12977 con el fin de realizar el seguimiento de la calidad de la docencia.	
<b>Descripción / Criterios de aceptación</b>	
Debe automatizar la gestión de cursos académicos, así como de los submódulos que forman, y debe estar coordinada e integrada con el resto de sistemas que administro	

Epic del administrador

A continuación se desglosa el epic citado anteriormente en las diferentes historias de usuario que se han contemplado.

<b>Título</b>	<b>US 5.1</b>
Como administrador quiero crear cursos académicos	
<b>Descripción / Criterios de aceptación</b>	
Las titulaciones sobre las que quiero abrir un curso académico deben estar previamente creadas.	

Nuevo curso académico

**Título**

**US 5.2**

Como administrador quiero poder editar los cursos académicos existentes.

**Descripción / Criterios de aceptación**

Debe permitir cambiar el término, es decir, el periodo lectivo en el que se imparte dicha titulación

Editar curso académico

**Título**

**US 5.3**

Como administrador quiero poder eliminar los cursos académicos pasados

**Descripción / Criterios de aceptación**

El borrado ha de ser lógico por si en un futuro debo recuperar los datos

Eliminar curso académico

**Título**

**US 5.4**

Como administrador quiero poder restaurar los cursos académicos pasados

**Descripción / Criterios de aceptación**

El borrado ha de ser lógico por si en un futuro debo recuperar los datos, es decir, debe existir un botón para restaurar

Restaurar curso académico

**Título****US 5.5**

Como administrador quiero poder reutilizar los cursos académicos existentes para de este modo automatizar su apertura año tras año.

**Descripción / Criterios de aceptación**

Dar de alta un curso académico completo es un *issue* costoso.

Mejora la eficiencia de la herramienta y automatiza las tareas del administrador.

Copiar curso académico

**Título****US 5.6**

Como administrador quiero crear cursos en los cursos académicos abiertos

**Descripción / Criterios de aceptación**

Cada curso es único dentro un curso académico

Nuevo curso

**Título****US 5.7**

Como administrador quiero poder editar los cursos de los cursos académicos

**Descripción / Criterios de aceptación**

Debe permitir cambiar el coordinador y la asignatura a la que representa

Editar curso

**Título****US 5.8**

Como administrador quiero poder eliminar los cursos que forman los cursos académicos pasados

**Descripción / Criterios de aceptación**

El borrado ha de ser lógico por si en un futuro debo recuperar los datos

Eliminar curso

**Título****US 5.9**

Como administrador quiero poder restaurar los cursos que forman los cursos académicos pasados

**Descripción / Criterios de aceptación**

El borrado ha de ser lógico por si en un futuro debo recuperar los datos, es decir, debe existir un botón para restaurar

Restaurar curso

**Título****US 5.10**

Como administrador quiero crear grupos en los cursos

**Descripción / Criterios de aceptación**

Los grupos deben distinguirse dentro de un curso. Cada grupo tiene asociado una lista de profesores y otra de alumnos. La asignación de participantes se realiza a través de archivos CSV

Nuevo grupo

**Título****US 5.11**

Como administrador quiero poder eliminar grupos de los cursos

**Descripción / Criterios de aceptación**

Debe permitir restaurar el grupo con las actividades que contenía antes de ser eliminado

Eliminar grupo

**Título****US 5.12**

Como administrador quiero poder reutilizar los grupos existentes para automatizar la apertura de más grupos dentro de un curso.

**Descripción / Criterios de aceptación**

Dar de alta un grupo es un *issue* costoso, comparado con la opción de copiar un grupo preexistente con las actividades ya definidas.

Copiar grupo

**Título****US 5.13**

Como administrador quiero exportar todas las actividades internas del sistemas para futuras necesidades

**Descripción / Criterios de aceptación**

Debe generarse un fichero CSV

Exportar actividades

**Título****US 5.14**

Como administrador quiero añadir titulaciones al sistema

**Descripción / Criterios de aceptación**

Las titulaciones representan distintos estudios. La información debe estar accesible para cualquier usuario identificado.

Añadir titulación

**Título****US 5.15**

Como administrador quiero eliminar titulaciones

**Descripción / Criterios de aceptación**

Al eliminar una titulación los cursos académicos asociados también se eliminarán

Eliminar titulación

**Título****US 5.16**

Como administrador quiero restaurar titulaciones

**Descripción / Criterios de aceptación**

También debe permitir restaurar las titulaciones y toda su información

Restaurar titulación

**Título**

**US 5.17**

Como administrador quiero poder editar titulaciones

**Descripción / Criterios de aceptación**

La edición debe realizarse a través de una interfaz cómoda y precisa, realizando comprobaciones sobre los cambios realizados

Editar titulaciones

**Título**

**US 5.18**

Como administrador quiero poder importar la información relativa a cada titulación

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Importación de titulaciones

**Título**

**US 5.19**

Como administrador quiero poder exportar las titulaciones del sistema

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Exportación de titulaciones

**Título**

US 5.20

Como administrador quiero añadir módulos a las titulaciones

**Descripción / Criterios de aceptación**

Las módulos de una titulación deben estar claramente separados pues representan la estructura interna de la misma

Añadir módulo

**Título**

US 5.21

Como administrador quiero eliminar módulos de una titulación

**Descripción / Criterios de aceptación**

Al eliminar un módulo las materias asociadas también se eliminarán.

Eliminar módulo

**Título**

US 5.22

Como administrador quiero restaurar los módulos de una titulación

**Descripción / Criterios de aceptación**

Debe permitir restaurarlos posteriormente junto a sus materias asociadas

Restaurar módulo

**Título****US 5.23**

Como administrador quiero poder editar los módulos de las titulaciones

**Descripción / Criterios de aceptación**

La edición debe realizarse a través de una interfaz cómoda y que realice comprobaciones sobre los campos modificados

Editar módulo

**Título****US 5.24**

Como administrador quiero poder importar la información relativa a los módulos de una titulación

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Importación de módulos

**Título****US 5.25**

Como administrador quiero poder exportar los módulos de las titulaciones

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Exportación de módulos

**Título**

US 5.26

Como administrador quiero añadir materias a los módulos

**Descripción / Criterios de aceptación**

Las materias de un módulo deben estar claramente separados pues representan la estructura interna de la misma

Añadir materia

**Título**

US 5.27

Como administrador quiero eliminar materias de un módulo

**Descripción / Criterios de aceptación**

Al eliminar una materia las asignaturas también se eliminarán

Eliminar materia

**Título**

US 5.28

Como administrador quiero eliminar materias de un módulo

**Descripción / Criterios de aceptación**

Debe permitir restaurarlas posteriormente junto a la información asociada

Restaurar materia

**Título**

**US 5.29**

Como administrador quiero poder editar las materias de los módulos

**Descripción / Criterios de aceptación**

La edición debe realizarse a través de una interfaz cómoda y que realice comprobaciones sobre los campos modificados

Editar materia

**Título**

**US 5.30**

Como administrador quiero poder importar la información relativa a las materias de un módulo

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Importación de materias

**Título**

**US 5.31**

Como administrador quiero poder exportar las materias de los módulos para futuras necesidades

**Descripción / Criterios de aceptación**

Ambas acciones se realizarán mediante archivos CSV

Exportación de materias

**Título****US 5.32**

Como administrador quiero añadir asignaturas a las materias

**Descripción / Criterios de aceptación**

Las asignaturas de una materia deben estar claramente ligadas entre sí, es decir, deben tratar sobre un tema similar o complementario

Añadir asignatura

**Título****US 5.33**

Como administrador quiero eliminar asignaturas de una materia

**Descripción / Criterios de aceptación**

Al eliminar un asignatura los cursos asociados también se eliminarán

Eliminar asignatura

**Título****US 5.33**

Como administrador quiero restaurar asignaturas de una materia

**Descripción / Criterios de aceptación**

Debe permitir restaurarlas posteriormente junto a su información asociada

Restaurar asignatura

**Título**

**US 5.34**

Como administrador quiero poder editar las asignaturas de las materias

**Descripción / Criterios de aceptación**

La edición debe realizarse a través de una interfaz cómoda y que realice comprobaciones sobre los campos modificados

Editar asignatura

**Título**

**US 5.35**

Como administrador quiero poder importar la información relativa a las asignaturas de una materia

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Importación de asignaturas

**Título**

**US 5.36**

Como administrador quiero poder exportar las asignaturas de las materias

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Exportación de asignaturas

**Título**

**US 5.37**

Como administrador quiero añadir competencias a las titulaciones

**Descripción / Criterios de aceptación**

Las competencias de una titulación representan las habilidades profesionales que los alumnos adquieren mediante su estudio

Añadir competencias a las titulaciones

**Título**

**US 5.38**

Como administrador quiero añadir competencias a las asignaturas de una titulación

**Descripción / Criterios de aceptación**

Las competencias de una asignatura representan las habilidades profesionales que los alumnos adquieren mediante su estudio. Deben coincidir con las competencias asociadas a la titulación

Añadir competencias a las asignaturas

**Título**

**US 5.39**

Como administrador quiero eliminar competencias en una titulación o asignatura

**Descripción / Criterios de aceptación**

Al eliminar una competencia los resultados de aprendizaje asociados también se eliminarán. También debe permitir restaurarlas posteriormente

Eliminar competencias

**Título**

**US 5.40**

Como administrador quiero poder editar las competencias de las titulaciones y asignaturas

**Descripción / Criterios de aceptación**

La edición debe realizarse a través de una interfaz cómoda y que realice comprobaciones sobre los campos modificados

Editar competencias

**Título**

**US 5.41**

Como administrador quiero poder importar competencias a las titulaciones y asignaturas

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Importación de competencias

**Título**

**US 5.42**

Como administrador quiero poder exportar las competencias de las titulaciones y asignaturas

**Descripción / Criterios de aceptación**

Se realizará mediante archivos CSV

Exportación de competencias

**Título****US 5.43**

Como administrador quiero añadir metas de aprendizaje a las competencias

**Descripción / Criterios de aceptación**

Las metas de aprendizaje de una competencia son un requisito fundamental del sistema, ya que los nuevos planes de estudios deben encontrar un modo de representarlas

Añadir metas de aprendizaje a las competencias

**Título****US 5.44**

Como administrador quiero eliminar metas de aprendizaje de las competencias

**Descripción / Criterios de aceptación**

Al eliminar una meta de aprendizaje, los resultados de aprendizaje asociados en los distintos grupos deben ser eliminados

Eliminar metas de aprendizaje

**Título****US 5.45**

Como administrador quiero añadir usuarios al sistema

**Descripción / Criterios de aceptación**

Este procedimiento debe ser o bien manual o bien mediante la importación de archivos CSV. A su vez debe asignarles los diferentes roles de usuarios

Añadir usuario

**Título**

**US 5.46**

Como administrador quiero poder bloquear y desbloquear a los usuarios que están dados de alta en el sistema

**Descripción / Criterios de aceptación**

Al bloquear a un usuario, éste pierde la capacidad de autenticarse en el sistema y por ende acceder a la web. Al desbloquearlo, el usuario recupera sus permisos

Bloquear/Desbloquear usuario

**Título**

**US 5.47**

Como administrador quiero poder editar a los usuarios del sistema

**Descripción / Criterios de aceptación**

La edición debe realizarse a través de una interfaz cómoda y que realice comprobaciones sobre los campos modificados

Editar usuario

**Título**

**US 5.48**

Como administrador quiero poder exportar la lista de usuarios del sistema para así poder llevar un seguimiento de control

**Descripción / Criterios de aceptación**

Debe generar un archivo CSV con todos los datos de los mismos

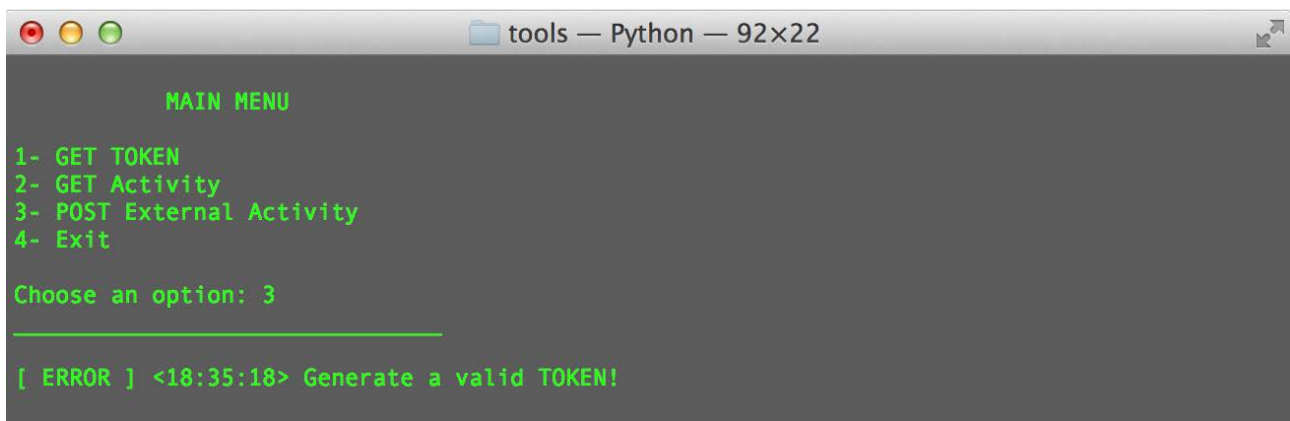
Exportación de usuarios

## Apéndice B: API DALGS

El objetivo de este apéndice es el de explicar la herramienta API\_DALGS que se incluye en la carpeta *tools*. La herramienta proporcionada permite demostrar y verificar el funcionamiento de la API REST que se ha implementado en DALGS para la interacción de aplicaciones externas con nuestro sistema.

Como se muestra en el menú principal, las opciones de las que disponemos son las de autenticarse en el sistema, ver un actividad previamente creada o añadir una actividad externa.

El usuario una vez lanzada la aplicación deberá autenticarse en el sistema para poder interactuar con DALGS. En caso contrario, mostrará un mensaje de error indicando que debe generar su TOKEN de acceso.



```
tools — Python — 92x22

MAIN MENU
1- GET TOKEN
2- GET Activity
3- POST External Activity
4- Exit

Choose an option: 3

[ ERROR ] <18:35:18> Generate a valid TOKEN!
```

Para ello nos dirigimos a la primera opción:

```
tools — Python — 92x23

1. GENERATE TOKEN

-----LOGIN-----
Username: admin
Password:
-----
Port: 8080
[ DEBUG ] <18:35:39> http://localhost:8080/dalgs/oauth/token?grant_type=password&client_id=restapp&client_secret=restapp&username=admin&password=admin

[ INFO ] <18:35:39> Status: 200
[ INFO ] <18:35:39> TOKEN: 02ed44aa-baab-4da8-8aca-bc1ea108fc57
```

Tras autenticarnos en el sistema, obtendremos el TOKEN necesario para llevar a cabo el resto de peticiones. El TOKEN generado sólo estará operativo durante 120 segundos, por lo que pasado ese tiempo deberemos volver a generarlo.

Si seleccionamos la segunda opción recibiremos un JSON con los datos de la actividad.

```
tools — bash — 93x23

2. GET ACTIVITY

Id Activity: 1
[ DEBUG ] <23:52:31> Status: 200
[ INFO ] <23:52:31> Activity:
{"info": {"code": "ACT1", "name": "Casos de Uso", "description": "Desarrollar diagramas de casos de uso"}, "attachments": [], "id": 1, "isDeleted": false, "learningGoalStatus": [{"learningGoal": {"info": {"code": "LEA1", "name": "learning 1", "description": "descripcion learning 1"}, "id": 1, "isDeleted": false}, "weight": 10}]}
```

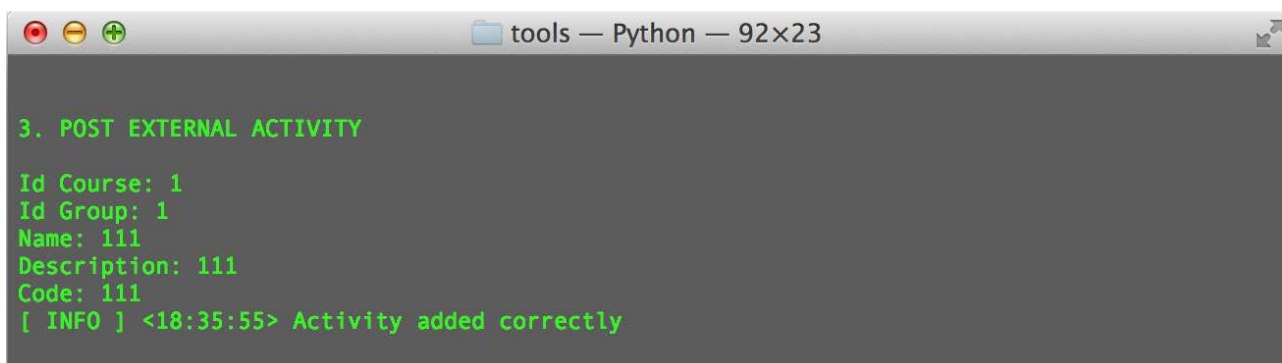
En caso de que la actividad no exista obtendremos un *mensaje de error* como respuesta.

```
tools — bash — 93x23

2. GET ACTIVITY

Id Activity: 45
[ DEBUG ] <23:52:21> Status: 200
[ ERROR ] <23:52:21> Activity with ID=45 no exists
```

Por último, si deseamos informar sobre una nueva actividad externa la herramienta nos pedirá cierta información que cotejará con la base de datos y posteriormente la añadirá a la colección de actividades externas del grupo o curso en cuestión.



```
tools — Python — 92x23

3. POST EXTERNAL ACTIVITY
Id Course: 1
Id Group: 1
Name: 111
Description: 111
Code: 111
[ INFO ] <18:35:55> Activity added correctly
```

El código único de la actividad (*code*) no es necesario, ya que por motivos de persistencia se ha decidido generar automáticamente. Hay que tener en cuenta que las actividades de terceros que se integren con este sistema no tendrán este problema ya que los datos que se manejan se encuentran ubicados en un servidor externo de acceso global para todas las aplicaciones conectadas, en nuestro caso se encuentran ubicados en un máquina virtual (VM).

El administrador, coordinador del curso o los profesores del grupo serán los encargados de transformar dicha actividad externa a actividad interna desde la propia aplicación, DALGS.



# Referencias

- [ 1 ] **BOE-A-2009-12977 Available:** <http://www.boe.es/boe/dias/2009/08/04/pdfs/BOE-A-2009-12977.pdf>
- [ 2 ] **FDI - UCM. Información docente. Available:** <http://informatica.ucm.es/informacion-docente>
- [ 3 ] **Tutorial de Spring Security mediante ACLs. Available:** <http://krams915.blogspot.com.es/2011/01/spring-security-3-full-acl-tutorial.html>
- [ 4 ] **Tutorial de Spring MVC, Desarrollando una aplicacion Spring Framework MVC paso a paso. Available:** <http://www.davidmarco.es/spring-mvc>
- [ 5 ] **Spring Security: OAuth2. Available:** <http://spring-security-oauth.codehaus.org/oauth2.html>
- [ 6 ] **Super-csv. Available:** <https://github.com/super-csv/super-csv>
- [ 7 ] **Ejemplo básico de Spring MVC con Maven. Available:** <https://hop2croft.wordpress.com/2011/09/10/ejemplo-basico-de-spring-mvc-con-maven/>
- [ 8 ] **Java Persistence API. Available:** [http://es.wikipedia.org/wiki/Java\\_Persistence\\_API](http://es.wikipedia.org/wiki/Java_Persistence_API)
- [ 9 ] **JPA. Available:** <http://www.genbetadev.com/frameworks/jpa-vs-hibernate>
- [ 10 ] **Git. Available:** <https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>
- [ 11 ] **User Stories. Available:** <http://yodiz.com/help/agile-user-stories-and-groomed-product-backlog/>
- [ 12 ] **¿Qué es el Proceso de Bolonia? Available:** <http://www.infouma.uma.es/docs/eees.pdf>
- [ 13 ] **Protocolos de evaluación y documentos de ayuda; Aneca. Available:** <http://www.aneca.es/Programas/VERIFICA/Verificacion-de-Grado-y-Master/Protocolos-de-evaluacion-y-documentos-de-ayuda>
- [ 14 ] **Bologna Process - European higher education area. Available:** <http://www.ehea.info/article-details.aspx?ArticleId=3>
- [ 15 ] **Maven. Available:** <http://es.wikipedia.org/wiki/Maven>
- [ 16 ] **MySQL. Available:** [www.mysql.com](http://www.mysql.com)
- [ 17 ] **Ejemplo de JPA. Available:** <http://www.arquitecturajava.com/ejemplo-de-jpa/>

- [ 18 ] **Un recorrido por Spring Security. Available:** <https://federicojcdm.wordpress.com/2010/03/23/un-recorrido-por-spring-security-3-0/>
- [ 19 ] **Moodle. Available:** <https://moodle.org/plugins/>
- [ 20 ] **OAuth– Autorización segura de APIs con OAuth (Open Authorization); Oscar García. Available:** <http://www.elclubdelprogramador.com/2012/04/23/oauth-autorizacion-segura-de-apis-con-oauth-open-authorization/>
- [ 21 ] **Ficheros CSV. Available:** <https://es.wikipedia.org/wiki/CSV>
- [ 22 ] **Modelo relacional. Available:** [http://es.wikipedia.org/wiki/Base\\_de\\_datos\\_relacional](http://es.wikipedia.org/wiki/Base_de_datos_relacional)
- [ 23 ] **ORM. Available:** [http://es.wikipedia.org/wiki/Mapeo\\_objeto-relacional](http://es.wikipedia.org/wiki/Mapeo_objeto-relacional)
- [ 24 ] **JPA. Available:** [http://es.wikipedia.org/wiki/Java\\_Persistence\\_API](http://es.wikipedia.org/wiki/Java_Persistence_API)
- [ 25 ] **MySQL. Available:** <http://es.wikipedia.org/wiki/MySQL>
- [ 26 ] **The Great Java Application Server Debate with Tomcat, JBoss, GlassFish, Jetty and Liberty Profile. Simon Maple. Available:** <http://zeroturnaround.com/rebellabs/the-great-java-application-server-debate-with-tomcat-jboss-glassfish-jetty-and-liberty-profile/>
- [ 27 ] **Apache Tomcat. Available:** <http://es.wikipedia.org/wiki/Tomcat>
- [ 28 ] **Arquitecturas multi-tenant. Available:** <http://gravitar.biz/tecnologia-negocios/arquitecturas-multi-tenant/>
- [ 29 ] **Experience API. Available:** <http://tincanapi.com/overview/>
- [ 30 ] **Mozilla Open Badges. <http://openbadges.org>**
- [ 31 ] **PBKDF2. Available:** <http://en.wikipedia.org/wiki/PBKDF2>
- [ 32 ] **Desarrollo ágil de software. Available:** [http://es.wikipedia.org/wiki/Desarrollo\\_ágil\\_de\\_software](http://es.wikipedia.org/wiki/Desarrollo_ágil_de_software)
- [ 33 ] **Desarrollo por etapas. Available:** [http://es.wikipedia.org/wiki/Desarrollo\\_por\\_etapas](http://es.wikipedia.org/wiki/Desarrollo_por_etapas)

- [ 34 ] Lucidchart. Available: <https://www.lucidchart.com>
- [ 35 ] Spring Tool Suite. Available: <https://spring.io/tools>
- [ 36 ] Sequel Pro. Available: <http://www.sequelpro.com>
- [ 37 ] Introducción a Maven. Available: <http://algo3.uqbar-project.org/temario/01-intro-UI/maven>
- [ 38 ] J2EE, Webopedia. Available: <http://www.webopedia.com/TERM/J/J2EE.html>
- [ 39 ] J2EE, Oracle. Available: <https://docs.oracle.com/cd/E19636-01/819-4627/6n6p1r5n6/index.html>
- [ 40 ] Java Server Pages, Wikipedia. Available: [http://es.wikipedia.org/wiki/JavaServer\\_Pages](http://es.wikipedia.org/wiki/JavaServer_Pages)
- [ 41 ] CSS, Wikipedia. Available: [http://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)
- [ 42 ] Qué es Python, Miguel Ángel Álvarez. Available: <http://www.desarrolloweb.com/articulos/1325.php>
- [ 43 ] Proyectos ágiles. Available: <http://www.proyectosagiles.org/que-es-scrum>
- [ 44 ] Google Developers. Available: <https://developers.google.com/identity/toolkit/>



# Bibliografía

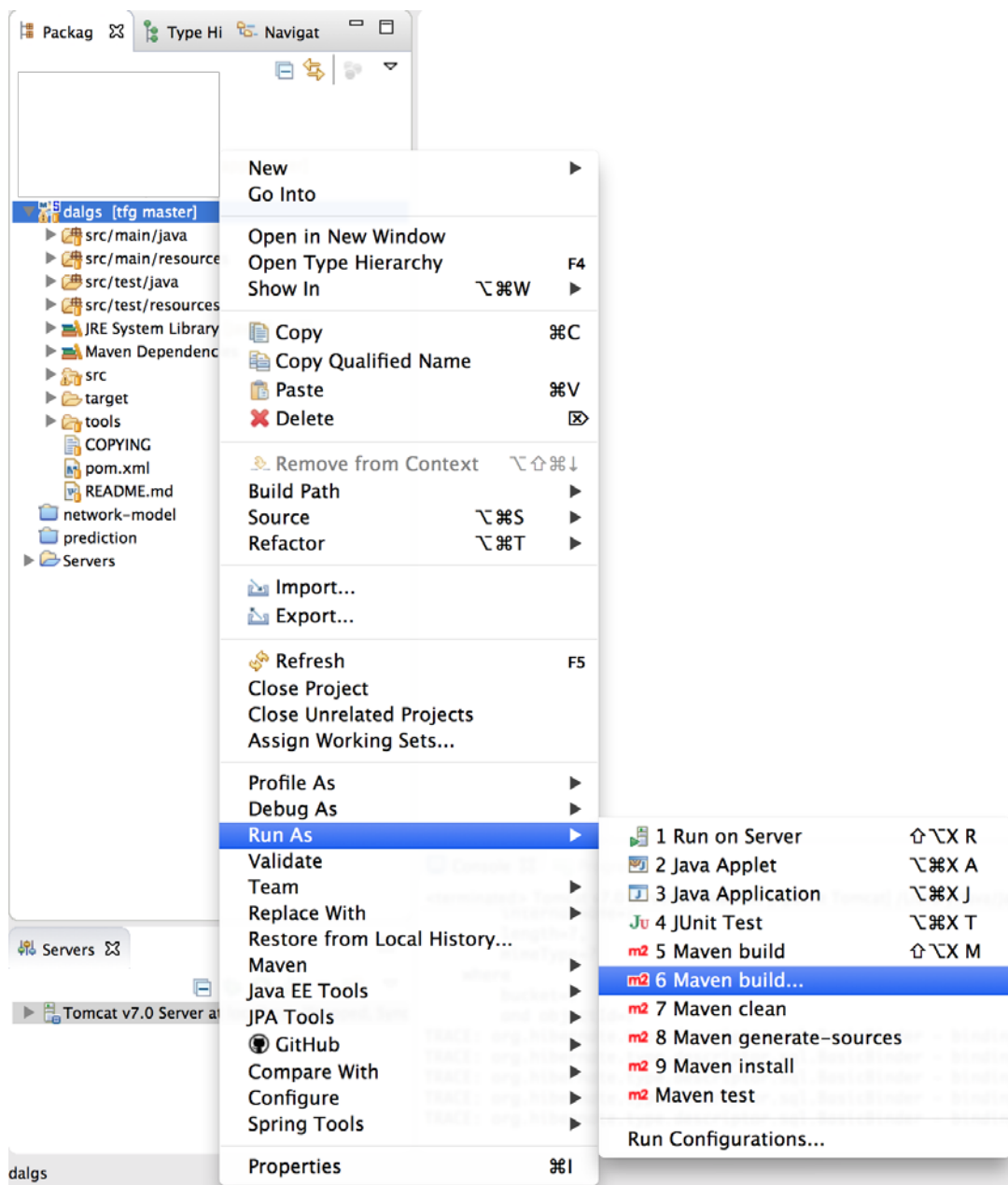
- [ 1 ] Chris Schaefer , Clarence Ho , Rob Harrop; Pro Spring Security, 4a ed. ed. Apress
  
- [ 2 ] Peter Mularien; Spring Security 3 (20 de mayo de 2010), ed. Packt
  
- [ 3 ] Chris Schaefer, Clarence Ho, Rob Harrop; Pro Spring, Fourth Edition , ed. Apress.  
Available: <http://0-proquest.safaribooksonline.com.cisne.sim.ucm.es/book/programming/java/9781430261520>
  
- [ 4 ] Ben Alex, Luke Taylor; Spring Security, Reference Documentation. Available:  
<http://docs.spring.io/spring-security/site/docs/3.0.x/reference/domain-acls.html>
  
- [ 5 ] The OAuth 2.0 Protocol, draft-ietf-oauth-v2-05. Available: <http://tools.ietf.org/html/draft-ietf-oauth-v2-05>
  
- [ 6 ] Spring MVC, Spring-framework-reference API. Available:  
<http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>
  
- [ 7 ] Amuthan G; Spring MVC Beginner's Guide, ed. Packt Publishing. Available: <http://0-proquest.safaribooksonline.com.cisne.sim.ucm.es/book/programming/java/9781783284870>
  
- [ 8 ] User Stories Applied: For Agile Software Development Paperback – March 11, 2004, Mike Cohn. Available: <http://www.amazon.com/User-Stories-Applied-Software-Development/dp/0321205685>
  
- [ 9 ] User Story Mapping: Discover the Whole Story, Build the Right Product; Jeff Patton, Peter Economy. Available: <http://www.amazon.com.mx/User-Story-Mapping-Discover-Product-ebook/dp/B00NF07FHS>



# Anexo A: Manual de Instalación

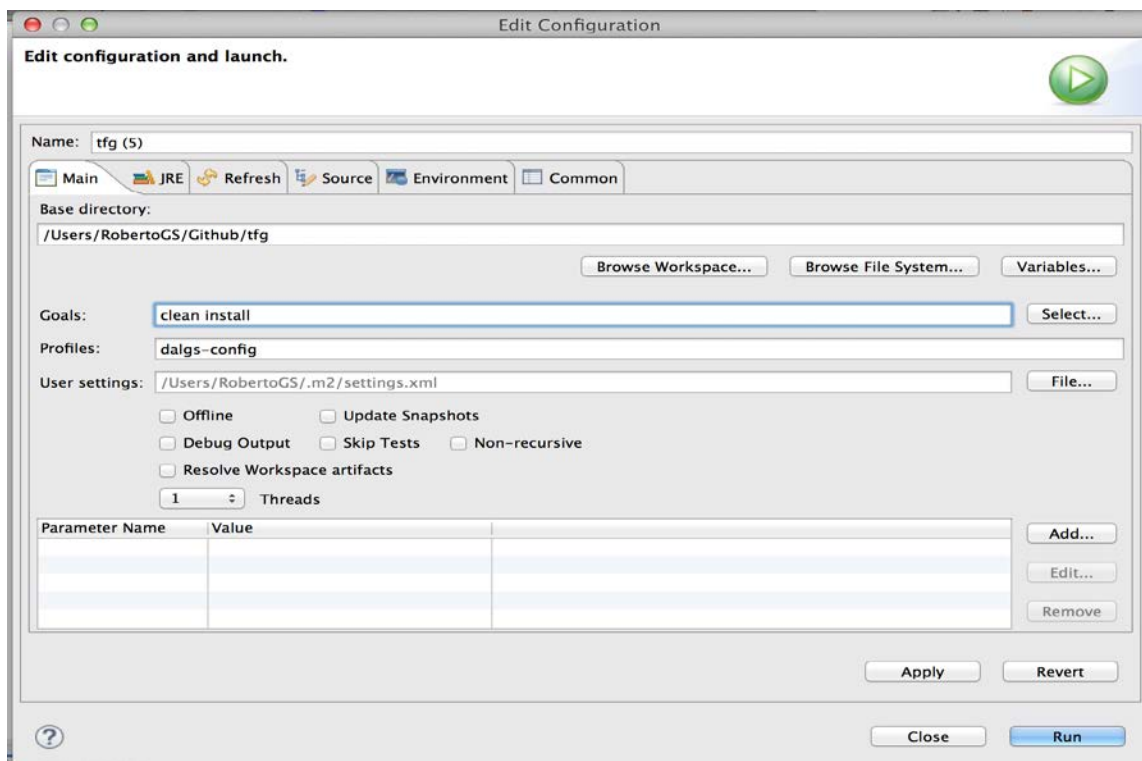
## 1. Generación del archivo ejecutable .war

En primer lugar, es necesario generar un fichero `.war` del proyecto. Para llevarlo a cabo, hay que ir a Eclipse y seleccionar el workspace en el que se encuentra el proyecto. Una vez hecho esto, en el “*Package Explorer*” se selecciona el proyecto del cual se quiere obtener el `.war` y se pulsa sobre él, usando la función secundaria del ratón. De esta manera aparece un menú desplegable con distintas opciones, de las cuales hay que seleccionar “*Run As*”-> “*Maven build*”, tal y como se muestra en la imagen siguiente.



Posteriormente aparece una ventana con información del proyecto y en la que hay que incluir “clean install” en el campo **Goals**.

- **Clean**: Esta opción se encarga de limpiar los artefactos creados por compilaciones anteriores
- **Install**: Esta opción se encarga de instalar el paquete en el repositorio local, para su uso como una dependencia en otros proyectos a nivel local



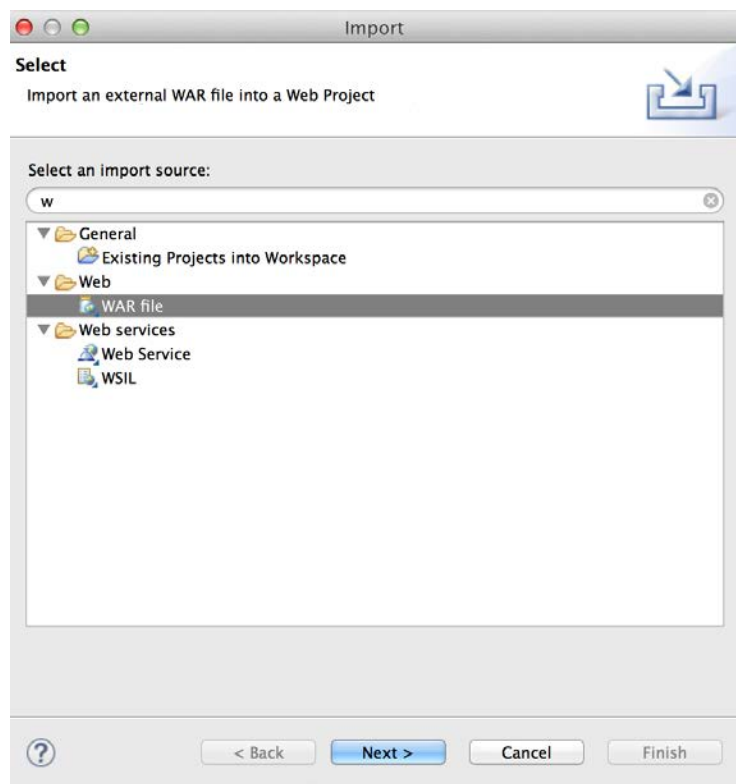
Una vez hecho esto, se aplican los cambios y se ejecuta pulsando el botón “Run”. Si se ha realizado correctamente tendrá que aparecer el archivo `.war` en la carpeta “`target`” del proyecto.

```
Console Progress
<terminated> tfg (4) [Maven Build] /Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/bin/java (9/6/2015 13:09:27)
[INFO] --- maven-install-plugin:2.4:install (default-install) @ dalgs ---
[INFO] Installing /Users/RobertoGS/Github/tfg/target/dalgs-1.0.0-BUILD-SNAPSHOT.war to /Users/RobertoGS/
[INFO] Installing /Users/RobertoGS/Github/tfg/pom.xml to /Users/RobertoGS/.m2/repository/es/ucm/fdi/dalg
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.148 s
[INFO] Finished at: 2015-06-09T13:09:34+01:00
[INFO] Final Memory: 25M/439M
[INFO] -----
```

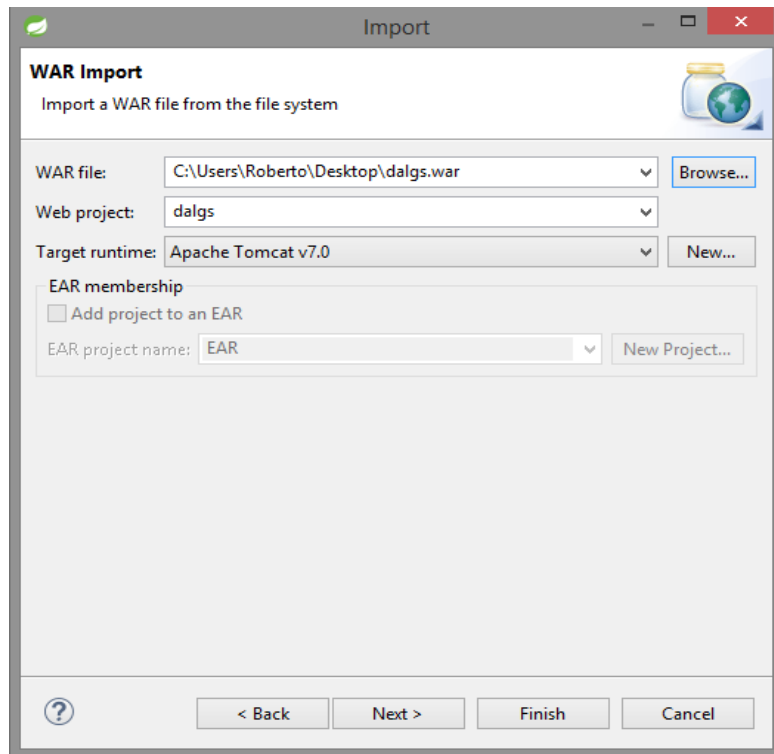
Una vez obtenido el WAR, existen dos opciones. Una consiste en importar el *war* para modificar parámetros que dependen del servidor en el que se desea desplegar la aplicación o de la base de datos de la que va a hacer uso. En caso de no tener que realizar ninguna modificación se puede saltar al paso 4, para llevar a cabo el despliegue del WAR.

## 2. Importación del proyecto

El segundo paso consiste en importar el fichero WAR. Para ello, dentro de Eclipse se selecciona en el menú superior: file ->import. Una vez hecho esto, se muestra una ventana para seleccionar el tipo de fichero. Tal y como se muestra en la siguiente imagen, hay que escoger la opción web y después seleccionar WAR file.



En la siguiente ventana se escoge la ruta del fichero WAR y el servidor tal y como se muestra en la imagen siguiente.



Después de esto solo falta pulsar en finalizar y el WAR se habrá importado.

Dentro del proyecto encontrará tanto la herramienta de prueba del web service como un conjunto de archivos CSV que corresponden a los formatos de importación y exportación de datos del sistema.

### 3. Importación del Proyecto (BIS)

Otra opción viable sería descargar el proyecto desde el siguiente repositorio de Github:

<https://github.com/romoAlmudena/dalgs>

En dicho repositorio se encuentran las últimas actualizaciones generadas sobre el proyecto.

GitHub, Inc. [US] <https://github.com/romoAlmudena/dalgs>

This repository Search Pull requests Issues Gist

romoAlmudena / dalgs Unwatch 1 Star 0 Fork 2

sha — Edit

335 commits 3 branches 0 releases 3 contributors

branch: master dalgs / +

File	Commit Message	Time
src	un poco de limpieza	4 days ago
tools	Cambios API_DALGS.py	4 days ago
.gitignore	Modifying .gitignore to ignore .springBeans	23 days ago
COPYING	Añadiendo licencia GNU GPL AFFERO 3.0 y la configuración de maven.	2 months ago
README.md	Arreglado storage de attachs para las actividades y otros bugs	13 days ago
pom.xml	Cambios API_DALGS.py	4 days ago

SSH clone URL  
git@github.com:romo/

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

**Download ZIP**

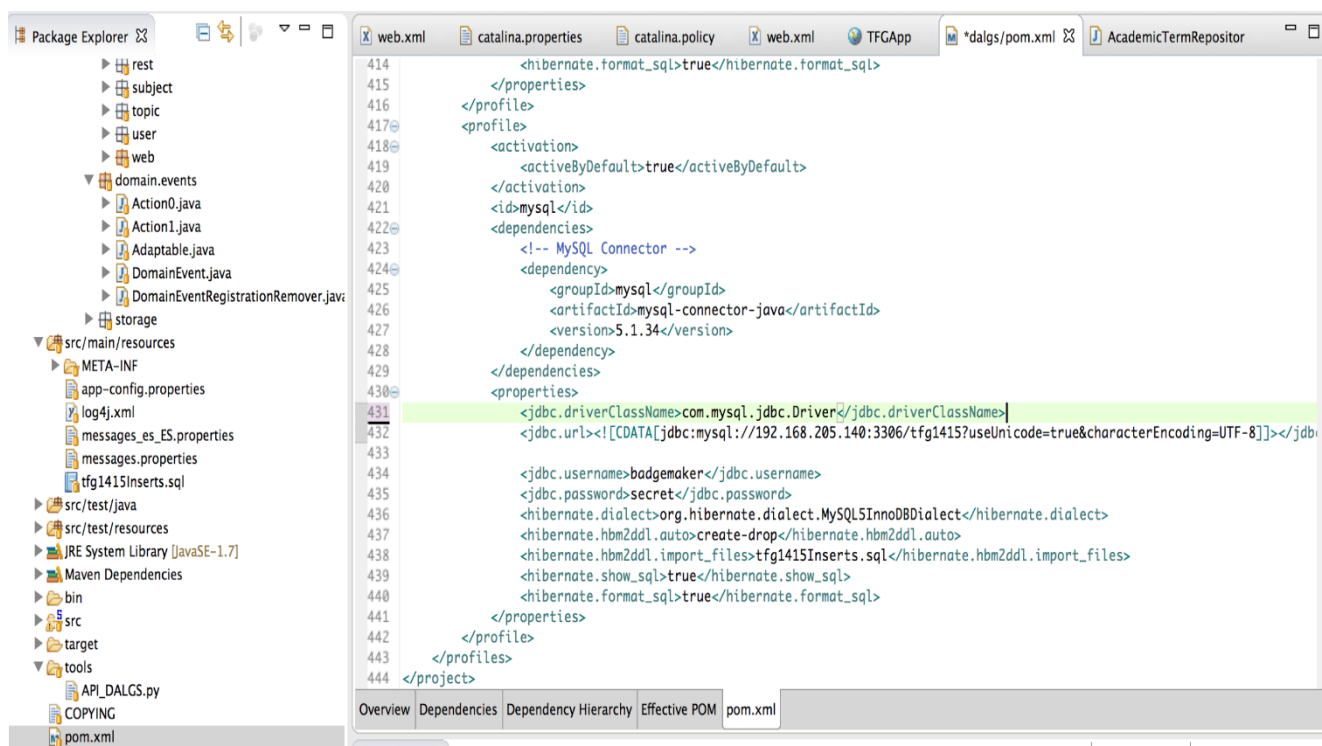
El zip que se descarga está listo para ser importado directamente en eclipse como proyecto Maven.

## 4. Configuración del entorno

Para realizar modificaciones en los parámetros del servidor o de la base de datos, es necesario acceder al fichero pom.xml del proyecto.

### 4.1 pom.xml

Como puede verse a continuación, nuestro fichero *pom* contiene, a parte de las dependencias maven, los datos de acceso a nuestra base de datos y algunos parámetros de configuración JPA.



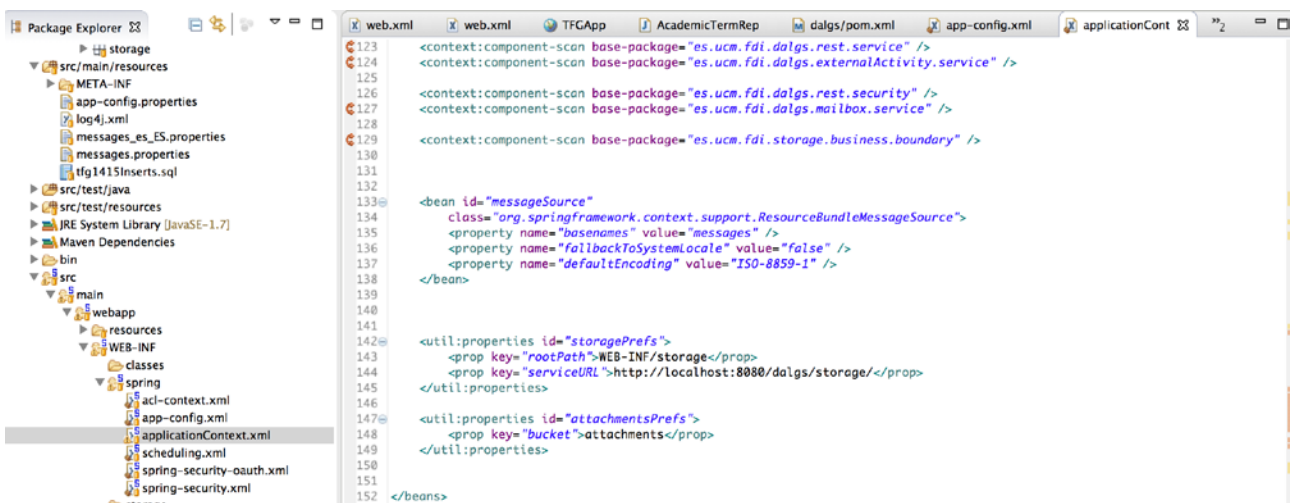
```
414     <hibernate.format_sql>true</hibernate.format_sql>
415   </properties>
416 </profile>
417 <profile>
418   <activation>
419     <activeByDefault>true</activeByDefault>
420   </activation>
421   <id>mysql</id>
422   <dependencies>
423     <!-- MySQL Connector -->
424     <dependency>
425       <groupId>mysql</groupId>
426       <artifactId>mysql-connector-java</artifactId>
427       <version>5.1.34</version>
428     </dependency>
429   </dependencies>
430 </properties>
431 <jdbc.driverClassName>com.mysql.jdbc.Driver</jdbc.driverClassName>
432 <jdbc.url><![CDATA[jdbc:mysql://192.168.205.140:3306/tfg1415?useUnicode=true&characterEncoding=UTF-8]]></jdbc
433
434 <jdbc.username>badgemaker</jdbc.username>
435 <jdbc.password>secret</jdbc.password>
436 <hibernate.dialect>org.hibernate.dialect.MySQLInnoDBDialect</hibernate.dialect>
437 <hibernate.hbm2ddl.auto>create-drop</hibernate.hbm2ddl.auto>
438 <hibernate.hbm2ddl.import_files>tfg1415Inserts.sql</hibernate.hbm2ddl.import_files>
439 <hibernate.show_sql>true</hibernate.show_sql>
440 <hibernate.format_sql>true</hibernate.format_sql>
441 </properties>
442 </profile>
443 </profiles>
444 </project>
```

## 4.2 applicationContext.xml

Este paso incluye la creación del directorio storage dentro del PATH escogido. Para ello se ha de acceder al fichero **applicationContext.xml** y establecer el valor para la propiedad "serviceURI".

En este caso la URL es <http://localhost:8080/dalgs/storage/>

Posteriormente hay que crear el directorio que se encargará de almacenar los mensajes descargados, así como de los recursos de las actividades.



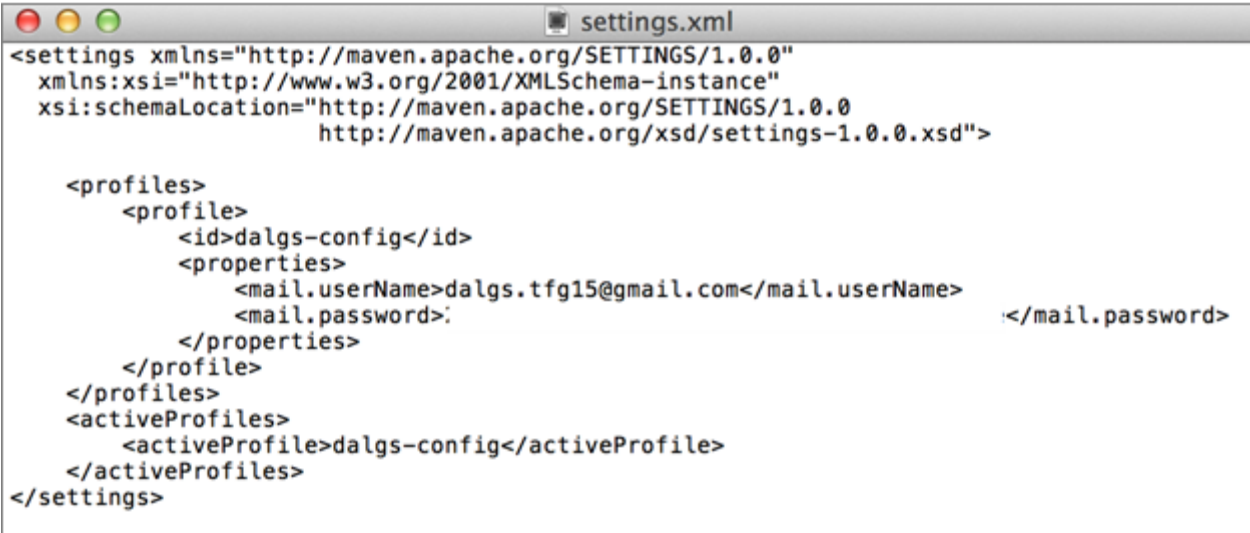
### 4.3 settings.xml

En este caso se ha usado un fichero *“settings.xml”* para guardar la información de conexión al correo electrónico [dalgs.tfg15@gmail.com](mailto:dalgs.tfg15@gmail.com), es decir, contiene el usuario y la contraseña con la que loguearse a la cuenta de correo.

Cada vez que el sistema se comunica con la cuenta de correo, recupera esta información del fichero.

La ruta del archivo tiene que ser */User/XXX/.m2/settings.xml* donde maven cogerá los datos de acceso al correo.

Este fichero tiene definido un *profile* compuesto por un identificador y una serie de propiedades, entre ellas los datos de acceso al correo - nombre de usuario y contraseña.

A screenshot of a text editor window titled "settings.xml". The window contains XML code for a Maven settings file. The code defines a profile named "dalgs-config" with properties for "mail.userName" (dalgs.tfg15@gmail.com) and "mail.password". The profile is set as the active profile. The XML is well-formed with proper opening and closing tags.

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <profiles>
    <profile>
      <id>dalgs-config</id>
      <properties>
        <mail.userName>dalgs.tfg15@gmail.com</mail.userName>
        <mail.password>: </mail.password>
      </properties>
    </profile>
  </profiles>
  <activeProfiles>
    <activeProfile>dalgs-config</activeProfile>
  </activeProfiles>
</settings>
```

## 5. Despliegue

Llegados a este punto, solo faltaría coger el archivo .war que se incluye y desplegarlo en nuestro servidor.

Mensaje: OK

**Gestor**

Listar Aplicaciones      Ayuda HTML de Gestor      Ayuda de Gestor      Estado de Servidor

**Aplicaciones**

Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/dalgs	Ninguno especificado	DALGS	false	0	Arrancar Parar Recargar Replegar
/docs	Ninguno especificado	Tomcat Documentation	true	0	Expirar sesiones sin trabajar ≥ 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

**Desplegar**

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Desplegar

Archivo WAR a desplegar

Seleccione archivo WAR a cargar    Seleccionar archivo    dalgs...HOT.war

Desplegar

Como puede verse, en este caso y a pesar de estar generado correctamente el .war , Tomcat es incapaz de desplegarlo. Esto es debido a que los archivos de configuración citados en el anterior apartado deberían de adaptarse a las rutas del propio servidor y no a las local, las del STS (definidas anteriormente).



# Glosario

## 1. ACL

*Acrónimo de Access Control List*

Se trata de término de seguridad para determinar los permisos de acceso sobre los recursos software.

## 2. API

*Acrónimo de Application Programming Interface*

Se trata de un conjunto de funciones y métodos públicos que ofrece una biblioteca para poder ser utilizada desde otro software.

## 3. ECTS

*Acrónimo de European Credit Transfer and Accumulation System*

Actual sistema de las universidades europeas para convalidar asignaturas

## 4. SaaS

*Acrónimo de Software as a Service*

Modelo de distribución software en el que los datos y la lógica de negocio se hospedan en una empresa proveedora TIC lo que hace que acceso a los datos pueda realizarse desde cualquier cliente a través de internet.

## 5. IT

*Acrónimo de Information technology*

Conjunto de recursos tecnológicos para manipular la información. Estos recursos son las redes, los terminales y los servicios en su conjunto.

## 6. US

### *Acrónimo de User Story*

Representación de un requisito de software escrito en lenguaje natural y entendible por todas las personas, sin la necesidad de tener conocimientos tecnológicos.

## 7. DBMS

### *Acrónimo de Database Management System*

Los sistemas de gestión de bases de datos son aplicaciones software que interactúan con el usuario, otras aplicaciones y la base de datos en sí para capturar y analizar los datos.