



*Facultad
de
Ciencias*

**DESARROLLO DE UNA APLICACIÓN WEB
PARA LA GESTIÓN DE TRABAJOS DE FIN
DE GRADO**

**(Development of a web application for final
thesis)**

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Raúl Ceballos Díez

Director: Esteban Stafford Fernández

Septiembre - 2017

RESUMEN

En pleno año 2017, se hace casi imprescindible el uso de herramientas informáticas que nos ayuden en la gestión de muchas de nuestras tareas diarias. Las herramientas nos permiten agilizar y facilitar procesos que realizados de una manera tradicional podrían demorarse y no ser lo suficientemente eficientes.

En nuestro caso, estas tareas corresponden a la gestión de los Trabajos de Fin de Grado (TFG), anteriormente denominados Proyectos de Fin de Carrera (PFC), realizados en la Facultad de Ciencias de la Universidad de Cantabria.

Es por ello por lo que el objetivo de este proyecto es desarrollar y construir un sistema que haga de nexo común entre alumnos, profesores y empresas. En el caso de los alumnos, proporcionando información acerca de sus TFG y facilitando el contacto con sus responsables y, de cara al profesorado y a las empresas, buscando simplificar el proceso de propuesta y gestión.

Para poder desarrollar este sistema, haremos uso de un CMS (Content Management System) que, debidamente configurado y adaptado a las necesidades de nuestro sistema, nos servirá de pilar fundamental para el desarrollo de la aplicación.

Palabras Clave: aplicación web, pagina web, CMS, Drupal, Trabajos de Fin de Grado.

ABSTRACT

In the year 2017, it is almost necessary to use computer tools to help us manage many of our daily tasks. The tools allow us to streamline and facilitate processes that done in a traditional way could be delayed and not be efficient enough.

In our case these tasks correspond to the management of thesis' projects realized in the Faculty of Sciences of the University of Cantabria.

The main goal of this project is to develop and build a system that serves as a common place between students, teachers and companies. In the case of students informing of the proposed projects and making it easier to contact with those responsible, and for the faculty and companies seeking to simplify the process of project proposal and management.

In order to develop this system, we will use a CMS (Content Management System), which properly configured and adapted to the needs of our system will serve as a fundamental pillar for the development of the application.

Keywords: web application, web page, CMS, Drupal, Thesis.

ÍNDICE GENERAL

1. MOTIVACIÓN Y CONTEXTO DEL PROYECTO.....	8
1.1. Introducción.....	8
1.2. Objetivos.....	9
2. DESCRIPCIÓN DEL SISTEMA Y ANALISIS DE REQUISITOS.....	10
2.1. Descripción del sistema	10
2.2. Análisis de requisitos	11
2.2.1. Actores del sistema	11
2.2.2. Requisitos funcionales.....	12
2.2.3. Requisitos no funcionales.....	13
3. METODOLOGÍA DE DESARROLLO	14
3.1. Fases planificadas del proyecto	14
4. ESPECIFICACIÓN DE CASOS DE USO.....	16
4.1. Diagramas de caso de uso	16
4.1.1. Diagrama para el rol Usuario Anónimo/Alumno.....	16
4.1.2. Diagrama para rol Empresa	16
4.1.3. Diagrama para rol Profesor	17
4.1.4. Diagrama para rol Miembro CAG	17
4.1.5. Diagrama para rol Presidente CAG	17
4.2. Plantillas de caso de uso	18
5. CONTENT MANAGEMENT SYSTEM (CMS)	25
5.1. Introducción al CMS.....	25
5.1.1. Principales ventajas del uso de un CMS	25
5.2. Estructura de un CMS.....	26
5.3. Elección CMS	26
5.3.1. Drupal.....	27
5.3.2. Wordpress	28
5.3.3. Mi elección: Drupal	29
6. HERRAMIENTAS, LENGUAJES Y TECNOLOGÍAS UTILIZADAS	31
6.1. Lenguajes	31
6.1.1. PHP	31
6.1.2. HTML.....	31
6.1.3. CSS.....	32
6.1.4. JavaScript.....	32
6.2. Herramientas.....	33

6.2.1. Servidor Web: Bitnami Drupal Stack	33
6.2.2. Navegador Web: Mozilla Firefox.....	33
6.2.3. Editor de Código: Atom	34
6.2.4. Editor de Imágenes: Adobe Photoshop	34
6.2.5. Gestor FTP: FileZilla	34
6.2.6. Violet UML Editor.....	34
7. DISEÑO E IMPLEMENTACIÓN	35
7.1. Tipos de Contenido (Nodo)	35
7.2. Módulos.....	36
7.2.1. Views.....	36
7.2.2. Rules.....	38
7.2.3. Workflow.....	40
7.2.4. Workflow fields.....	41
7.2.5. LDAP	41
7.2.6. SMTP Authentication	42
7.2.7. Twitter	42
7.2.8. User Reference.....	43
7.2.9. Node Access User Reference.....	43
7.2.10. View Unpublished.....	43
7.2.11. Captcha	43
7.3. Bloques	44
7.4. Roles y Permisos	45
7.5. Apariencia	45
7.5.1. Nuestro Theme: Esteem	46
8. PRUEBAS Y VALIDACIONES.....	47
8.1. Pruebas unitarias.....	47
8.2. Pruebas de integración	47
8.3. Pruebas de sistema	47
8.4. Pruebas de aceptación.....	48
9. CONCLUSIONES Y TRABAJOS FUTUROS	49
9.1. Conclusiones	49
9.2. Trabajos futuros	49
9.2.1. Drupal 8 y versiones posteriores.....	49
9.2.2 “Universalizar” la herramienta	49
BIBLIOGRAFÍA	50

ÍNDICE DE TABLAS

Tabla 1: Actores del sistema.....	12
Tabla 2: Requisitos funcionales	13
Tabla 3: Requisitos no funcionales	13
Tabla 4: Iteraciones planificadas del sistema	15
Tabla 5: Plantilla Inicio de Sesión	18
Tabla 6: Plantilla Listar Proyectos Ofertados.....	18
Tabla 7: Plantilla Listar Proyectos Asignados.....	19
Tabla 8: Plantilla Listar Proyectos Finalizados	19
Tabla 9: Plantilla Ver Información del Proyecto.....	19
Tabla 10: Plantilla Crear Proyecto	20
Tabla 11: Plantilla Listar Mis Proyectos.....	20
Tabla 12: Plantilla Listar Proyectos Codirigidos.....	21
Tabla 13: Plantilla Listar Proyectos a Codirigir	21
Tabla 14: Plantilla Editar Proyecto	21
Tabla 15: Plantilla Eliminar Proyecto	22
Tabla 16: Plantilla Finalizar Proyecto	22
Tabla 17: Plantilla Asignar Proyecto.....	22
Tabla 18: Plantilla Desasignar Proyecto.....	23
Tabla 19: Plantilla Añadir Comentario	23
Tabla 20: Plantilla Establecer Codirector.....	24
Tabla 21: Plantilla Listar Proyectos Eliminados	24
Tabla 22: Listar Proyectos Pendientes de Aprobación	24
Tabla 23: Rule Aprobar Proyecto	38
Tabla 24: Rule Cargar Datos Tutor	38
Tabla 25: Rule Cargar Datos Empresa	38
Tabla 26: Rule Notificación Proyecto UC.....	39
Tabla 27: Rule Notificación Proyecto Empresa	39
Tabla 28: Rule Notificación Nuevo Comentario.....	39
Tabla 29: Rule Cargar Campo Estado	39
Tabla 30: Rule Redirección tras Login Profesor	39
Tabla 31: Rule Redirección tras Login CAG	39
Tabla 32: Rule Notificación Caducidad de Proyecto	40
Tabla 33: Rule Deshabilitar Comentarios	40
Tabla 34: Rule Proyecto Cancelado Presidente CAG.....	40
Tabla 35: Rule Proyecto Cancelado Autor.....	40
Tabla 36: Rule Carga Datos Codirector y Notificación	40
Tabla 37: Bloques definidos manualmente en el sistema	45

ÍNDICE DE FIGURAS

Figura 1: Captura actual del sistema (pfciencias.unican.es)	8
Figura 2: Ciclo Ideal de un proyecto en el sistema	11
Figura 3: Esquema del Modelo Iterativo Incremental [3]	14
Figura 4: Diagrama de casos de uso para el rol Anónimo/Alumno.....	16
Figura 5: Diagrama de casos de uso para el rol Empresa	16
Figura 6: Diagrama de casos de uso para el rol Profesor	17
Figura 7: Diagrama de casos de uso para rol Miembro CAG.....	17
Figura 8: Diagrama de casos de uso para el rol Presidente CAG.....	17
Figura 9: Capas de un CMS	26
Figura 10: Definición para código JQuery	32
Figura 11: Estructura de Capas Drupal.....	35
Figura 12: Configuración del tipo Proyecto.....	36
Figura 13: Transiciones de Estados.....	41
Figura 14: Configuración para el estado Propuesto	41
Figura 15: Captura de un twitt publicado	42
Figura 16: Inicio de Sesión con Captcha	43
Figura 17: Bloques definidos para nuestro theme (Esteem)	44
Figura 18: Theme Garland	46

1. MOTIVACIÓN Y CONTEXTO DEL PROYECTO

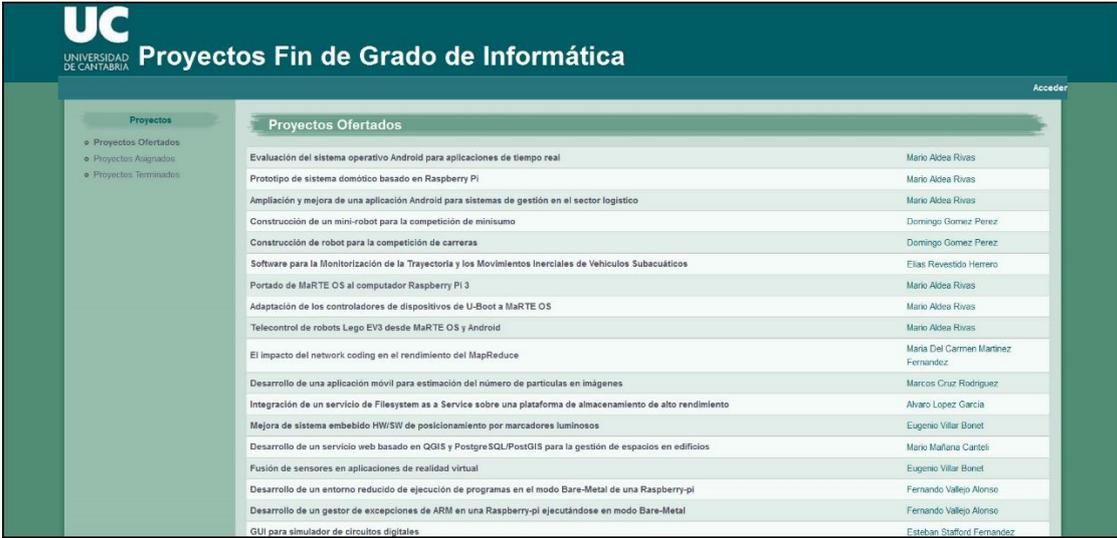
1.1. Introducción

En la Universidad de Cantabria (UC) se es consciente de la importancia que en la actualidad tienen las herramientas de Internet, y es por esta razón por la que desde hace años el núcleo de la Universidad funciona en base a estas herramientas. Dentro de los servicios web que esta institución ofrece, podemos destacar, entre otros, la recientemente renovada Web Oficial (www.unican.es), el Campus Virtual, el Aula Virtual, el repositorio UCrea y Moodle.

Como se puede observar, existen un conjunto de servicios gestionados por la Universidad cuya finalidad no es otra que la de ayudar a los usuarios a cubrir sus necesidades. Sin embargo, hay aspectos, como la gestión de la información acerca de los TFG, que no disponían de un soporte oficial global por parte de la Universidad, por lo que han sido las propias Escuelas y Facultades las que han tenido que buscar una solución a esta carencia.

Debido a esta situación, la Facultad de Ciencias de la Universidad de Cantabria decidió crear, en 2012, un sistema exclusivo para los miembros de Ingeniería Informática para gestionar e informar acerca de los aspectos más relevantes de los TFG. Años más tarde, se decidió adaptar dicho sistema a las otras titulaciones impartidas en la Facultad.

Pasado el tiempo, nace la necesidad de mejorar y adaptar el sistema anteriormente creado para la gestión de los TFG con el fin de corregir los fallos descubiertos durante sus años de uso, así como de dar una respuesta a las nuevas necesidades requeridas.



The screenshot shows the 'Proyectos Fin de Grado de Informática' website. The header includes the UC logo and the text 'UNIVERSIDAD DE CANTABRIA' and 'Proyectos Fin de Grado de Informática'. A navigation menu on the left lists 'Proyectos', 'Proyectos Ofertados', 'Proyectos Asignados', and 'Proyectos Terminados'. The main content area is titled 'Proyectos Ofertados' and contains a table with the following data:

Proyecto	Responsable
Evaluación del sistema operativo Android para aplicaciones de tiempo real	Mario Aldea Rivas
Prototipo de sistema doméstico basado en Raspberry Pi	Mario Aldea Rivas
Ampliación y mejora de una aplicación Android para sistemas de gestión en el sector logístico	Mario Aldea Rivas
Construcción de un mini-robot para la competición de minisumo	Domingo Gomez Perez
Construcción de robot para la competición de carreras	Domingo Gomez Perez
Software para la Monitorización de la Trayectoria y los Movimientos Inerciales de Vehículos Subacuáticos	Elias Revestido Herrero
Portado de MaRTE OS al computador Raspberry Pi 3	Mario Aldea Rivas
Adaptación de los controladores de dispositivos de U-Boot a MaRTE OS	Mario Aldea Rivas
Telecontrol de robots Lego EV3 desde MaRTE OS y Android	Mario Aldea Rivas
El impacto del network coding en el rendimiento del MapReduce	Maria Del Carmen Martinez Fernandez
Desarrollo de una aplicación móvil para estimación del número de partículas en imágenes	Marcos Cruz Rodriguez
Integración de un servicio de Filesystem as a Service sobre una plataforma de almacenamiento de alto rendimiento	Alvaro Lopez Garcia
Mejora de sistema embebido HW/SW de posicionamiento por marcadores luminosos	Eugenio Villar Bonet
Desarrollo de un servicio web basado en QGIS y PostgreSQL/PostGIS para la gestión de espacios en edificios	Mario Mallana Cartelli
Fusión de sensores en aplicaciones de realidad virtual	Eugenio Villar Bonet
Desarrollo de un entorno reducido de ejecución de programas en el modo Bare-Metal de una Raspberry-pi	Fernando Vallejo Alonso
Desarrollo de un gestor de excepciones de ARM en una Raspberry-pi ejecutandose en modo Bare-Metal	Fernando Vallejo Alonso
GUI para simulador de circuitos digitales	Esteban Stafford Fernandez

Figura 1: Captura actual del sistema (pfciencias.unican.es)

1.2. Objetivos

Una vez hecha la introducción, y con el fin de ofrecer una solución a los problemas encontrados, se definen los siguientes objetivos generales de la nueva versión del sistema:

- Facilitar a los alumnos de la Facultad de Ciencias de la UC la búsqueda y posterior elección de los Trabajos de Fin de Grado (TFG).
- Mejorar la difusión y conocimientos de los TFG ofertados por la Facultad de Ciencias.
- Mejorar, simplificar y agilizar el proceso de propuesta y aprobación de proyectos, tanto de profesores como de empresas.
- Mejorar el sistema de comunicación entre las empresas y la Facultad de Ciencias.
- Adaptar el entorno web a las especificaciones e interfaces actuales, especialmente en el acceso a través de dispositivos móviles o tabletas.

2. DESCRIPCIÓN DEL SISTEMA Y ANÁLISIS DE REQUISITOS

En este apartado, se describen las funcionalidades del sistema y su comportamiento. En primer lugar, se realizará un análisis detallado de los requisitos, tanto funcionales como no funcionales, y, posteriormente, de la metodología aplicada durante el desarrollo del proyecto.

2.1. Descripción del sistema

El sistema cuenta con dos secciones ampliamente diferenciadas: la parte pública y la parte privada o de acceso restringido.

Para la parte pública, nos encontramos con una aplicación web que es accesible a través de cualquier navegador. En dicha aplicación, se presenta al usuario una página inicial donde escoger la titulación que desee dentro de las ofrecidas por la Facultad de Ciencias, que son, en la actualidad, las siguientes: Grado en Física, Grado en Ingeniería Informática y Grado en Matemáticas.

Una vez escogida la titulación, el sistema mostrará el listado de proyectos ofertados. Se podrá consultar, además, el listado de proyectos ya asignados a los alumnos, así como de aquellos trabajos ya finalizados por los estudiantes de la Facultad, que servirán de referencia a los alumnos de cursos posteriores.

De cada uno de los proyectos, se podrá consultar información más detallada, siendo la información más relevante y útil para el usuario la información acerca del autor (tutor) del proyecto. Ésta servirá para el contacto con el profesor o empresa responsable del proyecto propuesto.

Aun así, la parte que más peso tendrá en el sistema es aquella bajo acceso restringido. El acceso a esta zona estará limitado a profesores de la Universidad, empresas autorizadas y a los administradores del sistema.

En esta parte es en la que más roles actúan, por lo que es la más compleja. Los profesores y empresas poseen una interfaz personalizada dónde pueden crear proyectos. Éstos, una vez creados, deben pasar por una fase de aprobación que gestiona la Comisión Académica de Grado (CAG) que, mediante anotaciones, sugiere cambios sobre lo propuesto. Ésta es la única fase en la que se puede modificar el proyecto, ya que, una vez que el presidente de la CAG apruebe un proyecto, el contenido del mismo no será modificable. Establecido el proyecto como ofertado, podrá pasar por tres estados diferentes más: asignado, finalizado o eliminado. Otra relevante funcionalidad del sistema es la capacidad de establecer un codirector a cualquiera de los proyectos propuestos, bien sean propuestos por una empresa o por profesores de la Universidad. El codirector solo podrá ser un profesor de la Universidad.

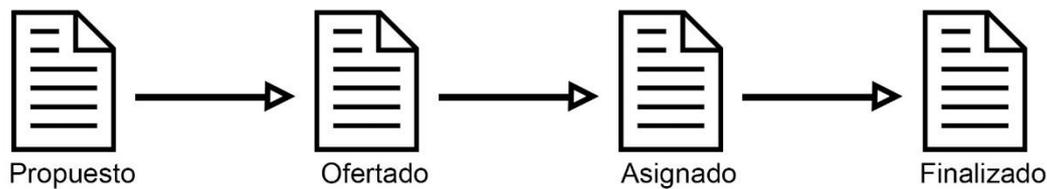


Figura 2: Ciclo Ideal de un proyecto en el sistema

Los miembros de la CAG ven aumentada su interfaz añadiendo un apartado con el listado de los proyectos que se encuentran aún pendientes de aprobación. Sobre estos proyectos los miembros de la CAG hacen anotaciones valorativas, tanto positivas, como negativas. Éstas son recibidas por el autor del proyecto y los otros miembros de la CAG. Finalmente, dentro de este grupo existe un sub-rol específico para el presidente de la CAG que, además de realizar las funciones anteriormente descritas, es quien decide aprobar finalmente un proyecto.

2.2. Análisis de requisitos

Los requisitos funcionales definen el comportamiento de un sistema software y sus componentes. Un requisito es una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. [1]

El análisis de requisitos es una etapa clave en el ciclo del proyecto, ya que se estima que su coste es alrededor del 10-15% del total. Un error en los requisitos puede llegar a ser más costoso que un error en el código, debido a que se arrastra en las demás del proyecto. [2]

2.2.1. Actores del sistema

Los requisitos de un software suelen ser una combinación compleja de los requisitos de diferentes actores en diferentes niveles de una organización y del entorno en el cual operará el software [3]. Debido a esta complejidad en la variedad de roles que posee nuestro sistema, se ha decidido realizar una pequeña tabla introductoria con los diferentes actores (roles) del sistema.

Actor	Descripción
Usuario público/Alumno	Tiene acceso a la información pública del sitio, puede ver información acerca de los proyectos. Su cometido es informarse para posteriormente contactar con el tutor. Principalmente alumnos.
Empresa	Debe estar autorizada previamente por el administrador. Puede proponer proyectos y gestionar el mismo pero necesitará de un codirector que pertenezca a la Universidad.
Profesor	Puede proponer proyectos, se encarga tras su aprobación de la tutorización del mismo, aunque también puede codirigir proyectos externos de empresas o de otros profesores de la Universidad.
Miembro de la CAG	Realiza comentarios sobre los proyectos que se encuentran propuestos, estos comentarios servirán para determinar la aprobación o revisión del proyecto.
Presidente de la CAG	Posee las mismas características que un miembro de la CAG. Además es el responsable de aprobar finalmente los proyectos propuestos, su

	decisión en principio vendrá determinada por los comentarios hechos por los distintos miembros de la CAG.
Administrador del sistema	Es el súper-usuario del sistema, en principio sus tareas se limitan al mantenimiento y gestión del sitio, aunque tendrá plenos poderes para ejecutar cualquier acción disponible para los demás actores.

Tabla 1: Actores del sistema

2.2.2. Requisitos funcionales

Una vez explicados los roles del sistema, describimos los diferentes requisitos identificados en el sistema y que deberán ser cumplidos a la finalización del mismo.

ID	Descripción
RF01	El sistema mostrará públicamente el listado de proyectos ofertados.
RF02	El sistema mostrará públicamente el listado de proyectos asignados.
RF03	El sistema mostrará públicamente el listado de proyectos finalizados.
RF04	El sistema mostrará el contenido detallado de cada uno de los proyectos, en cualquiera de sus posibles estados. Entendiéndose por contenido: nombre, descripción, persona de contacto, correo de contacto, empresa, departamento y etiquetas identificativas.
RF05	El sistema mostrará información acerca del codirector del proyecto (nombre y correo electrónico) si éste se encuentra en un proyecto.
RF06	El sistema mostrará el nombre del alumno asignado cuando el proyecto se encuentre en estado asignado.
RF07	El sistema mostrará la opción de visualización de la memoria mediante un enlace al repositorio <i>UCrea</i> cuando el proyecto se encuentre en estado finalizado.
RF08	Un profesor podrá crear propuestas de proyecto para su aprobación.
RF09	Una empresa autorizada podrá crear propuestas de proyecto para su aprobación.
RF10	El sistema mostrará un listado de los proyectos que el profesor o empresa tenga creados.
RF11	El sistema notificará por correo electrónico a los miembros de la CAG cuando un proyecto sea propuesto.
RF12	El sistema mostrará un listado de todos los proyectos que han sido propuestos por profesores y empresas externas que aún estén pendientes de aprobación.
RF13	Los miembros de la CAG podrán realizar anotaciones sobre un proyecto propuesto.
RF14	El sistema notificará por correo electrónico al autor y a los miembros de la CAG cuando se realicen anotaciones sobre un proyecto.
RF15	El profesor podrá modificar el contenido de un proyecto siempre y cuando no haya sido aprobado aún por el presidente de la CAG.
RF16	El sistema notificará por correo electrónico al autor del proyecto cuando este sea aprobado por la CAG.
RF17	El profesor y presidente de la CAG podrán asignar diferentes estados al proyecto una vez éste haya sido aprobado. Siendo los posibles estados: ofertado, asignado, finalizado y eliminado.
RF18	El sistema mostrará al presidente de la CAG un listado de los proyectos eliminados.
RF19	El sistema mostrará al profesor un listado de proyectos de otros profesores y empresas que se encuentren sin codirector.
RF20	Un profesor podrá codirigir proyectos creados por otros profesores o empresas autorizadas.
RF21	El presidente de la CAG podrá asignar un codirector a cualquier proyecto.

RF22	El profesor/empresa autor del proyecto podrá asignar un codirector a su proyecto.
RF23	El sistema notificará por correo electrónico al profesor cuando éste haya sido asignado como codirector de un proyecto.
RF24	El presidente de la CAG podrá escoger a otros miembros para que formen parte del rol miembro de la CAG.
RF25	El sistema notificará por correo electrónico al autor y a los miembros de la CAG cuando un proyecto propuesto esté próximo a caducar (2 años).
RF26	El sistema no permitirá solicitudes de registro de usuarios.
RF27	El administrador del sistema podrá editar la estética de la página.
RF28	El administrador del sistema podrá gestionar los roles del sistema.
RF29	El administrador del sistema deberá mantener actualizado el sistema.

Tabla 2: Requisitos funcionales

2.2.3. Requisitos no funcionales

Anteriormente hemos descrito comportamientos del sistema y en este apartado describiremos las restricciones sobre las funciones o servicios ofrecidos por el sistema. Nos podemos referir a ellos también como requisitos de calidad del sistema.

En la Tabla 3 se identifican nuestros requisitos no funcionales, especificando para cada uno de ellos una categoría y una importancia concreta:

ID	Descripción	Categoría	Importancia
RNF1	La ruta de autenticación del sistema no será la dada por defecto en el sistema.	Seguridad	○ Media
RNF2	El sistema adaptará los elementos visuales al tamaño de la ventana del navegador.	Accesibilidad, Portabilidad	○ Alta
RNF3	El sistema se adaptará a los distintos dispositivos actuales (ordenador, tableta y teléfono móvil).	Accesibilidad, Portabilidad	○ Alta
RNF4	La hoja de estilos (CSS) deberá estar definida mediante unos patrones claros y no recurrentes.	Rendimiento	○ Media
RNF5	Las imágenes del sistema deberán estar correctamente comprimidas para no ralentizar el sistema.	Rendimiento	○ Media
RNF6	El sistema utilizará el directorio de la UC para autenticar a los usuarios PAS y PDI	Seguridad	○ Alta
RNF7	La interfaz del sistema usará menús intuitivos y fácilmente reconocibles para los usuarios.	Usabilidad	○ Media

Tabla 3: Requisitos no funcionales

3. METODOLOGÍA DE DESARROLLO

Todo proyecto software debe definir una metodología que sirva para desarrollar correctamente nuestro sistema. El uso de las metodologías está pensado para mejorar la productividad durante el desarrollo del sistema, así como para obtener una mayor calidad en el producto final.

Para desarrollar nuestro sistema, se ha decidido emplear una metodología de Desarrollo Iterativo Incremental, que consiste en dividir el proyecto en diferentes iteraciones, que también podrían ser entendidas como “mini proyectos”. Todos estos mini proyectos poseen una estructura interna de trabajo similar, que se muestra en el siguiente esquema.

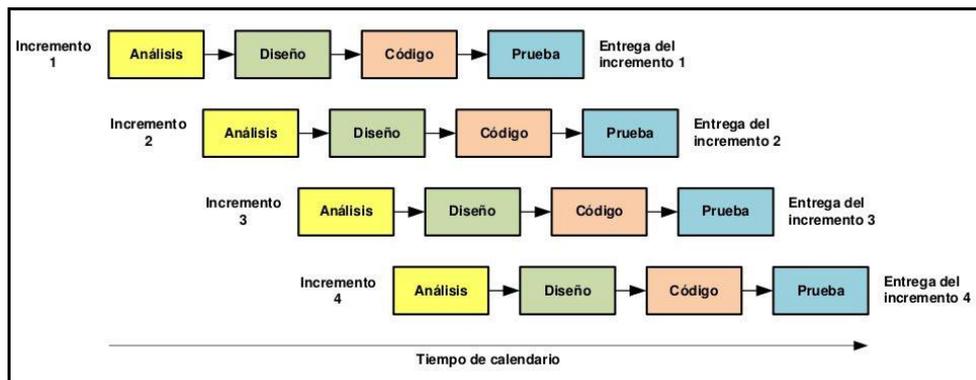


Figura 3: Esquema del Modelo Iterativo Incremental [3]

Como se observa en la Figura 3, se realizan entregas periódicas del proyecto. Cada una de esas entregas o iteraciones corresponden a lo que llamaremos también como versiones del proyecto. Al final de cada iteración se obtienen una versión. Gracias a este tipo de metodologías, es posible conocer la situación real de nuestro proyecto.

3.1. Fases planificadas del proyecto

Al final de cada una de las fases de nuestro proyecto, se deben satisfacer varios de los requisitos funcionales descritos con anterioridad, siendo en la versión final donde aparecen todos ellos satisfechos.

Es por ello por lo que se recomienda realizar una planificación de las distintas fases, especificando los distintos requisitos que se acometerán en cada una de ellas. Esa información es mostrada en la Tabla 4:

Fase	Descripción	Requisitos englobados
1	Implementación del concepto “Proyecto” y sus diferentes estados en el sistema. Creación de los roles del sistema.	RF01-RF10.
2	Gestión de los permisos en cada estado del proyecto.	RF15 y RF17.
3	Creación de la funcionalidad específica para cada rol del sistema.	RF12, RF13, RF18, RF24 y RF26- RF29.
4	Implementación notificaciones en el sistema.	RF11, RF14, RF16, RF23 y RF25.
5	Implementación de la codirección en los proyectos.	RF19-RF22.
6	Modificación y adaptación estética del sistema.	RNF2, RNF3-RNF5 y RNF7.
7	Integración del inicio de sesión mediante LDAP.	RNF6

Tabla 4: Iteraciones planificadas del sistema

4. ESPECIFICACIÓN DE CASOS DE USO

En este apartado, se describe el comportamiento general de nuestra aplicación, especificando las diferentes actividades y sus funciones. Para ello, se usarán diferentes diagramas de casos de uso que indican las diferentes funcionalidades que tendrá el sistema. Posteriormente, mediante el uso de plantillas, se describirán los casos de uso en un mayor detalle para ayudar a la comprensión total del mismo.

4.1. Diagramas de caso de uso

A continuación, se describen los diferentes casos de uso del sistema. Para que estos se puedan apreciar de una manera más clara y, debido a las características de nuestro sistema, decidimos separarlos por roles.

4.1.1. Diagrama para el rol Usuario Anónimo/Alumno

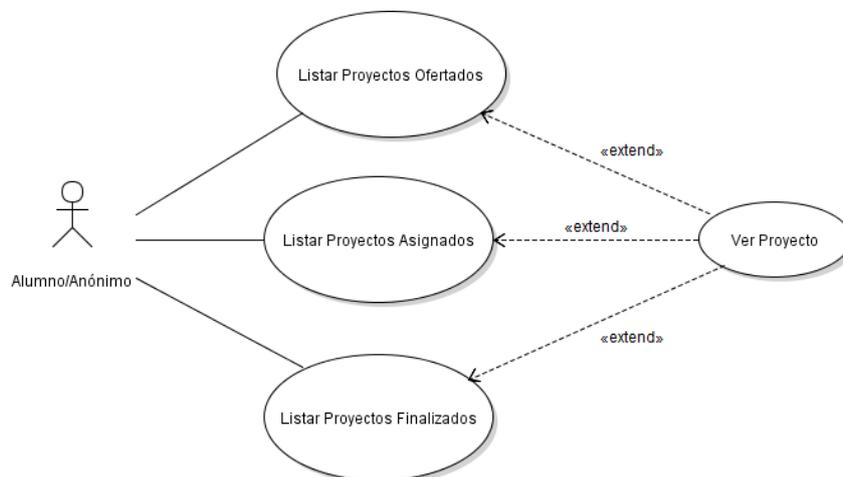


Figura 4: Diagrama de casos de uso para el rol Anónimo/Alumno

4.1.2. Diagrama para rol Empresa

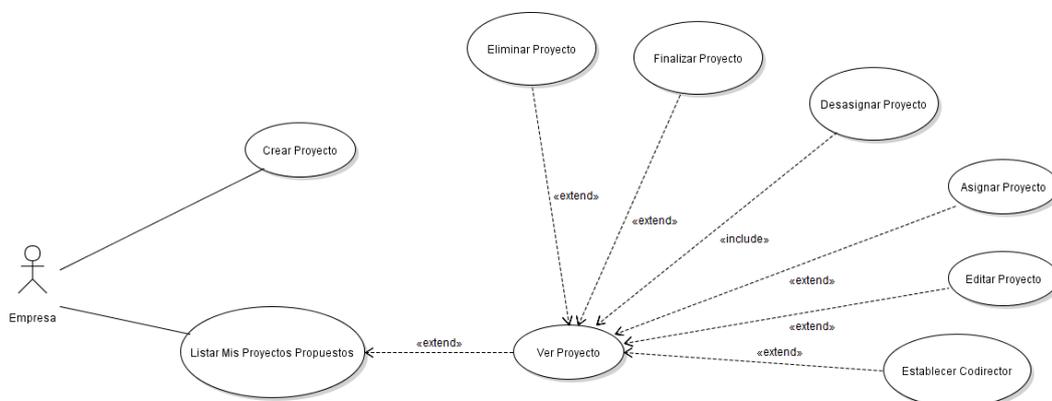


Figura 5: Diagrama de casos de uso para el rol Empresa

4.1.3. Diagrama para rol Profesor

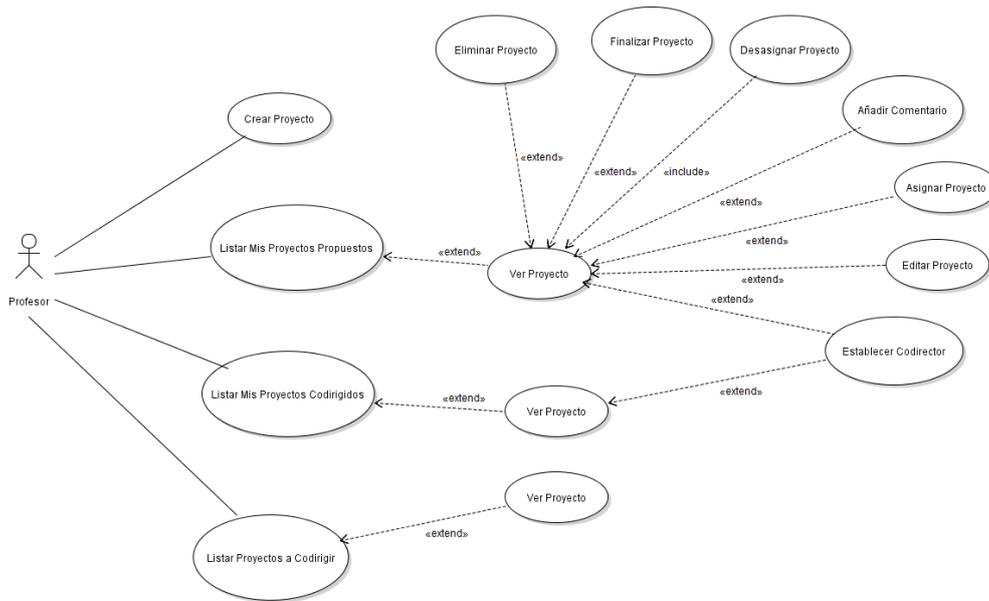


Figura 6: Diagrama de casos de uso para el rol Profesor

4.1.4. Diagrama para rol Miembro CAG

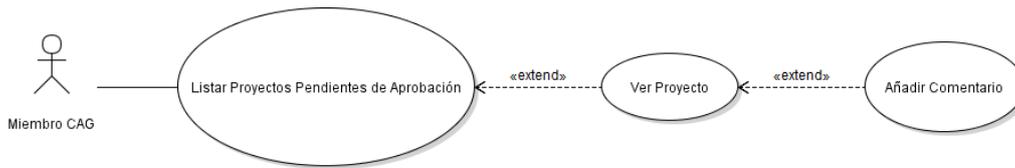


Figura 7: Diagrama de casos de uso para rol Miembro CAG

4.1.5. Diagrama para rol Presidente CAG

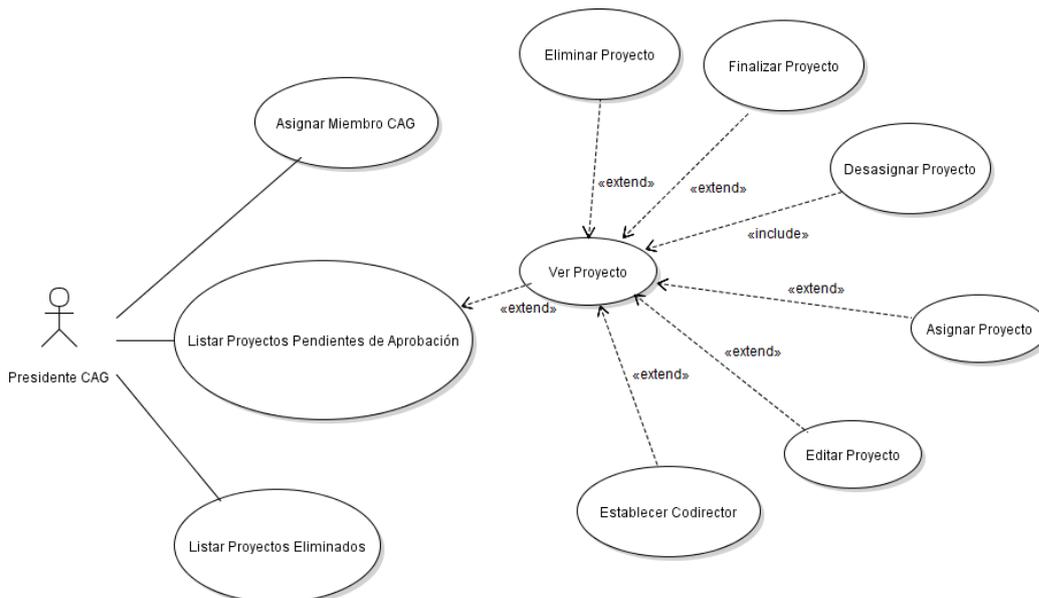


Figura 8: Diagrama de casos de uso para el rol Presidente CAG

4.2. Plantillas de caso de uso

Para mayor comprensión de los casos de uso que aparecen en los diagramas, se detallan a continuación mediante plantilla los diferentes casos de uso que existen en el sistema.

Plantilla	
ID	TFG01
Nombre	Inicio de Sesión
Descripción	El usuario inicia sesión en el sistema.
Actores	Administrador, Profesor, Empresa, Miembro CAG y Presidente de la CAG.
Precondiciones	El usuario tiene que poseer cuenta en la aplicación (Administrador y Empresa) o estar dado de alta en la base de datos PAS/PDI de la Universidad de Cantabria.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario pulsa la opción “Iniciar Sesión”. 2. El sistema muestra un formulario con dos campos a rellenar por el usuario (Nombre de usuario y contraseña) además de un Captcha. 3. El sistema carga la página personal del usuario en función de su rol.
Flujo alternativo	2.2. El sistema muestra un mensaje indicando que ha habido un error en el inicio de sesión.

Tabla 5: Plantilla Inicio de Sesión

Plantilla	
ID	TFG02
Nombre	Listar Proyectos Ofertados
Descripción	El usuario visualiza el listado de proyectos ofertados en la titulación.
Actores	Todos
Precondiciones	Ninguna
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción del menú “Proyectos Ofertados”. 2. El sistema muestra un listado con los proyectos que actualmente se encuentran ofertados a los alumnos.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos ofertados en ese momento.

Tabla 6: Plantilla Listar Proyectos Ofertados

Plantilla	
ID	TFG03
Nombre	Listar Proyectos Asignados
Descripción	El usuario visualiza el listado de proyectos asignados en la titulación.
Actores	Todos.
Precondiciones	Ninguna.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción del menú “Proyectos Asignados”. 2. El sistema muestra un listado con los proyectos que actualmente se encuentran asignados a alumnos.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos asignados en ese momento.

Tabla 7: Plantilla Listar Proyectos Asignados

Plantilla	
ID	TFG04
Nombre	Listar Proyectos Finalizados
Descripción	El usuario visualiza el listado de proyectos que han sido finalizados en la titulación.
Actores	Todos.
Precondiciones	Ninguna.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción del menú “Proyectos Finalizados”. 2. El sistema muestra un listado con los proyectos que ya han sido finalizados.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos finalizados en ese momento.

Tabla 8: Plantilla Listar Proyectos Finalizados

Plantilla	
ID	TFG05
Nombre	Ver Información del Proyecto
Descripción	El usuario visualiza la información específica de un proyecto creado en el sistema.
Actores	Todos.
Precondiciones	Deben existir proyectos en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario, a través de uno de los diferentes listados de proyectos, hace clic sobre el nombre de uno de ellos. 2. El sistema identifica el proyecto que ha sido pulsado. 3. El sistema carga la página específica que muestra la información de dicho proyecto.

Tabla 9: Plantilla Ver Información del Proyecto

Plantilla	
ID	TFG06
Nombre	Crear Proyecto
Descripción	El usuario selecciona la opción de crear proyecto, el sistema muestra un formulario con diferentes campos en función del rol.
Actores	Profesor y Empresa.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Crear Proyecto”. 2. El sistema carga un formulario con campos a rellenar por el usuario. 3. El usuario rellena los campos requeridos. 4. El sistema registra el proyecto en la base de datos del sistema y pasa a estado ‘Propuesto’ (No publicado). 5. El sistema añade el proyecto al listado “Mis Proyectos”. 6. El sistema envía un correo a los Miembros CAG, Presidente CAG y al usuario propietario del proyecto (Profesor o Empresa).
Flujo alternativo	4.3. El sistema muestra un mensaje indicando que existen campos requeridos que no han sido rellenados.

Tabla 10: Plantilla Crear Proyecto

Plantilla	
ID	TFG07
Nombre	Listar Mis Proyectos Propuestos
Descripción	El usuario visualiza el listado de proyectos que ha propuesto con independencia del estado en el que se encuentre.
Actores	Profesor y Empresa.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Mis Proyectos” 2. El sistema carga un listado con los proyectos que han sido creados por el profesor o empresa.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos propuestos por el usuario.

Tabla 11: Plantilla Listar Mis Proyectos

Plantilla	
ID	TFG08
Nombre	Listar Mis Proyectos Codirigidos
Descripción	El usuario visualiza el listado de proyectos en los que está establecido como codirector.
Actores	Profesor.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Mis Proyectos Codirigidos” 2. El sistema carga un listado con los proyectos que están siendo codirigidos por el profesor.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos codirigidos por el usuario.

Tabla 12: Plantilla Listar Proyectos Codirigidos

Plantilla	
ID	TFG09
Nombre	Listar Proyectos a Codirigir
Descripción	El usuario visualiza el listado de proyectos que se encuentran disponibles para codirigir.
Actores	Profesor.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Ver Proyectos a codirigir” 2. El sistema carga un listado con los proyectos que se encuentran sin codirector actualmente.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos a codirigir en este momento.

Tabla 13: Plantilla Listar Proyectos a Codirigir

Plantilla	
ID	TFG10
Nombre	Editar Proyecto
Descripción	El usuario selecciona edita el contenido de un proyecto.
Actores	Profesor, Empresa y Presidente CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema. En el caso del Profesor/Empresa debe ser el autor del Proyecto o el Codirector del mismo.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Editar” dentro de un proyecto. 2. El sistema carga solos los campos que son editables en ese estado y para ese rol. 3. El usuario escribe y actualiza los campos que necesite. 4. El sistema registra los cambios en la base de datos.
Flujo alternativo	4.1. El sistema muestra un mensaje indicando que existen campos requeridos que no han sido rellenados.

Tabla 14: Plantilla Editar Proyecto

Plantilla	
ID	TFG11
Nombre	Eliminar Proyecto
Descripción	El usuario establece como eliminado un proyecto.
Actores	Profesor, Empresa y Presidente de la CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema. En el caso del Profesor/Empresa debe ser el autor del Proyecto.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Eliminar Proyecto” dentro de un proyecto. 2. El sistema solicita rellenar un campo de justificación. 3. El usuario rellena el campo requerido. 4. El sistema retira el proyecto y lo añade a la lista de Proyectos Eliminados.

Tabla 15: Plantilla Eliminar Proyecto

Plantilla	
ID	TFG12
Nombre	Finalizar Proyecto
Descripción	El usuario establece como finalizado un proyecto.
Actores	Profesor, Empresa y Presidente de la CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema. En el caso del Profesor/Empresa debe ser el autor del Proyecto. El proyecto debe estar previamente en estado “Asignado”.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Finalizar Proyecto” dentro de un proyecto. 2. El sistema establece el proyecto como Finalizado y lo añade a la lista de Proyectos Finalizados.

Tabla 16: Plantilla Finalizar Proyecto

Plantilla	
ID	TFG13
Nombre	Asignar Proyecto
Descripción	El usuario establece asignado un proyecto.
Actores	Profesor, Empresa y Presidente de la CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema. En el caso del Profesor/Empresa debe ser el autor del Proyecto. El proyecto debe estar previamente en estado “Ofertado”.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Asignar Proyecto” dentro de un proyecto. 2. El sistema solicita rellenar el campo “Alumno Asignado”. 3. El usuario rellena el campo requerido. 4. El sistema establece el proyecto como Asignado y lo añade a la lista de Proyectos Asignados.

Tabla 17: Plantilla Asignar Proyecto

Plantilla	
ID	TFG14
Nombre	Desasignar Proyecto
Descripción	El usuario establece de nuevo el proyecto como ofertado.
Actores	Profesor, Empresa y Presidente de la CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema. En el caso del Profesor/Empresa debe ser el autor del Proyecto. El proyecto debe estar previamente en estado "Asignado".
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción "Volver a Asignar Proyecto" dentro de un proyecto. 2. El sistema borra el nombre del alumno anteriormente asignado. 3. El sistema establece el proyecto como Ofertado y lo añade a la lista de Proyectos Ofertados.
Flujo alternativo	-

Tabla 18: Plantilla Desasignar Proyecto

Plantilla	
ID	TFG15
Nombre	Añadir Comentario
Descripción	El usuario añade un comentario a un proyecto del sistema.
Actores	Profesor, Empresa, Miembro CAG y Presidente CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema. En el caso del Profesor/Empresa debe ser el autor del Proyecto.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona el proyecto sobre el que desea añadir un comentario. 2. El sistema carga en la parte inferior un cuadro de texto. 3. El usuario rellena el cuadro de texto y pulsa el botón enviar. 4. El sistema guarda el comentario en el sistema. 5. El sistema envía un correo a los Miembros CAG, Presidente CAG y al usuario propietario del proyecto (Profesor o Empresa).
Flujo alternativo	

Tabla 19: Plantilla Añadir Comentario

Plantilla	
ID	TFG16
Nombre	Establecer Codirector
Descripción	El usuario establece un codirector para el proyecto.
Actores	Profesor, Empresa y Presidente CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema. En el caso del Profesor/Empresa debe ser el autor del Proyecto. El proyecto debe estar en estado “Ofertado” o “Asignado”.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona el proyecto sobre el que desea establecer un codirector. 2. El usuario pulsa editar. 3. El usuario rellena el campo codirector. 4. El sistema guarda los datos en el sistema. 5. El sistema otorga permisos de edición al codirector. 6. El sistema envía un correo al usuario referenciado.
Flujo alternativo	4.1. El profesor referenciado no existe en el sistema.

Tabla 20: Plantilla Establecer Codirector

Plantilla	
ID	TFG17
Nombre	Listar Proyectos Eliminados
Descripción	El usuario visualiza el listado de proyectos que han sido eliminados en el sistema.
Actores	Presidente CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Proyectos Eliminados” 2. El sistema carga un listado con los proyectos que han sido eliminados en el sistema.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos eliminados en el sistema.

Tabla 21: Plantilla Listar Proyectos Eliminados

Plantilla	
ID	TFG18
Nombre	Listar Proyectos Pendientes de Aprobación.
Descripción	El usuario visualiza el listado de proyectos que han sido propuestos y que aún no han sido aprobados.
Actores	Miembros CAG y Presidente CAG.
Precondiciones	El usuario tiene que estar correctamente autenticado en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Proyectos Pendientes de Aprobación” 2. El sistema carga un listado con los proyectos que han sido se encuentran pendientes de aprobar.
Flujo alternativo	2.1. El sistema muestra un mensaje indicando que no existen proyectos eliminados en el sistema.

Tabla 22: Listar Proyectos Pendientes de Aprobación

5. CONTENT MANAGEMENT SYSTEM (CMS)

La mejor opción para desarrollar nuestro sistema era mediante el uso de un CMS. Sus siglas corresponden en inglés a *Content Management System*, que en castellano se traduce como Sistema de Gestión de Contenidos. A continuación, especificaremos en mayor detalle qué posibilidades nos ofrece un CMS y sus principales opciones, además de justificar por qué nos decantamos por uno de ellos finalmente.

5.1. Introducción al CMS

Un CMS se define como un software que nos permite crear y administrar contenidos de un modo cooperativo, principalmente páginas web. La principal virtud es su facilidad de uso, tanto para usuarios con conocimientos informáticos, como para aquellos que no están tan familiarizados con ellos.

Para el usuario, el funcionamiento del CMS es sencillo. Generalmente, la herramienta permite crear, editar y publicar cualquier tipo de información en la página web. A diferencia de una web estática, la información del contenido no reside en los ficheros HTML almacenados en el sistema, sino que se encuentra almacenada en una base de datos y la información se carga a medida que se haga necesario su uso.

Un CMS posee un sistema de flujo de trabajo (workflow) que permite a varias personas trabajar simultáneamente sobre el sistema. Además, posibilita el control de cada uno de esos contenidos por separado y gestionar cómo y cuándo se hacen públicos, siendo gestionado esto último por una jerarquía de roles del sistema.

Los CMS constituyen, en la actualidad, una herramienta de fuerte impacto debido a la popularidad de las páginas web donde interactúa un gran número de usuarios y es por esta razón por lo que se hace necesario establecer ciertas restricciones mediante el uso de permisos. Por lo general, estos sistemas están compuestos de dos partes: la parte pública y la parte privada.

- Parte pública: Es aquella parte a la que cualquier persona puede acceder tan sólo conociendo la ruta de la página.
- Parte privada: Generalmente se accede mediante un usuario y una contraseña específicos. Esta parte es la utilizada para la gestión del contenido y de la propia página web.

5.1.1. Principales ventajas del uso de un CMS

Además de lo anteriormente citado, existen aspectos que son relevantes a la hora de escoger un CMS como base de un proyecto.

- Extensible: Mediante módulos o plugins es bastante sencillo ir implementando nuevas funcionalidades al sistema sin alterar todo lo desarrollado anteriormente.
- Acceso desde cualquier lugar: Siempre y cuando poseamos una conexión a internet y hagamos uso de un navegador podremos gestionar contenido en el sistema.

- Facilidad de publicación: Un usuario sin necesidad de poseer grandes conocimientos puede publicar fácil y rápidamente con los permisos adecuados.
- Seguridad: Todo el peso recae en el CMS, que es quien se ocupa automáticamente de este aspecto. Cualquier posible agujero de seguridad es corregido rápidamente por la empresa propietaria del CMS y una solución puesta a disposición del usuario inmediatamente.
- Soporte Search Engine Optimization (SEO): Los CMS poseen características internas que, con una simple configuración por parte del administrador, harán que nuestro sitio esté bien posicionado para los sistemas de búsqueda.

5.2. Estructura de un CMS

Un CMS sigue una estructura clásica de 3 capas (ver Figura 9).

- Capa de Datos: Los datos quedan almacenados en una base de datos. En esta capa se encuentra todo el contenido que ha sido publicado en la web, así como los parámetros de configuración, categorías, usuarios...etc.
- Capa de Aplicación: Es la capa que actúa de intermediaria entre la capa de datos y la capa de diseño. Los ficheros pertenecientes a esta capa extraen la información desde la capa de datos y la trasladan a la capa de diseño para mostrárselo al usuario.
- Capa de Presentación: Define la estética de la aplicación web. Sobre esta capa se muestra la información almacenada en la capa de datos y que fue extraída en la capa lógica. Cuando accedemos a una página web a través de un navegador sólo vemos el resultado final, por lo que todas las capas anteriormente descritas son invisibles para el usuario final.

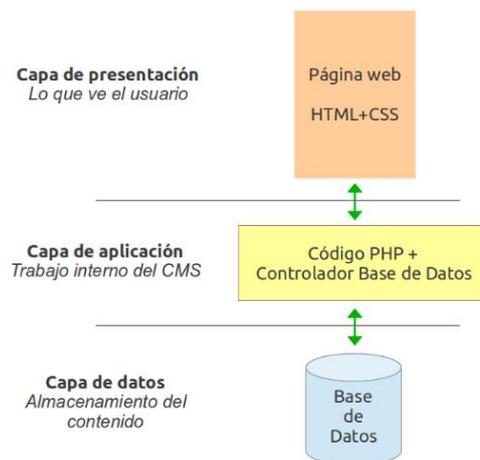


Figura 9: Capas de un CMS

5.3. Elección CMS

Establecido el proyecto sobre un CMS, uno de los puntos más relevantes es la elección del mismo. Existen muchos tipos de CMS, de hecho, se estima que existen más de 1000.

Obviamente, es inviable analizar todos ellos. Por ello, nuestro análisis se centrará en dos de los más punteros: *Drupal* y *Wordpress*.

5.3.1. Drupal

Drupal es un sistema de gestión de contenidos de código abierto cuyo uso se centra, en su mayoría, en aquellos sitios web de organizaciones que desean gestionar comunidades de usuarios y otros aspectos de la web 2.0, como son: foros, wikis, redes sociales, encuestas o votaciones. Este CMS es recomendable, también, para la difusión de artículos, imágenes u otros archivos.

Su diseño, por tanto, está pensado, especialmente, para el desarrollo y gestión de comunidades en Internet, aunque su versatilidad posibilita la producción de otro tipo de sitios web. Todo ello reunido en una plataforma que ofrece la oportunidad de editar, escribir y publicar contenidos de una manera muy simple y entendible por la mayor parte de los usuarios.

El proyecto *Drupal* es un software de código abierto. Cualquiera puede descargar, usar, trabajar y compartirlo con otros. Se basa en principios como la colaboración, el globalismo y la innovación. Se distribuye bajo los términos de la GNU General Public License (GPL). No hay cargos por licencias, nunca. *Drupal* siempre será gratis. [4]

5.3.1.1. Aspectos destacables de Drupal

- Versatilidad: Es un sistema que nos permite hacer casi cualquier proyecto que tengamos en mente. Se puede adaptar a cualquier tipo de contenido.
- Gran comunidad de desarrolladores: Importante sobre todo a la hora de resolver dudas ante problemas que nos puedan surgir mientras desarrollamos. Muchas veces no consiste en saber todo sino apoyarse en quien si sepa.
- Amplia librería de módulos: Aunque el núcleo de *Drupal* posee ya unos módulos por defecto que son bastante potentes siempre podremos complementar nuestro sistema con más módulos, todos ellos son descargables desde la web oficial y poseen información relativa a versiones, problemas y dudas en general.
- Potente para uso en comunidades: Si el proyecto necesita una fuerte gestión de registros, roles y usuarios está claro que *Drupal* es el entorno adecuado.
- Alta seguridad: Es de sus competidores directos el más seguro. Aunque es tarea del administrador que esta virtud se conserve con el tiempo realizando las actualizaciones concretas que el sistema requiera.

5.3.1.2. Carencias de Drupal

- Curva de aprendizaje: Es bastante alta respecto a otros CMS, tanto en su uso como sobre todo en su parte de desarrollo. No es un aspecto puramente negativo, pero que puede resultar un bloqueo en las primeras fases de desarrollo con *Drupal*.
- Interfaz de administración poco intuitiva: En un primer momento puede llegar incluso a desesperar porque es difícil encontrar muchos aspectos de la configuración. En algunos momentos la ordenación de algunos elementos no parece lógica.

- Poca variedad de temas: Este aspecto es principalmente relevante cuando se compara con su principal competidor: *Wordpress*. La biblioteca de temas de *Drupal* no destaca ni por su multitud de temas ni por la vistosidad de los mismos.

5.3.2. Wordpress

WordPress es un sistema de gestión de contenidos que permite la creación de cualquier tipo de sitio web, aunque se utiliza, especialmente y con gran relevancia, en el desarrollo de blogs, gracias a la gran facilidad en su manejo. Por este motivo, es muy recomendable para los principiantes y novatos o para quien apenas tenga conocimientos informáticos. De hecho, es el CMS más utilizado para la producción de blogs y el más popular entre todos los sistemas de gestión de contenidos que existen.

Es el programa idóneo para blogs debido también a su estructura, ya que se organiza en entradas que se hallan ordenadas por fecha de publicación y clasificadas en categorías. El blog puede ser diseñado fácilmente y con una amplia libertad por parte del usuario, ya que existe una gran cantidad de plantillas disponibles en la plataforma *WordPress*, todas ellas (o la inmensa mayoría) basadas en los principios de pragmatismo, sencillez e incluso elegancia.

Sin embargo, el manejo intuitivo y la estructura simple no son los únicos rasgos que definen esta plataforma, pues *WordPress* cuenta con ciertos complementos (o plugins) que potencian su uso y que pueden ser gratuitos o de pago. [4]

5.3.2.1. Aspectos destacables de WordPress

- Múltiples Autores: Debido a su origen como blog está pensando para soportar una gran cantidad de autores diferentes.
- Gran Librería de Plugins: Hay pocas cosas que *WordPress* no pueda hacer con su extensa librería de plugins.
- Amigable con el Usuario: La interfaz de usuario de *WordPress* es fácil de usar y muy intuitiva.
- Buen SEO: Aunque condicionado al número de plugins que uses en tu sitio web. Un número muy elevado de plugins podrían afectar al posicionamiento.
- Fácil de personalizar: El sistema de plantillas para *WordPress* está hecho para que sea fácil de personalizar. Se requieren conocimientos de HTML y CSS.
- Flexible: *WordPress* puede servir para cualquier tipo de web gracias a los plugins y a las plantillas personalizadas. Puede servir como portfolio personal, como web de una compañía, como tienda para compras o como un simple blog.

5.3.2.2. Carencias de WordPress

- Seguridad: El software recibe actualizaciones de seguridad de forma periódica, pero de aun así vulnerable. Existen plugins adicionales que nos pueden ayudar a no calificar este aspecto de problema.
- Incompatibilidad con plugins antiguos: El núcleo de *WordPress* está en constante actualización. Si usamos plugins, han de ser actualizados casi en paralelo para asegurarnos una correcta funcionalidad de los mismos.

- Capacidad limitada de administrar contenido: En caso de que se publicasen cientos de entradas diferentes diarias, el sistema *WordPress* se podría ver seriamente afectado en su rendimiento.

5.3.3. Mi elección: Drupal

En base a la información recabada sobre ambos CMS, es necesario decantarse por unos de ellos para comenzar a desarrollar nuestro proyecto. La elección ha de ser una decisión objetiva y debidamente justificada, por lo que debemos dejar a un lado aspectos como el hecho de que la versión anterior fuese desarrollada en *Drupal* o las propias preferencias personales.

WordPress, por defecto, ofrece una posibilidad de creación de contenido bastante limitada, ya que su núcleo tan sólo provee de entradas de blog y de páginas simples. Si quisiésemos crear tipos de datos personalizados, necesarios en nuestro proyecto, tendríamos que recurrir a plugins de terceros. En este aspecto, *Drupal*, gracias a su versión 7, permite editar los tipos de datos ya creados añadiendo campos personalizados o, si lo preferimos, desarrollar los nuestros desde cero de una manera bastante intuitiva y sin sobrecargar nuestro CMS con más módulos. Este último aspecto es bastante determinante en nuestro proyecto, ya que poseemos y necesitamos definir muchos campos específicos.

Si anteriormente citábamos la posibilidad de añadir campos personalizados como uno de los aspectos determinantes, existe otro al que nos podemos referir sin duda como el más importante: la gestión de permisos y roles. *WordPress*, debido a su origen, posee un sistema de roles muy básico. Pensado para gestionar blogs, nos provee cinco roles por defecto: Administrador, Editor, Autor, Colaborador y Suscriptor, no dejándonos crear nuevos roles a menos que instalemos plugins de terceros, y siendo la capacidad de otorgar permisos a éstos, muy simple. En este aspecto, *Drupal* nos permite crear roles sin necesidad de módulos adicionales, además de permitirnos gestionar con los permisos el control de casi cualquier elemento de nuestra página en función del rol.

Desde el punto de vista de la interfaz, ésta es clara y simple para los usuarios que utilizarán la aplicación. Ambos sistemas utilizan una interfaz específica para los temas administrativos del sitio. Siendo honestos, a nivel administrativo el *Dashboard* (Escritorio) de *WordPress* es francamente bueno ya que su interfaz simplifica mucho la gestión respecto del panel administrativo de *Drupal*. Pero es fuera del rol de administrador donde *Drupal* marca la diferencia con *WordPress*. Este último utiliza el mismo *Dashboard* para el resto de usuarios, tan sólo limita el número de opciones disponibles en función del rol, pero la interfaz administrativa se mantiene. Tampoco nos permite establecer nuevos bloques personalizados. *Drupal*, sin embargo, permite crear una interfaz totalmente personalizada, adaptada a la interfaz de nuestro sitio y coordinada con los roles, mostrando sus bloques y menús personalizados. Este aspecto está directamente relacionado con los aspectos anteriormente citados, roles y permisos.

Para complementar los aspectos teóricos que parecían casi definitivos, decidimos desarrollar una versión muy mínima del proyecto en ambos sistemas. Debido a que muchos de los requisitos aparecen en esta parte inicial, el sacrificio de tiempo iba a ser mínimo y la información obtenida de los desarrollos muy valiosa. Dicha prueba sirvió para corroborar todo lo anteriormente citado, ya que *WordPress* quedaba descolgado desde un principio en un aspecto tan importante como la gestión de roles. Determinante para nuestro sistema.

En resumen, a pesar de que, en un principio, influenciado por conocimientos previos, *WordPress* podría haberse decantado como el CMS escogido, finalmente no ha sido así. Tanto la teoría como la práctica lo han desmontado. *WordPress* es un entorno muy amigable en el que todo es muy sencillo si no se sale de su camino marcado, es decir, si el proyecto es relativamente sencillo pero, en cuanto la complejidad aumenta, algunos aspectos se complican y se hace necesario la frecuente instalación de plugins que sobrecargan el sistema. Además, en muchos casos, estos no obtienen de forma eficiente el objetivo deseado. En este aspecto, la potencia y las posibilidades de *Drupal* son casi infinitas.

6. HERRAMIENTAS, LENGUAJES Y TECNOLOGÍAS UTILIZADAS

A continuación, se describirán las diferentes herramientas, lenguajes y tecnologías implicadas en el desarrollo del proyecto.

6.1. Lenguajes

Aunque en este proyecto el desarrollo de código puro no es amplio, es necesario mencionar los diferentes lenguajes que son utilizados para hacer funcional la aplicación.

6.1.1. PHP

PHP (acrónimo recursivo de PHP: *Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.



Logo 1: PHP

Lo que distingue a PHP de otros lenguajes como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente del que proviene. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué existe por debajo. [5]

Drupal 7 requiere la versión 5.2.5 o superior, pudiéndose usar una versión inferior con los parches de seguridad adecuados. De todas formas, Drupal 7 recomienda usar siempre que sea posible la versión 5.4 o superior.

Drupal hace uso de las siguientes extensiones PHP para su correcto funcionamiento: *date*, *dom*, *filter*, *gd*, *hash*, *json*, *pcre*, *pdo*, *session*, *SimpleXML*, *SPL* y *xml*.

6.1.2. HTML



Logo 2: HTML

Sus siglas HTML significan *HyperText Markup Language*. Es un estándar para páginas web basado en etiquetas (marcas). Nació en 1990 con el objetivo de facilitar a científicos de diferentes universidades el acceso a documentos de investigación. Su creador fue Tim Berners-Lee, quien jamás pensó que su proyecto había sentado las bases de la web que conocemos en la actualidad.

Drupal 7 comenzó a ser desarrollado cuando aún HTML5 no se encontraba estandarizado. Es por ello por lo que se encuentra desarrollado bajo una versión anterior: HTML 4.01. Afortunadamente, existe un módulo llamado HTML5 Tools que nos ayuda a que HTML5 sea soportado. [6]

6.1.3. CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. [7]

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.



Logo 3: CSS

6.1.4. JavaScript

Es un lenguaje de programación usado en sistemas web que nos permite desarrollar páginas web dinámicas. Es lenguaje de programación interpretado, es decir, no es necesario compilar su código para poder ejecutarlo. Un código JavaScript es ejecutado directamente en nuestro navegador.

El uso de JavaScript en *Drupal* está regido de manera oficial mediante los siguientes principios:

- Todas las páginas deben ser perfectamente funcionales sin scripts. JavaScript proporciona alternativas o suplementos, no reemplazos para elementos estándar.
- No hay presencia de JavaScript directamente en las páginas. El código JavaScript solo se ejecuta mediante el enlace al archivo específico.



Logo 4: Javascript

6.1.4.1. JQuery

Es probablemente uno de los *frameworks* más populares de JavaScript. Un *framework* lo podríamos definir como un marco de desarrollo, el cual contiene unas librerías con unas funciones ya implementadas. Su uso por delante de *JavaScript* nativo es debido a que facilita y simplifica tareas complejas.

Afortunadamente, la librería *JQuery* (jquery.js) viene incluida ya en *Drupal* por lo que no tenemos que preocuparnos de importarla. Eso sí, si se quiere declarar alguna función *JQuery* en un archivo propio y no tener conflictos con otras librerías, se debería hacer bajo el siguiente formato: [8]

```
(function ($) {  
  // All your code here  
}(jQuery));
```

Figura 10: Definición para código JQuery

6.2. Herramientas

En esta sección se describen las herramientas que han sido necesarias para el desarrollo del conjunto de todas las partes del proyecto, tanto de la parte de desarrollo propia del proyecto como de la documentación durante y posterior al mismo.

6.2.1. Servidor Web: Bitnami Drupal Stack

Para poder desarrollar nuestro proyecto es necesario que poseamos un servidor. En nuestro caso para desarrollar el proyecto antes de su versión definitiva ha sido necesario el uso de un servidor de carácter local, es por ello por lo que se hizo uso de la aplicación *Bitnami Drupal Stack*. Se trata de una herramienta de software libre y multiplataforma que nos proporciona un servidor local con un servidor web (*Apache*) y con una base de datos (*MySQL*).



Logo 5: Bitnami

6.2.1.1. Apache

Es un servidor web HTTP de código abierto, para plataformas las plataformas más conocidas (*Unix, Microsoft Windows, Macintosh*, etc.). El hecho de que sea multiplataforma es importante, ya que es igualmente operativo en el servidor local bajo *Windows* como en el servidor final que usa *Linux*.

6.2.1.2. MySQL

Es un sistema gestor de base de datos relacional. En la mayoría de servidores y plataformas, se encuentra implementado por lo que carece de problemas de compatibilidad. Posee aspectos destacables respecto de otros gestores, tales como: bases de datos multiusuario a través de la web, escalabilidad y sobremanera su fácil despliegue y configuración.

6.2.2. Navegador Web: Mozilla Firefox

Firefox es un navegador web libre y de código abierto desarrollado para *Linux, Android, MacOS* y *Microsoft Windows* por *Mozilla*. Utiliza el motor *Gecko*, que renderiza las páginas web e implementa actuales y futuros estándares web. Un navegador web, *Mozilla* en este caso, es esencial para el desarrollo del proyecto, ya que la parte administrativa de *Drupal* funciona bajo la necesidad de un navegador y, de igual manera, es necesario para testear la estética final de la aplicación.

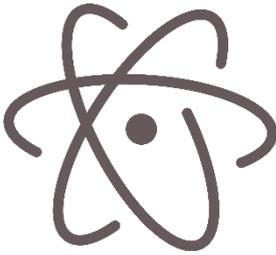


Logo 6: Mozilla
Firefox

6.2.4.1. Inspector de Elementos

Es un módulo que viene integrado en el propio navegador desde la versión 12. Nos permite conocer y editar el código HTML, la hoja de estilos (CSS) o funciones JavaScript de cualquier página web. Es útil, ya que nos permite hacer modificaciones en tiempo real que no afectan al código original pero que nos ayudan a visualizar cómo quedarán nuestros cambios si finalmente decidimos aplicarlos en nuestros archivos locales.

6.2.3. Editor de Código: Atom



Atom es un editor de código de fuente gratuito para *MacOS*, *Linux*, y *Windows* con soporte para múltiples plugins. Estos complementan al editor con funcionalidades que ayudan al usuario, como por ejemplo, un autocompletado de instrucciones. Los paquetes son *Open Source* y posee una gran comunidad. La primera versión de este editor fue lanzada en 2015.

Logo 7: Atom editor Se ha utilizado en este proyecto sobre todo por la comodidad y facilidad en su uso. Posee determinados plugins específicos del lenguaje en el que se trabaje que hacen de esta herramienta unas de las más versátiles en su campo.

6.2.4. Editor de Imágenes: Adobe Photoshop

Potente software informático que permite la edición y el retoque de imágenes. Es probablemente la aplicación mundialmente más famosa dedicada a este aspecto.

Se utilizó en una primera fase de maquetación de la interfaz del proyecto, así como de los logos o imágenes que aparecen en el sistema o en la cuenta de *Twitter*. Es un apoyo vital incluso para la realización de este documento.

6.2.5. Gestor FTP: FileZilla



Logo 8: *Filezilla*

FileZilla es un cliente FTP multiplataforma de código abierto y software libre, licenciado bajo la Licencia Pública General de GNU. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS).

Inicialmente, su uso no fue necesario debido a que mediante nuestro servidor local actuábamos directamente sobre los archivos alojados en nuestro equipo sin necesidad de un gestor FTP. Más tarde, su uso fue vital para subir los archivos al nuevo servidor. Este aspecto nos permitía la posibilidad de obtener un *Feedback* exterior que ayudase a determinar posibles fallos o mejoras de la aplicación en un estado bastante avanzado.

6.2.6. Violet UML Editor

Violet es un editor UML con una serie de beneficios. Es muy fácil de aprender a utilizarlo debido a su interfaz simplista e intuitiva. Produce contenido de gran calidad y es totalmente gratuito. [9]



Logo 9: *Violet UML Editor*

Este editor ha sido utilizado para realizar todos los diagramas de casos de uso que aparecen en este documento y que han sido necesarios a la hora de realizar el proyecto.

Es probable que no sea la herramienta más potente del mercado, pero en nuestro caso fue la más práctica, ya que en su simplicidad estaba su practicidad.

7. DISEÑO E IMPLEMENTACIÓN

Como hemos descrito con anterioridad, el diseño e implementación se realizará mediante el CMS *Drupal* en su versión 7. Para que nuestro sistema sea funcional necesitamos algo más que las especificaciones y configuraciones que trae *Drupal 7* por defecto. Para entender mejor *Drupal* debemos conocer su esqueleto. En la Figura 11, se muestra la construcción por capas de *Drupal*.

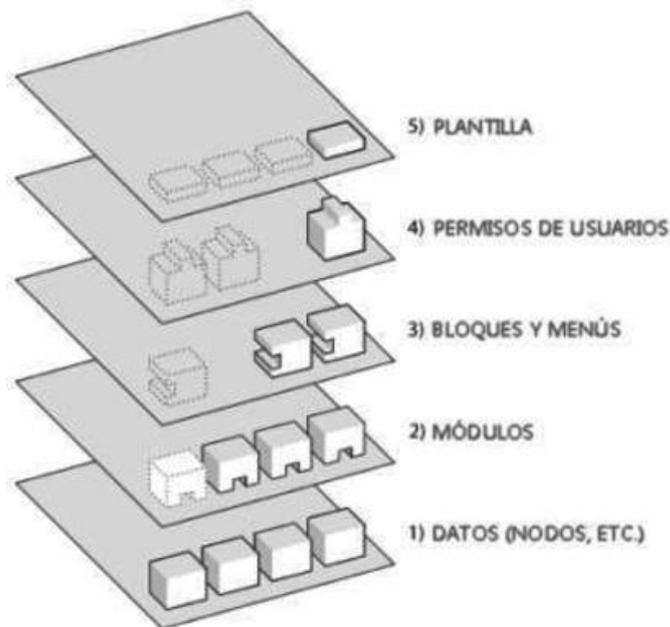


Figura 11: Estructura de Capas *Drupal*

A continuación, se da una descripción más detallada de los aspectos principales de la estructura interna de *Drupal*.

7.1. Tipos de Contenido (Nodo)

Se podría entender el nodo como la capa más básica de *Drupal*. Los nodos son la base de nuestro sistema de información. La porción de contenido mínima en *Drupal*.

Cuando instalamos *Drupal* contamos con dos tipos de contenido creados por defecto: *páginas* y *artículos*. El tipo de contenido *página* dispone básicamente de dos campos: título y cuerpo. En este caso se trata de un tipo de contenido muy sencillo orientado a crear páginas básicas. El tipo de contenido *artículo* es similar al anterior, añadiendo la posibilidad de subir una imagen y definir qué etiquetas deseamos usar para clasificarlo.

El potencial de *Drupal* reside en la posibilidad de crear tipos de contenido personalizados. Añadiendo los campos necesarios y su tipo, sin necesidad de sobrecargar el sistema con módulos especiales. Esto es un aspecto diferencial respecto de otros CMS, como ya habíamos mencionado anteriormente.

En nuestro caso, sólo necesitamos crear un tipo de contenido personalizado, llamado *Proyecto*. El sistema por defecto nos permite reordenar los campos, decidir si queremos que aparezca

una etiqueta al lado del campo y dónde queremos que aparezca. Las posibilidades son muy amplias.

En la Figura 12 podemos observar la definición de todos los campos que forman parte de nuestro tipo de contenido *Proyecto*, así como de la tipología de cada campo.

ETIQUETA	NOMBRE DE SISTEMA	TIPO DE CAMPO	CONTROL	OPERACIONES	
+ Opciones de ruta URL	path	Path module form elements			
+ Nombre del Proyecto	title	Elemento del módulo Node			
+ Descripción del Proyecto	body	Texto largo y resumen	Área de texto con un resumen	editar	eliminar
+ Email de Contacto	field_email	Texto	Campo de texto	editar	eliminar
+ Persona de Contacto	field_nombre	Texto	Campo de texto	editar	eliminar
+ Departamento	field_departamento	Texto	Campo de texto	editar	eliminar
+ Estado	field_estado	Texto	Campo de texto	editar	eliminar
+ Empresa	field_empresa	Texto	Campo de texto	editar	eliminar
+ Flujo de trabajo	workflow	Workflow module form			
+ Asignado a	field_asignado	Texto	Campo de texto	editar	eliminar
+ Enlace memoria	field_memoria	Texto largo	Área de texto (varias filas)	editar	eliminar
+ Cod	field_cod	User reference	Autocomplete text field	editar	eliminar
+ Nombre Codirector	field_nombre_codirector	Texto	Campo de texto	editar	eliminar
+ Agregar nuevo campo					
<input type="text"/>		- Seleccione un tipo de campo -		- Seleccione un control -	
Etiqueta		Tipo de datos a almacenar.		Elemento de formulario para editar los datos.	

Figura 12: Configuración del tipo Proyecto

7.2. Módulos

Los módulos son uno de los elementos más potentes que nos ofrece *Drupal*. Son elementos que nos permiten añadir nuevas funcionalidades a *Drupal* para adaptarlo a las necesidades de nuestro sistema. Básicamente, su uso nos abre un abanico de posibilidades casi infinitas para realizar cualquier tarea.

Los módulos se clasifican en dos tipos: los *Core Modules* y los *Contributed Modules*. Los primeros son los que vienen por defecto con *Drupal*, si bien son minoría, son los más necesarios. Los *Contributed Modules* son los desarrollados por la comunidad que da soporte a *Drupal* y que no vienen por defecto con *Drupal*. Para conseguir este tipo de módulos, deberemos visitar la web oficial de *Drupal*, descargarlos e instalarlos. Para reafirmar más la importancia de estos últimos cabe destacar que muchos de los módulos que en versiones anteriores fueron *Contributed Modules* forman parte de los *Core Modules* en las versiones actuales.

A continuación, detallaremos los módulos que han sido necesarios para desarrollar nuestro sistema. Todos ellos pertenecientes a la categoría *Contributed Modules*.

7.2.1. Views

Con este módulo tenemos un constructor de consultas que permite mostrar vistas de nodos almacenados en *Drupal*. A través de la interfaz de usuario de *Views*, podemos seleccionar el origen de los datos, los campos a mostrar, las condiciones por las que filtrar, ordenación y muchos otros aspectos. En resumen, es la alternativa a escribir consultas manualmente en SQL e incluirlas en código PHP [10]. *Views* puede mostrar información tanto a nivel de página como de bloque.

Views es esencial y vital en nuestro sistema debido a que la gran mayoría de páginas muestran como contenido un listado de nodos, del tipo proyecto en nuestro caso. La visualización de éstos viene condicionada por diferentes criterios en función de la página. Importante también es la posibilidad de limitar el acceso a las *views* en función del rol al que pertenezca el usuario. De esta manera, nadie que no posea el rol requerido podrá acceder a la página y visualizar información no autorizada.

En el desarrollo de nuestro sistema han sido necesarios definir los siguientes *Views*, todos ellos en formato de página:

- Proyectos Ofertados: Muestra un listado de los proyectos ofertados. Se incluye el nombre del proyecto, el tutor responsable y el departamento al que pertenece. Si hacemos clic sobre el nombre accedemos a la página específica del proyecto en cuestión.
- Proyectos Asignados: Muestra un listado de los proyectos asignados. Se incluye el nombre del proyecto, el tutor responsable y el alumno asignado. Si hacemos clic sobre el nombre accedemos a la página específica del proyecto en cuestión.
- Proyectos Finalizados: Muestra un listado de los proyectos finalizados. Se incluye el nombre del proyecto, el tutor responsable y el alumno asignado. Si hacemos clic sobre el nombre accedemos a la página específica del proyecto en cuestión.
- Proyectos Eliminados: Muestra un listado de los proyectos eliminados. Se incluye el nombre del proyecto y el tutor responsable. Si hacemos clic sobre el nombre accedemos a la página específica del proyecto en cuestión.
- Mis Proyectos: Muestra un listado de los proyectos que cada profesor o empresa ha creado. En esta tabla se incluye el nombre del proyecto y el estado en el que se encuentra. Si hacemos clic sobre el nombre accedemos a la página específica del proyecto en cuestión.
- Mis Proyectos Codirigidos: Muestra un listado de los proyectos en los que el profesor aparece como codirector. Se incluye el nombre del proyecto, tutor responsable, empresa y el estado en el que se encuentra. Si hacemos clic sobre el nombre accedemos a la página específica del proyecto en cuestión.
- Proyectos Pendientes: Muestra un listado de los proyectos que han sido propuestos por profesores o empresas y se encuentran aún pendientes de aprobación por parte del presidente de la CAG. Se incluye el nombre del proyecto, el profesor/empresa autor, la empresa y si existen nuevos comentarios. Si hacemos clic sobre el nombre accedemos a la página específica del proyecto en cuestión.
- Codirigir Proyecto: Muestra un listado de los proyectos creados por profesores o empresas que se encuentran sin codirector. Se incluye el nombre del proyecto, tutor responsable y su correo de contacto.

- Asignar Miembro CAG: Muestra un listado de los profesores registrados en el sistema que pueden optar a ser miembros de la CAG. Se incluye nombre del profesor, nombre de usuario y un botón de editar.

7.2.2. Rules

Rules nos permite definir la ejecución de acciones en función de si ocurren eventos o condiciones determinadas. Nos permite, por ejemplo, autorizar la publicación de contenido, enviar emails, mostrar mensajes por pantalla y un largo etcétera.

Es uno de los módulos más importantes de nuestro sistema debido a la dependencia que tienen muchos otros módulos respecto de él.

A continuación, se describen los elementos *Rules* que han sido necesarios para satisfacer las distintas funcionalidades requeridas por el sistema.

Nombre	Aprobar Proyecto
Condición	Cuando el nodo <i>Proyecto</i> hace una transición de estado 'Propuesto' a estado 'Ofertado'.
Acciones	Publica contenido en el sistema. El contenido pasa a ser visible para cualquier usuario. Envía un email al profesor/empresa que lo propuso y a los miembros de la CAG notificando la aprobación.

Tabla 23: *Rule* Aprobar Proyecto

Nombre	Cargar Datos Tutor
Condición	Cuando el usuario tiene el rol profesor y crea un proyecto.
Acciones	Establece el nombre, correo, departamento y empresa que posee el profesor en el perfil.
Anotaciones	Esta acción hace más cómoda la creación de un proyecto por parte de un profesor ya que debe rellenar menos campos.

Tabla 24: *Rule* Cargar Datos Tutor

Nombre	Cargar Datos Empresa
Condición	Cuando el usuario tiene el rol empresa y crea un proyecto.
Acciones	Establece el nombre de la empresa que posee la empresa en el perfil.
Anotaciones	Esta acción hace más cómoda la creación de un proyecto por parte de la empresa ya que debe rellenar menos campos.

Tabla 25: *Rule* Cargar Datos Empresa

Nombre	Notificación Nuevo Proyecto UC
Condición	Cuando un profesor rellena el formulario de creación de un nuevo proyecto.
Acciones	El sistema envía un correo electrónico a los miembros de la CAG y al profesor indicándoles que se ha realizado correctamente la propuesta de proyecto.

Tabla 26: *Rule* Notificación Proyecto UC

Nombre	Notificación Nuevo Proyecto Empresa
Condición	Cuando una empresa rellena el formulario de creación de un nuevo proyecto.
Acciones	El sistema envía un correo electrónico a los miembros de la CAG y al responsable de la empresa indicándoles que se ha realizado correctamente la propuesta de proyecto.

Tabla 27: *Rule* Notificación Proyecto Empresa

Nombre	Notificación Nuevo Comentario
Condición	Cuando un profesor/empresa o miembro de la CAG realiza una anotación en un proyecto.
Acciones	El sistema envía un correo electrónico a los miembros de la CAG y al profesor/empresa propietaria indicándoles que se ha realizado una nueva anotación en el proyecto.

Tabla 28: *Rule* Notificación Nuevo Comentario

Nombre	Cargar Campo Estado
Condición	Cuando existe un cambio de estado en el proyecto.
Acciones	Establece en el campo “estado” el nombre del nuevo estado del proyecto.
Anotaciones	Es necesario debido a un bug del módulo Workflow.

Tabla 29: *Rule* Cargar Campo Estado

Nombre	Redirección tras Login Profesor
Condición	Cuando el profesor entra correctamente con sus credenciales en el sistema.
Acciones	El sistema redirige al usuario a la página “Mis proyectos”.

Tabla 30: *Rule* Redirección tras Login Profesor

Nombre	Redirección tras Login CAG
Condición	Cuando el miembro CAG entra correctamente con sus credenciales en el sistema.
Acciones	El sistema redirige al usuario a la página “Proyectos Pendientes”.

Tabla 31: *Rule* Redirección tras Login CAG

Nombre	Notificación Caducidad de Proyecto
Condición	Cuando la fecha de creación del proyecto está próxima a cumplir dos años.
Acciones	El sistema notifica mediante correo electrónico al profesor/empresa que su proyecto está próximo a caducar.

Tabla 32: *Rule* Notificación Caducidad de Proyecto

Nombre	Deshabilitar Comentarios
Condición	Cuando el proyecto pasa a estado “Ofertado”.
Acciones	Se deshabilitan y ocultan los comentarios del proyecto.

Tabla 33: *Rule* Deshabilitar Comentarios

Nombre	Proyecto Cancelado Presidente CAG
Condición	El presidente CAG establece el proyecto en estado “Cancelado”.
Acciones	Se establece el contenido como no publicado. Se notifica por correo electrónico al profesor/empresa autor del proyecto.

Tabla 34: *Rule* Proyecto Cancelado Presidente CAG

Nombre	Proyecto Cancelado Autor
Condición	El profesor o la empresa establecen el proyecto en estado “Cancelado”.
Acciones	Se establece el contenido como no publicado. Se notifica por correo electrónico al presidente de la CAG.

Tabla 35: *Rule* Proyecto Cancelado Autor

Nombre	Carga Datos Codirector y Notificación
Condición	Se establece un valor en el campo “Codirector”.
Acciones	Establece en la información del codirector, el nombre y el correo que tiene el profesor en su perfil. Se notifica por correo electrónico al nuevo codirector.

Tabla 36: *Rule* Carga Datos Codirector y Notificación

7.2.3. Workflow

Es un módulo que permite asignar un flujo de estados a un tipo de contenido. Es esencial para controlar la publicación de contenido por parte de los usuarios. Su funcionamiento se basa en estados. Un nodo va pasando por diferentes estados, dentro de los cuales se pueden realizar unas acciones específicas. La transición entre estados va asociada a los roles, no todos los usuarios pueden cambiar el estado del nodo.

En nuestro sistema se establecen un total de cinco estados diferentes, que como es obvio coinciden con los estados especificados en el análisis de requisitos. Los estados son: Propuesto, Ofertado, Asignado, Finalizado y Cancelado. Este último estado no es relevante salvo para posteriores estadísticas a nivel de administración.

7.2.3.1. Transiciones entre estados

Un aspecto que debe estar bien definido es qué transiciones son posibles y quién puede realizarlas. En la Figura 13 podemos observar un informe completo de todas estas transiciones posibles y de quién puede realizarlas.

FROM / TO →	PROPUESTO	OFERTADO	ASIGNADO	FINALIZADO	ELIMINADO
(creation)	<input checked="" type="checkbox"/> (author) <input type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa
Propuesto		<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input checked="" type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa
Ofertado	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa		<input checked="" type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input checked="" type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa
Asignado	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input checked="" type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input checked="" type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa		<input checked="" type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input checked="" type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input checked="" type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa
Finalizado	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa		<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa
Eliminado	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	<input type="checkbox"/> (author) <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input type="checkbox"/> Empresa	

Figura 13: Transiciones de Estados

7.2.4. Workflow fields

Es una extensión complementaria al módulo *Workflow*. Gracias a ella, podemos determinar si un campo es editable o visible en función del estado en el que se encuentra el nodo y del rol al que pertenece el usuario que lo está visualizando en ese momento.

Por ejemplo, la configuración de los campos *Nombre del Proyecto* y *Descripción del Proyecto* en el estado propuesto se muestra en la Figura 14.

TIPO DE CONTENIDO	NOMBRE DEL CAMPO	VISIBLE	EDITABLE	SELECT
Proyecto (proyecto)	Nombre del Proyecto (title)	<input checked="" type="checkbox"/> author <input checked="" type="checkbox"/> usuario anónimo <input checked="" type="checkbox"/> usuario autenticado <input checked="" type="checkbox"/> administrador <input checked="" type="checkbox"/> Profesor <input checked="" type="checkbox"/> Miembro CAG <input checked="" type="checkbox"/> Presidente CAG <input checked="" type="checkbox"/> Empresa	<input checked="" type="checkbox"/> author <input type="checkbox"/> usuario anónimo <input type="checkbox"/> usuario autenticado <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input checked="" type="checkbox"/> Empresa	all, ninguno, toggle, visible, editable
Proyecto (proyecto)	Descripción del Proyecto (body)	<input checked="" type="checkbox"/> author <input checked="" type="checkbox"/> usuario anónimo <input checked="" type="checkbox"/> usuario autenticado <input checked="" type="checkbox"/> administrador <input checked="" type="checkbox"/> Profesor <input checked="" type="checkbox"/> Miembro CAG <input checked="" type="checkbox"/> Presidente CAG <input checked="" type="checkbox"/> Empresa	<input checked="" type="checkbox"/> author <input type="checkbox"/> usuario anónimo <input type="checkbox"/> usuario autenticado <input checked="" type="checkbox"/> administrador <input type="checkbox"/> Profesor <input type="checkbox"/> Miembro CAG <input type="checkbox"/> Presidente CAG <input checked="" type="checkbox"/> Empresa	all, ninguno, toggle, visible, editable

Figura 14: Configuración para el estado Propuesto

7.2.5. LDAP

Este método de autenticación es el más adecuado en entornos de red medios y grandes. Los usuarios se tendrán que autenticar cuando accedan a sitios web introduciendo un nombre de usuario y contraseña válidos. Las credenciales son verificadas contra un servidor externo, usando el Protocolo Ligero de Acceso a Directorios (LDAP).

La autenticación LDAP es útil si ya existe un servicio de directorio en su red y no quiere mantener cuentas de usuario y contraseñas adicionales para el acceso web. [11]

Por ello, y debido a que la Universidad nos permite usar su directorio para validar a nuestros usuarios, utilizaremos este módulo. De este modo evitamos la incomodidad de usar una cuenta y unas credenciales independientes a las universitarias. La primera vez que un profesor intente acceder con sus credenciales universitarias de manera correcta el sistema se creará una cuenta local en la aplicación, pero no almacenará su contraseña. La validación siempre se hará contra el LDAP propiedad de la Universidad.

7.2.6. SMTP Authentication

Este módulo permite a *Drupal* evitar la función `PHPmail()` y enviar correo electrónico directamente a un servidor SMTP. El módulo es compatible con la autenticación SMTP e incluso puede conectarse a servidores que utilizan SSL si son compatibles con PHP.

Configuramos este módulo con una cuenta de correo creada expresamente para el proyecto (*tfg.ciencias@gmail.com*). A través de esta cuenta, debidamente configurada a nivel de permisos en Gmail y a nivel de parámetros en nuestro sistema, nos será posible enviar correos en nombre del sistema cada vez que se requiera una realizar una notificación.

7.2.7. Twitter

El módulo Twitter nos permite vincular una cuenta de *Twitter* en nuestro sistema, otorgando al sistema permisos de lectura y escritura sobre la cuenta vinculada.

Debido a los tiempos que corren y tratando de adaptarnos a un mundo donde las redes sociales cada vez poseen más peso, se decide crear una cuenta de *Twitter* (*@TFGCiencias*). La cuenta actuará de espejo y altavoz de la aplicación web.

El mecanismo de funcionamiento es sencillo, se publicará un nuevo mensaje en *Twitter*, como el que se muestra en la Figura 15, cada vez que uno de los proyectos de la Facultad pase a estar ofertado a los alumnos.

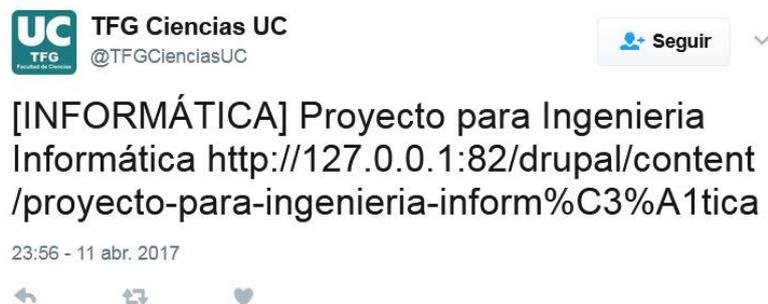


Figura 15: Captura de un twitt publicado

Para que el sistema funcione correctamente, debemos establecer un vínculo entre nuestro sistema y la cuenta de *Twitter*. Para ello, es necesario registrar una aplicación en *Twitter*, ésta no será una aplicación como tal y tan solo lo haremos para conseguir una serie de parámetros necesarios en la configuración del módulo, estos parámetros son: *Consumer key* y *Consumer*

secret. Obteniendo estos parámetros, autorizamos a nuestro sistema para que pueda escribir en *Twitter* a través del sistema, y de esta manera publicar el contenido automáticamente.

7.2.8. User Reference

Este módulo añade una nueva tipología a los campos personalizados de nuestro nodo *Proyecto*. A través de este campo se nos permite referenciar a otros usuarios. Además, posee la función de autocompletado, lo cual facilita mucho la búsqueda de un usuario en concreto.

7.2.9. Node Access User Reference

Permite otorgar permisos específicos al usuario que sea referenciado dentro de un campo de tipo *User Reference*. El módulo nos permite escoger entre permisos de lectura, actualización y borrado para los usuarios referenciados.

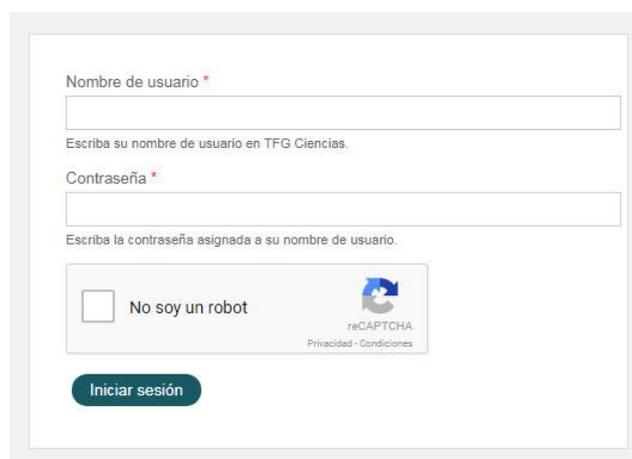
7.2.10. View Unpublished

Cuando se genera contenido en el sistema, éste nunca se publica automáticamente. Esto hace que el contenido solo sea visible para el autor del mismo. Gracias a este módulo podemos establecer que otros usuarios puedan ver contenido no publicado.

Este módulo es importante en la gestión de los proyectos en estado propuesto, ya que en ese estado los proyectos aún no están publicados y necesitamos que otros roles además del autor puedan visualizarlo para evaluarlo.

7.2.11. Captcha

Un captcha es una prueba de desafío-respuesta colocada dentro de formularios web para determinar si el usuario es humano. El propósito de captcha es bloquear envíos de formularios mediante robots, que generan spam en los sistemas [12]. El módulo captcha puede configurarse en cualquiera de los formularios de nuestro sistema. En nuestro caso se ha configurado para el formulario de inicio de sesión como se puede ver en la Figura 16.



The image shows a login form with the following elements:

- A text input field labeled "Nombre de usuario *" with a red asterisk, containing the placeholder text "Escriba su nombre de usuario en TFG Ciencias."
- A text input field labeled "Contraseña *" with a red asterisk, containing the placeholder text "Escriba la contraseña asignada a su nombre de usuario."
- A reCAPTCHA challenge box containing a checkbox labeled "No soy un robot" and the reCAPTCHA logo with the text "reCAPTCHA Privacidad - Condiciones".
- A dark green button labeled "Iniciar sesión".

Figura 16: Inicio de Sesión con Captcha

7.3. Bloques

Los bloques son contenedores de información que pueden situarse en diferentes partes de la web. Estos bloques son independientes entre sí. Podemos poseer tantos bloques como deseemos y estos podrán estar activos o no. Uno de los principales atractivos de los bloques es su versatilidad ya que pueden almacenar cualquier contenido que le especifiquemos bajo etiquetas HTML, siendo posible añadir simplemente texto o imágenes como en el caso de nuestro sistema. Además de los bloques que podemos generar nosotros como administradores, también el sistema y los módulos que vayamos añadiendo nos proveerán de bloques predefinidos.

Como hemos especificado con anterioridad estos bloques se pueden situar en diferentes partes de nuestra interfaz. Estos lugares reciben internamente el nombre de regiones, y la distribución de las mismas va establecida en función del tema que se encuentre activo en el sistema. Las regiones por defecto son: *Left sidebar*, *Right sidebar*, *Content*, *Header* y *Footer*. En el caso de nuestro sistema, bajo el theme Esteem, las regiones son más variadas, como vemos en la Figura 17.

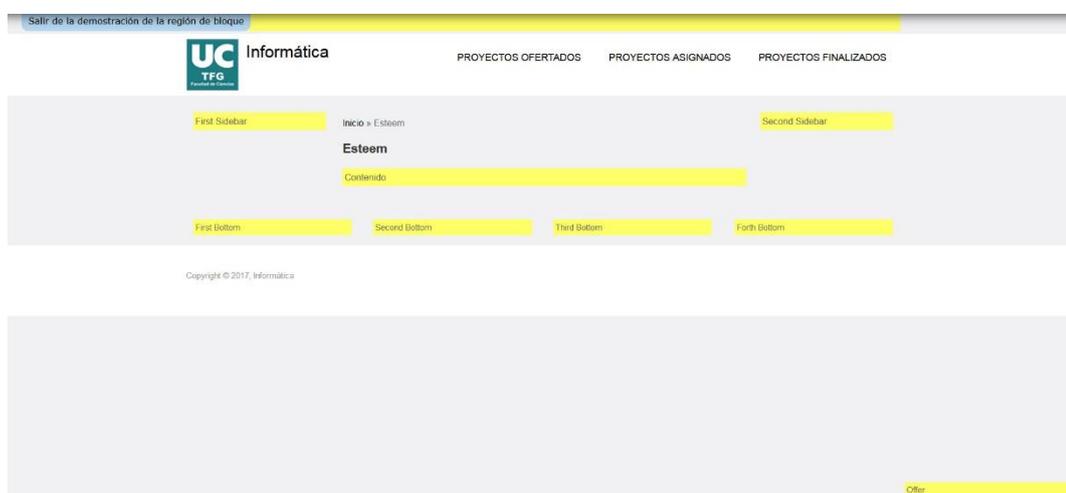


Figura 17: Bloques definidos para nuestro theme (Esteem)

Centrándonos en nuestro sistema pasaremos a describir en la Tabla 37 los bloques que han sido creados.

Nombre	Descripción	Situación
Menú CAG	Menú con opciones específicas para los miembros de la CAG.	Second Sidebar
Menú Profesor	Menú con opciones específicas para el rol profesor.	Second Sidebar
Menú Empresa	Menú con opciones específicas para el rol empresa.	Second Sidebar
Bloque Login	Posee una imagen con el icono de usuario. Al hacer clic nos envía a la página de inicio de sesión.	First Bottom
Bloque Información TFG	Posee una imagen con el icono de documentos. Al hacer clic nos envía a la página de la Facultad de Ciencias.	Second Bottom
Bloque Información Fechas	Posee una imagen con el icono de calendario. Al hacer clic nos envía a un documento oficial con las fechas más relevantes del curso 17/18.	Third Bottom
Bloque Twitter	Posee una imagen con el icono de usuario. Al hacer clic nos envía a la página Twitter de @TFGCiencias	Fourth Bottom

Tabla 37: Bloques definidos manualmente en el sistema

7.4. Roles y Permisos

Estamos ante uno de los aspectos más críticos de un gestor de contenidos (CMS). Mediante los roles y los permisos somos capaces de gestionar las operaciones y el contenido que un usuario podrá ejecutar o acceder. El objetivo es evitar que un usuario pueda ejecutar acciones o acceder a contenido que no le son permitidos y que podrían comprometer la seguridad del sistema.

Los usuarios se agrupan por roles, éstos ya fueron descritos con anterioridad en este documento (ver apartado 2.2.1). Cada rol posee unos permisos específicos que podemos determinar y limitar. Estos permisos son los que establecen si un rol puede ejecutar o no determinadas operaciones o puede visualizar determinado contenido dentro del sistema.

La gestión de permisos es un aspecto controlado por el administrador. Es su tarea velar por la coherencia de los mismos, ya que cualquier incoherencia en la configuración de permisos puede hacer que nuestro sistema sea altamente vulnerable.

En este apartado, es imposible exponer y explicar cada uno de los permisos que pueden establecerse en el sistema ya que en el momento de cierre del sistema eran más de cincuenta.

7.5. Apariencia

La gestión de la apariencia reside en lo que *Drupal* conoce como *themes* y son los principales responsables de la estética o estilo con que se mostrará la información al usuario.

Drupal viene por defecto con dos temas básicos. El primero de ellos es *Garland*. Es, probablemente, el tema más famoso de *Drupal*. Ha sido utilizado como tema por defecto desde su versión 5, lanzada ya en un lejano año 2006. Como se puede ver en la Figura 18, es un tema demasiado genérico.



Figura 18: *Theme Garland*

El otro tema instalado por defecto es *Stark*, un tema exclusivo para la administración interna de nuestro sistema. Esto es un aspecto novedoso que incorporó la versión 7 de *Drupal*, ya que hace una separación entre la estética de administración y la pública. De esta manera, los administradores poseen un tema perfectamente adaptado a las labores de administración del sistema que ayuda a la gestión del mismo con independencia del tema que se les esté mostrando a los usuarios en ese momento.

7.5.1. Nuestro Theme: Esteem

Debido al aspecto básico del tema público por defecto (*Garland*) decidimos buscar una alternativa. El tema escogido fue *Esteem*. Los aspectos más destacados para su elección fueron:

- Poseer una versión estable y definitiva para *Drupal 7*, con lo cual, sobre el papel nos ahorraríamos posibles problemas de incompatibilidad.
- Estar definido como “Mantenido activamente” por *Drupal*. Esto nos asegura soporte de la comunidad en caso de futuros problemas.
- Tener una licencia completamente gratuita.
- Un estilo minimalista, basado en colores solidos que nos facilitara su adaptación a la imagen y colores corporativos de la Universidad.
- Incorporar en su hoja de estilos (CSS) propiedades *Responsive Design*.

La personalización posterior del *Theme* ha sido bastante sencilla. Se ha buscado, sobre todo, no perder la simplicidad del tema, se han adaptado los colores principales del tema al verde corporativo de la Universidad, se ha adaptado un nuevo logo y se han realizado cambios menores en la hoja de estilos (CSS).

8. PRUEBAS Y VALIDACIONES

En cualquier metodología de desarrollo software, es necesario dedicar un apartado referente a las pruebas y validaciones del software.

En nuestro caso, al desarrollar el proyecto bajo un tipo de metodología iterativa incremental hace necesario que las verificaciones se hagan al final de cada una de las iteraciones programadas. Con esto, conseguimos saber con certeza si cada nueva iteración planificada del sistema es compatible con las anteriormente desarrolladas.

Con estas pruebas se permite identificar los aspectos donde nuestro sistema pueda fallar o generar conflicto. Se busca mediante las pruebas por iteraciones que estos fallos sean percibidos en las fases iniciales del proyecto para que así éstos no sean más complejos de resolver.

8.1. Pruebas unitarias

Las pruebas unitarias se han realizado a medida que se iban completando las iteraciones programadas del sistema (ver apartado 3.1). Se ha comprobado que cada una de las iteraciones satisfacía los requisitos que involucraba. En este aspecto quizás la iteración más compleja haya sido la que gestionaba los permisos en cada estado del proyecto.

8.2. Pruebas de integración

Otro aspecto importante es la integración de los diferentes módulos del proyecto, ya que puede darse el caso de que todas las partes funciones por separado y pasen todas las pruebas unitarias y después en el momento que se junten todas las partes el resultado final no sea el esperado. Es por ello imprescindible realizar estas pruebas al final de cada una de las iteraciones. De esta manera, verificamos cuanto antes que el desarrollo de nuestro sistema está siendo positivo.

8.3. Pruebas de sistema

Representan las pruebas finales de nuestro sistema. Es aquí donde debemos verificar el comportamiento de nuestro sistema en conjunto y comprobar que los requisitos no funcionales (ver apartado 3.2) son satisfactoriamente cumplidos.

8.3.1. Pruebas de rendimiento

Evaluamos el comportamiento del sistema en diferentes navegadores (Firefox, Chrome e Internet Explorer).

Principalmente, se buscaba obtener unos tiempos de carga óptimos para el sistema en todos los navegadores. Afortunadamente, nuestro sistema posee apenas imágenes, independientemente de ese dato se ha buscado una correcta compresión de las mismas para obtener unos tiempos de carga reducidos.

8.4. Pruebas de aceptación

Las pruebas de aceptación se focalizan en cumplir satisfactoriamente los requisitos funcionales definidos que afectan a los usuarios finales.

En nuestro caso los usuarios finales son los profesores de la Facultad y empresas externas. La mayor parte de estas pruebas han sido realizadas por compañeros del entorno universitario que han accedido a la aplicación y han aportado su correspondiente *feedback*.

9. CONCLUSIONES Y TRABAJOS FUTUROS

Es positivo, al finalizar la realización de un proyecto, hacer un análisis de lo que ha supuesto para nosotros. Es por ello por lo que debemos identificar los aspectos positivos, las dificultades encontradas y posibles mejoras que quizás sea interesante aplicar en un futuro.

9.1. Conclusiones

El desarrollo de este proyecto tenía como principal misión actualizar la aplicación anteriormente desarrollada. Ésta se realizó sobre una plataforma que, actualmente, carece de soporte. Por ello hubo que realizar el proceso desde cero, lo cual afectó al tiempo de desarrollo, pero indirectamente lo hizo más satisfactorio, ya que se han podido ejecutar todas las fases de un desarrollo software.

A título personal, ha sido un proyecto enriquecedor. Aunque en un principio tuve dudas respecto de su desarrollo, finalmente puedo valorarlo gratamente. Durante el mismo, he descubierto que muchos aspectos o fases del desarrollo que quizás anteriormente no había valorado en su correcta medida son de vital importancia para alcanzar el fin de un proyecto de una manera satisfactoria. Otro aspecto importante ha sido la obligación que he tenido gracias al proyecto de documentarme en temas o herramientas que no había estudiado durante mi paso por la Universidad, complementando de esta manera mi formación académica.

9.2. Trabajos futuros

9.2.1. Drupal 8 y versiones posteriores

En el momento de comenzar a desarrollar el proyecto, nos encontramos con otro dilema, ya que existía la posibilidad de poder desarrollar el proyecto bajo la última versión presentada por *Drupal*: *Drupal 8*. Finalmente nos decantamos por *Drupal 7* por su estabilidad y compatibilidad de módulos, pero es inevitable la transición a *Drupal 8* en un futuro. Si tenemos como referencia la versión 6, ésta fue oficial en febrero de 2008 y fue abandonada en febrero de 2016 [13], es decir, 8 años de soporte. Si aplicamos esos datos a *Drupal 7*, tendríamos garantizado soporte hasta 2019, ya que su fecha de publicación fue enero de 2011[14]. Es decir, el paso a *Drupal 8* no tiene por qué ser inminente, pero sí debe estar contemplado de cara al futuro.

9.2.2 “Universalizar” la herramienta

Podría ser interesante que la aplicación fuese usada no solo en la Facultad de Ciencias, sino en cualquier escuela o facultad de la Universidad de Cantabria. Investigando un poco, actualmente existen facultades o escuelas que no poseen sistema alguno o, que si lo poseen, no es común a todos, sino que va en función de sus respectivos departamentos.

BIBLIOGRAFÍA

- [1] Mario Piattini. Análisis y diseño detallado de aplicaciones informáticas de gestión. 1ª ed. RA-MA Editorial, 1996.
- [2] Roger Pressman. Ingeniería del Software un enfoque práctico. McGraw Hill 7º Edición, 2002.
- [3] Sommerville. Ingeniería del Software. Addison-Wesley. 7º Edición, 2005.
- [4] Sandra Fernández Moreno. ¿Qué es un CMS - WordPress, Joomla, Drupal...? <http://www.valortop.com/blog/cms-wordpress-joomla-drupal>
- [5] Web Oficial PHP. ¿Qué es PHP? <https://secure.php.net/manual/es/intro-what-is.php>
- [6] Web Oficial Drupal. HTML5 Tools. https://www.drupal.org/project/html5_tools
- [7] LibrosWeb. ¿Qué es CSS? https://librosweb.es/libro/css/capitulo_1.html
- [8] Web Oficial Drupal. Managing JavaScript in Drupal 7 <https://www.drupal.org/docs/7/api/javascript-api/managing-javascript-in-drupal-7>
- [9] Violet UML Editor. <https://sourceforge.net/projects/violet/>
- [10] Pedro Cambra. Módulo Views: Qué es, cómo se instala y cómo crear nuestra primera vista. <http://cambrico.net/drupal/modulo-views-que-es-como-se-instala-y-como-crear-nuestra-primer-vista>
- [11] IPCop Firewall. Autenticación LDAP. <http://www.ipcop.org/2.0.0/es/admin/html/proxy-auth-ldap.html>
- [12] Christian Córdoba. Evita el spam en tu web con reCAPTCHA de *Google* <https://www.acceso.com/2017/04/evita-el-spam-en-tu-web-con-invisible-recaptcha-de-google.html>
- [13] Web Oficial Drupal. Drupal 6 end-of-life announcement <https://www.drupal.org/drupal-6-eol>
- [14] Web Oficial Drupal. Drupal 7 Release. <https://www.drupal.org/node/1015392>