



Universidad de Valladolid
Universidad de Valladolid

E.T.S Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática de Sistemas

Rediseño y actualización de un servidor web basado en el CMS Joomla!

Autor:

D. Sergio Cárcel Fernando

Tutor:

Dr. Jesús M. Vegas Hernández

Cotutor:

D. Ramiro González Catón

Resumen

Este TFG se desarrolla para el Grupo de Óptica Atmosférica, que es parte del departamento de Física Teórica, Atómica y Óptica de la Universidad de Valladolid. Consiste en la actualización del servidor web que ya tienen implementado gracias al CMS (Gestor de Contenidos) Joomla!.

Este servidor web es de vital importancia para ellos, ya que en la base de datos de su servidor guardan todos sus artículos de investigación, publicados y algunos que todavía no se han publicado, de esta manera pueden acceder a esos datos en cualquier momento y no depender de fuentes externas con el consiguiente peligro de la pérdida de esos datos.

A partir de esa versión que ya tienen implementada en Joomla!, hay que crear otro servidor web desde cero, pero con una versión de Joomla! más actual. El diseño y estructura de la página tienen que ser lo más parecido posible a esa versión antigua, introduciendo los cambios que el cliente crea oportunos. En este caso, el cliente y cotutor es Ramiro González Catón, miembro de dicho Grupo de Óptica Atmosférica.

La versión de Joomla! con la que estaba implementado el servidor web era la 1.7, versión muy obsoleta ya que hoy en día todos los servidores implementados con Joomla! utilizan versiones 3.x. En nuestro caso se ha elegido la versión 3.2.4, una versión bastante estable y que incluye numerosas mejoras respecto a versiones antiguas.

Además de actualizar el servidor web en sí, hay que tener en cuenta que contaba con tres componentes externos creados expresamente para esa versión de Joomla! obsoleta. Por lo tanto, estos tres componentes hay que actualizarlos completamente, para adaptarlos a la nueva versión. El cambio de la estructura y la forma de acceder a esos archivos a cambiado completamente ya que han modificado su plataforma interna de una forma bastante profunda. El primer gran cambio surgió con la versión 2.5 y volvieron a remodelarla con las versiones 3.x.

Aunque estos tres componentes son independientes a la hora de instalarlos y usarlos, tienen una funcionalidad conjunta que por separado no tiene sentido tenerlos, ya que uno se encarga de mostrar todos los artículos de investigación (artículos, tesis doctorales, congresos y otras publicaciones), otro componente se encarga de introducir en diferentes formatos estos artículos en la base de datos y el último sirve para relacionar nombres de autores con los nombres exactos por los que figuran en la base de datos.

Índice general

1. Introducción	3
1.1. Objetivos y Motivación del TFG	3
1.1.1. Objetivos	3
1.1.2. Motivaciones	4
1.1.3. Propuesta	4
1.1.4. Suposiciones y Restricciones	4
2. Estado del Arte	7
2.1. Joomla! vs CMS	7
2.2. Joomla!	9
2.3. Arquitectura Joomla!	10
2.3.1. Diagrama de Capas	10
2.3.2. Patrón MVC	11
2.3.3. Modelo Entidad-Relación	13
2.3.4. Casos de Uso y Diagramas de Clases	13
2.3.5. Primer Acercamiento	14
2.3.6. Privacidad y Usuarios (Niveles de Acceso y Grupos)	18
2.3.7. Módulos, Menús y Páginas Joomla!	21
3. Plan de Desarrollo del Proyecto	23
3.1. Plan de Desarrollo Software	23
3.1.1. Visión General	24
3.1.2. Plan de Iteración de la Etapa de Inicio	26
3.1.3. Plan de Iteración de la Etapa de Elaboración	27
3.1.4. Plan de Iteración de la Etapa de Construcción	28
3.1.5. Plan de Iteración de la Etapa de Transición	29
3.2. Seguimiento del Proyecto	29
3.2.1. Evolución del Proyecto	29
3.3. Organización del Proyecto	31
3.3.1. Interfaces Externas	31
3.3.2. Estructura Interna	31
3.3.3. Roles y Responsabilidades	31
3.4. Plan de Control y Riesgos	33
3.4.1. Plan de Control	33
3.4.2. Plan de Riesgos	33
3.5. Herramientas Utilizadas	37

4. Análisis de la Aplicación	39
4.1. Requisitos	39
4.1.1. Requisitos de Información	39
4.1.2. Requisitos Funcionales	41
4.1.3. Requisitos no Funcionales	43
4.2. Casos de Uso	45
4.2.1. Diagrama de Casos de Uso	45
4.2.2. Actores	46
4.2.3. Descripción de los Casos de Uso	46
4.2.4. Escenarios	59
4.3. Modelo de Dominio	62
4.3.1. Descripción de las Clases de Análisis	62
4.4. Modelo Entidad Relación	66
5. Diseño de la Aplicación	67
5.1. Diagrama de Despliegue	68
5.2. Diagrama de Clases de Diseño	68
5.2.1. Componente Com_bibtex	69
5.2.2. Componente Com_editarAutores	70
5.2.3. Componente Com_verArticulos	71
5.3. Diagramas de Secuencia	72
5.3.1. Introducir Artículo Manualmente	73
5.3.2. Introducir Artículo por DOI	74
5.3.3. Rellenar Autores	75
5.3.4. Ver Artículos	76
5.3.5. Ver Bibtex	77
5.3.6. Ver Txt	78
5.3.7. Ver Cicyt	79
5.3.8. Editar Bibtex	80
6. Implementación del Proyecto	81
6.1. Implementación	81
6.1.1. Plantilla	81
6.1.2. Componentes	82
6.1.3. Componentes Propios	82
7. Batería de Pruebas	89
7.1. Introducción	89
7.2. Caja Blanca	90
7.2.1. Base de Datos	90
7.2.2. Parámetros	90
7.2.3. Interfaz	90
7.3. Caja Negra	90
8. Conclusiones y Trabajo Futuro	93
8.1. Conclusiones	93
8.2. Trabajo Futuro	94

Bibliografía	95
A. Manual de Instalación	97
I..1. Requisitos del Sistema	97
I..2. Instalación de la Aplicación	97
I.a. Manual de Usuario	102
I.a.1. Requisitos del Sistema	102
I.a.2. Acceso al Sistema	102
I.a.3. Descripción de la Interfaz de Usuario	102
I.a.4. Añadir artículos en diferentes idiomas	109
B. Contenido del CD	115

Capítulo 1

Introducción

Contenido

1.1. Objetivos y Motivación del TFG	3
1.1.1. Objetivos	3
1.1.2. Motivaciones	4
1.1.3. Propuesta	4
1.1.4. Suposiciones y Restricciones	4

Resumen: *Este capítulo trata de ubicar el marco de trabajo en el que se desarrolla este proyecto y describir los objetivos principales.*

1.1. Objetivos y Motivación del TFG

1.1.1. Objetivos

El objetivo principal de este TFG es la actualización del servidor web del Grupo de Óptica Atmosférica (GOA) de la Universidad de Valladolid. Este servidor está implementado con Joomla!, un gestor de contenidos (CMS). La versión con la que contaba este servidor era la 1.7, una versión muy obsoleta.

Como versión de Joomla! se ha elegido la 3.2.4 que es una de las mas modernas actualmente; la gran ventaja es que desde la versión 3.x se puede ir actualizando desde el *backend* del CMS con un simple click y cuenta con numerosos *plugins*, componentes y *templates* gratuitos ya compatibles con esta versión, teniendo menos fallos de seguridad que otras versiones ya obsoletas.

A parte de actualizar este servidor web hay que actualizar los tres componentes creados por el administrador del sistema y cliente por el que ha sido encargado este TFG, Ramiro González Catón, debido a que no son compatibles con esta versión y hay que reescribir el código y cambiar la estructura de los mismos.

También hay que añadirle a uno de ellos la funcionalidad extra por la que se puedan guardar en la base de datos del sistema los artículos de investigación introduciendo el DOI, identificador

único de estos artículos, que se conecte al WOS (Web Of Science) mediante una API creada en PHP y extraiga la información de ese artículo y la guarde en nuestra propia base de datos.

1.1.2. Motivaciones

La principal motivación que me llevó a la elección de este TFG es, además de profundizar en el aprendizaje de Joomla!, crear estos tres componentes compatibles con esta versión y liberarlos a la comunidad Joomla! para que otros grupos de investigación puedan instalarlos en sus servidores web y puedan gracias a ellos poder administrar sus propios artículos, ya que no existe nada parecido actualmente y considero que hay mucha gente interesada en esto mismo. Tener un acceso siempre que quieran a información de esta índole es beneficioso para la comunidad investigadora.

1.1.3. Propuesta

Creación de un servidor web en Joomla! 3.2.4 con una plantilla responsiva (que se adapta al dispositivo desde el que se accede) para el acceso desde tablets y smartphones.

El sistema permitirá ver las diferentes secciones del menú, permitiendo al usuario iniciar sesión en el sistema si está dado de alta en el mismo o registrarse en el, siempre esperando una confirmación por parte del administrador, es decir, hasta que el administrador no le dé de alta en el sistema no podrá iniciar sesión. Cualquier usuario podrá buscar los artículos de investigación que desee, ya sean artículos, tesis doctorales, congresos u otras publicaciones, pudiéndolos filtrar por autor y/o año y primer autor.

Un usuario que haya iniciado sesión en el sistema podrá, entre otras cosas, publicar artículos de investigación, rellenando los campos del formulario correspondientes del artículo. La segunda manera de poder guardar en el sistema una publicación es introduciendo el artículo con el formato Bibtext adecuado. La tercera manera para hacerlo, es seleccionar un archivo con extensión .bib para poder subir varias publicaciones a la vez, siempre que estén en ese formato, y guardarlo en el sistema. Y la cuarta manera y última consiste en introducir el DOI de la publicación, de esta manera el sistema realizará una búsqueda en la base de datos del WOK, conseguirá el artículo y lo introducirá en la base de datos. Si ha iniciado sesión en el sistema también podrá rellenar los autores, es decir, relacionar los autores introducidos en las publicaciones con el nombre exacto del autor almacenado en la base de datos. De esta manera no tendremos diferentes nombres para un mismo autor.

1.1.4. Suposiciones y Restricciones

En el siguiente apartado se detallaran las suposiciones y restricciones que tiene el proyecto en sí. Con respecto a las suposiciones, el usuario al que está orientado este portal web, es un usuario con conocimientos básicos sobre informática y navegación a través de la red, por lo que tiene que estar implementada una interfaz sencilla, sin muchas variaciones respecto a la versión anterior, a la vez que completa y funcional, para llegar a todo tipo de usuario, siguiendo los patrones HCI1. La conexión de red suponemos que es estable y no hay ningún tipo de fallo producido por un desastre natural.

Abordando el tema de las restricciones, centrado en la gestión de un proyecto, se pueden apreciar diferentes restricciones de tiempo, restricciones de coste y restricciones de alcance. Con restricciones de tiempo están las fechas marcadas para la presentación del proyecto así como la exigencia del cliente de que el servidor web este actualizado lo más tempranamente posible.

Teniendo en cuenta las restricciones de coste, hay que mencionar que la carga de trabajo que hay que dedicarle a un proyecto de esta envergadura es grande, además de tener que compaginarlo con las diversas asignaturas, que nos ocuparán una carga de trabajo bastante elevada también.

En lo referente al coste económico, no hay ningún tipo de restricción, ya que con el equipo informático del que se dispone en la escuela, mi equipo informático propio y el hosting web que el cliente me montó en un servidor propiedad del Grupo de Óptica Atmosférica no tengo ninguna restricción de esta índole.

Por último, si tenemos en cuenta la restricción de alcance, se llevará a cabo el plan marcado por el cliente, hasta alcanzar los objetivos del proyecto que se exigen.

Capítulo 2

Estado del Arte

Contenido

2.1. Joomla! vs CMS	7
2.2. Joomla!	9
2.3. Arquitectura Joomla!	10
2.3.1. Diagrama de Capas	10
2.3.2. Patrón MVC	11
2.3.3. Modelo Entidad-Relación	13
2.3.4. Casos de Uso y Diagramas de Clases	13
2.3.5. Primer Acercamiento	14
2.3.6. Privacidad y Usuarios (Niveles de Acceso y Grupos)	18
2.3.7. Módulos, Menús y Paginas Joomla!	21

Resumen: *En este capítulo se muestra como es el CMS Joomla!, la comparación con otros CMS y su arquitectura de una manera bastante detallada así como sus ventajas y desventajas.*

2.1. Joomla! vs CMS

Lo primero de todo, hay que hablar sobre los diferentes modos de crear una pagina web. Durante el periodo comprendido entre 1998-2007, la manera mas utilizada de crear las páginas web era utilizando HTML, CSS y añadiendole tecnología flash.

10 años después, las cosas han cambiado para mejor. Los gestores de contenido (CMS) dominan actualmente en lo que a paginas HTML estáticas se refiere. Aunque la gente con conocimientos de programación se decanta más por los diferentes frameworks PHP.

Las tres ventajas que tienen los CMS para la creación de paginas web son:

- **Baratos:** La mayoría de ellos son gratuitos y de código abierto.
- **Rápida creación:** Se puede construir una página web bastante aceptable en poco tiempo.
- **Facilidad:** Para una página web estática y simple no se necesita tener muchos conocimientos de programación.

A continuación se va a comparar los que son considerados como los tres CMS más utilizados, y sus principales diferencias:

- **WordPress:** Es la mejor opción para personas con poco o ningún conocimiento técnico, está orientado a paginas de tamaño medio y blogs.
- **Joomla!:** Es la mejor opción para páginas orientadas a e-commerce y que utilicen acceso a base de datos, pero requiere un conocimiento más elevado de programación.
- **Drupal:** Es el más potente de los tres, pero también exigen un conocimiento más elevado de programación y de aprendizaje.

Aquí se muestra una imagen con detalles interesantes de la comparación de estos tres CMS:



Figura 2.1: Imagen comparación de los tres CMS. Disponible en (<http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>)

2.2. Joomla!

Joomla! es un potente gestor de contenidos de código abierto desarrollado en PHP y liberado bajo licencia GPL, que puede funcionar sin problema alguno en la mayoría de los servidores web.

No requiere el mismo nivel técnico que requiere un Framework, pero en cambio, si se quiere crear un componente y salirse de algo delimitado o ya implementado por el gestor en sí, al no estar bien documentado, no es tarea fácil llevarlo a cabo.

Como Drupal y WordPress, Joomla! tiene multitud de plugins y plantillas para elegir, por lo que puedes personalizar tu página web casi de cualquier manera que desees. Otras de las razones principales por lo que la gente elige Joomla! son las siguientes:

- **Redes sociales:** Este es quizás el mayor beneficio de Joomla!. De los tres CMS nombrados, Joomla! hace que sea el más fácil para crear las redes sociales. Las redes sociales son un poderoso elemento de valor para muchas webs, y con Joomla!, se puede crear red social extremadamente rápido y fácil.
- **Páginas de comercio electrónico:** Si por el contrario se quiere crear una tienda online, esto se puede hacer de una manera muy simple con Joomla!, ya que lo integra de forma nativa y la conexión con la base de datos se realiza de forma automática, siendo una manera simple y rápida de poner a punto este tipo de negocios con Joomla!.
- **Gratis y de código abierto:** Una de las ventajas respecto a WordPress es que puede instalarse en servidores propios, aunque WordPress en cambio te ofrece el hosting y el dominio de forma gratuita.

2.3. Arquitectura Joomla!

2.3.1. Diagrama de Capas

Joomla basa su arquitectura física en tres grandes capas y cada capa esta formado por piezas de software independientes:

- **Capa de extensiones:** Formada por lo que serían las extensiones de Joomla! (módulos y componentes) y los templates. Son programas independientes uno de otro con una función clara y específica, desarrollados con el framework de Joomla! para ejecutarse sobre alguna de las aplicaciones de la capa inferior.
- **Capa de aplicación:** Esta formada por lo que englobaría a prácticamente todo el CMS. Se trata de 4 aplicaciones básicas creadas a partir del framework que proporcionan una serie de funcionalidades para la gestión de contenidos y una estructura sobre la que ejecutar extensiones. Estas cuatro aplicaciones son:
 - *Instalador de Joomla:* Su función es la de instalar todo el sistema Joomla! sobre un servidor web. Una vez que realiza dicha función es eliminado.
 - *Frontend o parte pública:* Aplicación encargada de crear y gestionar toda la parte que presentará de cara al público a los usuarios.
 - *Backend o parte privada:* Es la aplicación encargada de gestionar todos los parámetros de configuración del frontend. Además el backend permite la ejecución de extensiones, que permiten funciones específicas de administración de sitios web: gestión de ficheros, copias de seguridad, etc.
 - *Control remoto mediante XML-RPC.*

- **Capa del framework:** Formada por librerías externas, el framework propiamente dicho y una serie de plugins que permiten extender sus funcionalidades, es el “lenguaje común” a través del cual el resto de aplicaciones se entienden entre sí y mediante el cual pueden realizar las diferentes funciones.

A continuación, se muestra en el siguiente diagrama (ver [figura 2.2](#)), que muestra la arquitectura física de Joomla! con las tres capas por las que se compone:

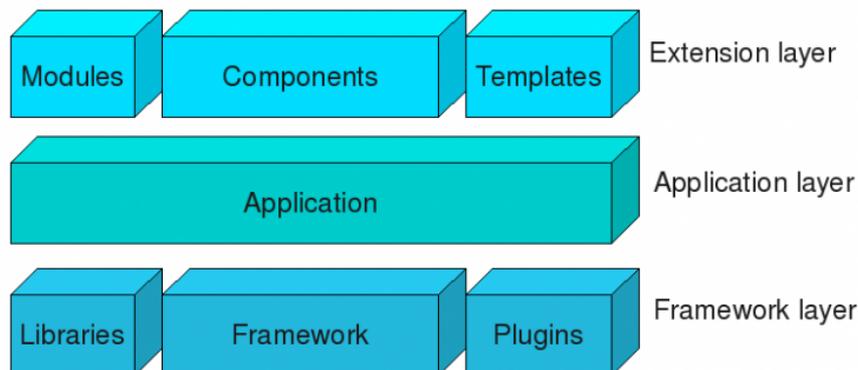


Figura 2.2: Arquitectura física Joomla!, disponible en (http://es.wikieducator.org/Archivo:Capas_joomla.png)

2.3.2. Patrón MVC

Estas tres capas (extensiones, aplicación y framework) interactúan entre sí de acuerdo con una serie de patrones establecidos. El patrón base que usan todos los componentes en Joomla! es el MVC (Modelo-Vista-Controlador).

Al estar desarrollado bajo una arquitectura basada en el patrón MVC lo que permite:

- Interactuar directamente con la parte de vista de Joomla!, lo que permite sobrescribir desde el template de Joomla! la parte de vista de un componente, módulo o plugin, lo que permite un gran nivel de personalización en el desarrollo de los templates.
- Un desarrollo de componentes módulos y plugins basados en la arquitectura base del CMS.
- Actualizaciones rápidas en caliente, actualizando los elementos requeridos para los cambios de versión con la plataforma funcionando

A continuación, se muestra en el siguiente diagrama (ver [figura 2.3](#)), que muestra la arquitectura del patrón MVC en un componente cualquiera de Joomla!:

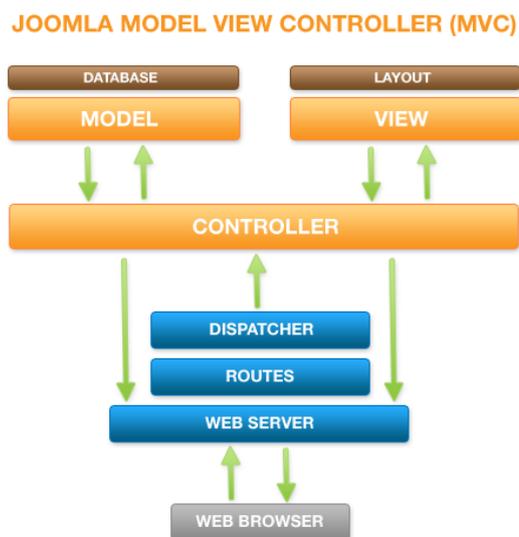
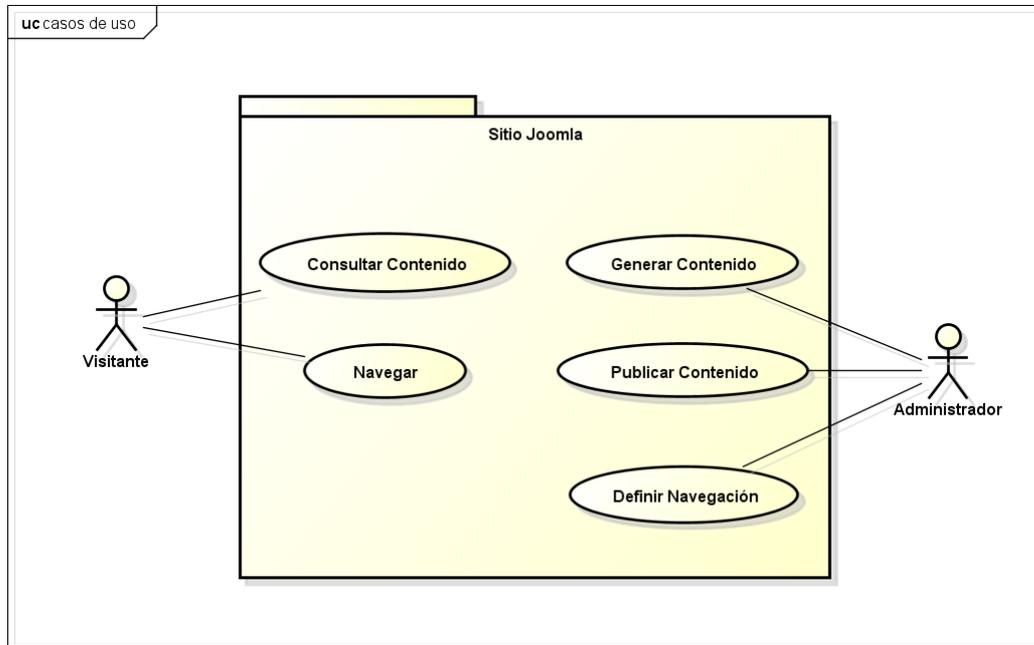


Figura 2.3: Arquitectura patrón MVC en Joomla!, disponible en (<http://goo.gl/KE1qOG>)

2.3.5. Primer Acercamiento

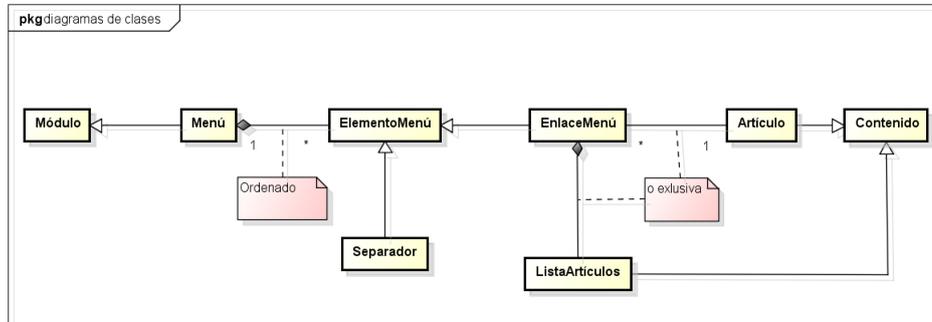
En esta primera figura (ver [figura 2.5](#)) vamos a reducir el alcance de las funcionalidades de Joomla! y considerar los siguientes casos de uso:



Para satisfacer las necesidades del usuario Invitado (acceder a la web, al menú y al contenido), el actor administrador tendrá que actuar sobre los distintos elementos de un sitio web Joomla!, vamos a representar a los elementos con los diagramas de clases.

Menús

Los Menús de Joomla! solo constan en realidad de enlaces a los contenidos del sitio web. Para simplificar el diagrama (ver figura 2.8), no se muestran las clases derivadas de la clase abstracta Menú de enlace, que corresponden a diferentes clases derivadas de la clase Contenido.

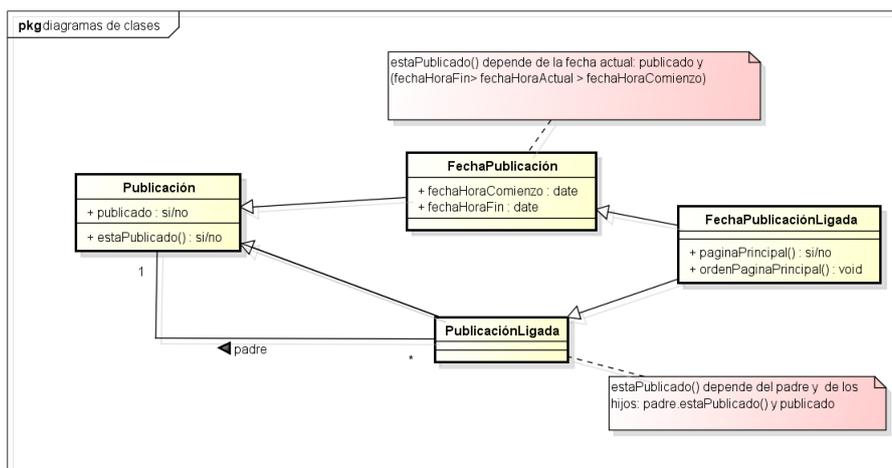


La Publicación de los Contenidos en los Menús

Aunque el administrador puede crear y gestionar el contenido del sitio web y definir cómo usar el sitio, el usuario invitado no puede acceder a estos elementos. Queda por definir el contenido que se va a publicar y los menús en los que se va a publicar (paso absolutamente necesario en Joomla!).

Publicación

Vamos a hablar de la publicación en Joomla! de manera detallada, ya que sólo un conocimiento completo del mecanismo permite al administrador obtener el resultado deseado. Consideramos en este diagrama (ver figura 2.9) una clase Publicación con un método estaPublicado() que devuelve sí o no:

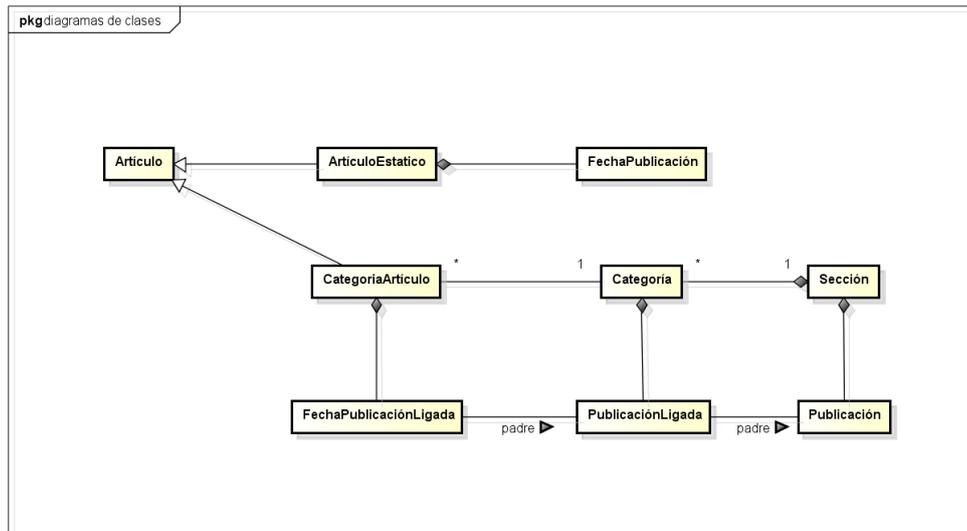


Un objeto de la clase Publicación será asociado a cada elemento del sitio Joomla! que pueda publicarse. Los objetos de la clase FechaPublicaciónLigada tienen además los métodos paginaIni-

cio() y ordenPaginaInicio() que determinarán si el método estaPublicado()=si, su presencia en la página de Inicio y en que orden respectivamente.

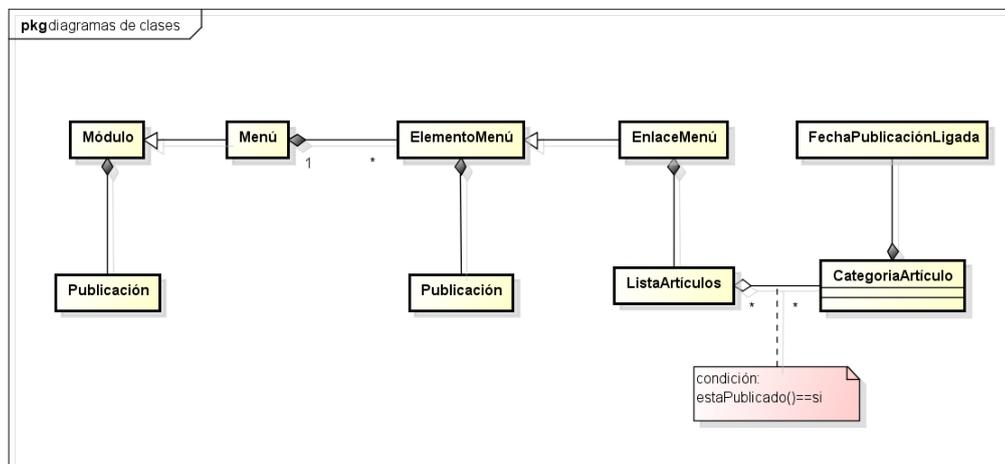
Artículos Publicados

Como podemos ver en el siguiente diagrama (ver [figura 2.10](#)) la publicación de CategoríaArtículo en Joomla! depende de la publicación de la Categoría y de la publicación de la Sección de esa categoría. Por lo que hay que tener esas relaciones en cuenta:



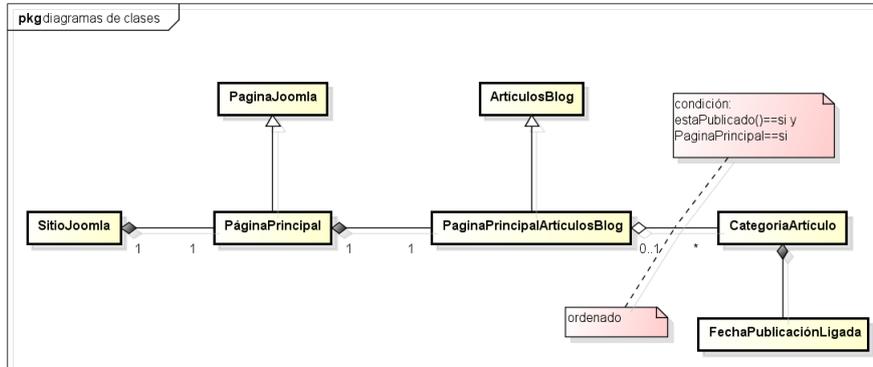
Los Menús y la Lista de Artículos Publicados

La publicación de la clase Menú derivada de la clase Módulo se gestiona para la publicación de los Módulos, la clase ListaArtículos se gestiona a través de la publicación de ElementoMenú, en la siguiente figura (ver [figura 2.11](#)) se muestra:



La Página de Inicio

Para finalizar, nos queda por resolver el caso particular de la página principal de la web construida con Joomla!, esto se muestra en la figura (ver [figura 2.12](#)):



2.3.6. Privacidad y Usuarios (Niveles de Acceso y Grupos)

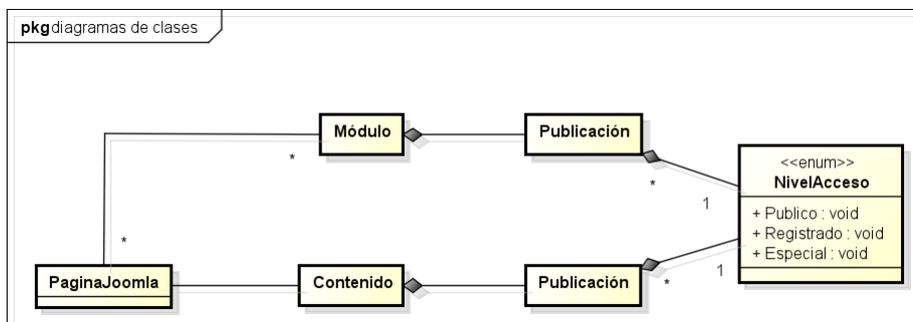
La confidencialidad y los usuarios se consideran en conjunto, porque cada elemento de la página puede estar asociada con un nivel de acceso que hará que sea accesible o no accesible de acuerdo con el grupo de usuarios a los que el actor que desea consultar pertenece.

Privacidad (Niveles de Acceso)

Hay 3 niveles de acceso generales:

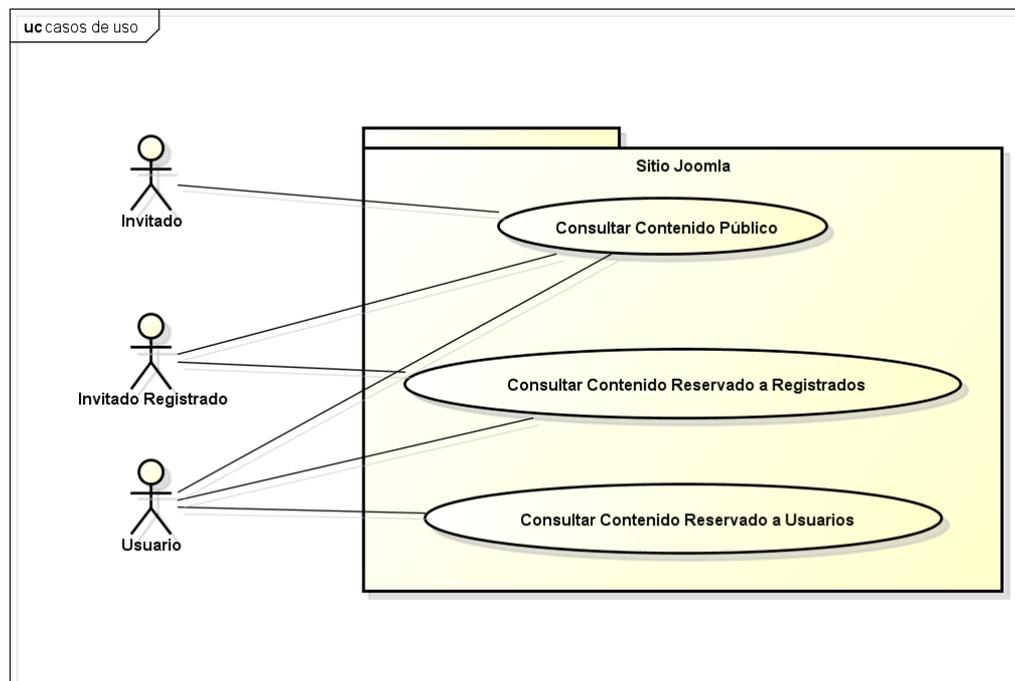
- El nivel Publico
- El nivel Registrado
- El nivel Especial

Vamos a incluir este diagrama (ver [figura 2.13](#)) que a la clase Publicación se le añade la clase NivelAcceso que es de tipo enumeración:



Una misma página de Joomla! puede tener elementos de diferente nivel de acceso. En cada uno de estos niveles de acceso es un tipo de actor, estos tipos son los siguientes:

- El actor Invitado
- El actor Invitado Registrado
- El actor Usuario



Los actores Invitado Registrado y el Usuario tienen que estar identificados por un usuario y una contraseña convencionales.

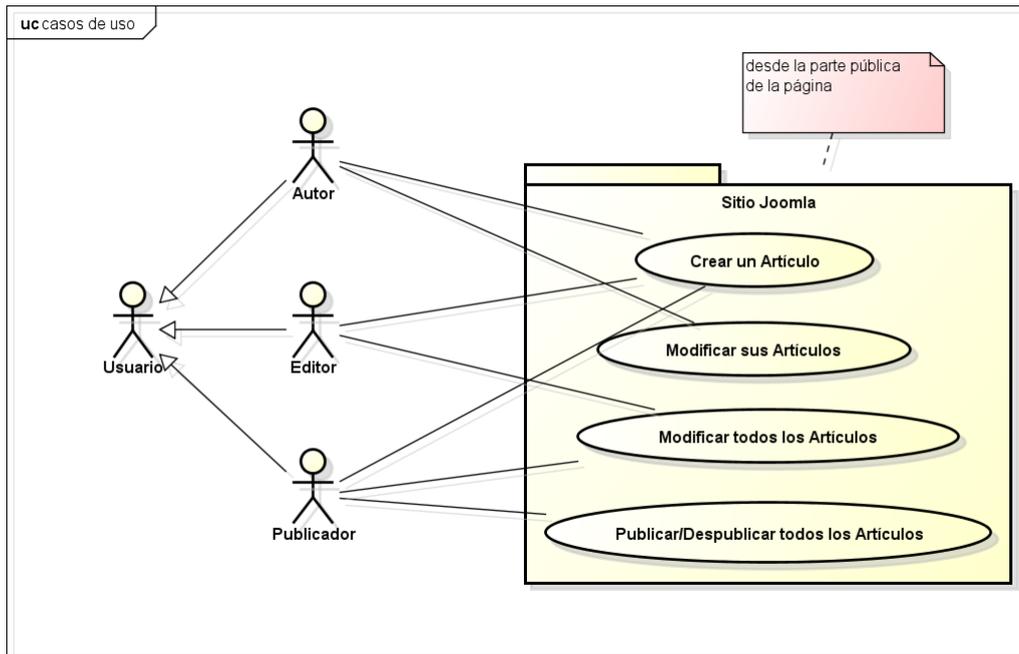
A nivel de interfaz de administración el Invitado Registrado corresponde al grupo Registrado y el actor Usuario corresponde al resto de grupos de usuarios: Autor, Editor, Publicador, Manager, Administrador y Super Administrador.

Usuarios (Grupos)

Un actor Usuario va a ser capaz de editar el sitio, se puede intervenir esencialmente en el artículo y/o en el nivel Administración. Hay seis tipos diferentes de actores Usuarios.

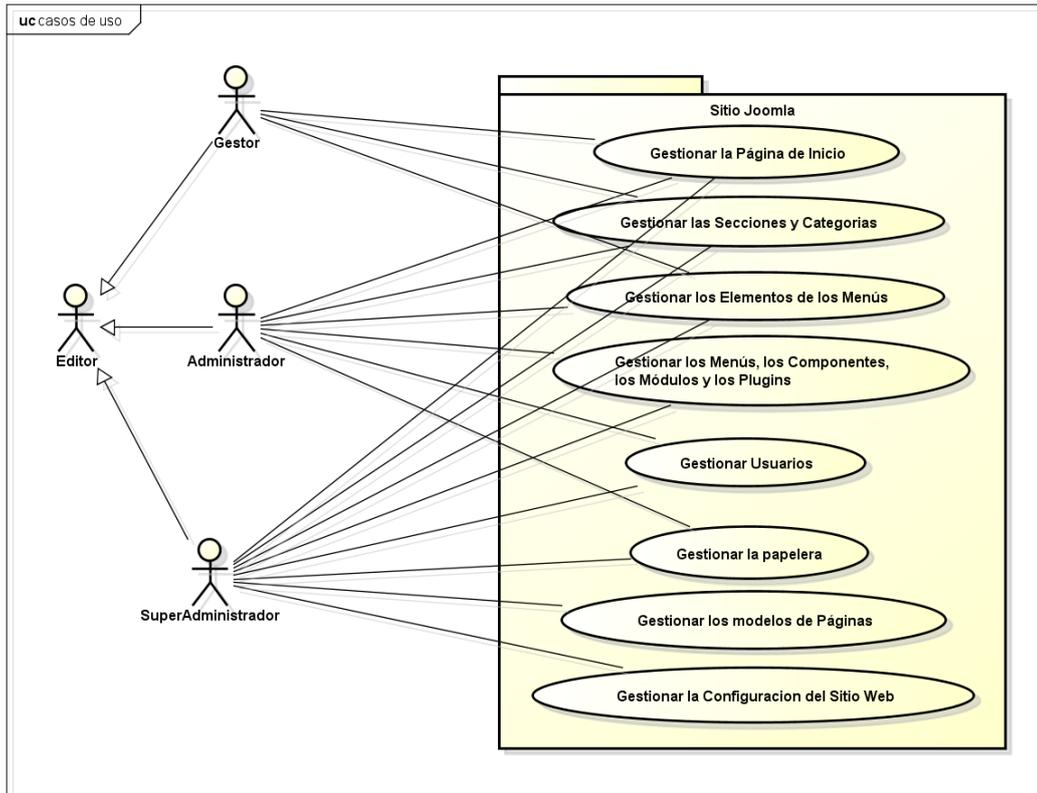
Los Usuarios y los Artículos

A continuación podemos ver un diagrama con los usuarios involucrados en la parte pública de Joomla! o frontend. Se puede ver que el actor Usuario se diferencia en tres tipos de actores: Autor, Editor y por último Publicador (ver [figura 2.15](#)):



Los Usuarios y la Administración

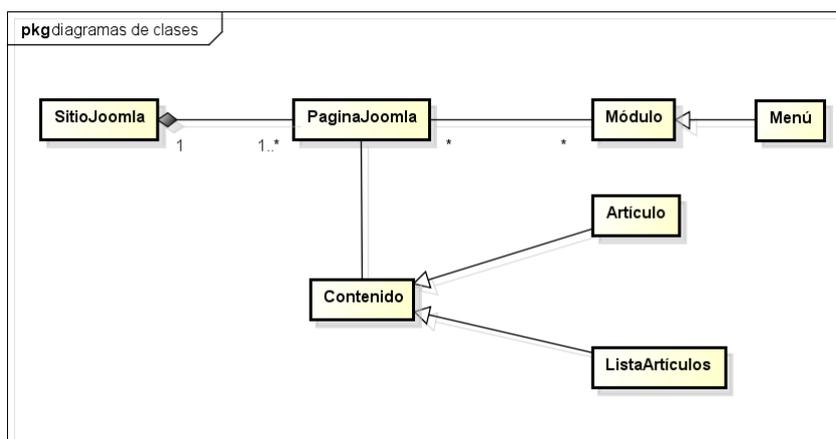
A continuación podemos ver un diagrama con los usuarios que participan en la sección de administración de Joomla! o backend. Se puede ver como el actor Editor puede ser de tres tipos diferentes, con sus consiguientes diferencias. Estos actores son Gestor, Administrador y por último Super Administrador (ver [figura 2.16](#)).



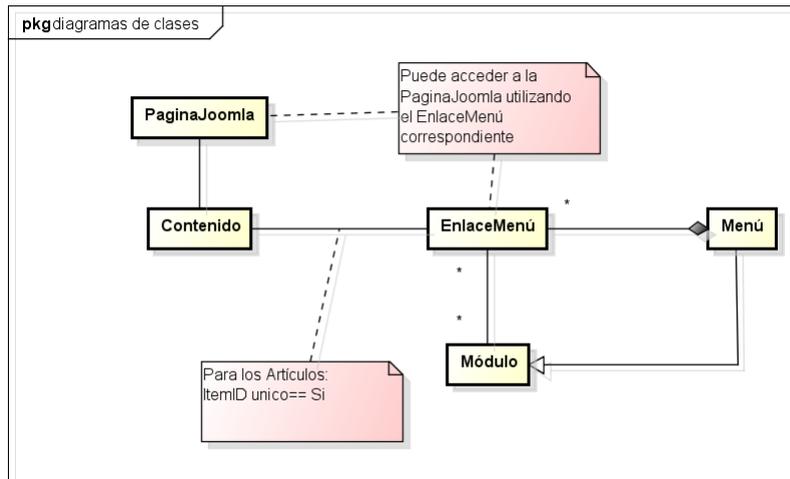
2.3.7. Módulos, Menús y Páginas Joomla!

Consideramos que los módulos son idénticos para todas las páginas del sitio web. Ahora vamos a ver que podemos asignar a cada página una lista de los diferentes módulos de otras páginas.

Y nos da como resultado este diagrama (ver [figura 2.17](#)):



De hecho, la asociación entre la clase Módulo y la clase PáginaJoomla es a través de la clase EnlaceMenú que proporciona acceso a la clase PáginaJoomla correspondiente, como podemos ver en este diagrama (ver [figura 2.18](#)).



Capítulo 3

Plan de Desarrollo del Proyecto

Contenido

3.1. Plan de Desarrollo Software	23
3.1.1. Visión General	24
3.1.2. Plan de Iteración de la Etapa de Inicio	26
3.1.3. Plan de Iteración de la Etapa de Elaboración	27
3.1.4. Plan de Iteración de la Etapa de Construcción	28
3.1.5. Plan de Iteración de la Etapa de Transición	29
3.2. Seguimiento del Proyecto	29
3.2.1. Evolución del Proyecto	29
3.3. Organización del Proyecto	31
3.3.1. Interfaces Externas	31
3.3.2. Estructura Interna	31
3.3.3. Roles y Responsabilidades	31
3.4. Plan de Control y Riesgos	33
3.4.1. Plan de Control	33
3.4.2. Plan de Riesgos	33
3.5. Herramientas Utilizadas	37

Resumen: *En este capítulo se muestra la planificación que se ha seguido para desarrollar el proyecto así como los problemas que han surgido y han podido alterar dicha planificación.*

3.1. Plan de Desarrollo Software

Se llama plan de desarrollo software al documento que gestiona un proyecto, definiendo para ellos las técnicas, actividades y tareas necesarias para concluir satisfactoriamente dicho proyecto. En esta sección se pretende dar a exponer los elementos básicos de un proyecto y de esta manera detallar los aspectos básicos tal y como han sido entendidos y formulados.

3.1.1. Visión General

Como se puede ver en los objetivos marcados para este proyecto, consta de diversos puntos que al final forman uno solo, ya que por separado no realizarían ninguna función, que sería todo el funcionamiento de la web con todas sus funciones y componentes, por lo que se tratarán como un conjunto. Además una parte importante de la planificación son las **reuniones semanales** que se mantenían con el cliente, en este caso Ramiro González Catón.

Para el desarrollo del proyecto se ha seguido el proceso UPEDU, por lo que se contará con una fase de inicio, una de elaboración, otra de construcción y por último una fase de transición. Los objetivos de cada una de estas fases les podemos definir como:

■ Inicio:

- Comprensión del problema.
- Análisis de los requisitos, riesgos y definición del modelo de casos de uso del sistema.
- Realizar estimaciones iniciales de planificación y costes.

■ Elaboración:

- Definición de los diagramas de clases, el modelo de datos e implementación de la interfaz de usuario y de los casos de uso más importantes, consiguiendo así versiones de prueba ya utilizables y testeables y reduciendo los costes de desarrollo.
- Revisión de los requisitos.

■ Construcción:

- Añadir el total de funcionalidades al prototipo considerando los nuevos requisitos surgidos de la revisión.
- Elaborar el plan de pruebas.

■ Transición:

- Entrega de la documentación al cliente.
- Conseguir que el cliente final acepte el producto y sea capaz de utilizarlo y entenderlo.
- Prueba del sistema en el entorno de explotación.

Además de tener distintas fases, hay que entregar una serie de artefactos en cada una de ellas, para llevar a cabo todos los términos del proyecto. A continuación se muestra esa lista:

■ Fase de Inicio:

- Lista de riesgos.
- Especificación de requisitos software.
- Especificación de casos de uso.
- Plan de iteración inicial.
- Plan de iteración de elaboración.

- Plan de desarrollo software.
- Documento de seguimiento del proyecto.
- **Fase de Elaboración:**
 - Documento general de análisis.
 - Documento de arquitectura software.
 - Elaboración de casos de uso.
 - Plan de pruebas.
 - Casos de prueba.
 - Plan de iteración de construcción.
- **Fase de Construcción:**
 - Modelo de implementación.
 - Plan de iteración de transición.
 - Aplicación final.
- **Fase de Transición:**
 - Manual de usuario.
 - Manual de instalación.

3.1.2. Plan de Iteración de la Etapa de Inicio

El objetivo principal de esta fase es saber el problema al que nos enfrentamos, identificar los recursos necesarios para poder llevar a cabo el proyecto y definir una planificación completa del trabajo que se va a llevar a cabo.

Cuando se termine esta fase tienen que estar perfectamente definidos los requisitos del proyecto, los casos de usos principales y su consiguiente análisis de riesgos. Lo fundamental es saber el trabajo que hay que realizar y cómo llevarlo a cabo.

Aquí se muestra el inicio y duración previsto para la **fase de inicio** (ver [figura 3.1](#) para vista de las tareas y ver [figura 3.2](#) para vista de Gantt):

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	▲ Inicio	32,5 días	lun 17/02/14	sáb 22/03/14	
2	▲ Primera iteración	22,38 días	lun 17/02/14	mié 12/03/14	
3	plan de iteración de inicio	5 días	lun 17/02/14	vie 21/02/14	
4	Conocimientos técnicos del software	2 días	sáb 22/02/14	lun 24/02/14	3
5	reunion semanal cliente	1 hora	lun 24/02/14	lun 24/02/14	4
6	Plan de desarrollo software	3 días	lun 24/02/14	jue 27/02/14	5
7	Definición de riesgos de software	4 días	jue 27/02/14	lun 03/03/14	6
8	reunion semanal cliente	1 hora	lun 03/03/14	lun 03/03/14	7
9	Recogida de requisitos	5 días	lun 03/03/14	sáb 08/03/14	8
10	Primera versión de casos de uso	3 días	sáb 08/03/14	mar 11/03/14	9
11	reunion semanal cliente	1 hora	mié 12/03/14	mié 12/03/14	10
12	▲ Segunda iteración	10,13 días	mié 12/03/14	sáb 22/03/14	2
13	Lista de Riesgos software	2 días	mié 12/03/14	vie 14/03/14	11
14	Especificación de requisitos software	3 días	vie 14/03/14	lun 17/03/14	13
15	reunion semanal cliente	1 hora	lun 17/03/14	lun 17/03/14	14
16	Plan de pruebas	2 días	lun 17/03/14	mié 19/03/14	15
17	Plan de iteración de elaboración	1 día	mié 19/03/14	jue 20/03/14	16
18	Informe de seguimiento	2 días	jue 20/03/14	sáb 22/03/14	17

Figura 3.1: Duración de las tareas de la fase de Inicio.

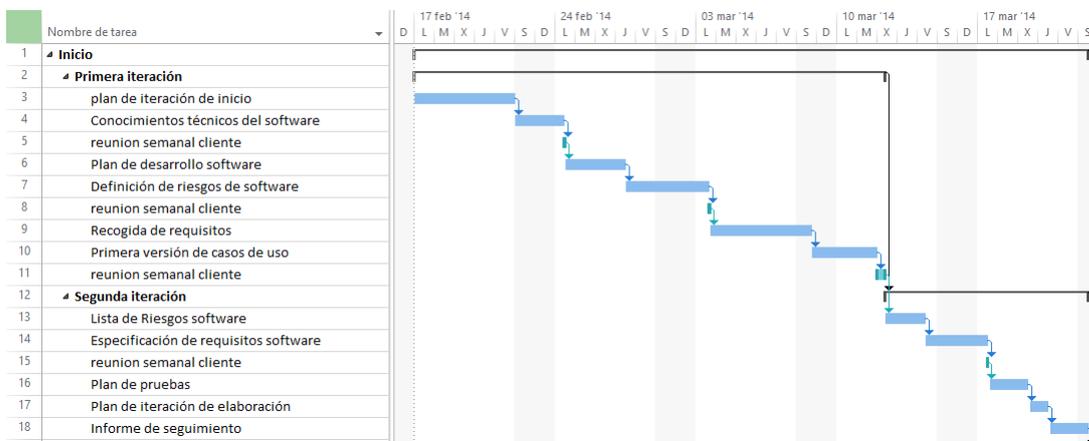


Figura 3.2: Diagrama de Gantt de la fase de Inicio.

3.1.3. Plan de Iteración de la Etapa de Elaboración

Cuando se termine esta fase, se tiene que haber obtenido la mayor parte del análisis y un primer prototipo de las interfaces de usuario, así como la implementación de los casos de uso más importantes.

Aquí se muestra el inicio y duración previsto para la **fase de elaboración** (ver [figura 3.3](#) para vista de las tareas y ver [figura 3.4](#) para vista de Gantt):

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
19	Elaboración	47,63 días	sáb 22/03/14	sáb 10/05/14	1
20	Primera iteración	24,38 días	sáb 22/03/14	mié 16/04/14	12
21	Revisión de los requisitos	3 días	sáb 22/03/14	mar 25/03/14	18
22	reunion semanal cliente	1 hora	mar 25/03/14	mar 25/03/14	21
23	Revisión del modelo de casos de uso	3 días	mar 25/03/14	vie 28/03/14	22
24	Diagrama de análisis	4 días	vie 28/03/14	mié 02/04/14	23
25	reunion semanal cliente	1 hora	mié 02/04/14	mié 02/04/14	24
26	Diseño inicial del sistema	4 días	mié 02/04/14	dom 06/04/14	25
27	Diagramas de diseño	5 días	dom 06/04/14	vie 11/04/14	26
28	Estudio de la arquitectura de los componentes	5 días	vie 11/04/14	mié 16/04/14	27
29	reunion semanal cliente	1 hora	mié 16/04/14	mié 16/04/14	28
30	Segunda iteración	23,25 días	mié 16/04/14	sáb 10/05/14	20
31	Documento de análisis del proyecto	4 días	mié 16/04/14	lun 21/04/14	29
32	Documento de diseño del sistema	4 días	lun 21/04/14	vie 25/04/14	31
33	reunion semanal cliente	1 hora	vie 25/04/14	vie 25/04/14	32
34	Documento de arquitectura software	3 días	vie 25/04/14	lun 28/04/14	33
35	Plan de iteración de construcción	2 días	lun 28/04/14	mié 30/04/14	34
36	reunion semanal cliente	1 hora	mié 30/04/14	mié 30/04/14	35
37	Realización del prototipo inicial	10 días	mié 30/04/14	sáb 10/05/14	36

Figura 3.3: Duración de las tareas de la fase de Elaboración.

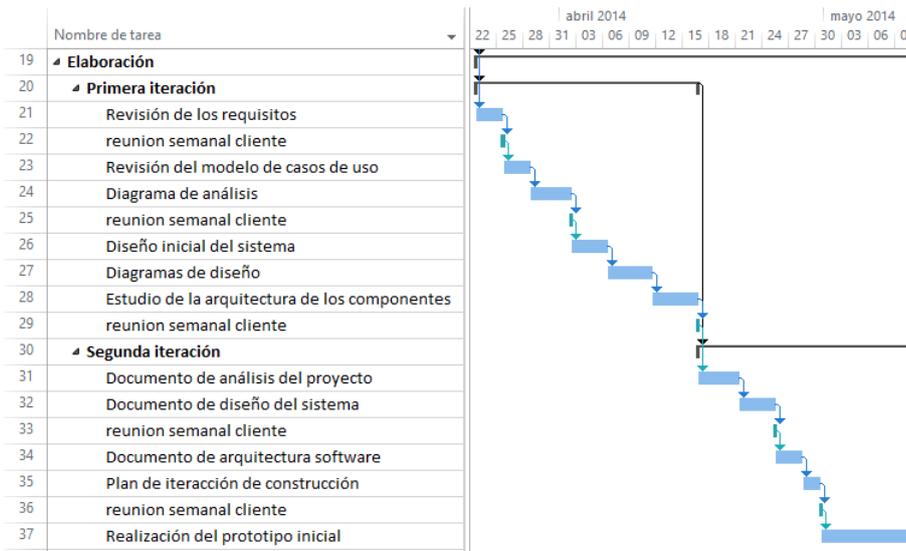


Figura 3.4: Diagrama de Gantt de la fase de Elaboración.

3.1.4. Plan de Iteración de la Etapa de Construcción

El objetivo de esta fase es construir todas las funcionalidades de la aplicación. El producto final se desarrolla en dos iteraciones como en las fases anteriores, haciendo las pruebas correspondientes después de cada una de ellas. Al final de esta fase se han añadido todas las funcionalidades al prototipo anterior, añadiéndole los nuevos requisitos que hayan podido surgir en las revisiones anteriores.

En la [figura 3.5](#) podemos ver las actividades que componen la fase, y en la [figura 3.6](#), el diagrama de Gantt resultante:

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
38	▲ Construcción	56,63 días	dom 11/05/14	mar 08/07/14	19
39	▲ Primera iteración	32,25 días	dom 11/05/14	vie 13/06/14	30
40	Revisión documento de análisis	8 días	dom 11/05/14	lun 19/05/14	37
41	reunion semanal cliente	1 hora	lun 19/05/14	lun 19/05/14	40
42	Revisión documento de diseño	7 días	lun 19/05/14	lun 26/05/14	41
43	reunion semanal cliente	1 hora	lun 26/05/14	lun 26/05/14	42
44	Modelo inicial implementación	15 días	lun 26/05/14	mié 11/06/14	43
45	Pruebas	2 días	mié 11/06/14	vie 13/06/14	44
46	▲ Segunda iteración	24,38 días	vie 13/06/14	mar 08/07/14	39
47	reunion semanal cliente	1 hora	vie 13/06/14	vie 13/06/14	45
48	Modelo final implementación	18 días	vie 13/06/14	mié 02/07/14	47
49	reunion semanal cliente	2 horas	mié 02/07/14	mié 02/07/14	48
50	Pruebas	2 días	mié 02/07/14	vie 04/07/14	49
51	Plan de iteración de Transición	2 días	vie 04/07/14	dom 06/07/14	50
52	Informe de seguimiento	2 días	dom 06/07/14	mar 08/07/14	51

Figura 3.5: Duración de las tareas de la fase de Construcción.



Figura 3.6: Diagrama de Gantt de la fase de Construcción.

3.1.5. Plan de Iteración de la Etapa de Transición

Durante la fase de transición se le entregará al cliente el manual de usuario y el manual de instalación del sistema. No se le hace presentación del sistema porque el cliente ya ha ido testeando las pruebas del mismo.

En la [figura 3.7](#) podemos ver las actividades que componen la fase, y en la [figura 3.8](#) el diagrama de Gantt resultante:

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
53	Transición	8,13 días	mar 08/07/14	jue 17/07/14	38
54	reunion semanal cliente	1 hora	mar 08/07/14	mar 08/07/14	52
55	Manual de Usuario	3 días	mar 08/07/14	vie 11/07/14	54
56	Manual de Instalación	3 días	vie 11/07/14	mar 15/07/14	55
57	Informe de seguimiento	2 días	mar 15/07/14	jue 17/07/14	56

Figura 3.7: Duración de las tareas de la fase de Construcción.

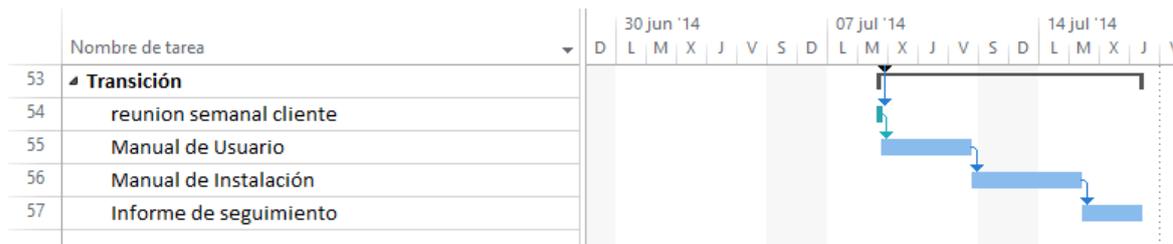


Figura 3.8: Diagrama de Gantt de la fase de Construcción.

3.2. Seguimiento del Proyecto

El objetivo de esta sección es comparar la evolución del proyecto real con la planificación llevada a cabo, e indicar porqué motivos se ha tenido que modificar dicha planificación.

3.2.1. Evolución del Proyecto

Inicialmente el proyecto estaba previsto comenzar el 17 de febrero de 2014 y terminarlo el 14 de julio de 2014. El proyecto ha sufrido un retraso, debido en gran parte a no disponer del tiempo estimado para la realización del mismo, y también debido a que en una parte de la implementación del proyecto dependíamos de un organismo externo para obtener permisos a una base de datos y tardaron unas tres semanas en concedernos dichos permisos. A parte de eso en la adaptación de los componentes que había que crear hubo problemas al no haber suficiente documentación para la creación de los mismos al tratarse de una versión del CMS muy actual.

En la [tabla 3.2.1](#) podemos ver las diferencias entre el tiempo previsto para la realización de este proyecto, y el tiempo que realmente se ha tardado en realizar:

Fase	Iteración	Duración Estimada	Duración Real
Inicio	1	22	24
	2	10	10
	Total	32	34
Elaboración	1	24	24
	2	23	28
	Total	47	52
Construcción	1	32	48
	2	24	36
	Total	56	84
Transición	1	8	8

3.3. Organización del Proyecto

3.3.1. Interfaces Externas

La interfaz externa de la aplicación planteada será desarrollada con Joomla! 3.2.4, un gestor de contenidos muy utilizado actualmente para la creación de páginas web. Esta herramienta es muy potente y facilita mucho la labor del programador. La interfaz web tendrá un aspecto minimalista y una disposición intuitiva para una fácil interacción con el usuario.

Además la plantilla o template que se ha escogido para este proyecto es responsiva, lo que para el acceso desde smartphones o tabletas deja una interfaz mucho más limpia y acorde con la plataforma desde la que se accede. El usuario podrá acceder a la página y navegar a través de los menús disponibles, pero solo aquellos usuarios que se registren y el administrador los dé de alta podrán tener acceso a toda la funcionalidad de la web.

El administrador será el encargado de confirmar el registro de los usuarios que considere necesarios mediante un click en el enlace que le llegará a su correo para confirmar el registro de ese usuario. Además tendrá acceso al panel de control del portal web pudiendo añadir artículos, noticias e incluso modificando la página a su antojo de una manera bastante sencilla.

3.3.2. Estructura Interna

En este subapartado se explicará como estará estructurado el proyecto, de manera interna. Como es un proyecto realizado por una persona, en las fases de planificación, análisis, diseño e implementación, todas las tareas serán realizadas por mi. Más adelante detallaré su estructura desglosada.

Las tareas que voy a tener que llevar a cabo me situarían como analista en la fase de planificación y análisis, diseñador en una parte de la fase de análisis y en la fase de diseño y programador en la fase de implementación. Estos son los roles que me toca asumir a lo largo de este proyecto.

3.3.3. Roles y Responsabilidades

En esta sección, se van a detallar los roles que desempeñaré en cada momento a lo largo de todo el proyecto. Tenemos cuatro fases importantes dentro del proyecto: plan de proyecto, documento de análisis, documento de diseño e implementación.

- **FASE 1: Plan de proyecto.**

Rol	Responsable	Responsabilidades
Jefe de proyecto	Sergio Cárcel Fernando	Planificación de las etapas del proyecto.
Analista	Sergio Cárcel Fernando	Elaboración del resumen del sistema y los requisitos, análisis de las interfaces externas y plan de control de trabajo y realización de plan de riesgos y plan de calidad.

Cliente	Ramiro González Catón	Funcionalidades que debe cumplir el sistema.
---------	-----------------------	--

■ **FASE 2: Documento de análisis.**

Rol	Responsable	Responsabilidades
Jefe de proyecto	Sergio Cárcel Fernando	Revisión de los requisitos del sistema y control de los documentos elaborados.
Diseñador	Sergio Cárcel Fernando	Realización de casos de uso.
Analista	Sergio Cárcel Fernando	Realización de casos de uso y revisión de requisitos, realización de casos de uso y modelos de dominio.
Cliente	Ramiro González Catón	Enumeración de funcionalidades del sistema.

■ **FASE 3: Documento de diseño.**

Rol	Responsable	Responsabilidades
Jefe de proyecto	Sergio Cárcel Fernando	Control de los documentos elaborados por el equipo de proyecto y revisión del diagrama de clases final.
Diseñador	Sergio Cárcel Fernando	Realización de diagramas de secuencia.
Analista	Sergio Cárcel Fernando	Elaboración de modelo de comportamiento y arquitectura física, realización de clases de diseño.
Cliente	Ramiro González Catón	Detalles que debe cumplir el sistema.

■ **FASE 4: Implementación.**

Rol	Responsable	Responsabilidades
Jefe de proyecto	Sergio Cárcel Fernando	Revisión de funcionamiento de sistema acorde a los requisitos y objetivos.
Programador	Sergio Cárcel Fernando	Diseño de prototipo y desarrollo de código.
Diseñador	Sergio Cárcel Fernando	Batería de pruebas de caja blanca y guía de usuario.

Cliente	Ramiro González Catón	Batería de pruebas de caja negra, comprobación de funcionalidades y comprobación final diseño del sistema.
---------	--------------------------	--

3.4. Plan de Control y Riesgos

3.4.1. Plan de Control

- La documentación estará alojada en Dropbox¹ para tener copias de seguridad en caso de pérdida o corrupción de datos.
- El código fuente desarrollado tendrá copias de seguridad en la nube y a todo el proyecto de Joomla! se le realizarán *backups* periódicos con la herramienta externa Akeeba Backup².
- Los requisitos serán revisados al final de cada fase para comprobar que el desarrollo se está ajustando a lo establecido en etapas anteriores.
- El cliente irá probando las diferentes versiones de la página para dar el visto bueno o ir añadiendo o modificando requisitos o detalles, ya sean de implementación o de diseño.

3.4.2. Plan de Riesgos

- **Riesgo 1:**

FORMULARIO ADMINISTRACIÓN DEL RIESGO		
Identificador: 1	Fecha: 26/3/2014	Categoría: Riesgos de Proyecto.
Título: Fallos de personal.	Probabilidad: Baja.	Consecuencia: Mayor carga de trabajo en un menor tiempo ya que solo hay una persona implicada en el proyecto.
Fase: En todos los hitos.		

Valoración del riesgo

Descripción del Riesgo: Posibilidad de que un miembro del equipo por diferentes razones no pueda realizar su función.

Contexto del Riesgo: Este riesgo se puede producir en cualquier fase del proyecto al sucederse una situación inesperada con cualquier miembro del equipo.

Análisis del Riesgo: El impacto de este riesgo repercute aumentando la carga de trabajo en un menor tiempo influyendo en la calidad del producto a realizar.

¹Almacenamiento en la nube Dropbox: www.dropbox.com

²Componente Joomla! para realizar copias de seguridad

Planificación del Riesgo
<u>Estrategia:</u> Reducción del riesgo, se tratará.
<u>Plan de acción del riesgo:</u> Al no poder suplantar al personal que no puede realizar las funciones, se aumentarían las horas/hombre por día para llegar a los objetivos marcados .

■ **Riesgo 2:**

FORMULARIO ADMINISTRACIÓN DEL RIESGO		
Identificador: 2	Fecha: 26/3/2014	Categoría: Riesgos de Proceso.
Título: Planificaciones del tiempo en cada tarea.	Probabilidad: Media	Consecuencia: Aumento de las horas extras.
Fase: Todos los Hitos.		

Valoración del riesgo
<u>Descripción del Riesgo:</u> Posibilidad de una mala predicción de cada una de las tareas del proyecto.
<u>Contexto del Riesgo:</u> Este riesgo se puede producir en cualquier fase del proyecto debido a una mala planificación inicial.
<u>Análisis del Riesgo:</u> El impacto de este riesgo puede repercutir las tareas posteriores a la afectada, exigiendo un cambio de planificación.

Planificación del Riesgo
<u>Estrategia:</u> Reducción del riesgo.
<u>Plan de acción del riesgo:</u> En caso de una mala predicción será necesario aumentar la carga de trabajo del único miembro en lo que a esa fase se refiere y menor dedicación de tiempo a otras tareas.

■ **Riesgo 3:**

FORMULARIO ADMINISTRACIÓN DEL RIESGO		
Identificador: 3	Fecha: 26/3/2014	Categoría: Riesgos de Proceso.
Título: Decisión errónea de requisitos.	Probabilidad: Media	Consecuencia: Realización de cambios continuos en los requisitos.
Fase: Segundo hito.		

Valoración del riesgo
<u>Descripción del Riesgo:</u> Posible mal análisis de los requisitos.
<u>Contexto del Riesgo:</u> Este riesgo se puede producir en la fase de análisis al no realizar correctamente el análisis de requisitos.
<u>Análisis del Riesgo:</u> El impacto de este riesgo puede repercutir en la calidad del producto a entregar al no ajustarse adecuadamente a las especificaciones deseadas.

Planificación del Riesgo
<u>Estrategia:</u> Protección del riesgo.
<u>Plan de acción del riesgo:</u> Realización de una revisión de requisitos para reducir la probabilidad de riesgos incorrectos.

■ **Riesgo 4:**

FORMULARIO ADMINISTRACIÓN DEL RIESGO		
Identificador: 4	Fecha: 26/3/2014	Categoría: Riesgo del producto.
Título: Fallo de hardware.	Probabilidad: Baja.	Consecuencia: Retraso de la realización de tareas afectadas por dicho hardware.
Fase: En todos los hitos.		

Valoración del riesgo
<u>Descripción del Riesgo:</u> Posibilidad de no disponibilidad del hardware en el momento necesitado.
<u>Contexto del Riesgo:</u> Este riesgo se puede producir en cualquier fase debido a problemas no controlables en el funcionamiento del hardware.
<u>Análisis del Riesgo:</u> El impacto de este riesgo puede producir retraso en la realización de tareas afectadas por dicho hardware.

Planificación del Riesgo
<u>Estrategia:</u> Aceptación del riesgo.
<u>Plan de acción del riesgo:</u> No se realizará ninguna acción ya que el coste asociado a cubrir el riesgo es excesivo a la probabilidad de que ocurra.

■ **Riesgo 5:**

FORMULARIO ADMINISTRACIÓN DEL RIESGO		
Identificador: 5	Fecha: 26/3/2014	Categoría: Riesgo del producto.
Título: Fallo de software.	Probabilidad: Baja.	Consecuencia: Retraso de la realización de tareas afectadas por dicho software.
Fase: En todos los hitos.		

Valoración del riesgo
<u>Descripción del Riesgo:</u> Posibilidad de funcionamiento incorrecto del software debido a un problema externo (virus, error de mantenimiento, etc.).
<u>Contexto del Riesgo:</u> Este riesgo se puede producir en cualquier fase debido a fallos no controlables en del software.
<u>Análisis del Riesgo:</u> El impacto de este riesgo puede producir retraso en la realización de tareas afectadas por dicho software.

Planificación del Riesgo
<u>Estrategia:</u> Reducción del riesgo.
<u>Plan de acción del riesgo:</u> Se dispondrá de distintas aplicaciones para realizar la tarea implicada.

■ **Riesgo 6:**

FORMULARIO ADMINISTRACIÓN DEL RIESGO		
Identificador: 6	Fecha: 26/3/2014	Categoría: Riesgo de acceso a base de datos externa.
Título: Fallo al acceder al servicio web.	Probabilidad: Media.	Consecuencia: Retraso de la realización de la API debido al retraso al dar de alta la ip en el servidor donde está alojado el servicio web.
Fase: Fase de Implementación.		

Valoración del riesgo
<u>Descripción del Riesgo:</u> Posibilidad de lento proceso a la hora de poder acceder al servicio web, debido a que deben dar de alta la ip del servidor.
<u>Contexto del Riesgo:</u> Este riesgo se puede producir desde el momento que se llame para que den de alta la IP para acceder al servicio pero principalmente en la fase de implementación.
<u>Análisis del Riesgo:</u> El impacto de este riesgo puede producir retraso en la realización del caso de uso introducir artículo por el DOI.

Planificación del Riesgo
<u>Estrategia:</u> Aceptación del riesgo.
<u>Plan de acción del riesgo:</u> Esperar a que den de alta el servidor o en su caso obviar ese caso de uso debido a problemas externos.

3.5. Herramientas Utilizadas

A continuación describo las herramientas utilizadas para la realización de este proyecto:

- Herramientas para la documentación:
 - TexMaker: Herramienta $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ para la realización de la memoria.
 - Astah: Herramienta de modelado para la realización de todos los diagramas.
 - Microsoft Project 2013: Herramienta usada para la planificación del proyecto.
- Herramientas para el tratamiento de imágenes:
 - GIMP 2.8.6: Editor gráfico para el tratamiento de imágenes.
- Entorno de programación:
 - PhpDesigner 8: Entorno de desarrollo para realizar los componentes de la aplicación.
 - Joomla! 3.2.4 Stable: CMS utilizado para la creación del entorno web.
 - PHP, MYSQL, HTML, CSS: lenguajes de programación usados.
 - SQL Workbench: Herramienta con la que modelar los diagramas entidad relación.
 - Apache 2: Servidor HTTP de código abierto.
- Otras herramientas:
 - Dropbox: Copias en la nube de todos los ficheros creados y toda la documentación.
 - Akeeba Backup: Componente que se integra con Joomla! para poder hacer copias de seguridad periódicas.

Capítulo 4

Análisis de la Aplicación

Contenido

4.1. Requisitos	39
4.1.1. Requisitos de Información	39
4.1.2. Requisitos Funcionales	41
4.1.3. Requisitos no Funcionales	43
4.2. Casos de Uso	45
4.2.1. Diagrama de Casos de Uso	45
4.2.2. Actores	46
4.2.3. Descripción de los Casos de Uso	46
4.2.4. Escenarios	59
4.3. Modelo de Dominio	62
4.3.1. Descripción de las Clases de Análisis	62
4.4. Modelo Entidad Relación	66

Resumen: *A lo largo de este capítulo se hablará sobre los requisitos que definen el proyecto, la especificación de los casos de uso y a continuación se presentará el modelo de dominio. Finalmente se mostrará el modelo entidad relación de las tablas que afectan a los tres componentes creados que son en sí cuerpo del proyecto.*

4.1. Requisitos

En el siguiente apartado se detallan los requisitos que se han encontrado necesarios para el desarrollo y realización del resto del proyecto, acorde con las especificaciones del enunciado del proyecto.

4.1.1. Requisitos de Información

- **RI.1:** El sistema categorizará tres tipos de usuarios:

- Tipo 1: Visitante.
 - Tipo 2: Registrado.
 - Tipo 3: Administrador
- **RI.2:** El sistema deberá coleccionar la siguiente información sobre los usuarios registrados:
- Nombre de usuario
 - Apellido1, Apellido2
 - Contraseña
 - Dirección de correo
- **RI.3:** El sistema deberá coleccionar la siguiente información sobre los artículos:
- | | |
|-------------|---------------------|
| • Bibtexkey | • Zona abstracta |
| • Autor | • Crossref |
| • Título | • Palabras clave |
| • Revista | • DOI |
| • Año | • URL |
| • Volumen | • Citeseerurl |
| • Páginas | • Comentarios |
| • Número | • Veces citado |
| • Mes | • Factor de impacto |
| • EID | • Propietario |
| • Nota | |
- **RI.4:** El sistema deberá coleccionar la siguiente información sobre las tesis doctorales:
- | | |
|------------------|---------------------|
| • Bibtexkey | • Crossref |
| • Autor | • Palabras clave |
| • Título | • DOI |
| • Escuela | • URL |
| • Año | • Citeseerurl |
| • Tipo | • Comentarios |
| • Dirección | • Veces citado |
| • Mes | • Factor de impacto |
| • Nota | • Propietario |
| • Zona abstracta | |
- **RI.5:** El sistema deberá coleccionar la siguiente información sobre los congresos:

- Bibtexkey
- Autor
- Título
- Título del libro
- Año
- Editor
- Volumen
- Número
- Series
- Páginas
- Dirección
- Mes
- Organización
- Editor
- Nota
- Zona abstracta
- Crossref
- Palabras clave
- DOI
- URL
- Citeseerurl
- Comentarios
- Veces citado
- Factor de impacto
- Propietario

- **RI.6:** El sistema deberá coleccionar la siguiente información sobre otras publicaciones:

- Tipo de entrada
- Bibtexkey
- Autor
- Título
- Institución
- Año
- Tipo
- Número
- Dirección
- Mes
- Nota
- Zona abstracta
- Crossref
- Palabras clave
- DOI
- URL
- Citeseerurl
- Comentarios
- Veces citado
- Factor de impacto
- Propietario

4.1.2. Requisitos Funcionales

Autenticación

- **RF.1:** La aplicación implementará un sistema de autenticación y autorización para controlar el acceso de los usuarios a las areas que le corresponden según el nivel de acceso.
- **RF.2:** El sistema reconocerá durante toda la sesión las credenciales del usuario, de manera que pueda controlar a qué secciones o funcionalidades se le da acceso.
- **RF.3:** Los usuarios podrán autenticarse en la aplicación proporcionando su nombre de usuario y su password.

Registro

- **RF.4:** El sistema permitirá registrar nuevos usuarios desde el formulario de registro disponible para cualquier usuario que no se haya registrado previamente.
- **RF.5:** Para poder registrarse, el usuario deberá rellenar el formulario en el que se le pedirán los siguientes datos: nombre, email y contraseña (repetir el email y la contraseña de confirmación).
- **RF.5:** La contraseña se almacenara en la base de datos de forma encriptada para mayor seguridad.
- **RF.6:** La contraseña tiene que tener un mínimo de 4 caracteres.
- **RF.7:** Al formulario de registro se accederá a través de la pantalla de inicio de sesión de usuario.
- **RF.8:** Se necesitará la confirmación del administrador para dar de alta el registro.

Cuenta de Usuario

- **RF.9:** Los usuarios registrados tendrán a un area privada donde podrán modificar su perfil, (contraseña, nombre de usuario, email, idioma predeterminado, editor y zona horaria). Además tendrán acceso a diferentes areas del menú de usuario. Para poder acceder a enviar artículo y enviar enlace tendrá que ser usuario editor.
- **RF.10:** Los usuarios registrados, y por tanto miembros de GOA, podrán insertar artículos de investigación en la base de datos del sistema.

Base de Datos

- **RF.11:** El sistema permitirá introducir artículos en la base de datos.
- **RF.12:** El sistema permitirá introducir tesis doctorales en la base de datos.
- **RF.13:** El sistema permitirá introducir congresos en la base de datos.
- **RF.14:** El sistema permitirá introducir otras publicaciones en la base de datos.
- **RF.15:** El sistema permitirá introducir toda clase de artículos introduciendo el texto en formato bibtex.
- **RF.16:** El sistema permitirá introducir toda clase de artículos en archivos en el formato estandar de bibtex con extensión “.bib”.
- **RF.17:** El sistema guardará la información de los artículos introduciendo el DOI de los mismos, buscando la información en la base de datos del WOK e importándola a la base de datos propia del sistema .
- **RF.18:** El sistema permitirá relacionar seudónimos con los autores correspondientes o descartar los mismos.

- **RF.19:** El sistema mostrará todos los artículos que estén guardados en la base de datos.
- **RF.20:** El sistema permitirá filtrar las búsquedas para mostrar los artículos de la base de datos.
- **RF.21:** El sistema permitirá seleccionar mostrar los artículos en formato Bibtex, Txt o Cicyt si el usuario ha iniciado sesión en el sistema.
- **RF.22:** El sistema permitirá editar, ocultar o borrar un artículo si el usuario es el que creó el artículo o es el administrador
- **RF.23:** El sistema permitirá desocultar un artículo si el usuario es el administrador sino no se mostrará.

Frontend

- **RF.24:** El sistema permitirá cambiar el idioma del sitio de español a inglés y viceversa.
- **RF.25:** El sistema permitirá navegar a través de los diferentes elementos del menú.

4.1.3. Requisitos no Funcionales

Usabilidad

- **RNF.1:** El sistema deberá ser usable: que los usuarios no noten mucho cambio con la anterior versión para que se adapten rápido a la nueva interfaz, siendo limpia, clara pudiendo realizarse todas sus funciones con efectividad y eficiencia.
- **RNF.2:** La organización de los contenidos en el sitio web deberá ser coherente, sin cambiar el orden de los menús para que el usuario le resulte fácil encontrar lo que busque.
- **RNF.3:** El sistema mostrará los diferentes errores de la mejor manera posible para que se puedan subsanar.

Seguridad y Rendimiento

- **RNF.4:** El sistema ofrecerá mecanismos de seguridad para la autenticación de los usuarios.
- **RNF.5:** El acceso a la base de datos deberá estar protegido por una contraseña.
- **RNF.6:** El acceso al backend deberá estar protegido por una contraseña.
- **RNF.7:** El sistema tendrá la funcionalidad de hacer backups, para poder restaurar el sistema en caso de ataque externo o fallo.
- **RNF.8:** El sistema tendrá que responder en un tiempo razonablemente pequeño ya que la cantidad de información que se maneja en la base de datos es pequeña.
- **RNF.9:** El sistema deberá encriptar las contraseñas de los usuarios registrados en md5.

Diseño, Implementación y Disponibilidad

- **RNF.10:** El sistema ofrecerá una interfaz de usuario basada en HTML-CSS ya que será en entorno Web.
- **RNF.11:** El sistema gestionará la aplicación Web utilizando la tecnología PHP.
- **RNF.12:** El sistema gestionará el acceso a los datos con el gestor de bases de datos MySQL.
- **RNF.13:** El sistema deberá visualizarse en todos los navegadores, aunque correctamente en navegadores Web Internet Explorer, Mozilla Firefox y Google Chrome.
- **RNF.14:** El sistema será soportado en un servidor bajo una versión de Apache.
- **RNF.15:** El sistema estará diseñado bajo una arquitectura basada en MVC (modelo-vista-controlador).
- **RNF.16:** La conexión con el servidor se establecerá a través de un protocolo de comunicación TCP-IP.
- **RNF.17:** El sistema sera responsivo para acceder desde smartphones y tablets.
- **RNF.18:** El sistema tiene que funcionar las 24 horas, los 7 días a la semana.
- **RNF.18:** La funcionalidad de rellenar autores lo pueden usar todos los usuarios registrados ya que es tarea interna del GOA.

4.2. Casos de Uso

4.2.1. Diagrama de Casos de Uso

¹ A continuación, se muestra el diagrama de casos de uso elaborado acorde con los requisitos descritos en el punto anterior (ver 4.1).

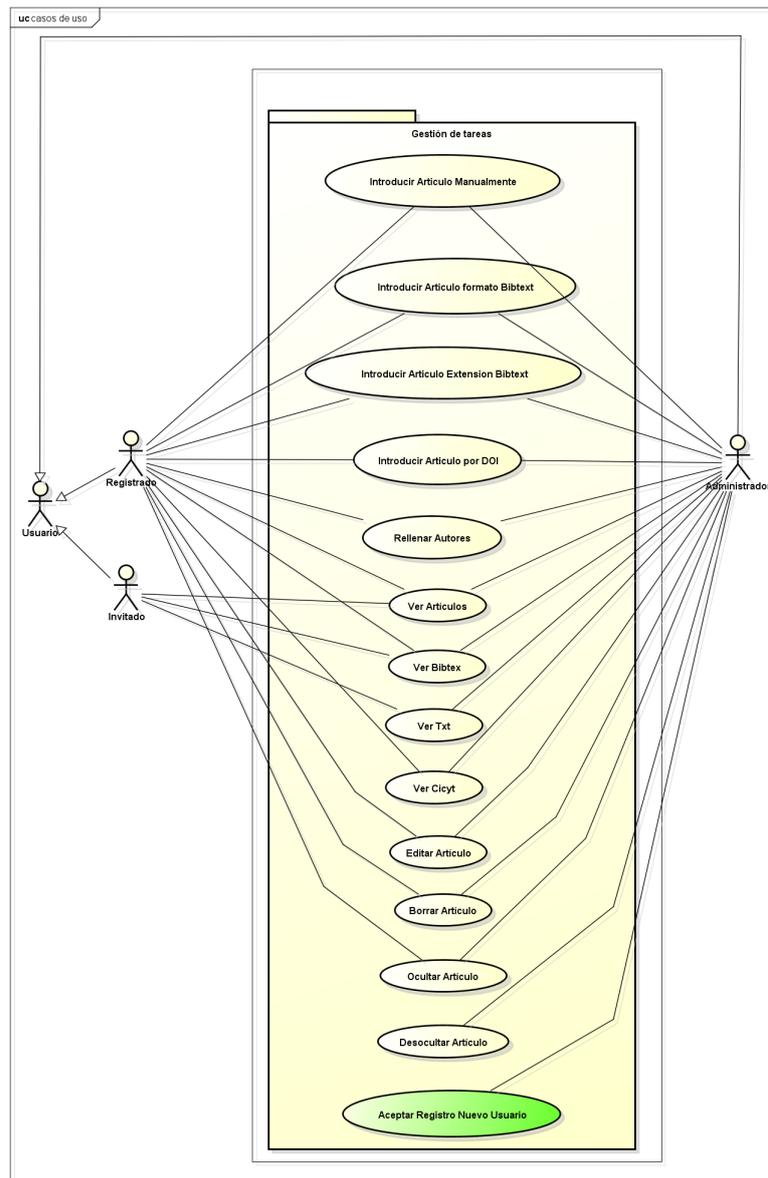


Figura 4.1: Diagrama de Casos de Uso.

¹Junto a este documento, se adjunta un fichero comprimido con las imágenes del diagrama de casos de uso, el modelo de dominio, y los diferentes diagramas de secuencia para apreciar mejor los detalles de cada uno.

4.2.2. Actores

ACT-0001	Visitante
Versión	2.0 (21/04/2014)
Autores	Cárcel Fernando, Sergio
Fuentes	Cárcel Fernando, Sergio
Descripción	Este actor representa <i>al usuario no registrado que entra a la página.</i>
Comentarios	Requisitos: RI.1

ACT-0002	Registrado
Versión	2.0 (21/04/2014)
Autores	Cárcel Fernando, Sergio
Fuentes	Cárcel Fernando, Sergio
Descripción	Este actor representa <i>al grupo de que usuarios que se ha registrado en el sistema y a sido aceptado.</i>
Comentarios	Requisitos: RI.1

ACT-0003	Administrador
Versión	2.0 (21/04/2014)
Autores	Cárcel Fernando, Sergio
Fuentes	Cárcel Fernando, Sergio
Descripción	Este actor representa <i>al usuario que puede modificar el sistema y dar de alta a usuarios nuevos, ya sea registrados o editores.</i>
Comentarios	Requisitos: RI.1

4.2.3. Descripción de los Casos de Uso

Introducir Artículo Manualmente

UC-0001	Introducir Artículo Manualmente
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados deseen introducir un artículo en el sistema ya sea del tipo artículo, tesis doctoral, congresos u otras publicaciones rellenando los campos de forma manual.</i>
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema.

Secuencia normal	Paso	Acción
	1	El actor Registrado (ACT-0002) <i>envía los campos requeridos para poder introducir en la base de datos el artículo, tesis, congreso u otra publicación.</i>
	2	El sistema <i>guarda el artículo en la base de datos y muestra un mensaje de artículo guardado.</i>
Postcondición	El artículo se guardo correctamente.	
Excepciones	Paso	Acción
	2	Si el sistema no guarda correctamente el artículo en la base de datos o ese artículo ya esta introducido en la base de datos, muestra un mensaje de error, y a continuación, este caso de uso vuelve al paso 1.
Comentarios	Requisitos: RF.10,RF.11,RF.12,RF.13	

Introducir Artículo Formato Bibtex

UC-0002	Introducir Artículo Formato Bibtext	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados deseen introducir un artículo en el sistema ya sea del tipo artículo, tesis doctoral, congresos u otras publicaciones relleno la caja de texto con el formato Bibtex.</i>	
Precondición	El actor Registrado (ACT-0002) o cualquiera de los otros dos citados anteriormente ha iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor Registrado (ACT-0002) <i>escribe en formato Bibtex el artículo, tesis, congreso u otra publicación.</i>
	2	El sistema <i>guarda el artículo en la base de datos y muestra un mensaje de artículo guardado.</i>
Postcondición	El artículo se guardo correctamente.	

Excepciones	Paso	Acción
	2	Si el sistema no guarda correctamente el artículo en la base de datos o ese artículo ya está introducido en la base de datos o por última instancia el formato Bibtex tiene algún error o errata, muestra un mensaje de error, y a continuación, este caso de uso vuelve al paso 1.
Comentarios	Requisitos: RF.10,RF.15	

Introducir Artículo Extensión Bibtex

UC-0003	Introducir Artículo extensión Bibtext	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados deseen introducir un artículo en el sistema ya sea del tipo artículo, tesis doctoral, congresos u otras publicaciones subiendo un archivo con extensión .bib</i> .	
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor Registrado (ACT-0002) <i>selecciona y sube un archivo con formato Bibtex, es decir ".bib", que contiene un artículo, tesis, congreso u otra publicación.</i>
	2	El sistema <i>guarda el artículo en la base de datos y muestra un mensaje de artículo guardado.</i>
Postcondición	El artículo se guardó correctamente.	
Excepciones	Paso	Acción
	2	Si el sistema no guarda correctamente el artículo en la base de datos o ese artículo ya está introducido en la base de datos o por última instancia el archivo ".bib" está dañado o contiene algún error, muestra un mensaje de error, y a continuación, este caso de uso vuelve al paso 1.
Comentarios	Requisitos: RF.10,RF.16	

Introducir Artículo Mediante DOI

UC-0004	Introducir Artículo extensión Bibtext	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados deseen introducir en el sistema ya sea un artículo, tesis doctoral, congresos u otras publicaciones mediante el DOI (identificador único)</i> .	
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor Registrado (ACT-0002) <i>introduce el DOI del artículo, tesis doctoral, congreso u otra publicación que quiere guardar en el sistema.</i>
	2	El sistema <i>busca con ese DOI en la base de datos del WOK, extrae la información, la guarda en la base de datos y muestra un mensaje de artículo guardado.</i>
Postcondición	El artículo se guardo correctamente.	
Excepciones	Paso	Acción
	2	Si el sistema no guarda correctamente el artículo en la base de datos o ese artículo ya esta introducido en la base de datos, puede que el DOI introducido sea incorrecto o en última instancia no lo encuentra en la base de datos del WOS, entonces muestra un mensaje de error, y a continuación, este caso de uso vuelve al paso 1.
Comentarios	Requisitos: RF.10,RF.17	

Rellenar Autores

UC-0005	Rellenar Autores	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados quiera enlazar el nombre o nombres de autores que ha introducido algún usuario al añadir un artículo con el nombre exacto que tiene ese autor en la base de datos para que no haya duplicidad y tenga concordancia.</i>	
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema <i>muestra todos los seudónimos pendientes de relacionar con los autores introducidos en la base de datos.</i>
	2	El actor Registrado (ACT-0002) o el administrador <i>selecciona el seudónimo con el nombre original si existe de entre todos los disponibles de la base de datos y le da a actualizar.</i>
	3	El sistema <i>relaciona el seudónimo con el nombre del autor que se le ha indicado, lo guarda en la base de datos y devuelve un mensaje de confirmación.</i>
Postcondición	El artículo se guardo correctamente.	
Excepciones	Paso	Acción
	2	Si el actor Registrado (ACT-0002) tiene seleccionada la casilla de ignorar este seudónimo se descarta y no se relaciona con ningún autor de la base de datos, y el caso de uso vuelve al paso 1 .
	3	Si el sistema no guarda correctamente al autor que corresponde con el seudónimo o hay un error, este caso de uso vuelve al paso 1.
Comentarios	Requisitos: RF.18	

Ver Artículos

UC-0006	Ver Artículos	
Dependencias	Actor Invitado (ACT-0001) Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados quiera ver los artículos guardados en la base de datos, ya sean artículos, tesis doctorales, congresos u otras publicaciones.</i>	
Secuencia normal	Paso	Acción
	1	El sistema <i>carga en forma de lista todos los artículos de la base de datos, sean del tipo que sean.</i>
	2	El actor, <i>sea del tipo que sea filtra la búsqueda por autor y año.</i>
	3	El sistema <i>hace la consulta con el filtro del actor y devuelve la lista con los artículos que cumplen esas condiciones.</i>
Postcondición	Los artículos se mostraron correctamente.	
Excepciones	Paso	Acción
	1	Si el sistema no puede acceder a la base de datos, muestra un mensaje de error y termina el caso de uso.
Comentarios	Requisitos: RF.19,RF.20	

Ver Bibtex

UC-0007	Ver Bibtex	
Dependencias	Actor Invitado (ACT-0001) Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados pulse el icono bibtex o el enlace para ver los artículos guardados en la base de datos, ya sean artículos, tesis doctorales, congresos u otras publicaciones, en formato bibtex.</i>	
Secuencia normal	Paso	Acción
	1	El actor, <i>sea del tipo que sea pulsa el icono de bibtex.</i>
	2	El sistema <i>carga en forma de lista todos los artículos de la base de datos, sean del tipo que sean con formato bibtex.</i>
Postcondición	Los artículos se mostraron correctamente.	
Excepciones	Paso	Acción
	1	Si el sistema no puede acceder a la base de datos, muestra un mensaje de error y termina el caso de uso.
	1.1	El usuario pulsa en el enlace bibtex debajo de un artículo en concreto.
	1.2	El sistema <i>carga ese artículo en formato bibtex.</i>
	1.3	El sistema <i>comprueba que el usuario sea el que guardo ese artículo en la base de datos o el administrador y le muestra las opciones de editar, ocultar y borrar.</i>
Comentarios	Requisitos: RF.21	

Ver Txt

UC-0008	Ver Txt	
Dependencias	Actor Invitado (ACT-0001) Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>cualquiera de los usuarios indicados pulse el icono txt para ver los artículos guardados en la base de datos, ya sean artículos, tesis doctorales, congresos u otras publicaciones , en formato txt.</i>	
Secuencia normal	Paso	Acción
	1	El actor, <i>sea del tipo que sea pulsa el icono de txt.</i>
	2	El sistema <i>carga en forma de lista todos los artículos de la base de datos, sean del tipo que sean con formato txt.</i>
Postcondición	Los artículos se mostraron correctamente.	
Excepciones	Paso	Acción
	1	Si el sistema no puede acceder a la base de datos, muestra un mensaje de error y termina el caso de uso.
Comentarios	Requisitos: RF.21	

Ver Cicyt

UC-0009	Ver Cicyt	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario Registrado (ACT-0002) o el administrador pulse el icono cicyt para ver los artículos guardados en la base de datos, ya sean artículos, tesis doctorales, congresos u otras publicaciones, en ese formato.</i>	
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema <i>comprueba que el usuario ha iniciado sesión y muestra el icono de cicyt.</i>
	2	El actor Registrado (ACT-0002) o el administrador pulsa el icono de cicyt.
	3	El sistema <i>carga en forma de lista todos los artículos de la base de datos, sean del tipo que sean en formato cicyt.</i>
Postcondición	Los artículos se mostraron correctamente.	
Excepciones	Paso	Acción
	3	Si el sistema no puede acceder a la base de datos, muestra un mensaje de error y termina el caso de uso.
Comentarios	Requisitos: RF.21	

Editar Artículo

UC-0010	Editar Artículo	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario Registrado (ACT-0002) o el administrador pulse el enlace de editar artículo para editar el artículo que se halla seleccionado al estar viendo el bibtex.</i>	
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema y en el caso del actor Registrado (ACT-0002) es el que introdujo ese artículo en la base de datos.	
Secuencia normal	Paso	Acción
	1	El sistema <i>comprueba que el usuario ha iniciado sesión y sea el que introdujo el artículo en la base de datos o sea el administrador y muestra el enlace de editar artículo.</i>
	2	El actor Registrado (ACT-0002) o el administrador pulsa el enlace editar artículo.
	3	El sistema <i>carga el formulario con los campos que están actualmente guardados para poder cambiarlos o añadir otros campos vacíos.</i>
	4	El actor Registrado (ACT-0002) o el administrador rellena los campos que quiera modificar o añadir y pulsa enviar
5	El sistema <i>actualiza la información del artículo en la base de datos.</i>	
Postcondición	El artículo se actualizó correctamente.	
Excepciones	Paso	Acción
	3	Si el sistema no puede acceder a la base de datos, muestra un mensaje de error y termina el caso de uso.
5	Si el sistema no puede guardar los datos en la base de datos, muestra un mensaje de error y vuelve al caso de uso 4.	
Comentarios	Requisitos: RF.22	

Borrar Artículo

UC-0011	Borrar Artículo	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario Registrado (ACT-0002) o el administrador pulse el enlace de borrar artículo para borrar el artículo que se halla seleccionado al estar viendo el bibtex.</i>	
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema y en el caso del actor Registrado (ACT-0002) es el que introdujo ese artículo en la base de datos.	
Secuencia normal	Paso	Acción
	1	El sistema <i>comprueba que el usuario ha iniciado sesión y sea el que introdujo el artículo en la base de datos o sea el administrador y muestra el enlace de borrar artículo.</i>
	2	El actor Registrado (ACT-0002) o el administrador pulsa el enlace borrar artículo.
	3	El sistema <i>muestra un mensaje de confirmación para borrar el artículo.</i>
	4	El actor Registrado (ACT-0002) o el administrador acepta ese mensaje de confirmación para borrar el artículo
	5	El sistema <i>borra el artículo de la base de datos y muestra un mensaje de artículo borrado.</i>
Postcondición	El artículo se borró correctamente.	
Excepciones	Paso	Acción
	5	Si el sistema no puede borrar los datos en la base de datos, muestra un mensaje de error y caso de uso termina.
Comentarios	Requisitos: RF.22	

Ocultar Artículo

UC-0012	Ocultar Artículo	
Dependencias	Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario Registrado (ACT-0002) o el administrador pulse el enlace de ocultar artículo para ocultar el artículo que se halla seleccionado al estar viendo el bibtex.</i>	
Precondición	El actor Registrado (ACT-0002) o el administrador ha iniciado sesión en el sistema y en el caso del actor Registrado (ACT-0002) es el que introdujo ese artículo en la base de datos.	
Secuencia normal	Paso	Acción
	1	El sistema <i>comprueba que el usuario ha iniciado sesión y sea el que introdujo el artículo en la base de datos o sea el administrador y muestra el enlace de ocultar artículo.</i>
	2	El actor Registrado (ACT-0002) o el administrador pulsa el enlace ocultar artículo.
	3	El sistema <i>muestra un mensaje de confirmación para ocultar el artículo.</i>
	4	El actor Registrado (ACT-0002) o el administrador acepta ese mensaje de confirmación para ocultar el artículo
5	El sistema <i>oculta el artículo en la página y muestra un mensaje de artículo oculto.</i>	
Postcondición	El artículo se ocultó correctamente.	
Excepciones	Paso	Acción
	5	Si el sistema no puede ocultar el artículo seleccionado, muestra un mensaje de error y caso de uso termina.
Comentarios	Requisitos: RF.22	

Desocultar Artículo

UC-0013	Desocultar Artículo	
Dependencias	Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario Administrador (ACT-0003) pulse el enlace de desocultar artículo para mostrar el artículo que se halla seleccionado al estar viendo el bibtex.</i>	
Precondición	El actor Administrador (ACT-0003) ha iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El sistema <i>comprueba que el usuario ha iniciado sesión y sea el administrador y muestra el enlace de desocultar artículo.</i>
	2	El actor Administrador (ACT-0003) pulsa el enlace desocultar artículo.
	3	El sistema <i>muestra un mensaje de confirmación para desocultar el artículo.</i>
	4	El actor Administrador (ACT-0003) acepta ese mensaje de confirmación para desocultar el artículo
5	El sistema <i>desoculta el artículo en la página y muestra un mensaje de artículo desocultado correctamente.</i>	
Postcondición	El artículo se desocultó correctamente.	
Excepciones	Paso	Acción
	5	Si el sistema no puede desocultar el artículo seleccionado, muestra un mensaje de error y caso de uso termina.
Comentarios	Requisitos: RF.23	

Aceptar Registro Nuevo Usuario

UC-0013	Aceptar Registro Nuevo Usuario	
Dependencias	Actor Invitado (ACT-0001) Actor Registrado (ACT-0002) Actor Administrador (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el actor Invitado (ACT-0001) se ha registrado y validado su email y el administrador tiene que aceptar a este usuario para que sea actor Registrado.</i>	
Precondición	El actor Invitado (ACT-0001) se registró correctamente y validó desde su correo que ese era su email correcto.	
Secuencia normal	Paso	Acción
	1	El sistema <i>recibe la confirmación de validación de email por parte del usuario Invitado y envía un email de confirmación al Administrador.</i>
	2	El actor Administrador (ACT-0003) pulsa en el enlace de validación final de usuario .
3	El sistema <i>valida al actor Invitado (ACT-0001) y pasa a ser un actor Registrado (ACT-0002) del sistema al cual podrá acceder con su usuario y contraseña, y envía un email de que ya está registrado en el sistema.</i>	
Postcondición	El usuario se introdujo correctamente en el sistema.	
Excepciones	Paso	Acción
	5	Si el sistema no puede registrar al usuario, muestra un mensaje de error y el caso de uso termina.
Comentarios	Requisitos: RF.8	

4.2.4. Escenarios

Introducir Artículo Manualmente: Juan es un miembro ya registrado e iniciada sesión en el sistema, se dirige al elemento del menú de usuario llamado “Introducir Artículo”. En la pestaña “Formulario”, selecciona una de las 4 subpestañas de “Artículo”, “Tesis doctoral”, “Congresos” u “Otras Publicaciones”, rellena el formulario con los campos que crea conveniente y por último pulsa “Enviar” para que el sistema lo introduzca en la base de datos.

Introducir Artículo Formato Bibtext: Juan es un miembro ya registrado e iniciada sesión en el sistema, se dirige al elemento del menú de usuario llamado “Introducir Artículo”. En la pestaña “Bibtext” rellena una caja de texto escribiendo el artículo, tesis, congreso u otra publicación, pero siempre en el formato Bibtext correspondiente, y por último pulsa “Añadir” para que el sistema lo introduzca en la base de datos.

Introducir Artículo Extension Bibtext: Juan es un miembro ya registrado e iniciada sesión en el sistema, se dirige al elemento del menú de usuario llamado “Introducir Artículo”. En la pestaña “Archivo” pulsa el botón “Seleccionar Archivo” y elige el archivo con extensión “.bib” que contiene varios artículos en formato Bibtext y por último pulsa “Enviar” para que el sistema lo introduzca en la base de datos.

Introducir Artículo por DOI: Juan es un miembro ya registrado e iniciada sesión en el sistema, se dirige al elemento del menú de usuario llamado “Introducir Artículo”. En la pestaña “DOI” introduce el DOI, que es un identificador único del artículo y por último pulsa “Añadir” para que el sistema busque en la base de datos del WOK el artículo, extraiga esa información y la introduzca en la base de datos del sistema.

Rellenar Autores: Daniel es un usuario registrado y una vez iniciada sesión en el sistema, sabe que es tarea interna de los usuarios enlazar los seudónimos de los autores con los autores reales introducidos en la base de datos, por lo que va dentro del menú de usuario al elemento rellenar autores, selecciona ese elemento de menú y se le despliega una lista con todos los seudónimos junto a un desplegable al lado de cada uno de ellos para relacionar ese seudónimo con el nombre por el que los autores de artículos están guardados en el sistema. Una vez elegido el que conozca le da actualizar y se relacionará en la base de datos.

Si por el contrario ve que hay uno que no corresponde con ninguno, pulsará la casilla de ignorar, actualizará, y ese seudónimo será descartado y desaparecerá de la lista de rellenar autores.

Ver Artículos: Jose es un usuario que le interesa saber el trabajo que hace el GOA y lo que publica, por lo que entra en la página web y se dirige al elemento de menú publicaciones. El sistema carga una lista con todos los artículos, tesis, congresos y otras publicaciones. Jose podrá filtrar las búsquedas por año, autor y seleccionar si solo quiere que de ese autor se le muestren los artículos que esta como primer autor.

Ver Bibtext: Jose es un usuario que le interesa saber que trabajo hace el GOA y que publica, y dentro de la pestaña publicaciones que ya estaba decide ver el Bibtext de los artículos, por lo que pincha en el icono de “Bibtext” y el sistema carga una lista con todos los artículos, sean del tipo que sean, con formato Bibtext. Luego decide filtrar más la búsqueda por lo que vuelve a publicaciones y allí quiere ver solo el Bibtext de un artículo en concreto. Pincha en el enlace “Bibtext” situado debajo de un artículo y le muestra solo el Bibtext de ese artículo.

Ver Txt: Jose es un usuario que le interesa saber que trabajo hace el GOA y que publica, y dentro de la pestaña publicaciones que ya estaba decide ver el Txt de los artículos, por lo que pincha en el icono de “Txt” y el sistema carga una lista con todos los artículos, sean del tipo que sean, con formato Txt.

Ver Cicyt: Daniel es un usuario registrado del sistema, inicia sesión en la página web y va a la pestaña publicaciones entonces el sistema el icono oculto de “Cicyt” pincha en el y le muestra todos los artículos del la base de datos, sean del tipo que sean en forma de lista y en ese formato.

Editar Artículo: Daniel es un usuario registrado del sistema, inicia sesión en la página web y va a la pestaña publicaciones, el sistema que ha cargado todos los artículos de la base de datos, él busca uno que había introducido el otro día en la base de datos y le faltaba rellenar un campo, por lo que pincha en el enlace “Bibtex” situado en la parte inferior de su artículo y al mostrarle en ese formato el artículo, el sistema le carga el botón de editar. Pincha en el y el sistema le muestra un formulario con todos los campos que puede tener un artículo y los que ya estaban rellenos, tienen esa información pero puede editarse. Daniel rellena lo que le interesa y cambia lo que cree correcto y le da a enviar. El artículo entonces será actualizado.

Borrar Artículo: Daniel es un usuario registrado del sistema, inicia sesión en la página web y va a la pestaña publicaciones, el sistema que ha cargado todos los artículos de la base de datos, él busca uno que había introducido el otro día en la base de datos y que quiere eliminar, por lo que pincha en el enlace “Bibtex” situado en la parte inferior de su artículo y al mostrarle en ese formato el artículo, el sistema le carga el botón borrar. Pincha en el y el sistema le pide una confirmación para que esté seguro de querer borrarlo. Daniel acepta la confirmación y el artículo es borrado.

Ocultar Artículo: Daniel es un usuario registrado del sistema, inicia sesión en la página web y va a la pestaña publicaciones, el sistema que ha cargado todos los artículos de la base de datos, él busca uno que había introducido el otro día en la base de datos y que quiere ocultarlo de momento, por lo que pincha en el enlace “Bibtex” situado en la parte inferior de su artículo y al mostrarle en ese formato el artículo, el sistema le carga el botón ocultar. Pincha en el y el sistema le pide una confirmación para que esté seguro de querer ocultarlo. Daniel acepta la confirmación y el artículo es ocultado.

Desocultar Artículo: Marcos es un usuario administrador del sistema, inicia sesión en la página web y va a la pestaña publicaciones, el sistema que ha cargado todos los artículos de la base de datos, él busca uno que había introducido el otro día un usuario registrado y había ocultado y que quiere desocultarlo, por lo que pincha en el enlace “Bibtex” situado en la parte inferior de su artículo y al mostrarle en ese formato el artículo, el sistema le carga el botón desocultar. Pincha en el y el sistema le pide una confirmación para que esté seguro de querer desocultarlo. Marcos acepta la confirmación y el artículo es desoculta correctamente.

Aceptar Registro Nuevo Usuario: Marcos es un usuario administrador del sistema, ha recibido un email para confirmar el registro de un usuario comprueba que el usuario es parte del GOA y pincha en el link, por lo que el usuario pasará desde ese momento a ser usuario registrado del sistema.

4.3. Modelo de Dominio

A continuación, se muestra en el siguiente diagrama (ver figura 4.2), el modelo de dominio del sistema que se va a implementar.

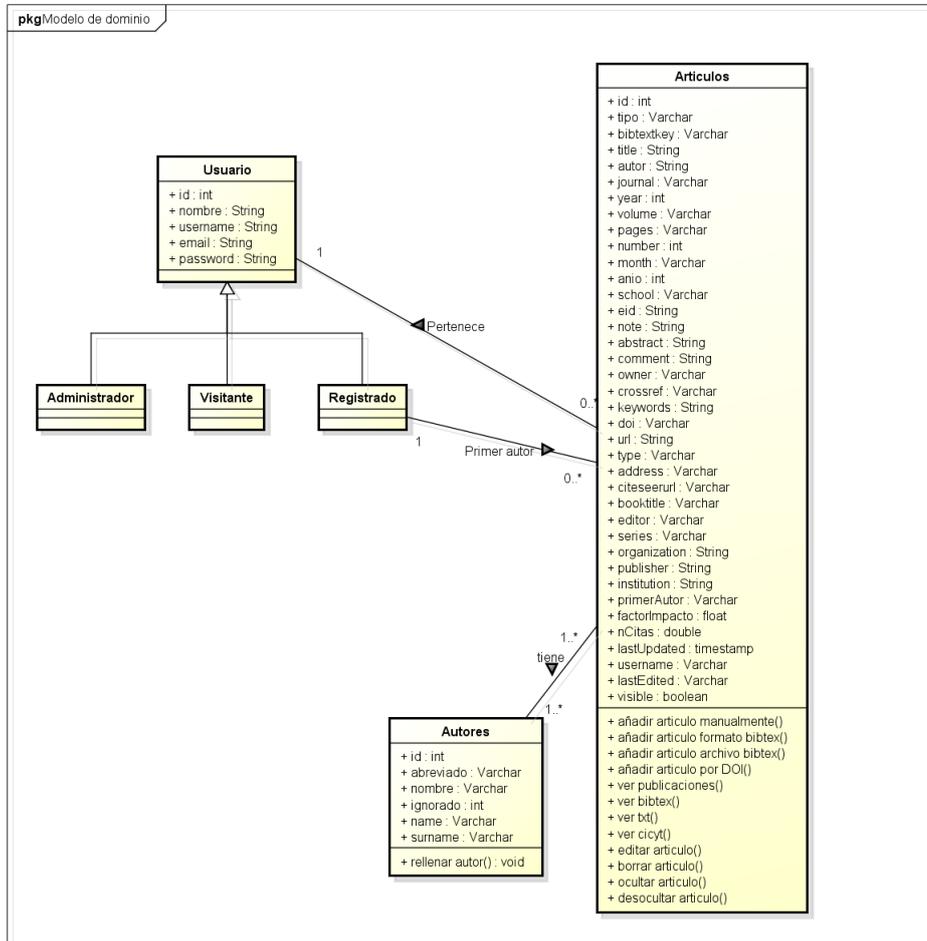


Figura 4.2: Modelo de dominio del sistema.

4.3.1. Descripción de las Clases de Análisis

	Artículos
Responsabilidades	Esta clase se encargará de guardar las publicaciones en la base de datos, así como la edición, el borrado y la lectura de las mismas.
Atributos	<p>id: Identificación única de esa publicación en la base de datos.</p> <p>tipo: Tipo de publicación.</p> <p>bibtexkey: Otra identificación única de esa publicación.</p> <p>title: Título de la publicación.</p>

	<p>autor: Autor de la publicación.</p> <p>journal: Revista publicación.</p> <p>year: Año de publicación.</p> <p>volume: Volumen en el que se encuentra.</p> <p>pages: Páginas de la publicación.</p> <p>number: Numero de la publicación.</p> <p>month: Mes de la publicación.</p> <p>school: Escuela donde se ha publicado.</p> <p>eid: Otro identificador para la publicación.</p> <p>note: Nota informativa de la publicación.</p> <p>abstract: Algo que añadir sobre la publicación.</p> <p>comment: Un comentario sobre el artículo.</p> <p>owner: Propietario.</p> <p>crossref: Código de referencia.</p> <p>keywords: Palabras clave.</p> <p>doi: Identificación única de esa publicación.</p> <p>url: Dirección web.</p> <p>address: Dirección.</p> <p>citeseurl: URLs donde esta citado el artículo.</p> <p>booktitle: Título del libro donde está publicado.</p> <p>editor: Nombre del editor.</p> <p>series: Número de serie que tiene la publicación.</p> <p>organization: Organización si la hay a la que pertenece esa publicación.</p> <p>publisher: Empresa que publica el artículo.</p> <p>institution: Institución desde la cual se hizo el artículo.</p> <p>primerAutor: Primer autor de la publicación.</p> <p>factorImpacto: Factor representativo de un artículo sobre una revista.</p> <p>nCitas: Número de citas en revistas del artículo.</p> <p>lastUpdated: Fecha ultima actualización.</p> <p>userName: Nombre del usuario que lo publicó.</p> <p>lastEdited: Fecha última edición.</p> <p>visible: Si el articulo esta visible u oculto.</p>
Operaciones	<p>Añadir artículo manualmente(): Guarda en la base de datos la publicación introducida rellenando el formulario, ya sea del tipo artículo, tesis, congreso u otra publicación.</p> <p>Añadir artículo formato bibtex(): Guarda en la base de datos la publicación introducida en formato bibtex ya sea del tipo artículo, tesis, congreso u otra publicación.</p>

Añadir artículo archivo bibtex(): Guarda en la base de datos las publicaciones que estén dentro del archivo bibtex con extensión .bib que se sube a la página, ya sean del tipo artículo, tesis, congreso u otra publicación.

Añadir artículo por DOI(): Guarda en la base de datos la publicación que introduciendo el DOI la buscará en la base de datos del WOK y la guardará en nuestra base de datos, ya sea del tipo artículo, tesis, congreso u otra publicación.

Ver Artículos(): Carga de la base de datos todas las publicaciones, sean del tipo que sea y las muestra al usuario.

Ver Bibtex(): Carga de la base de datos todas las publicaciones, en formato bibtex, o se puede mostrar solo el de una publicación concreta y se las muestra al usuario.

Ver Txt(): Carga de la base de datos todas las publicaciones, en formato txt y las muestra al usuario.

Ver Cicyt(): Si el usuario está registrado e iniciado sesión, carga de la base de datos todas las publicaciones, en formato Cicyt y las muestra a ese usuario.

Editar Artículo(): Permite editar una publicación introducida al administrador o al usuario que lo publicó.

Borrar Artículo(): Permite borrar una publicación introducida al administrador o al usuario que lo publicó.

Ocultar Artículo(): Permite ocultar una publicación introducida al administrador o al usuario que lo publicó.

Desocultar Artículo(): Permite desocultar una publicación oculta al administrador.

Autores	
Responsabilidades	Esta clase se encargará de guardar toda la información relacionada con los autores, así como relacionarlos con los que son miembros del GOA.
Atributos	<p>id: Identificación única del autor en la base de datos.</p> <p>abreviado: Nombre con iniciales y apellido.</p> <p>nombre: Nombre del autor completo.</p> <p>ignorado: Si el autor este no corresponde con ninguno o si.</p> <p>name: Iniciales del nombre.</p> <p>surname: Apellido del autor.</p>
Operaciones	Rellenar Autor(): Relaciona un autor introducido con el nombre que le corresponde exactamente en la base de datos con un miembro de GOA.

Usuarios	
Responsabilidades	Esta clase se encargará de guardar toda la información relacionada con los autores, así como relacionarlos con los que son miembros del GOA.
Atributos	<p>id: Identificación única del usuario.</p> <p>nombre: Nombre del usuario.</p> <p>username: Nombre con el que el usuario se identifica en la página.</p> <p>email: Email del usuario.</p> <p>password: Contraseña de acceso del usuario.</p>

4.4. Modelo Entidad Relación

Un modelo entidad relación es una colección de conceptos que se emplean para describir la estructura de una base de datos. Con este modelo, (ver [figura 4.3](#)), vamos a modelar la información que la aplicación web debe manipular.

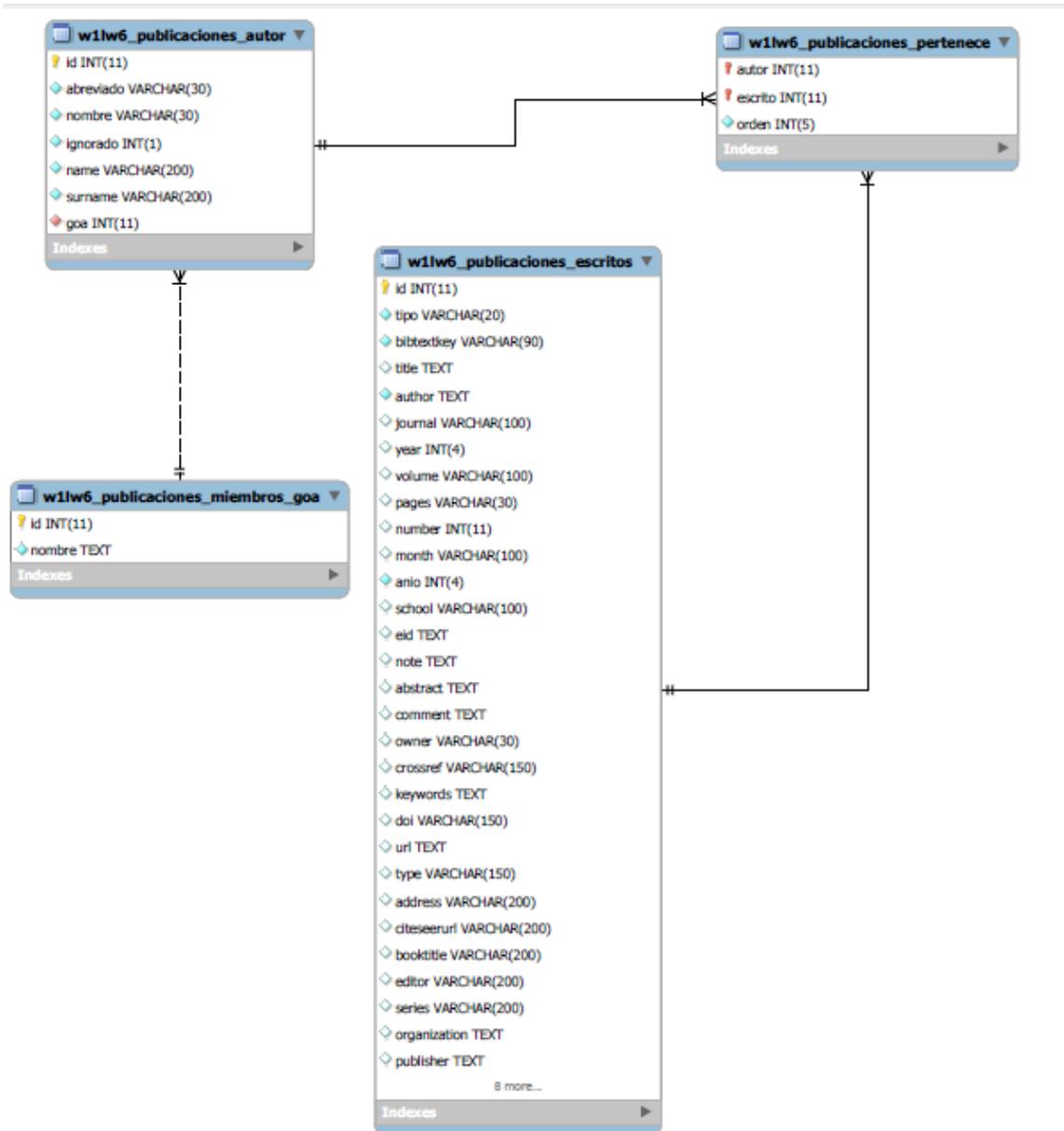


Figura 4.3: Modelo Entidad Relación del sistema.

Capítulo 5

Diseño de la Aplicación

Contenido

5.1. Diagrama de Despliegue	68
5.2. Diagrama de Clases de Diseño	68
5.2.1. Componente Com_bibtex	69
5.2.2. Componente Com_editarAutores	70
5.2.3. Componente Com_verArticulos	71
5.3. Diagramas de Secuencia	72
5.3.1. Introducir Artículo Manualmente	73
5.3.2. Introducir Artículo por DOI	74
5.3.3. Rellenar Autores	75
5.3.4. Ver Artículos	76
5.3.5. Ver Bibtex	77
5.3.6. Ver Txt	78
5.3.7. Ver Cicyt	79
5.3.8. Editar Bibtex	80

Resumen: *Este capítulo no muestra todo lo referente a la arquitectura de Joomla! en si, sino de los componentes creados para Joomla!, su descomposición en subsistemas para comprender en mayor profundidad el sistema creado.*

5.1. Diagrama de Despliegue

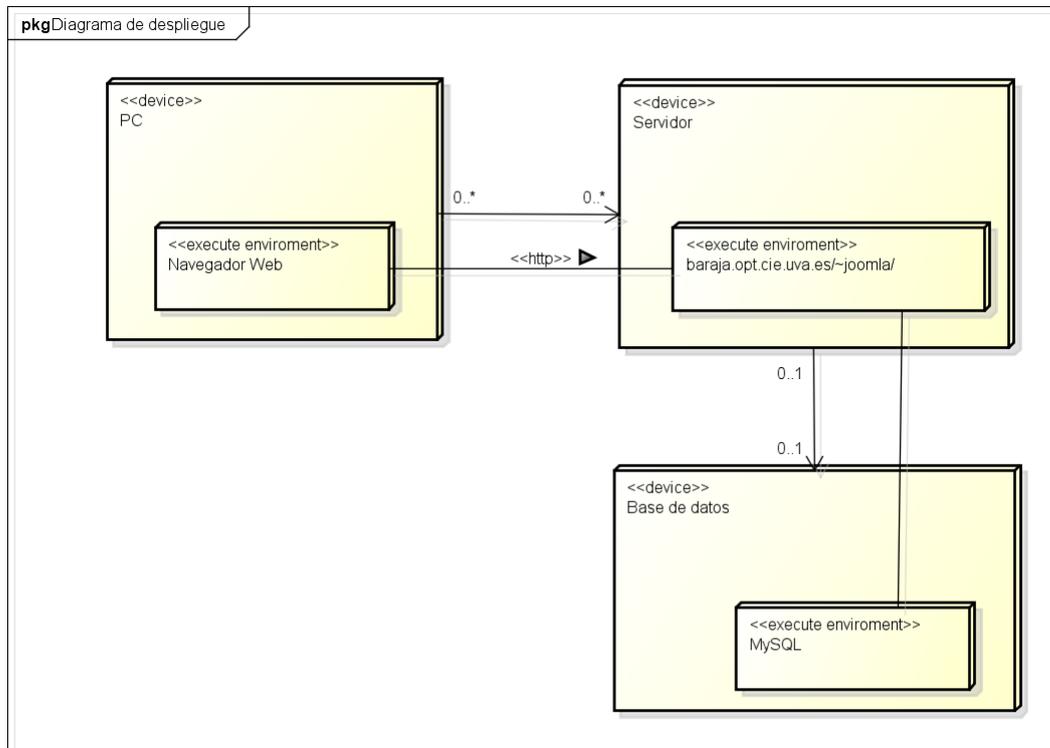
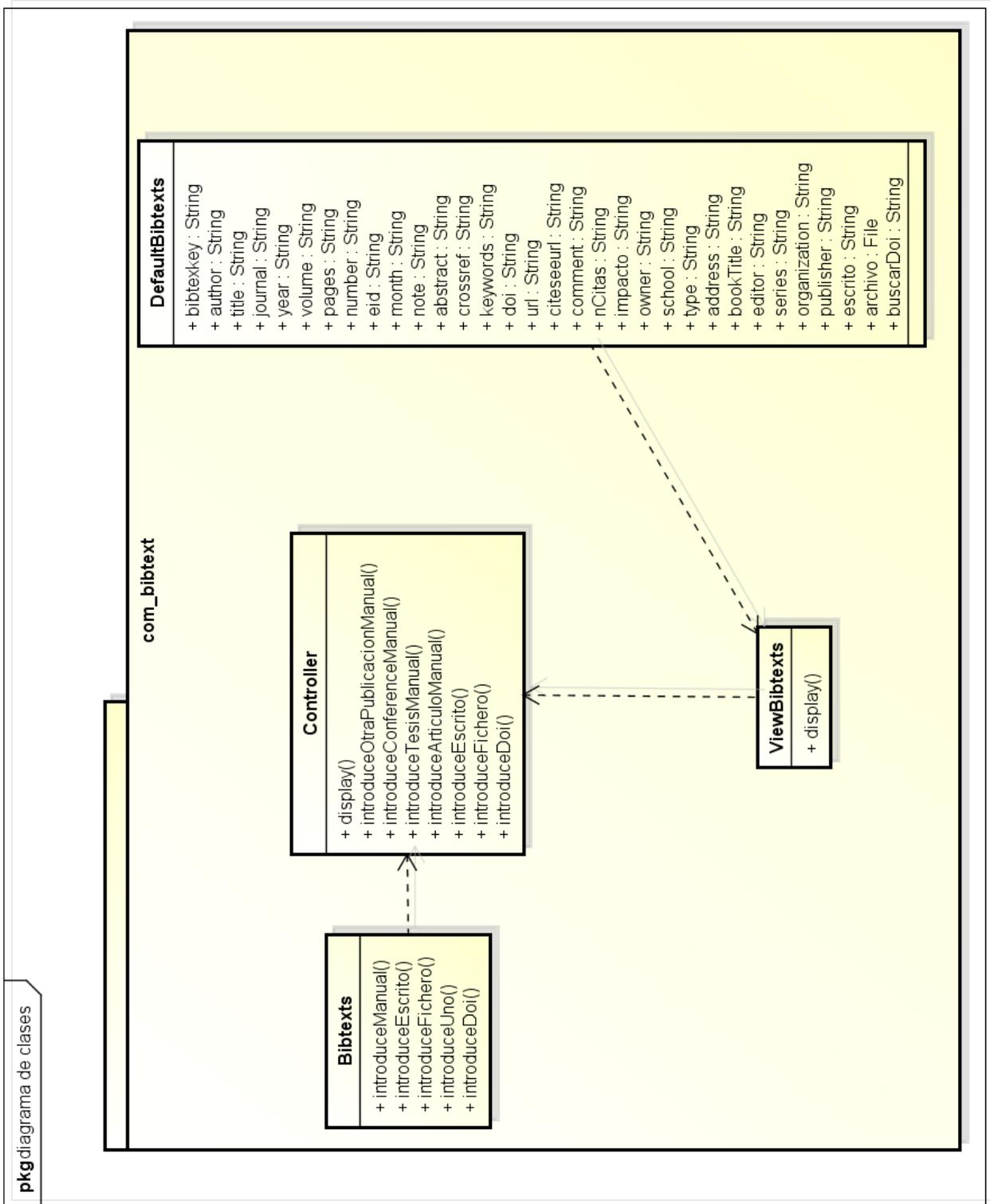


Figura 5.1: Diagrama de despliegue del sistema.

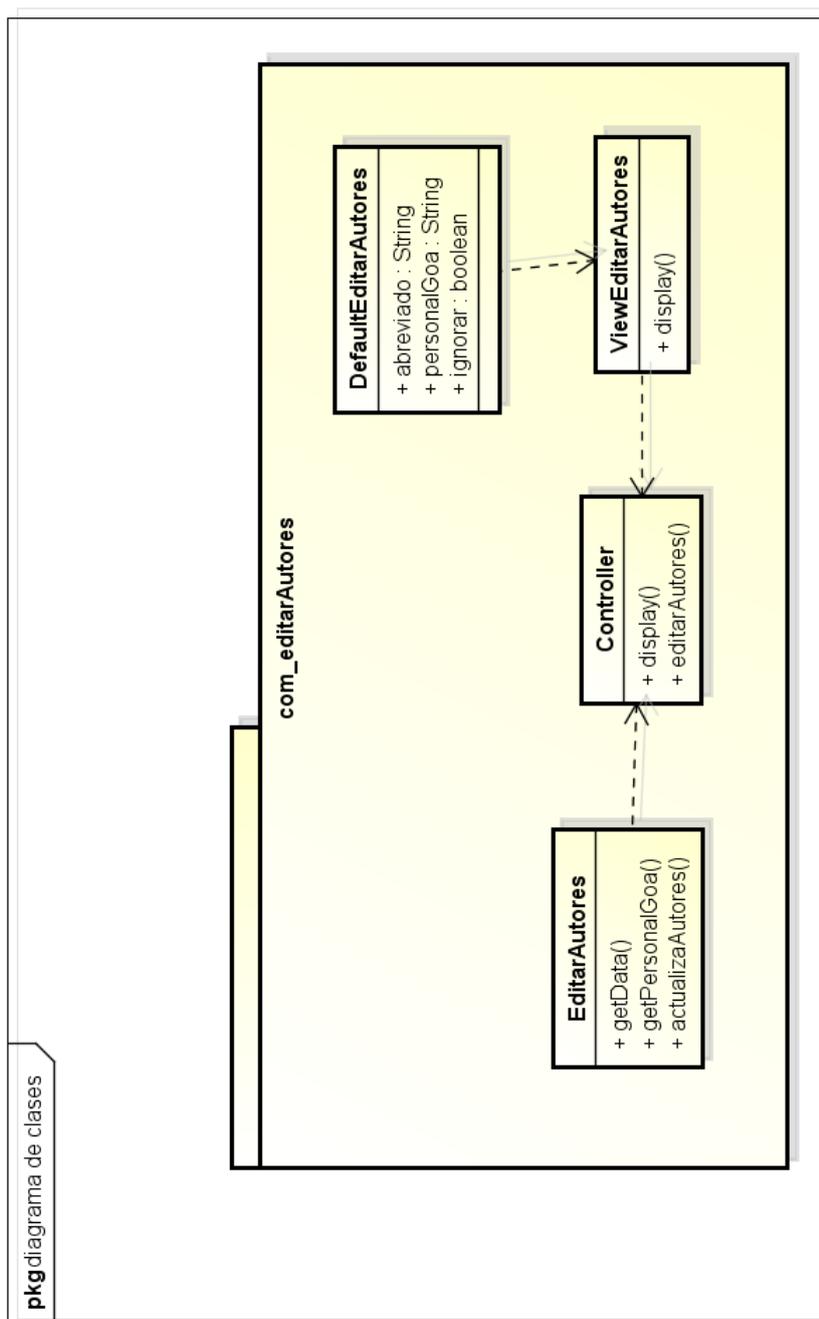
5.2. Diagrama de Clases de Diseño

Un diagrama de clases describe de forma estática, todos y cada uno de los componentes que forman el sistema. En un diagrama de clases podemos ver como se relaciona cada uno de los componentes del sistema y de qué operaciones y atributos están compuestos. De esta forma se captura la estructura lógica del sistema.

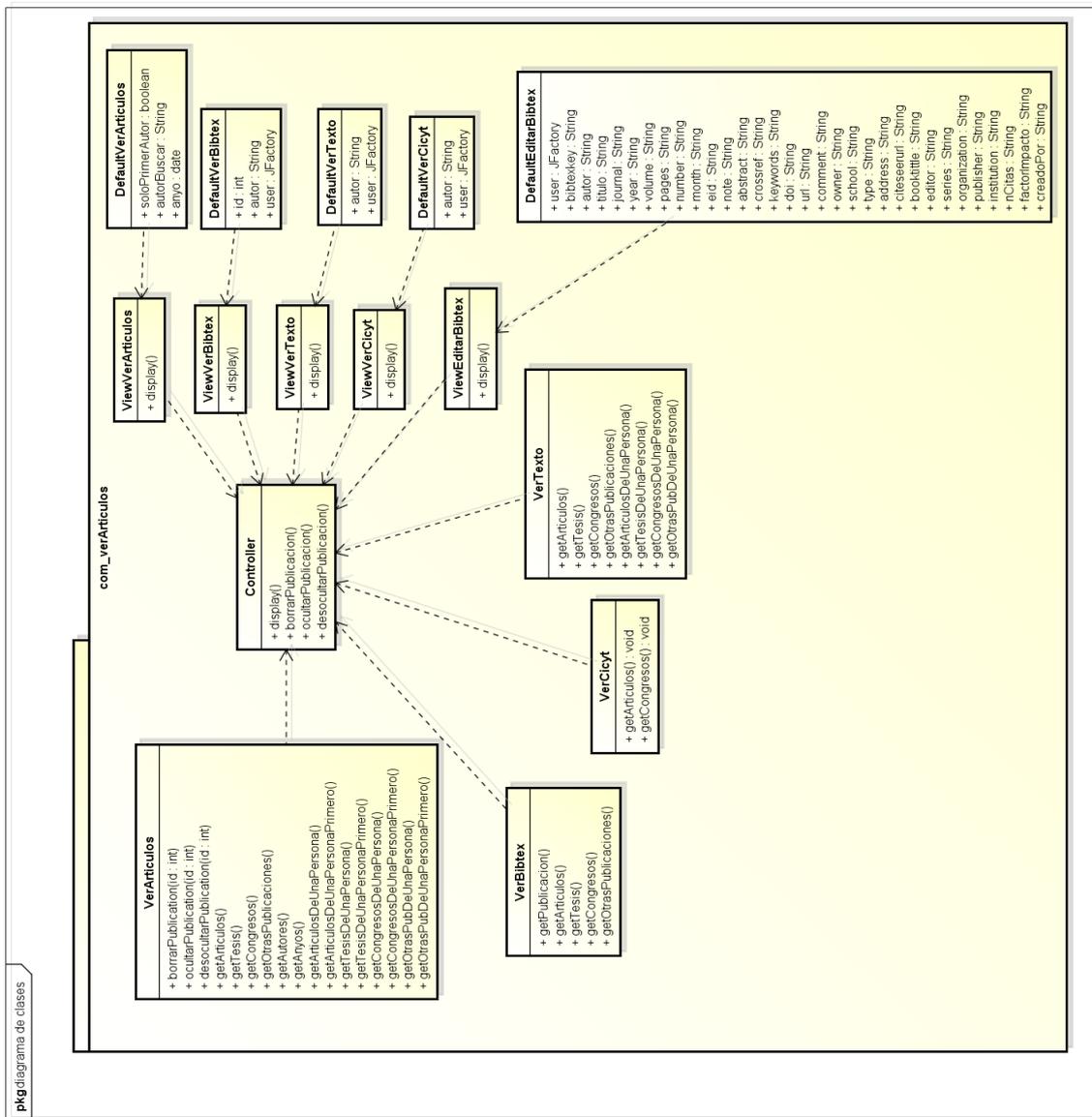
5.2.1. Componente Com_bibtex



5.2.2. Componente Com_editarAutores



5.2.3. Componente Com_verArticulos



5.3. Diagramas de Secuencia

La realización de los casos de uso, mediante diagramas de secuencia UML, nos permite apreciar cómo interactúan entre sí los distintos objetos para llevar a cabo el objetivo que se pretende cumplir.

Antes de mostrar los diagramas de secuencia de los casos, se va a mostrar el diagrama de secuencia del framework Joomla! donde la figura en verde representaría el diagrama de secuencia de cada caso de uso que son los que se muestran más adelante en nuestro caso (ver [figura 5.2](#))

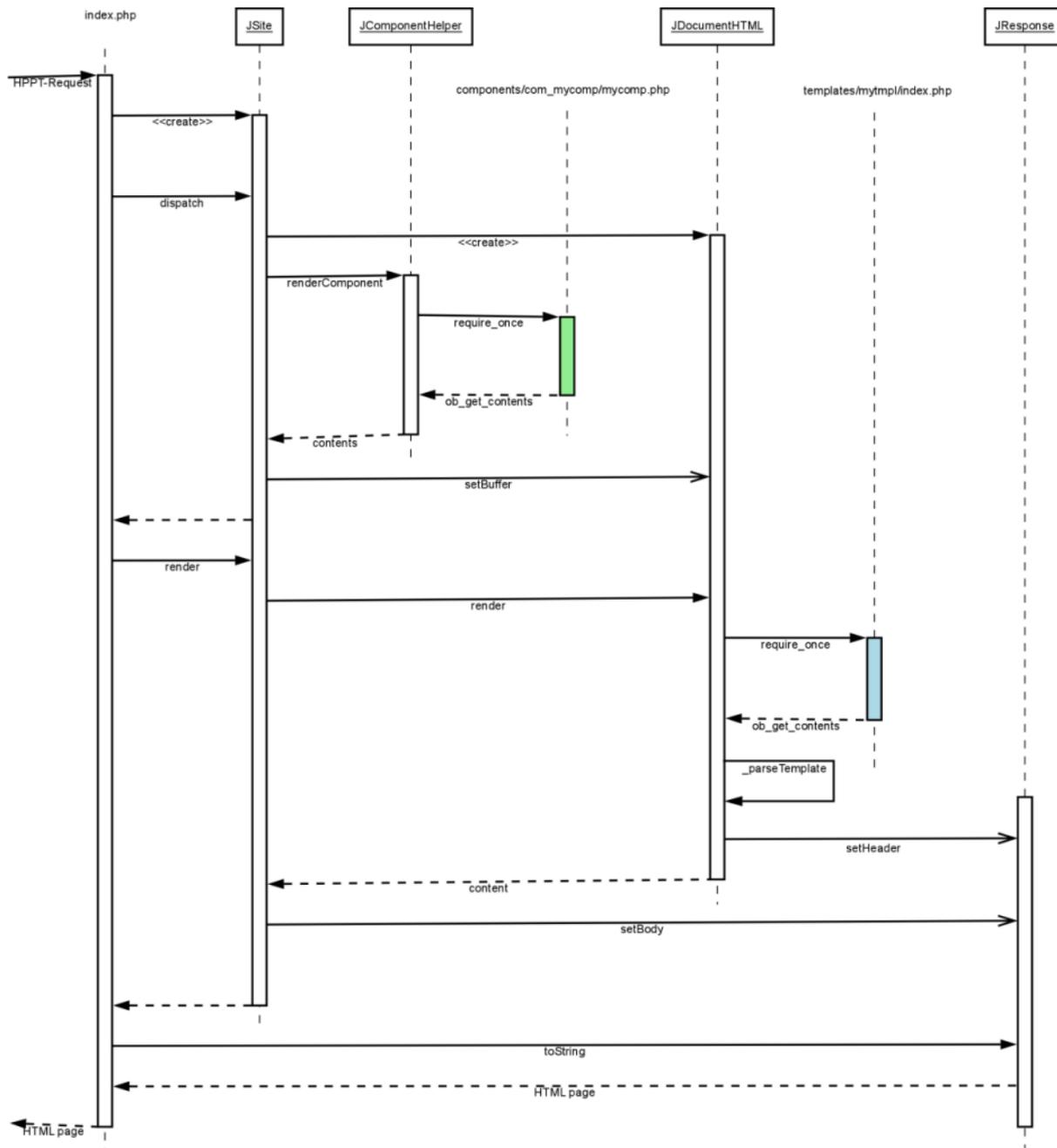
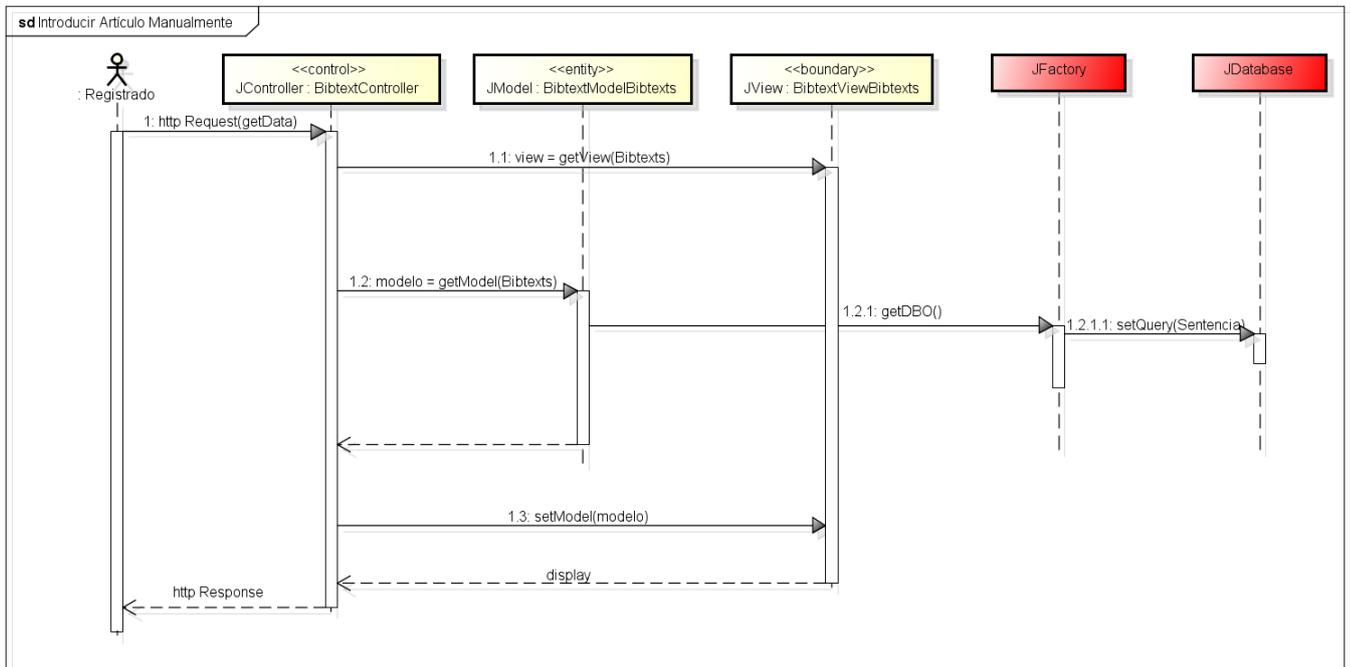


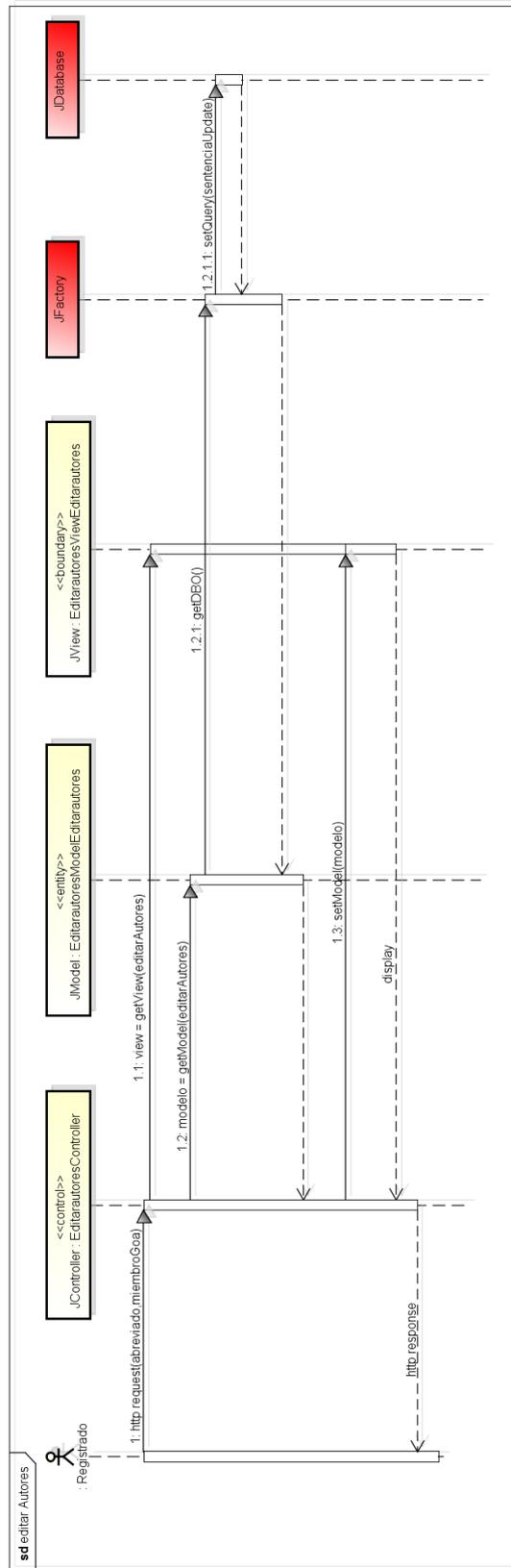
Figura 5.2: Diagrama de secuencia que muestra como funciona el framework de Joomla!, disponible en (http://docs.joomla.org/Component_Program_Flow)

5.3.1. Introducir Artículo Manualmente

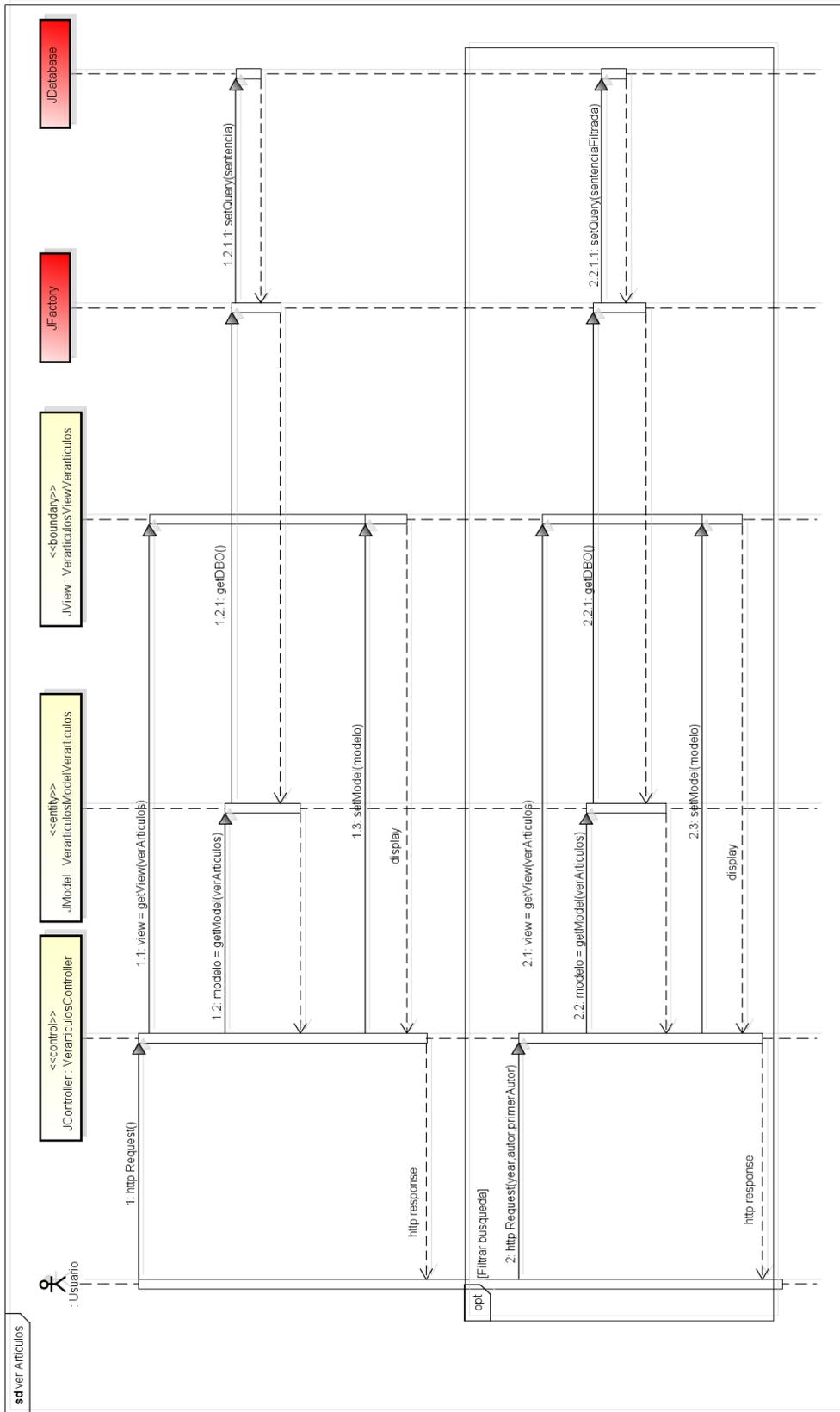


Los diagramas correspondientes a introducir el artículo en formato bibtext o extensión .bib serían iguales que el introducido anteriormente por lo que se van a obviar.

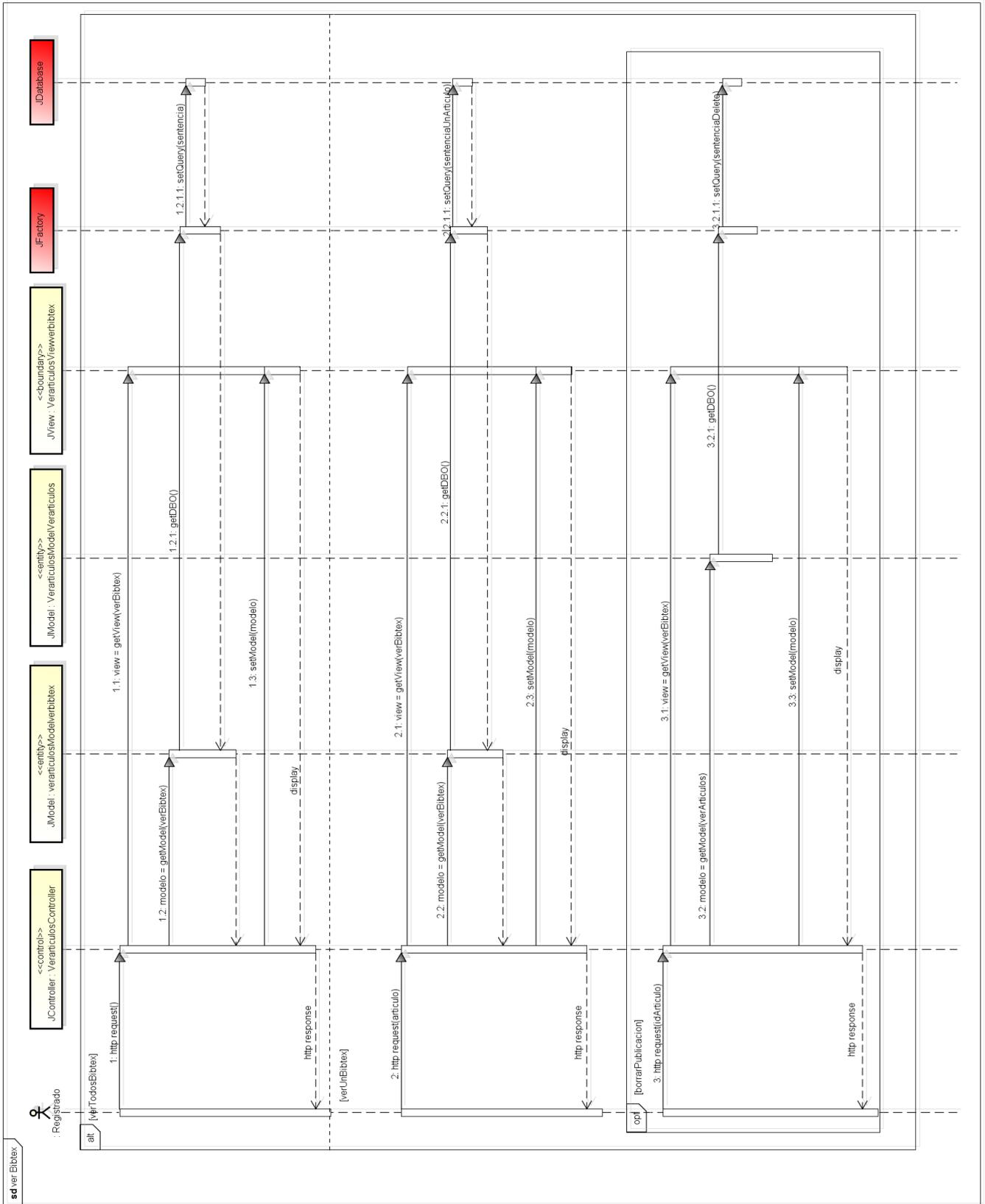
5.3.3. Rellenar Autores



5.3.4. Ver Artículos

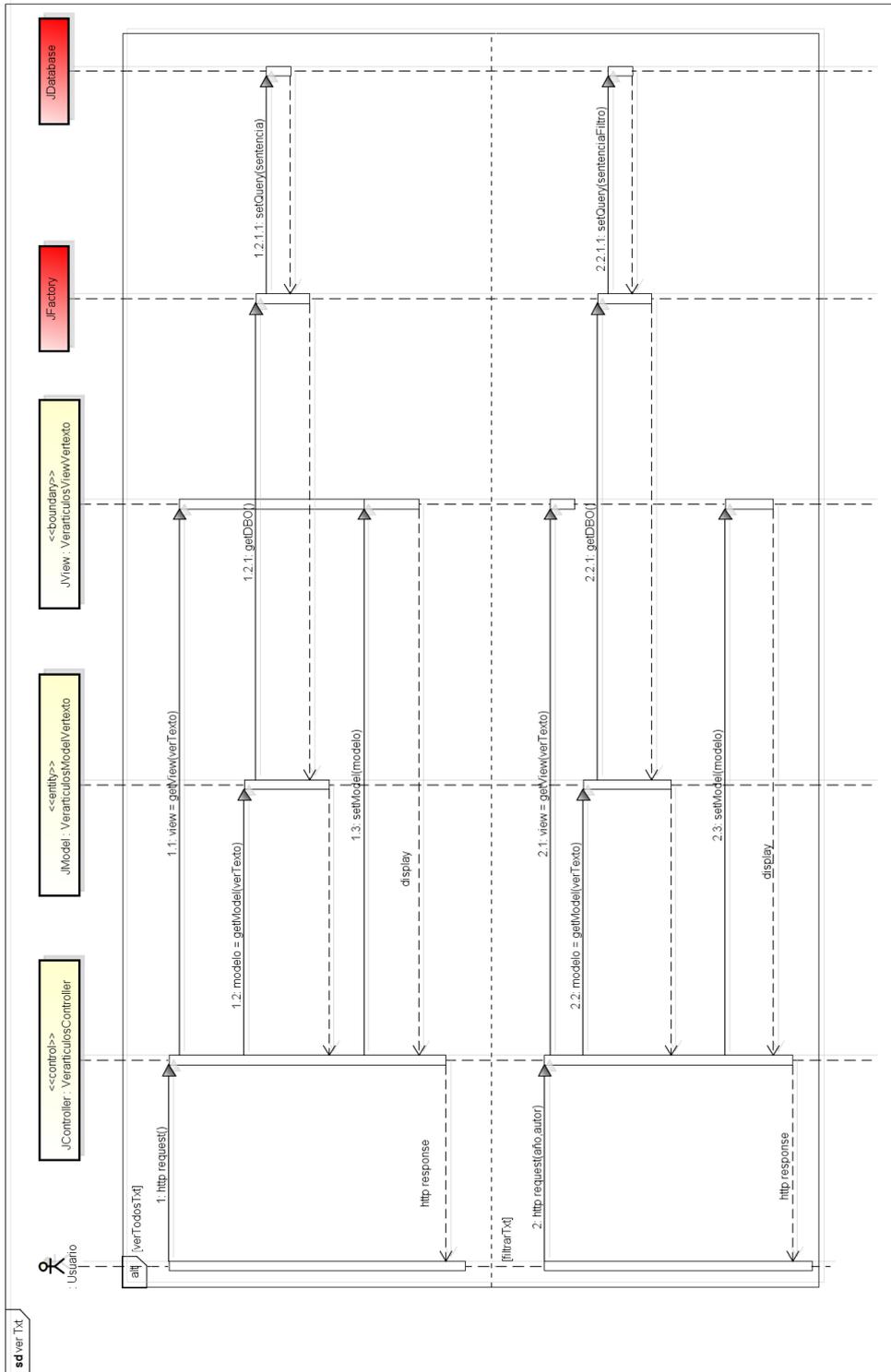


5.3.5. Ver Bibtex

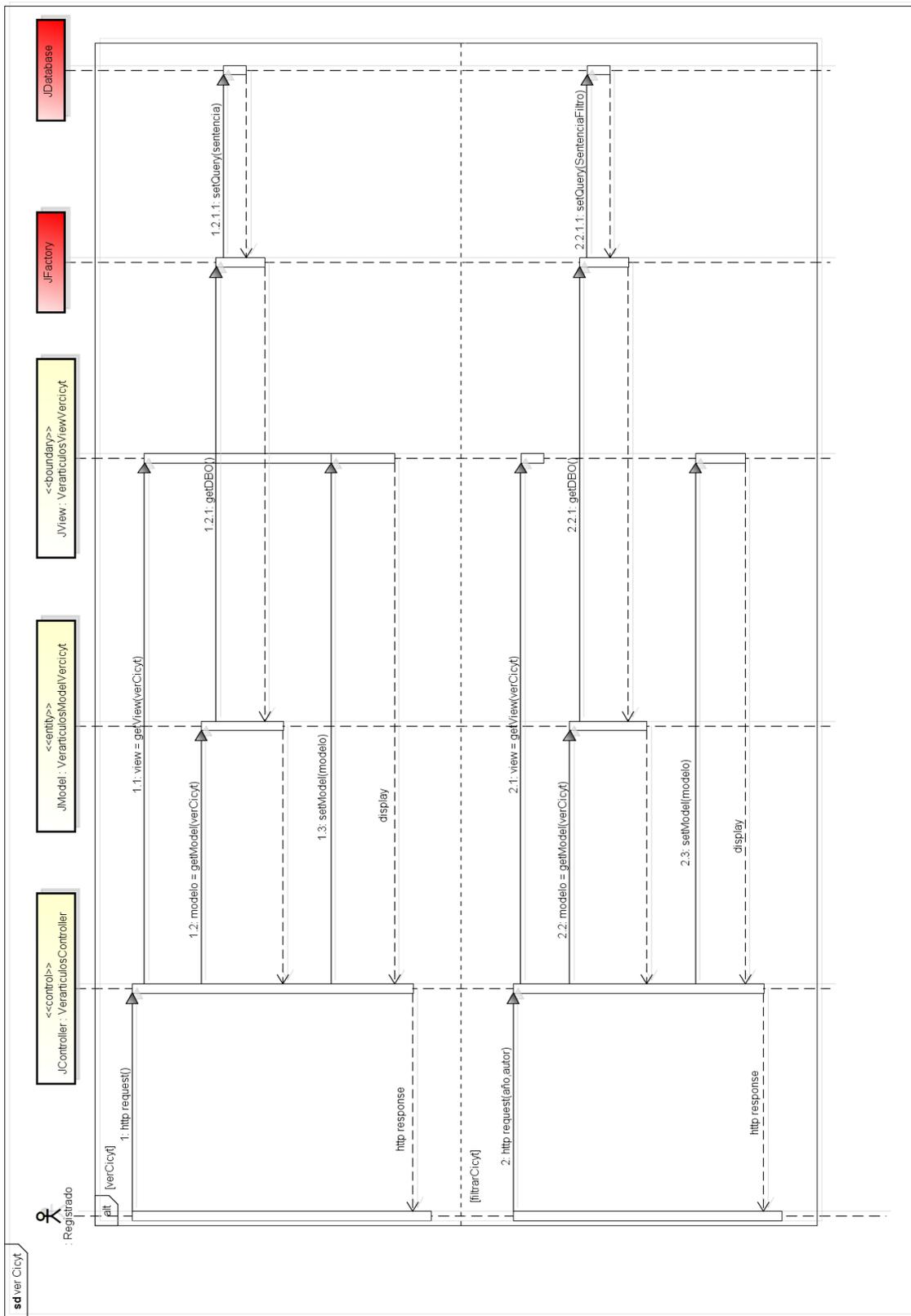


El diagrama de secuencia de desocultar artículo sería igual que la parte de ocultar artículo, incluida en ver bibtex, solo que le aparecería al administrador.

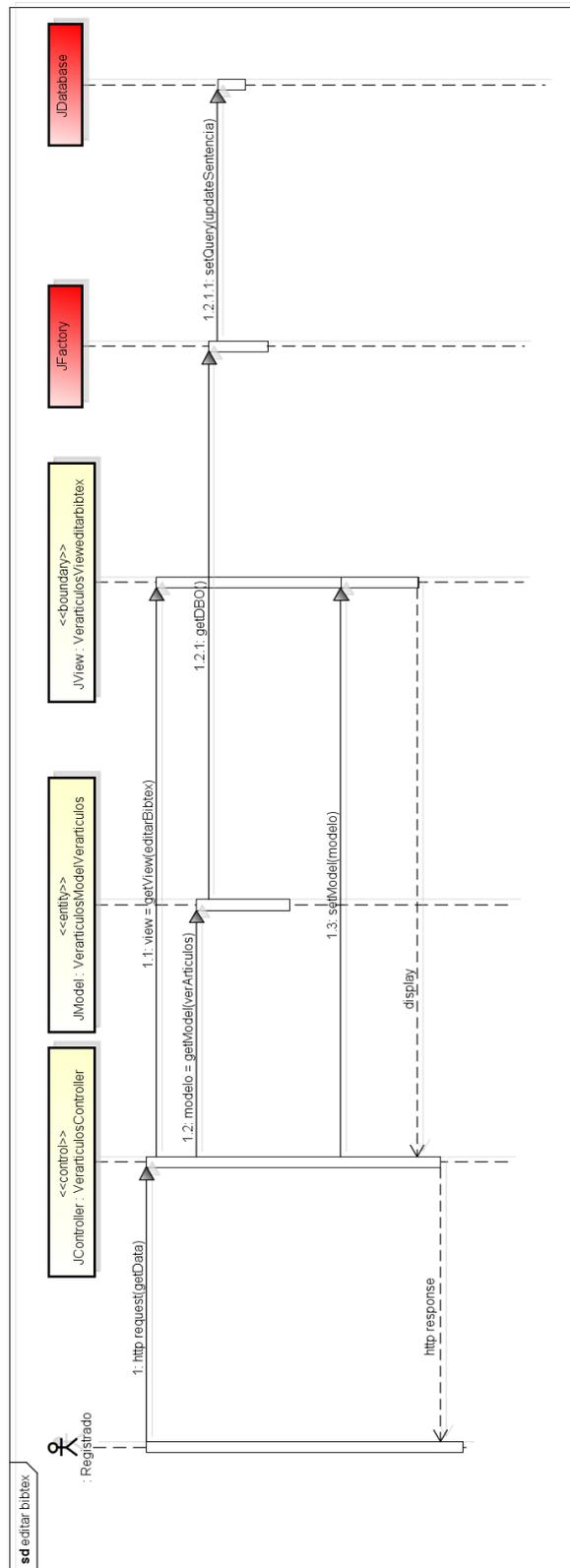
5.3.6. Ver Txt



5.3.7. Ver Cicyt



5.3.8. Editar Bibtex



Capítulo 6

Implementación del Proyecto

Contenido

6.1. Implementación	81
6.1.1. Plantilla	81
6.1.2. Componentes	82
6.1.3. Componentes Propios	82

Resumen: *En este capítulo se muestra todo lo referente a la implementación de los tres componentes creados ya que es la parte más de ingeniería de este proyecto, debido a que la creación de la página en si es más de diseño gráfico, por lo que aquí hay que centrarse en lo referente a la estructura de los componentes y lo más remarcable de su código.*

6.1. Implementación

A continuación se detallan los aspectos más importantes de la implementación de las acciones que realiza el sistema, ya que no se puede incluir todo el código de la aplicación debido a su extensión. Este capítulo va ha centrarse en lo que ha sido la creación e implementación de los tres componentes creados para la pagina web que son, **com_bibtext**, **com_editarautores** y por último **com_verarticulos**. Hay que centrarse en esta parte debido a que la implementación de lo que es Joomla! en sí se crea cuando se instala el CMS y no tiene mayor transcendencia para el proyecto. Eso sí, hay que mencionar por encima algunas elecciones que se han tenido que llevar a cabo para la configuración y correcta implementación de este proyecto.

6.1.1. Plantilla

Antes de nada hay que puntualizar que dentro de la multitud de plantillas (denominados templates) que se pueden instalar para el frontend de Joomla!, se ha elegido una que tuviera sobre todo dos factores clave.

El primero de estos factores es que tuviera el mayor número de posiciones posibles para los módulos, ya que como Joomla! se basa en módulos, cuantas más posiciones tenga predeterminadas

para los distintos módulos, mayor nivel de personalización tendrá, debido a que modificar la plantilla en sí puede ser una tarea bastante compleja para no descuadrar el resto de módulos y que todo quede perfectamente colocado a la vista del usuario sin depender del navegador que esté usando.

Y el segundo factor es que fuera responsiva, debido en gran parte a que el acceso a las páginas web hoy en día se suele hacer desde dispositivos móviles, ya sean smartphones o tablets, y que sea responsiva hace que el CSS adapte la pagina web a las características de tamaño del dispositivo y sea mucho mas usable por el usuario que entre a la web.

La template que se ha elegido en cuestión es **Shape5.vertex**, ya que cotejando y valorando respecto a otras es la que más se adaptaba a nuestras necesidades, debido a que cumple las dos características anteriormente citadas y además tiene bastantes parámetros de configuración.

6.1.2. Componentes

Como Joomla! es un CMS basado en módulos y componentes, a continuación simplemente se van a enumerar y dar una pequeña descripción de los componentes externos que se han instalado para esta página web. Son los siguientes:

- **Akeeba Backup**: Su función principal es hacer copias de seguridad para almacenar en caso de desastre o migración de la página a otro servidor web. Simplemente es un .zip que tendremos que colocar en nuestro servidor para poder ejecutarlo.
- **CoalaWeb Traffic**: Potente componente para todo lo que tiene que ver control del numero de visitas, IPs de las que provienen estadísticas de estas visitas y otras muchas funcionalidades.
- **DPCalendar**: Calendario para poder guardar eventos y recordar fechas importantes.

6.1.3. Componentes Propios

Aunque también son componentes, en estos se va a hacer más hincapié y se va a detallar los archivos que contienen cada uno de ellos ya que son los tres componentes propios que se han creado para esta versión de Joomla! 3.2.4, debido a que aunque ya existían para la página web antigua, la estructura de Joomla! y el core del CMS han cambiado y eran totalmente incompatibles, así que ha sido necesaria un cambio en el código y en la estructura de ficheros de cada componente; además se ha añadido a uno de ellos una funcionalidad nueva bastante atractiva a ojos de los usuarios de la página, facilitando la tarea de introducir los artículos en la base de datos.

Primeramente, antes de hablar de lo que sería el código y la funcionalidad de cada archivo, se va a detallar como sería la estructura interna de los componentes ya que es la base para que todo componente funcione en Joomla!.

Para poder instalar cualquier componente en Joomla! desde el gestor de extensiones, este tiene que estar comprimido en formato .zip y dentro de este archivo comprimido tiene que tener dos carpetas, una para lo que sería la funcionalidad del componente en el frontend, llamada **site**, y otra para la funcionalidad del componente en el backend, llamada **administrator**. Además tiene que tener un archivo .xml que tenga el nombre del componente. A continuación, se muestra en la siguiente captura (ver [figura 6.1](#)), la estructura de este .zip:

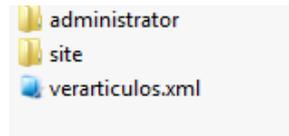


Figura 6.1: Estructura del componente.

Vamos a describir la estructura del .xml que sería el instalador del componente :

```
<?xml version="1.0" encoding="utf-8"?>
<extension type="component" version="3.0" method="upgrade">
  <name>com_verarticulos</name>
  <creationDate>2014-05-05</creationDate>
  <copyright>Copyright (C) 2014. Todos los derechos reservados.</copyright>
  <license>Licencia Pública General GNU versión 2 o posterior. Consulte LICENSE.txt</license>
  <author>sergio carcel fernando</author>
  <authorEmail>sergiocarfer@gmail.com</authorEmail>
  <authorUrl>http://</authorUrl>
  <version>1.0.0</version>
  <description></description>

  <install> <!-- Runs on install -->
    <sql>
      <file driver="mysql" charset="utf8">sql/install.mysql.utf8.sql</file>
    </sql>
  </install>
  <uninstall> <!-- Runs on uninstall -->
    <sql>
      <file driver="mysql" charset="utf8">sql/uninstall.mysql.utf8.sql</file>
    </sql>
  </uninstall>

  <files folder="site">
    <filename>index.html</filename>
    <filename>verarticulos.php</filename>
    <filename>controller.php</filename>
    <filename>router.php</filename>
    <folder>views</folder>
    <folder>models</folder>
    <folder>controllers</folder>
    <folder>helpers</folder>
  </files>
  <languages folder="site">
    <language tag="en-GB">language/en-GB.com_verarticulos.ini</language>
    <language tag="es-ES">language/es-ES.com_verarticulos.ini</language>
  </languages>
  <administration>
    XXX_XML_MENU_VIEW_START_AND_STOP_XXX
    <submenu>
      XXX_LOOP_XML_SUBMENU_VIEWS_START_AND_STOP_XXX
```

```
</submenu>
<files folder="administrator">
  <filename>access.xml</filename>
  <filename>config.xml</filename>
  <filename>controller.php</filename>
  <filename>index.html</filename>
  <filename>verarticulos.php</filename>
  <folder>controllers</folder>
  <folder>assets</folder>
  <folder>helpers</folder>
  <folder>models</folder>
  <folder>sql</folder>
  <folder>tables</folder>
  <folder>views</folder>
</files>
<languages folder="administrator">
  <language tag="en-GB">language/en-GB.com_verarticulos.ini</
  language>
  <language tag="en-GB">language/en-GB.com_verarticulos.sys.ini</
  language>
  <language tag="es-ES">language/es-ES.com_verarticulos.ini</
  language>
  <language tag="es-ES">language/es-ES.com_verarticulos.sys.ini</
  language>
</languages>
</administration>
</extension>
```

Vemos en este código que la primera parte está relacionada con todo lo que tiene que ver con datos de información. Luego está la parte de creación de tablas en la base de datos, y lo más importante es la parte donde se define la estructura de archivos para la parte de frontend en la carpeta “site” y la parte de administración en el backend de la carpeta “administrator”.

La estructura de la parte de administración es la siguiente:

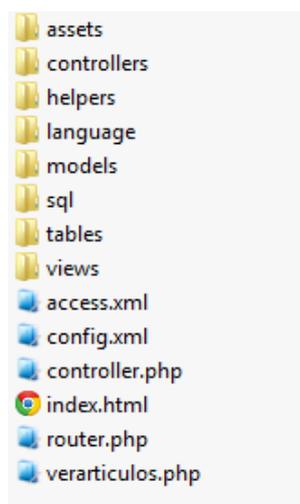


Figura 6.2: Estructura de la carpeta Administrator de un componente.

Mientras que la estructura de la parte del frontend es la siguiente:

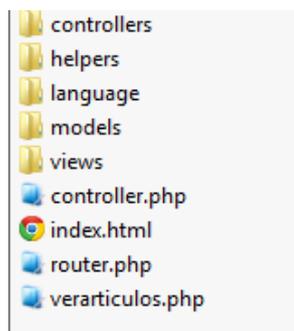


Figura 6.3: Estructura de la carpeta Site de un componente.

Cada uno de estos archivos y carpetas son importantes, aunque no todos son obligatorios. Se va a detallar que contiene cada carpeta y cada fichero de la parte de frontend:

- **archivo Verarticulos.php:** En el caso de la imagen anterior (ver [figura 6.3](#)), tendría este nombre, pero este archivo siempre tendrá el nombre del componente; es el primer archivo reconocido y leído por Joomla! después de la instalación. Se encargará de la reorientación de las tareas a otros controladores, la carga de los archivos helper, hojas de estilo, javascript, librerías de plugins y otras piezas básicas y necesarias de todo el componente.
- **archivo Router.php:** Permite configurar el sistema de enrutamiento del componente.
- **archivo Index.html:** Fichero vacío que debe estar dentro de todas las carpetas.
- **archivo Controller.php:** Es el controlador genérico.
- **carpeta Controllers:** Esta carpeta ha sido uno de los cambios añadidos en Joomla! 3.x, porque desde esta versión es posible crear los controladores como una clase con una única función. No es obligatorio hacerlo y se puede seguir creando como una clase con diversas funciones, aunque si el componente tiene un controlador con muchas funciones por tema de escalabilidad, es preferible hacerlo de esta manera.
- **carpeta Views:** Joomla! tiene una forma un poco particular de manejar las vistas. Hace uso de un controlador secundario para ayudar en el aspecto del renderizado de los datos. Dentro de la carpeta *views* hay una carpeta con el nombre de cada vista y dentro de ella hay un archivo con el nombre del render que usa, en nuestro caso **view.html.php**. Junto a ese fichero hay otra carpeta llamada *tmpl* que contiene el archivo `default.php` que está formado por todo el HTML que finalmente será mostrado. Además, en esa carpeta está el archivo `default.xml`, que le indica al gestor de menús de nuestro Joomla! que puede utilizar esta vista como una vista enlazable desde el menú de nuestro sitio web.

Aquí se muestra una pequeña parte de esa vista:

```
class VerarticulosViewVerarticulos extends JViewLegacy
{
```

```

// Sobrescribir visualización del método JView
public function display($tpl = null)
{
    //conecta con su modelo
    $model = $this->getModel();

    $articulos = $model->getArticulos();
    //Asignar datos a la vista
    $this->assignRef('articulos', $articulos);

    $tesis = $model->getTesis();
    $this->assignRef('tesis', $tesis);
}

```

Lo más importante a tener en cuenta es el nombre de la clase para poder comunicarse con el modelo, ya que viene dado siempre como *nombreDelControladorViewNombreDeLaVista* y tiene que coincidir el nombre de la vista con el nombre del modelo, sino no será capaz de comunicarse con el modelo y el controlador no podrá hacer esa conexión a menos que se especifique en éste que vista va con ese modelo. Pero por el sistema de nomenclaturas la conexión es automática.

- **carpeta Models:** Aquí se sitúan los modelos y tienen que tener el nombre que luego se asignará en la clase de la vista como ya se explicó anteriormente. Los modelos en Joomla! trabajan como la mayoría de los sistemas MVC y manejan la mayor parte de la manipulación de los datos y recuperación de los mismos.

Aquí se incluye una pequeña parte del código para que se vea el nombre de la clase, que en este caso tendrá que ser *NombreDelControladorModelNombreDelModelo*:

```

//nombre de la clase
class VerarticulosModelVerarticulos extends JModelList {

    //aquí estaran las funciones que se llaman para acceder a los datos.
}

```

- **carpeta Helpers:** Contiene clases que nos aportarán una serie de métodos para facilitar el desarrollo.
- **carpeta Language:** Contiene los ficheros de los idiomas que tengamos configurados.

Estos son los archivos y la estructura que contiene la parte del frontend; en la parte del backend, además de estos ficheros, contiene la configuración de las tablas que se añadieron a la base de datos si hubiera que añadirlas, y algunos ficheros de configuración extra, pero en definitiva el cuerpo es el mismo con la diferencia de que en esta parte se le añadiría el código para la administración o cambio de parámetros de la parte pública del componente.

En mi caso, la parte de administración no tiene nada más que el esqueleto para que en caso de que se quiera más adelante añadir alguna funcionalidad extra solo hubiera que añadir el código correspondiente.

Una vez visto como está estructurado, se pasa a explicar que hacen los componentes en sí, dividiendo la funcionalidad y las vistas que tiene cada componente:

- **Com_verarticulos:** Este componente cuenta con 5 vistas que son: **verArticulos**, **verBibtext**, **verTexto**, **verCicyt**, y por último, **editarBibtext**.
 - **verArticulos:** Muestra una lista con todos los artículos de investigación que hay en la base de datos, ya sean artículos, tesis doctorales, congresos u otras publicaciones, y se pueden filtrar por año, autor y al seleccionar por autor que solo se muestren las que ese autor sea el primer autor. Además esta es la vista principal y desde ella se acceden a las otras vistas en los iconos correspondientes.
 - **verBibtext:** Se puede acceder a esta vista de dos maneras, pinchando en el icono de bibtext que nos mostrará una lista de todos los artículos en el formato bibtext. Y la otra manera es desde un link de bibtext situado debajo de cada artículo en verArticulos y nos mostrará el bibtext solo de ese artículo en concreto.
 - **verTexto:** Se accede también desde verArticulos pulsando en la imagen de txt y en esta vista se muestran todos los artículos; primero los artículos, luego las tesis, luego los congresos y luego las otras publicaciones.
 - **verCicyt:** Para poder acceder a esta vista se tiene que estar registrado e iniciado sesión en el sistema, entonces se podrá ver una imagen desde la que accederemos a esta vista y es otro formato para poder ver todos los artículos de la base de datos, listado uno detrás de otro.
 - **editarBibtext:** A esta vista solo se puede acceder cuando se entra en la vista verBibtext de un solo artículo y el usuario que entra, o es el usuario que creó el artículo, o es un administrador y si es así le aparecerá un botón de editar; este botón le llevará a esta vista donde podrá editar los campos del artículo.
- **Com_editarautores:** Este componente tiene solo una vista y se creó con el simple propósito de que los autores tuvieran un solo nombre en la base de datos, ya que cuando la gente introducía autores no ponía exactamente el mismo nombre y surgió la idea de los seudónimos. Por lo que lo que este componente muestra es una lista con los seudónimos que todavía no han sido relacionados con los autores guardados en la base de datos. Muestra una lista de ellos y al lado un desplegable para que se seleccione el autor que corresponda; si no corresponde con ninguno se selecciona la casilla ignorar y ese seudónimo es borrado.
- **Com_bibtext:** Este componente tiene una sola vista también y es el de más valor de cara a la función que desempeña. Es en el que se introduce el artículo, sea del tipo que sea y se guarda en la base de datos. Para ello dentro de la misma vista tiene varias pestañas; 4 pestañas principales que son la de introducir artículo por formulario, por formato bibtext, por archivo .bib y la nueva funcionalidad añadida en esta nueva versión que es la de introducir el DOI y que busque en la base de datos del WOK ese artículo y lo introduzca en la base de datos del servidor de GOA. Dentro de la pestaña de formulario hay otras 4 subpestañas que contienen los campos a rellenar según si lo que se introduce es un artículo, una tesis doctoral, un congreso u otra publicación.

Capítulo 7

Batería de Pruebas

Contenido

7.1. Introducción	89
7.2. Caja Blanca	90
7.2.1. Base de Datos	90
7.2.2. Parámetros	90
7.2.3. Interfaz	90
7.3. Caja Negra	90

Resumen: *En este capítulo se enumerarán las pruebas realizadas a la aplicación durante la fase de implementación y una vez finalizada la misma, para comprobar y verificar que el sistema realiza todas sus funciones correctamente, o subsanar los errores que aparezcan.*

7.1. Introducción

En cualquier proceso de desarrollo de software, una parte fundamental es la fase de pruebas. Estas consisten en revisar la especificación, el diseño y la implementación del sistema para poder verificar que el sistema cumple con las expectativas iniciales.

La fase de pruebas ha ido combinándose con la codificación del sistema, por lo que muchos errores se han ido subsanando a la vez que se programaban ya que como Joomla! está basado en componentes unitarios, el sistema en sí ayuda a poder comprobar por separado estos pequeños sistemas software, por lo que el sistema ha ido mejorándose y depurando los diferentes errores que iban surgiendo a la vez que se construía.

Como muchas partes de lo que son los componentes son similares, al subsanar un error en uno de esos componentes, se evitaba que surgieran los mismos errores en el resto.

Al ser independiente un componente de otro un error que surgiera en uno no afectaba a ningún otro, exceptuando un error en alguna vista en alguno que hace que sistema entero no se muestre correctamente.

Se han realizado dos tipos de pruebas, las que se conoce el código del sistema o pruebas de caja blanca, o las que se obvia el código y solo se observa el sistema desde exterior, llamadas pruebas de caja negra.

7.2. Caja Blanca

Las pruebas de caja blanca como se basan en el estudio de los diferentes caminos por los que se puede ejecutar el código, el uso de recursos de los mismos, así como fallos en algún tipo de implementación y en definitiva abarcar tanto, solo se presenta un breve resumen de las mismas.

7.2.1. Base de Datos

- El esquema relacional diseñado se obvió ya que estaba diseñado y creado por el administrador y programador del antiguo sistema y se mantiene intacto.
- Operaciones con la base de datos (selección, modificación, inserción y eliminación).
- Prueba con diferentes parámetros para ver que se comportan correctamente.
- Comprobación de la correcta devolución del resultado esperado.

7.2.2. Parámetros

- Comprobación de paso de parámetros entre las diferentes partes del MVC, ya sean tanto tipos como valores de los parámetros.

7.2.3. Interfaz

- En esta parte habría que comprobar que el valor introducido por el usuario se recoge bien, y si el foco donde esta el cursor es correcto, pero de eso es encarga Joomla!, no ha habido que tocar nada.
- Validaciones en los cuadro de textos para que recoja los tipos que están permitidos, es decir, entero, string, etc.
- Campos obligatorios a rellenar para poder enviar formularios.
- Las secciones del menú funcionan correctamente y solo se muestran las que se deben de mostrar según el tipo de usuario que sea.
- Funcionan los enlaces correctamente, tanto externos como internos.
- Formato correcto, filtrado de datos correcto y si se recogen correctamente los valores introducidos por pantalla.
- Valores por defecto si el usuario deja en blanco un campo.

7.3. Caja Negra

Las pruebas de caja negra, fueron realizadas por el mismo grupo de óptica atmosférica cuando se iba terminando cada uno de los tres componentes. El de **com_bibtex**, el de **com_verarticulos** y el de **com_editarautores**.

De esta manera ellos iban introduciendo artículos, tesis doctorales, congresos y otras publicaciones. Probando las diferentes combinaciones posibles y rellenando los campos que creían oportunos en busca de algún fallo. Cuando el cliente me daba el visto bueno se pasaba al siguiente componente.

Este proceso se repitió tantas veces como el cliente creyó oportuno, hasta que cada componente estaba implementado con todas las funcionalidades y características que ellos consideraban correctos.

Hasta que no se daba el visto bueno de uno de ellos no se pasaba al siguiente. En lo referente a la página web en si también; en cada reunión semanal se iban diciendo los fallos o mejoras que querían en lo que a la interfaz y funcionalidades se refiere.

Capítulo 8

Conclusiones y Trabajo Futuro

Contenido

8.1. Conclusiones	93
8.2. Trabajo Futuro	94

Resumen: *Tras la realización de este TFG, lo que se muestra en este capítulo, es lo que podemos concluir y lo que, bajo nuestro punto de vista, podría dar lugar a futuras mejoras de la aplicación creada.*

8.1. Conclusiones

En lo referente a los objetivos marcados se han cumplido todos, se ha implementado la nueva página web basada en una versión de Joomla! 3.x, se ha mejorado la interfaz rediseñandola sin perder la esencia de la página anterior, se han adaptado los tres componentes a esta nueva versión y en lo referente a añadir el artículo introduciendo el DOI está implementada la API, pero debido a causas administrativas estamos a la espera de que nos validen nuestra IP en su servidor.

Lo que más nos ha sorprendido es la velocidad de acceso a la base de datos en comparación con la versión antigua que estaba implementada de Joomla!. Tardando en la versión antigua hasta 5-6 segundo en cargar los artículos. En esta nueva versión la carga de datos es prácticamente instantánea, por lo menos con la cantidad de datos que tiene actualmente nuestra base de datos.

Pero el mayor fallo que se le puede sacar a Joomla!, que a pesar de ser de código abierto, es que hay poca documentación en lo referente a lo que es el sistema en sí y más en lo referente a las últimas versiones. En muchos casos hay que tratar al CMS como si de una caja negra se tratase.

Y si lo que desea alguien es crearse su propio componente o funcionalidad para su página web, Joomla! no es la mejor opción, ya que la forma de crearlos no es nada intuitiva, hay que saber muy bien como funciona Joomla! y no hay muchos tutoriales de como crearlos más allá de un componente muy básico de paso de mensajes, sobre todo a partir de las versiones 3.x. En lo que a depuración se refiere no es nada funcional, ya que hay que subir el componente al servidor y probarlo para ver los fallos que tiene sin poder probarlos de manera local. Debido a que un componente es una parte del sistema Joomla! en sí, el solo no puede hacer nada.

Como conclusión final el haber creado estos componentes y pudiendo liberarlos a la comunidad Joomla!, beneficia principalmente a todo el sector de la investigación que publica este tipo de artículos, teniendo centralizados y de forma local todas las publicaciones que ellos deseen o consideren importantes para su labor.

8.2. Trabajo Futuro

Como trabajo futuro se expone a continuación la funcionalidad que se le debería añadir al componente **com_bibtex** ya que es el que introduce datos en nuestra base de datos y cuanto más automatizado, más fácil será para el usuario.

- Añadir la funcionalidad de poder introducir las revistas y los cuartiles en los que están los artículos, así como el factor de impacto actualizado. Esta funcionalidad se planteó en un principio, pero el servicio web que lo proporciona es de pago. Y habría que buscar una manera alternativa.
- Añadir la funcionalidad por la cual los autores de los artículos introducidos hagan referencia al nombre exacto del autor en la base de datos sin necesidad del componente **com_editarautores**, que actualmente es trabajo interno del personal del GOA que esta registrada en la página.

Bibliografía

- [1] ROGER PRESSMAN, *Ingeniería de Software: Un enfoque práctico*, quinta edición, McGraw-Hill, 2001 (capítulo 9)
- [2] IAN SOMMERVILLE, *Ingeniería de Software*, sexta edición, Addison-Wesley, 2002 (cap. 29)
- [3] MARTIN FOWLER, D. RICE, M. FOEMMEL, E. HIEATT, R. MEE, R. STAFFORD, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2002
- [4] CRAIG LARMAN, *Applying UML and patterns: an introduction to object oriented analysis and design and the unified process.*, Prentice-Hall, 2005
- [5] *Joomla Spanish*. [Consulta: 3 marzo 2014]. Disponible en: (<http://www.joomlaspanish.org/>)
- [6] *Vertex free Joomla Template*. [Consulta: 14 marzo 2014]. Disponible en: (http://www.shape5.com/product_details/free_templates/vertex_-_free_joomla_template.html)
- [7] *Curso de construcción de un sitio web desde cero con Joomla! 3*. [Consulta: 20 marzo 2014]. Disponible en: (<http://www.webempresa.com/curso-de-construccion-de-un-sitio-web-desde-cero-usando-joomla-3.html>)
- [8] *Multilinguaje y Seo con Joomla! 3.2*. [Consulta: 22 marzo 2014]. Disponible en: (<http://magazine.joomla.org/es/ediciones-anteriores/dic-2013/item/1657-multilinguaje-y-seo-con-joomla-3-2>)
- [9] *Configurar gmail como servidor smtp en Joomla 3*. [Consulta: 27 marzo 2014]. Disponible en: (<http://techandbeyond.wordpress.com/2013/04/20/configurar-gmail-como-servidor-smtp-en-joomla-3/>)
- [10] *Comprendre Joomla à l'aide d'UML*. [Consulta: 8 abril 2014]. Disponible en: (http://linformalibre.f2lt.fr/index.php?title=Comprendre_Joomla_%C3%A0_l%27aide_d%27UML)

- [11] *Developing a MVC component*. [Consulta: 15 abril 2014]. Disponible en: (http://docs.joomla.org/J3.x:Developing_a_MVC_Component/Introduction)
- [12] *Componente para Joomla! 3: Estableciendo el Escenario*. [Consulta: 17 abril 2014]. Disponible en: (<http://magazine.joomla.org/es/ediciones-antteriores/agosto-2013/item/1418-desarrollo-componente-joomla-3-preparando-escenario>)
- [13] *Componente para Joomla! 3: Comenzando el desarrollo*. [Consulta: 23 abril 2014]. Disponible en: (<http://magazine.joomla.org/es/ediciones-antteriores/agosto-2013/item/1421-componente-joomla-3-comenzando-el-desarrollo>)
- [14] *Componente para Joomla! 3: Funcionalidades*. [Consulta: 27 abril 2014]. Disponible en: (<http://magazine.joomla.org/es/ediciones-antteriores/agosto-2013/item/1425-desarrollo-de-un-componente-para-joomla-3-funcionalidades>)
- [15] *Componente para Joomla! 3: Funcionalidades*. [Consulta: 3 mayo 2014]. Disponible en: (<http://magazine.joomla.org/es/ediciones-antteriores/agosto-2013/item/1425-desarrollo-de-un-componente-para-joomla-3-funcionalidades>)
- [16] *La clase SoapClient*. [Consulta: 25 junio 2014]. Disponible en: (<http://php.net/manual/es/class.soapclient.php>)
- [17] RAMIRO GONZALEZ CATÓN. *Página Grupo de Óptica Atmosférica*. [Consulta: 25 julio 2014]. Disponible en: (<http://www.goa.es>)

Apéndice A

Manual de Instalación

Resumen: En este apéndice se muestra cómo instalar y utilizar los componentes creados.

I.1. Requisitos del Sistema

Para poder instalar estos componentes es necesario disponer de un servidor web donde tener tu hosting y tener un dominio asignado. Y en el tiene que estar instalado Joomla! 3.x. Sin estos requisitos los componentes no podrán ser instalados ya que necesitan del gestor de contenidos para ser usados.

I.2. Instalación de la Aplicación

Dentro de la carpeta software del CD-ROM que se entrega como parte de la documentación, hay una carpeta de componentes y dentro de ella están los tres archivos comprimidos cada uno para un componente. Para instalarlos hay que realizar el mismo proceso para cualquier componente que queramos instalar. Dentro de la parte de administración de nuestro Joomla!, en el menú situado en la parte superior vamos a **Extensiones->Gestor de extensiones** como podemos ver en la siguiente figura (A.1):

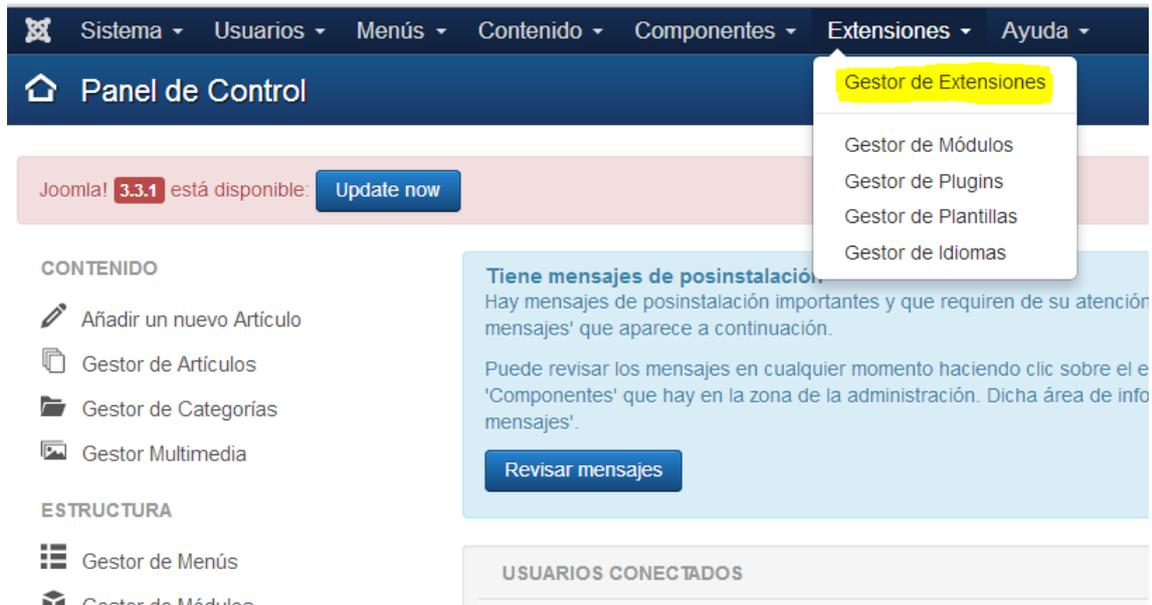


Figura A.1: Menú principal del backend.

Pulsando sobre Gestor de extensiones nos llevará a ese en el que vemos una serie de opciones, seleccionamos **Upload package file** como se ve en la figura (A.2):

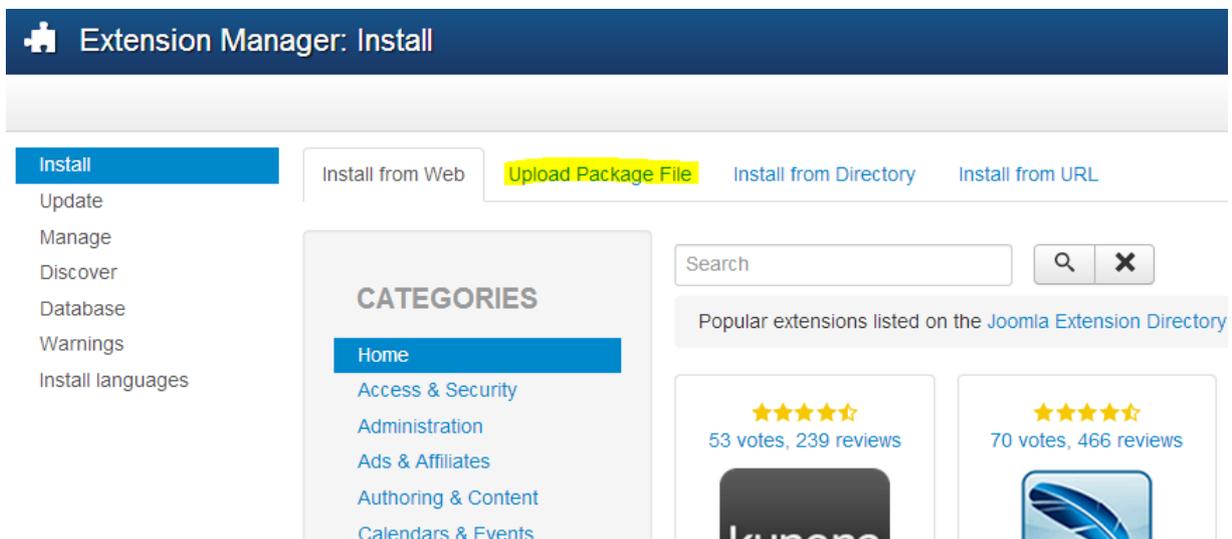


Figura A.2: Menú del gestor de extensiones

En este menú primero pulsaremos **seleccionar archivo** y seleccionaremos uno de los tres archivos comprimidos .zip como se ve en la figura (A.3):

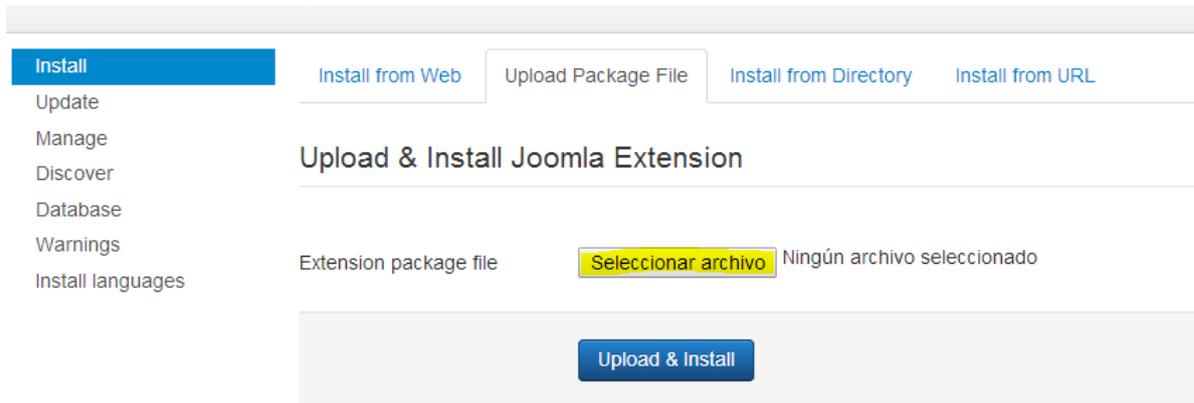


Figura A.3: Selección del archivo en el menú Upload Package File

Una vez seleccionado pulsaremos la opción de **Upload & Install** y se instalará el archivo como se ve en la figura siguiente (A.4):

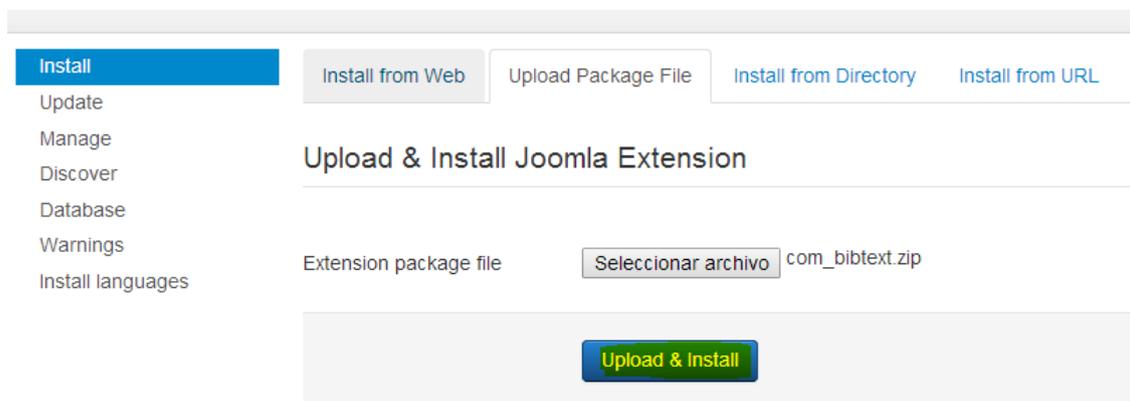


Figura A.4: Instalación del componente

Para finalmente avisarnos de que la instalación se ha completado con éxito y podremos comenzar a utilizar los componentes, ver (A.5):

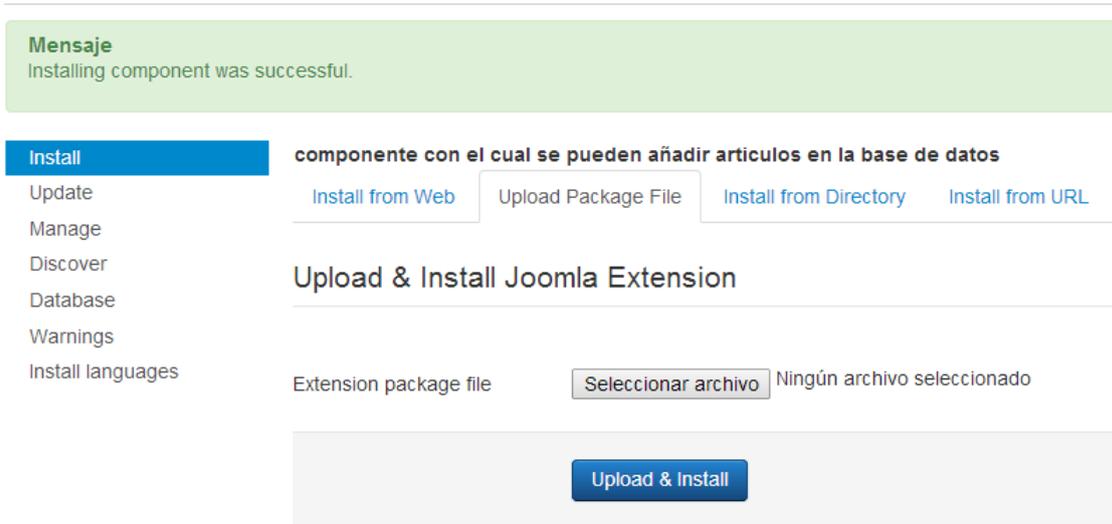


Figura A.5: Componente instalado correctamente

Para poder verlo tendremos que publicarlo al pulsar un elemento del menú que tengamos en la página web. La forma de realizarlo es la siguiente, primero vamos a la parte del backend **menú** y seleccionamos el menú que tengamos o queramos, como se ve en esta figura (A.6):



Figura A.6: Seleccionamos Menú

Seleccionamos de todos los elementos de Menú, el que queramos que nos lleve al componente, los elementos del menú tendrán este formato (A.7):

	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Publicaciones (Alias: publicaciones) Ver artículos » verarticulos		Publico	EN	Español (ES)	105
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Proyectos (Alias: proyectos) Artículos » Artículo simple		Publico	EN	Español (ES)	106
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Noticias (Alias: noticias) Artículos » Categoría Blog		Publico	EN	Español (ES)	107

Figura A.7: Seleccionamos Elemento del Menú

En el menú que nos aparece pinchamos en seleccionar tipo de elemento de menú, como se puede apreciar en esta figura (A.8):



Gestor de Menús: Editar Elemento del Menú

Aplicar Guardar & Cerrar Guardar & Nuevo Guardar como Copia Cerrar

Título Menú * Alias

Detalles Tipo de opciones para los Enlaces Opciones de visualización para la página Opciones Metadato

Tipo Elemento Menú * **Seleccionar**

Enlace

Ventana Destino

Plantilla Estilo

Figura A.8: Seleccionamos tipo de elemento del Menú

En el menú emergente que nos despliega seleccionamos el que tiene el nombre del componente, al hacer click en el, nos mostrará otra lista con opciones, elegimos la que tiene nuestro nombre de componente en nuestro caso la última, aquí se puede ver (A.9):



Figura A.9: Seleccionamos nombre de nuestro componente

Pulsamos guardar y cerrar y ya estaría publicado y listo.

I.a. Manual de Usuario

I.a.1. Requisitos del Sistema

Para que un usuario pueda acceder al sistema lo único que le hace falta es un dispositivo con acceso a Internet, ya sea ordenador de sobremesa, smartphone o tablet y tener un navegador como ya se indica en el RNF.13.

I.a.2. Acceso al Sistema

Para tener acceso a la página web introduciremos en la barra de direcciones de nuestro navegador la URL en la que se encuentre alojado nuestro sitio Joomla!. Actualmente está situado en baraja.opt.cie.uva.es/~joomla/, pero la intención futura es sustituirlo en el servidor antiguo en <http://goa.uva.es/>

I.a.3. Descripción de la Interfaz de Usuario

Estructura Página Inicial



Figura A.10: Pantalla Inicio

En esta figura (A.10) podemos ver la estructura de la página. Según entras en ella, tiene el menú situado a la izquierda donde podemos ver las diferentes secciones a las que acceder, además de las dos banderas que indican el cambio de idioma del sitio.

De todas esas secciones del menú voy a destacar dos, ya que las otras son meramente de exposición de contenido. **Publicaciones** que al pinchar en ella nos mostrará el componente creado **com.verarticulos**. En la figura siguiente (A.11) podemos ver lo que nos aparece al pinchar en **Publicaciones**:

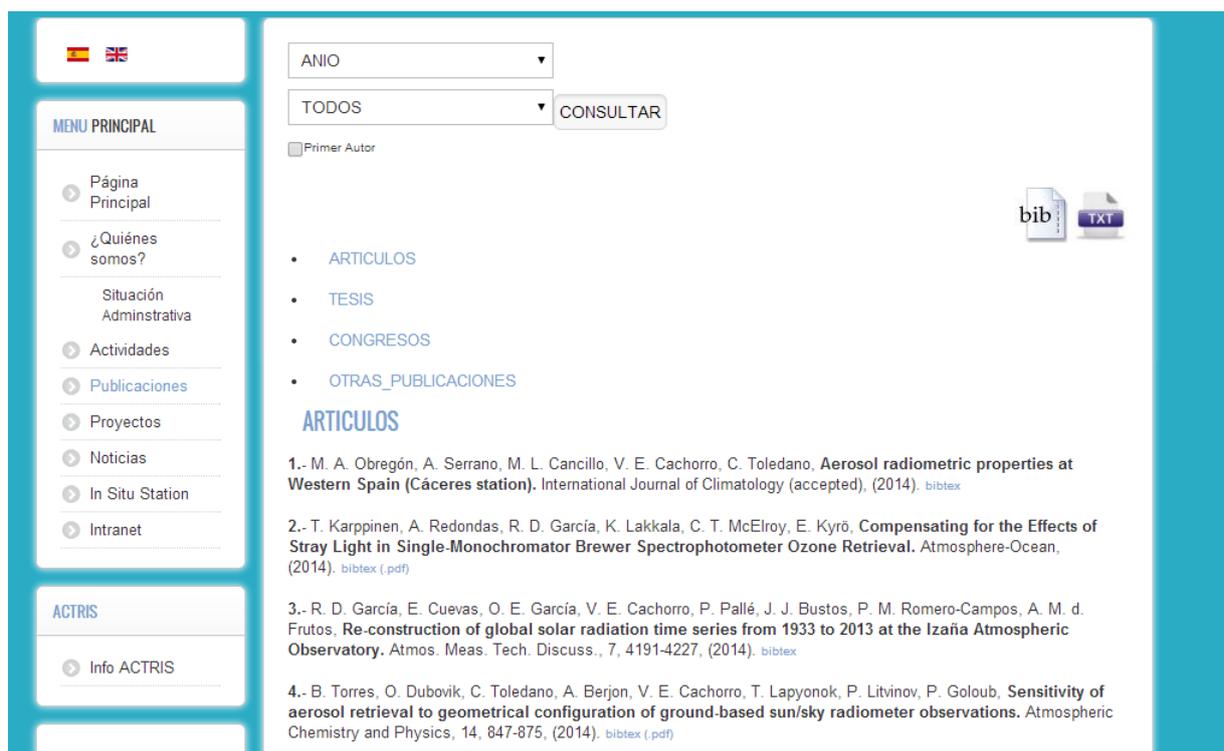
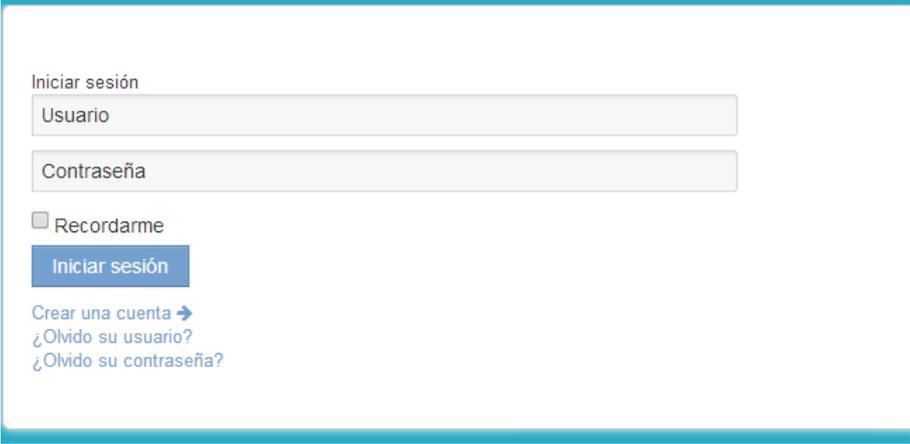


Figura A.11: componente ver artículos

Como podemos nos muestra una lista con todos los artículos, tesis doctorales, congresos y otras publicaciones. Arriba vemos que podemos filtrar la búsqueda por año y por autor, e incluso que al filtrar por autor solo salgan los que figuran como primer autor. Además vemos dos botones a la derecha arriba. Uno nos mostrará en formato **bibtex** cada artículo en forma listado y el otro en formato **txt** listado tal cual.

También se puede ver en esa imagen que debajo de cada artículo hay un enlace que pone **bibtex** el cual nos llevará al bibtex de ese artículo en concreto. Donde una vez iniciada sesión en el sistema podremos editar ocultar o borrar el artículo.

Estructura Zona Inicio de Sesión



The image shows a login and registration form with the following elements:

- Section title: "Iniciar sesión"
- Input field: "Usuario"
- Input field: "Contraseña"
- Checkbox: "Recordarme"
- Button: "Iniciar sesión"
- Link: "Crear una cuenta →"
- Link: "¿Olvido su usuario?"
- Link: "¿Olvido su contraseña?"

Figura A.12: Zona de Login y registro

En la imagen superior (A.12) podemos ver la zona para ingresar como usuario registrado y un enlace para registrarte, si estamos registrado rellenaremos usuario y contraseña para entrar en el sistema y si no estamos registrados pincharemos en **Crear una cuenta** y nos llevará a esta imagen (A.13) donde rellenaremos los campos pertinentes y nos registraremos esperando a que nos llegue el email de confirmación y luego nos acepte el administrador del sistema.



Registro de Usuario

* Campo obligatorio

Nombre: *

Usuario: *

Contraseña: *

Confirmar Contraseña: *

Correo electrónico: *

Confirmar correo electrónico: *

Figura A.13: Formulario de registro

Estructura Páginas Usuario Autenticado

Una vez iniciada la sesión la página principal habrá cambiado mostrándonos un menú más amplio y con más opciones como se ve en la siguiente imagen ([A.14](#)):



Figura A.14: Menú usuario autenticado

En el menú principal salen las mismas opciones y dos opciones a mayores meramente informativas. Pero dentro de **Publicaciones** se muestra nuevas opciones a tener en cuenta.



Figura A.15: Opción CICYT

Vemos en la imagen (A.15) que junto a las ya mencionadas opciones de **bibtex** y **txt**, aparece una nueva de **cicyt**. Esta opción nos mostrará la lista de todos los artículos con un formato diferente.

La otra opción como comenté antes está dentro del bibtex de cada artículo, donde nos aparecerá si somos un administrador o la persona que guardó el artículo en la base de datos las opciones de editar y ocultar el artículo, la persona que lo guardó en la base de datos tiene a mayores un botón para borrar el artículo.

```
@ARTICLE{asfafs,
  author = { , },
  title = {},
  year = {0},
  Times_Cited = {0},
  Impact_Factor = {0},
  lastUpdated = {2014-07-02 08:24:13 by prueba},
}
```

editar borrar ocultar

Figura A.16: Opciones en bibtex artículo

Si elegimos la opción de editar el artículo nos llevará a otra vista que podemos ver a continuación (A.17) con todos los campos que tiene un artículo para poder modificarlo:

BibtexKey:	asfafs
Author:	
Título:	
Journal:	
Year:	0
Volume:	
Pages:	
Number:	0
Month:	
EID:	
Note:	
Abstract:	
Crossref:	
Keywords:	
Doi:	

Figura A.17: Edición artículo

Dentro del menú de usuario aparte de opciones como para modificar el perfil del usuario entre otras, caben destacar dos elementos del menú **Introducir artículo** y **Rellenar autores**.

La opción de **Introducir artículo** nos muestra 4 pestañas principales que son **Formulario**, **Bibtex**, **Archivo** y **DOI**. Dentro de formulario tiene otras 4 opciones según lo que quieras introducir **Artículo**, **Tesis Doctoral**, **Congreso** y **Otras publicaciones**. Simplemente rellenaremos los campos que queramos que se guarden en la base de datos y pincharemos en el botón enviar (A.18):

Formulario Bibtex Archivo DOI

Rellena cada uno de los campos con los valores deseados:

Artículo Tesis Doctoral Congresos Otras Publicaciones

BibTexKey:

Author:

Título:

Journal:

Year:

Volume:

Pages:

Number:

Month:

EID:

Note:

Abstract:

Crossref:

Keywords:

Doi:

Url:

Citeeeerurl:

Comment:

Times_Cited:

Impact_Factor:

Owner:

Enviar

Figura A.18: Guardar artículo en la base de datos

En la opción de **bibtex** rellenaremos el cuadro de texto con el artículo en dicho formato (A.19):

Formulario Bibtex Archivo DOI

Introduce texto en formato bibtex para añadir el artículo:

Texto bibText:

Añadir

Figura A.19: Guardar artículo formato Bibtex

En la opción de **Archivo** seleccionaremos un archivo con extensión `.bib` que contendrá varios artículos en formato bibtex (A.20):

Figura A.20: Guardar varios artículos con formato bibtex

Y por último la nueva opción añadida a este componente es la de poder introducir artículos a la base de datos mediante su **DOI**. Simplemente introducimos ese numero identificativo del artículo y hará una búsqueda a la base de datos del WOS para tratarlo y meterlo en nuestra base de datos (A.21):

Figura A.21: Guardar artículo introduciendo el DOI

La otra opción a resaltar del menú de usuario es la de **Rellenar Autores** que nos permitirá enlazar los nombres que los usuarios introducen al meter un artículo con los que tenemos almacenados como autores en nuestra base de datos para que haya un solo nombre oficial (A.22):

pseudonimo	nombre	ignorar	
Romero-Campos, P.M., Ramos, R.	-Ninguno-	<input type="checkbox"/>	Actualizar
Frette, O.	-Ninguno-	<input type="checkbox"/>	Actualizar
Litvinov, P.	-Ninguno-	<input type="checkbox"/>	Actualizar
García, O.E.	-Ninguno-	<input type="checkbox"/>	Actualizar
Romero-Campos, P.M.	-Ninguno-	<input type="checkbox"/>	Actualizar

Figura A.22: Rellenando autores

I.a.4. Añadir artículos en diferentes idiomas

En esta sección se va a mostrar como el administrador desde el backend puede añadir los diferentes artículos para luego publicarlos en la página, tanto en inglés como en español.

Primero nos logueamos en la parte de la administración que se puede acceder a ella a través de la dirección web de la página añadiéndole /administrator y llegaremos a una página como esta donde iniciaremos sesión (A.23):



Figura A.23: Inicio como administrador en el backend

Una vez dentro del backend nos dirigiremos al menú superior de la interfaz, concretamente a la sección **Contenido-Gestor de Artículos** (A.24):

Estado	Título	Acceso	Asociación	Autor	Idioma	Fecha
<input checked="" type="checkbox"/>	In Situ EN (Alias: in-situ) Categoría: Categoría EN	Registrado	ES	Super Usuario	English (UK)	15-05-2014
<input checked="" type="checkbox"/>	Espectroradiometría y Teledetección EN (Alias: espectroradiometria-y-teledeteccion) Categoría: Categoría EN	Registrado	ES	Super Usuario	English (UK)	15-05-2014
<input checked="" type="checkbox"/>	Medida Aerosoles en Columna EN (Alias: medida-aerosoles-en-columna) Categoría: Categoría EN	Registrado	ES	Super Usuario	English (UK)	15-05-2014

Figura A.24: Sección para añadir artículos

Para añadir un nuevo artículo simplemente pinchamos en el botón verde de nuevo y nos llevara a la pantalla que nos permitirá añadir un artículo, ya sea en ingles, en español y en la categoría de noticias (A.25):

Figura A.25: Nuevo artículo

En esta pantalla podemos ver múltiples opciones a la hora de crear un artículo pero vamos a comentar solo las más importantes a la hora de crear un artículo y poder asociarlo con ese mismo artículo en otro idioma, y que, cambiando de idioma la página web nos lleve directamente a ese artículo en el idioma deseado.

A la derecha de la imagen superior vemos la categoría; seleccionaremos la categoría en la que queramos que esté, si es español la categoría será **Categoría ES**, si es inglés la categoría será **Categoría EN** y por último si queremos que nos salga en la sección de noticias seleccionaremos la categoría **Noticias**.

El estado es la segunda opción importante ya que si no está en **Publicado** nunca lo veremos en nuestra página web.

Y luego tenemos la opción **Idioma** que tendremos que seleccionar el idioma en el que hemos escrito el artículo, en nuestro caso solo tenemos disponible inglés y español.

En el cuadro situado a la izquierda escribiremos el contenido del artículo, con el formato que creamos necesario.

Otra opción muy importante está situada en la sección **Asociaciones** en la que seleccionaremos el artículo homólogo que estamos añadiendo pero en otro idioma (A.26):

Figura A.26: Ventana de asociación de un artículo con su homólogo en otro idioma

Una vez hecho esto simplemente pondremos título al artículo y pulsaremos el botón de **Guardar y cerrar** situado en la parte superior y ya habremos creado un artículo, haremos este proceso tantas veces como artículos queramos añadir.

Para poder publicar un artículo volveremos a utilizar el menú superior y seleccionaremos **Menú-Gestor de Menu** (A.27):

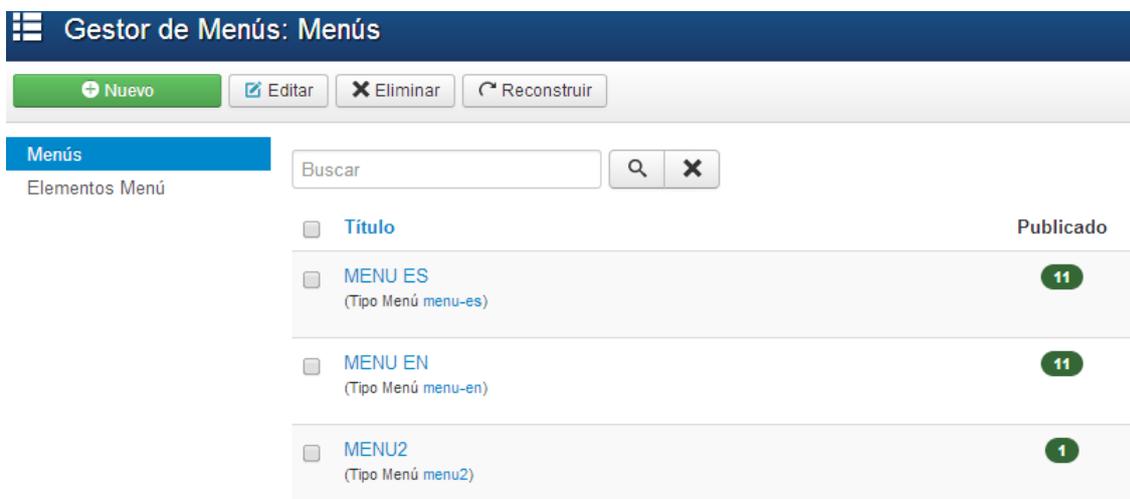


Figura A.27: Ventana de gestión de menús

Elegiremos el menú correspondiente, que como se ve hay dos menús de cada, uno para cada idioma y al seleccionar en el que queremos que aparezca pincharemos en el elemento de menú al que queramos enlazar el artículo (A.28):

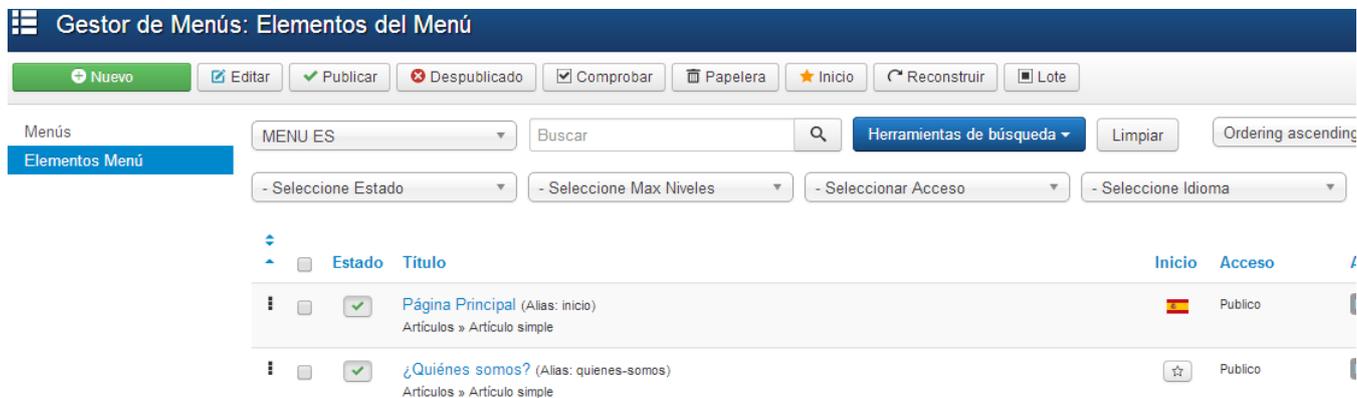


Figura A.28: Ventana elementos del menú

Una vez seleccionado el elemento del menú que queremos, nos llevara a la pantalla en la que configuraremos que enlace al artículo que hemos creado antes, para ello solo tendremos que seleccionar **Artículo Simple** en **Tipo de Elemento de Menú** y seguidamente en **Seleccionar el Artículo** elegiremos el artículo que queremos que aparezca, en las opciones de la derecha comprobaremos que es correcto el menú en el que queremos que aparezca y que el idioma es correcto. Y como ocurrió en la ventana de crear artículo en la pestaña **Asociaciones** seleccionaremos el elemento de menú homologa en el otro idioma (A.29):

Aplicar Guardar & Cerrar Guardar & Nuevo Guardar como Copia Cerrar

Título Menú * ¿Quiénes somos? Alias:

Detalles Opciones Artículo Tipo de opciones para los Enlaces Opciones de visualización para la página

Tipo Elemento Menú * Artículo simple Seleccionar

Seleccionar el Artículo * quienes somos ES Seleccionar Editar

Enlace index.php?option=com_content&vie

Ventana Destino Seguir igual

Plantilla Estilo - Usar por defecto

Figura A.29: Ventana selección de opción para que muestre el elemento de menú

Para finalizar haremos click en guardar y cerrar y nuestro artículo estará publicado y listo para poderlo consultar en nuestra página.

Apéndice B

Contenido del CD

Todos los ficheros que se han generado a lo largo del desarrollo del TFG han sido incluidos en el CD-ROM que ha sido entregado. He aquí una breve descripción de su árbol de directorios:

- **Aplicación:** En este directorio se encuentra un .zip con un backup de sitio web generado con Joomla!, ya configurado y con toda la estructura de la página, listo para poder ser instalado en cualquier servidor.
- **Componentes:** En este directorio se encuentran 3 carpetas con el nombre de cada uno de los componentes y dentro de cada una de ellas un .zip con cada uno de los 3 componentes creados para esta versión de Joomla!. Pueden ser instalados en cualquier servidor que tenga Joomla! 3.x
- **Diagramas:** En este directorio se incluyen todos los diagramas generados, ya sea con el Astah o con el Microsoft Project '13.
 - **Diagramas Astah:** Contiene los dos ficheros .asta , el que contiene todos los diagramas de los componentes y el que contiene los diagramas de lo que es Joomla! en sí. y una carpeta con todas las imágenes sueltas.
 - **Diagramas Project:** Contiene el fichero .mpp correspondiente al project con los diagramas de Gantt.
- **Memoria:** En esta carpeta se encuentra la memoria del TFG, incluyendo como apéndice los manuales de instalación y usuario.