

MAGICO

“Sistema de Mejora Automatizada para la Gestión de la Inteligencia y Conocimiento en la Organización”

Juan Antonio Montaña Pérez
Grado en Ingeniería Informática

Margarita Hospedales Salomó

Junio de 2014



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	MAGICO “Sistema de mejora automatizada para la gestión de la inteligencia y conocimiento en la organización”.
Nombre del autor:	Juan Antonio Montaña Pérez
Nombre del consultor:	Margarita Hospedales Salomó
Fecha de entrega (mm/aaaa):	06/2014
Área del Trabajo Final:	Gestión del Conocimiento
Titulación:	<i>Plan de Estudios del Estudiante</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>La información es quizás el principal activo empresarial en la actualidad. Durante décadas se ha tratado obtener un acceso directo e instantáneo a la misma, a la vez que se pretendía su integridad y seguridad. Por este motivo, han sido mucho los intentos por parte de las empresas de implantar sistemas de gestión de bases de datos, software de gestión empresarial o de gestión documental. Sin embargo estos intentos no han hecho si no ahondar en los principales problemas</p> <p>Este Trabajo Final de Grado versa sobre el desarrollo de una aplicación web con funcionalidades de Gestión Documental, Gestión de Tareas, Red Social y Gamificación. El Objetivo de la aplicación consiste en lograr una Mejora Automatizada de la Gestión de la Inteligencia y el Conocimiento en la Organización.</p>	

Abstract (in English, 250 words or less):

Information is perhaps the main business asset today. For decades we have tried for direct and instant access to it, while its integrity and security was intended. For this reason, there have been many companies attempts to implement database management systems, business management software and document management software. However, these attempts instead of being successfull have digged into the main problems.

This Final Project deals with the development of a web application with Document Management, Task Management, Gamification and Social Network functionalities. The goal of the application is to achieve an Automated Improvement of the Intelligence and Knowledge Management in the Organization.

Palabras clave (entre 4 y 8):

Gestión, Documento, Conocimiento, Inteligencia, Tareas, Social, Gamificación

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	3
1.3 Enfoque y método seguido.....	8
1.4 Planificación del Trabajo.....	19
1.5 Breve resumen de productos obtenidos.....	24
1.6 Breve descripción de los otros capítulos de la memoria.....	25
2. Resto de capítulos.....	26
2.1 Especificación de Requisitos.....	26
2.2 Especificación de la base de datos.....	37
2.3 Arquitectura del sistema.....	43
2.4 Diseño Web.....	50
2.4.1 Logotipo.....	50
2.4.2 Interfaz.....	51
2.5 Implementación.....	54
2.6 Valoración económica del proyecto.....	58
3. Conclusiones.....	60
4. Glosario.....	63
5. Bibliografía.....	65
6. Anexos.....	66

Lista de figuras

Ilustración 1: Sistema de Información	5
Ilustración 2: Fuente de valor	6
Ilustración 3: Planificación	21
Ilustración 4: Diagrama de Red	45
Ilustración 5: Ciclo de petición	49
Ilustración 6: Ejemplo de Interfaz	51
Ilustración 7: Página de Login	52
Ilustración 8: Pantalla de Cpanel	56
Ilustración 9: Herramientas de Bases de datos de Cpanel	56
Ilustración 10: Formulario de MySql	57
Ilustración 11: Ejemplo de PHPMyadmin	57

1. Introducción

1.1 Contexto y justificación del Trabajo

La Información, tal y conforme se detalla en el ensayo de divulgación de la Fundación BBVA “El impacto de las tecnologías de la información y la comunicación (TIC) en el crecimiento económico español” publicado en Febrero de 2007, es quizás el principal activo empresarial en la actualidad. Durante décadas se ha tratado obtener un acceso directo e instantáneo a la misma, a la vez que se pretendía su integridad y seguridad. Por este motivo, han sido mucho los intentos por parte de las empresas de implantar sistemas de gestión de bases de datos, software de gestión empresarial o de gestión documental. Sin embargo estos intentos no han hecho si no ahondar en dos de los principales problemas a los que se enfrenta una empresa a la hora de optimizar su gestión del Conocimiento, la fragmentación de los datos y el exceso de integración de software, tal y como se detalla en los textos de “Enterprise Integration” de Kent Sandoe, Gail Corbitt y Raymond Boykin (2 de abril de 2001) y que fue motivo de estudio en la asignatura de Integración de Sistemas del presente Grado en Ingeniería Informática.

La fragmentación de los datos viene provocada cuando disponemos de demasiadas bases de datos repartidas por diferentes departamentos. Por su parte, el exceso de integración de software ocurre cuando disponemos de demasiados productos de software que tienen supuestamente que funcionar juntos pero nunca fueron diseñados para ello.

Aparte de dichos problemas, a nivel interno las empresas sufren de falta de comunicación a distintos niveles. . Según la obra “The Essential Drucker” (2008), en la que se repasan las teorías de Peter

Drucker (19/11/1909–11/11/2005), el 60% de los problemas empresariales son consecuencia de una mala comunicación. Las empresas y los líderes necesitan de una buena comunicación para poder transmitir claramente sus ideas y proyectos. Esto que parece tan sencillo, muchas veces, es difícil alcanzar.

Los especialistas afirman que a mayor comunicación, mayor claridad, y a mayor claridad, mayor seguridad de lograr los objetivos. Pero, si este problema de comunicación ya es importante cuando se trata de hacer llegar nuestro mensaje a los clientes, mas importante es aún cuando el problema de comunicación es interno, ya que ni siquiera obtendríamos un mensaje final que transmitir a los clientes. ¿Cómo podría ayudar la gran capacidad comunicativa que poseen las Redes Sociales si las aplicamos en la empresa?

Por múltiples causas, tanto internas como externas, es común en muchas empresas una disminución en la motivación de sus empleados. En los últimos años se ha desarrollado una serie de técnicas consistentes en el empleo de mecánicas de juego en entornos y aplicaciones no lúdicas con el fin de potenciar la motivación, la concentración, el esfuerzo, la fidelización y otros valores positivos comunes a todos los juegos. En este sentido nos encontramos con el caso de éxito de Accenture España, consultoría de gestión, tecnología y outsourcing, decidió gamificar su gestión de recursos humanos para lograr una mayor participación e implicación por parte de sus empleados.

Su estrategia de Gamificación giró en torno a cinco programas:

1. **Accenture Land**, cuyo objetivo es recordar a la plantilla cuáles son los valores de la compañía.
2. **Tú eres el centro**, plataforma que recoge información sobre la propuesta de valor de la empresa,

3. **Tu Bienestar**, programa de salud dirigido a fomentar hábitos más saludables,
4. **Eco Challenge**, creado con la idea de fomentar un mayor compromiso medioambiental y
5. **ReValorízate**, plataforma de comunicación interna creada con el fin de lograr una relación más estrecha entre las diferentes jerarquía de la compañía.

Según datos aportados por Accenture España, la estrategia fue todo un éxito, logrando que la participación de la plantilla en las diferentes iniciativas de comunicación interna creciera un 20%.

Como solución a cada uno de los puntos anteriores podemos plantear una herramienta web con funcionalidades de gestión documental, tareas, recursos, red social empresarial y gamificación. Todo ello además aprovechando las características que nos brinda el Cloud Computing.

A lo largo de este proyecto se abordará el desarrollo de esta herramienta y su aplicación de cara a obtener una mejora en la Gestión del Conocimiento.

1.2 Objetivos del Trabajo

De cara a justificar y establecer los objetivos de proyecto, se debe previamente prestar atención a distintos términos que representarán los elementos base en torno a los que gira el proyecto.

En la actualidad la información juega un papel protagonista. Gracias a la comunicación en cualquiera de sus manifestaciones, la

información fluye de un lugar a otro y en espacios de tiempo cada vez más instantáneos. Las sociedades y los sistemas económicos gracias a las herramientas *TIC* (Tecnologías de la Información y Comunicación), son los principales motores generadores de información.

Como ejemplos del volumen de información que generamos ya en 2011 nos encontrábamos con que cada día se enviaban más de 188.000 millones de e-mails. Mientras, Facebook tenía 60 millones de actualizaciones o Twitter 140 millones de tweets. Estábamos hablando de 2.900 millones de cuentas de correo, una cantidad muy provocativa y digna de analizar en las corporaciones. (DATOS ENCONTRADOS EN LA WEB)

En materia de Sistemas de Información se consideran las definiciones de **Información, Sistema de Información, Conocimiento**.

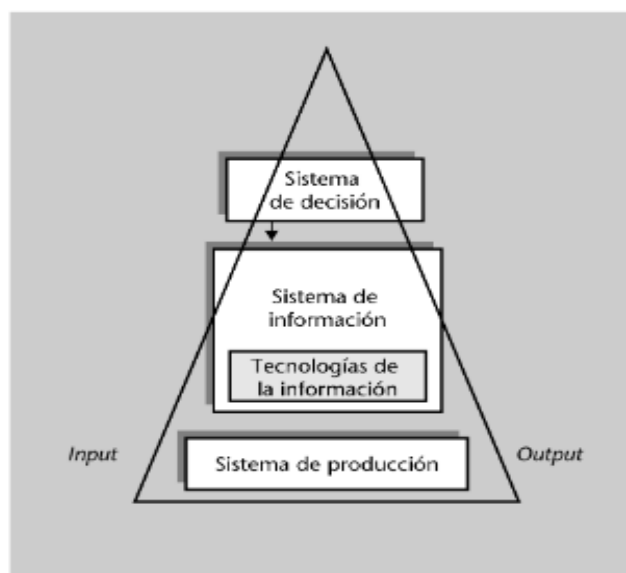
La **Información** constituye un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

Un **Sistema de Información** es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo. Dichos elementos formarán parte de alguna de las siguientes categorías:

- Personas
- Datos
- Actividades o técnicas de trabajo
- Recursos materiales en general (generalmente recursos informáticos o de comunicación aunque no necesariamente).

Todos estos elementos interactúan para procesar los datos (incluidos los procesos manuales y automáticos) y dan lugar a información más elaborada, que se distribuye de la manera más adecuada posible en una determinada organización, en función de sus objetivos. Este tipo de información ya más elaborada y las técnicas que surgen de la aplicación de la misma es lo que conocemos por **Conocimiento**.

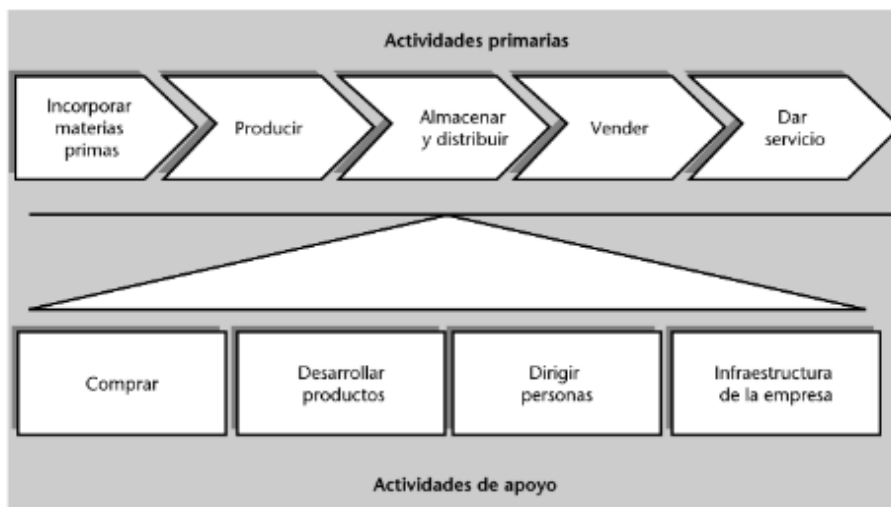
Las empresas con el fin de mantener su competitividad y productividad en los escenarios empresariales actuales se han visto obligadas a adaptarse a este nuevo escenario de la Sociedad de la Información y el Conocimiento.



Fuente: Pastor (s/d)

Ilustración 1: Sistema de Información

La figura anterior muestra como un Sistema de Información apoyado en las Tecnologías de la Información se sitúa en un lugar intermedio en el Sistema de Producción y el Sistema de Decisión lo cual lo convierte en un elemento clave alrededor de cual pivotan los demás sistemas empresariales. El sistema de Información además debe prestar apoyo a cada una de las actividades de la conocida Cadena de Valor tal y conforme se muestra en la siguiente ilustración.



Fuente: Andreu, Ricard, Valor (1996).

Ilustración 2: Fuente de valor

Por esta razón, la Gestión del Conocimiento comprendida como información más recursos humanos representará un nuevo paradigma de gestión empresarial durante los próximos años.

Sin embargo, a pesar del gran esfuerzo realizado por las empresas y del gran avance realizado en esta materia, éstas se enfrentan a distintos problemas que hace que su gestión del Conocimiento no se encuentre todo lo optimizada que se desearía.

Existen problemas como la dispersión o aislamiento de la información, mal reparto de las cargas de trabajo, falta de comunicación entre departamentos distintos que trabajan en un mismo proyecto u otros problemas como una progresiva desmotivación de los trabajadores.

Muchos de estos problemas se han tratado tradicionalmente de forma individual con mejor o peor resultado, pero hoy día podríamos aprovechar la tecnología existente para ofrecer una solución integral de apoyo de a la Gestión del Conocimiento en la empresa.

Este Trabajo de Final de Grado propone el Desarrollo de una Intranet Empresarial con herramientas de gestión documental, gestión de recursos, gestión de tareas, red social y gamificación con el fin de obtener una gestión documental centralizada, realizar seguimiento de tareas y actividades y mejorar la productividad de los empleados gracias a la red social y la gamificación.

1.3 Enfoque y método seguido

El enfoque pretendía una solución integral que englobara las siguientes características.

Gestión Documental

A pesar de que generamos millones de documentos ofimáticos a diario y podríamos llegar a pensar que la gestión documental está implícita en este hecho, la realidad es la siguiente respecto a la información:

- El 10% está en formatos informáticos estructurados.
- El 20% está en formatos informáticos no estructurados.
- El 70% está en soporte papel.
- Un volumen indeterminado, pero muy importante, está en la memoria de los trabajadores.

De cara a solucionar este hecho se plantea dentro del producto a desarrollar herramientas de gestión documental que ofrezcan las siguientes funcionalidades

- | |
|--|
| <ul style="list-style-type: none">- Almacenamiento centralizado- Clasificación y búsqueda por varios criterios- Control de versiones- Registro de actividad |
|--|

Gestión de Recursos

En empresas con cierto volumen de empleados y departamentos y una cantidad de recursos considerable tales como vehículos de

empresa, sala de juntas, ordenadores portátiles compartidos, proyectores para formación se suele dar el caso de que se cuente de ante mano con un determinado recurso de cara a realizar una tarea, pero, sin embargo, cuando llega el momento de disponer del mismo, este recurso está siendo utilizado por otra persona, por lo que es muy posible que la tarea no se pueda realizar según lo previsto.

En este sentido, el producto a desarrollar aportará soluciones a este conflicto a través de las siguientes funcionalidades:

- Base de datos de recursos
- Reserva de recursos mediante asignación libre
- Reserva de recursos mediante autorización
- Asignación de recursos a tareas
- Registro de actividad

Gestión de Tareas

Es habitual que en muchas empresas no se tenga claro quién debe realizar una tarea, qué plazo tiene para realizarla o a quién debe informar sobre el estado de la misma.

Por otro lado, también suele ser habitual cargar excesivamente de trabajo a determinados trabajadores de forma que el tiempo estimado de ejecución de las tareas diarias supere el 100% de la jornada laboral del trabajador, lo que hará que el trabajador no sea capaz de realizar todo el trabajo asignado en plazo y, por lo tanto, se la producción se verá afectada.

El producto a desarrollar aportará las siguientes soluciones a este conflicto a través de las siguientes funcionalidades:

- Seguimiento de Tareas
- Asignación de Tareas a Trabajadores

- Asignación de Tareas a Proyectos
- Notificaciones
- Monitor de Carga de Trabajo

Red Social

De cara a mejorar la comunicación interna de la empresa y de ofrecer un paso hacia la empresa 2.0, el producto a desarrollar dentro del ámbito de este proyecto incluirá características típicas de red social como las siguientes:

- Muro de actualidad
- Envío de mensajes
- Notificaciones

Gamificación

A pesar de que aún pocas empresas han apostado por procesos de **Gamificación**, se puede contrastar de que la Gamificación funciona, ya que se puede demostrar que las **dinámicas de juego** pueden aplicarse con éxito en cualquier proceso de la organización véase en Recursos Humanos o Producción.

En el ámbito de la Gamificación, una de las formas más comunes en las que se manifiesta la motivación de los trabajadores tras un proceso de Gamificación es la motivación extrínseca, la cual se identifica con los premios o las recompensas que obtenemos al participar en ciertos procesos.

En este sentido, el producto dispondrá de las siguientes funcionalidades en base a dotar de procesos de Gamificación a la herramienta final:

- | |
|--|
| <ul style="list-style-type: none">- Ranking- Sistema de puntuación basado en tareas |
|--|

De cara a realizar la aplicación y tras estudiar las tecnologías disponibles para dicho cometido se escogieron las tecnologías HTML5, CSS y JQuery, para el front end de nuestra aplicación y PHP para el backend, estando los datos almacenados en una base de datos Mysql.

HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 abre todo un abanico de posibilidades en cuanto a interactividad web y también convierte la web en una plataforma para albergar aplicaciones o web apps, tanto para PCs como para móviles y tablets.

Entre las ventajas del uso de HTML5 de cara a crear aplicaciones web nos encontramos las siguientes:

- HTML5 permite separar los elementos del sitio en dos planos, presentando un primer plano con las partes mas significativas y que menos tiempo tardan en cargar de manera completa.
- Permite la **renderización de las imágenes lo que incrementa significativamente las posibilidades de crear elementos gráficos de gran interacción para el consumidor**. Antes de esto solamente se podía insertar una imagen al código pero sin muchas posibilidades de modificarla mientras que las galerías eran enlaces a otras páginas o aplicaciones externas con varias limitantes.

- En la mayoría de los navegadores que utilizamos, el usuario debe tener instalado algún tipo de reproductor que le permita reproducir correctamente un video. Esto pronto será cosa del pasado ya que **solamente se necesitará un navegador que soporte HTML5 para poder disfrutar de todo tipo de contenido de video sin requerir el apoyo de otras aplicaciones** mejorando tiempo y estandarizando calidad de reproducción.
- La aplicación incluida en el HTML5 permite que la ubicación del usuario ya sea por GPS o por cualquier otro método esté disponible para el navegador. De esta manera se **incrementa significativamente la posibilidad de entregar contenido diferente dependiendo del idioma o la localización puntual del usuario** en el momento de acceder a un sitio sin tener que desarrollar una herramienta adicional para poderlo hacer.
- En el HTML5 se pueden guardar las aplicaciones para poder seguir trabajando aún cuando nuestros proveedores de Internet no cumplan con el servicio prometido.
- Gracias a este nuevo modelo se permite dar un mayor nivel de jerarquía a los elementos que así lo ameritan. **Se puede establecer con mayor claridad cada sección y así facilitar la vida a los buscadores dentro de nuestros sitios mejorando el desempeño SEO** de nuestro contenido.

La tecnología HTML5, suele ir de la mano de las hojas de estilo, en su última versión, CSS3. La especificación de CSS3 viene con interesantes novedades que permiten hacer webs más elaboradas y más dinámicas, con mayor separación entre estilos y contenidos. Da soporte de forma sencilla a muchas necesidades de las webs actuales. De hecho su utilización ofrece numerosas ventajas a la hora del desarrollo de entre las que destacaré las siguientes:

- Con una hoja de estilo es posible alterar la presentación de cada elemento sin tocar el código HTML, ahorrando esfuerzo y tiempo de edición.
- El lenguaje de las CSS posee y ofrece una gran variedad de herramientas de composición más potentes que HTML. Con HTML, el tamaño de la fuente se especifica con un sistema de medidas predeterminadas por el browser (en el ejemplo, **SIZE=5**), con las CSS hemos especificado el tamaño en puntos tipográficos (y podemos hacerlo en cm, pixeles, cuadratines, altura de la x, etc.). Más aún, las CSS permiten aplicar prácticamente todas las propiedades a cualquier elemento de la página, mientras que HTML sólo permite un número limitado de propiedades para cada elemento.
- Se evita tener que recurrir a trucos para conseguir algunos efectos. Con CSS no es necesario usar imágenes invisibles para hacer una sangría (la propiedad **text-indent** se encarga de eso) o usar una tabla para ubicar un elemento en determinado lugar de la pantalla (las CSS permiten posicionar con precisión cualquier elemento).
- El lenguaje de las Hojas de Estilo, aunque muy potente, es relativamente sencillo y fácil de aprender.

- Los documentos que usan CSS generalmente resultan más compactos.
- Las Hojas de Estilo pueden aplicarse de varias maneras y combinarse formando una cascada de estilos con la información de cada una.
- Pueden usarse con otros lenguajes de programación (como JavaScript) para conseguir efectos dinámicos en las páginas.
- Se pueden especificar Hojas de Estilo para diferentes navegadores y tipos de medios (impresos, braille, auditivos, etc.).
- El usuario con alguna discapacidad (o simplemente por preferencias) puede definir su propia Hoja de Estilo y la regla **!important** obliga a su navegador a suplantar la Hoja de Estilo del autor.

Javascript es un lenguaje de script o guión que nos permite mejorar infinitamente la interactividad y las características de la web. Su framework más conocido es jQuery, el cual facilita enormemente su utilización. Además nos añade la función ajax() que simplifica la integración de Ajax en nuestra aplicación web.

Recordemos que AJAX (*Asynchronous JavaScript And XML*) es una técnica para la creación de páginas web de forma rápida y dinámica. Permite que las páginas web se actualicen de forma asíncrona mediante el intercambio de pequeñas cantidades de datos con el servidor en segundo plano y sin necesidad de recargar la página web. Las webs que

no utilizan AJAX deben volver a cargarse si ha de cambiarse el contenido.

AJAX se basa en estándares de internet, y utiliza una combinación de:

- XMLHttpRequest (para intercambiar datos de forma asíncrona con el servidor).
- JavaScript / DOM (para mostrar / interactuar con la información)
- CSS (estilo de los datos)
- XML (utilizado a menudo como el formato de transferencia de datos)

Para el back end se utilizó PHP sobre todo por su conectividad ya que dispone de una amplia gama de librerías. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML. Como lenguaje de script de propósito general especializado para el desarrollo web, dispone de las siguientes ventajas:

- **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, esta listo para ser utilizado.

- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini
- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Respecto a la base de datos, opté por Mysql. **MySQL** es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones, según Wikipedia. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009. Las principales ventajas que encontramos gracias al uso de Mysql son entre otras:

- Gran integración en sistemas web basados en servidores web Apache y Php tanto en su versión Linux, como Windows. Lo que comúnmente se denomina sistema WAMP en el caso de Windows (Windows

Apache Mysql PHP) o LAMP (Linux Apache Mysql PHP) para Linux.

- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- Soporte a grandes bases de datos. Usamos MySQL Server con bases de datos que contienen 50 millones de registros. También conocemos a usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000.000.000 de registros.
- Se permiten hasta 64 índices por tabla (32 antes de la versión MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de la versión MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.

Por todas las razones anteriores se estimó oportuno el uso de HTML5, CSS3, Javascript y su framework JQuery, PHP y Mysql de cara al desarrollo de la aplicación web.

Por otro lado, al tratarse de una aplicación web se requería un servidor web para la publicación de la aplicación. En este caso, se utilizó un servidor virtual dedicado y el dominio appnware.net, ambos de mi propiedad.

1.4 Planificación del Trabajo

Para la consecución del proyecto se establecieron los siguientes hitos:

HITO 1: FASE 1

- Determinación del problema a resolver.
- Establecimiento de objetivos.
- Estudiar las aplicaciones similares en el mercado.
- Lectura del material
- Instalación del software necesario para la elaboración de la documentación del proyecto y para el desarrollo del proyecto en sí.
- Conocer la tecnología HTML5 de cara a fases más avanzadas del proyecto.
- Conocer la tecnología AJAX de cara a fases más avanzadas del proyecto.
- Entrega PAC 1.

HITO 2: FASE 2

- Repaso de los requisitos del enunciado de la propuesta y comparación con los requisitos actuales.
- Elaboración del análisis de requisitos.
- Diseño de diagramas UML pertinentes.
- Especificación de la base de datos.
- Profundizar sobre AJAX
- Profundizar sobre HTML5
- Sugerencias de Interfaces
- Entrega PAC 2.

HITO 3: FASE 3

- Realizar la codificación e implementación de la aplicación en función de los resultados obtenidos en la fase anterior.
- Integración del Interfaz elegido en la fase anterior con la implementación de la aplicación.
- Realización de pruebas de funcionamiento.
- Entrega PAC 3.

HITO 4: FASE 4

- Redacción de la memoria final del proyecto
- Elaboración de video de presentación del proyecto para su defensa ante el Tribunal.
- Entrega PAC 4

En base a los hitos anteriores se planificaron las tareas relativas a los mismos según la siguiente figura:

Nombre	Fecha de inicio	Fecha de fin
* Fase 1	28/02/14	19/03/14
◦ Determinación del problema a resolver	28/02/14	3/03/14
◦ Establecimiento de objetivos	4/03/14	6/03/14
◦ Estudio de aplicaciones similares	7/03/14	11/03/14
◦ Elaboración y Entrega PAC 1	12/03/14	19/03/14
◦ Instalación de Software	17/03/14	19/03/14
◦ Lectura del Material	28/02/14	19/03/14
◦ Estudio HTML 5	28/02/14	19/03/14
◦ Estudio AJAX	28/02/14	19/03/14
◦ Fase 2	20/03/14	23/04/14
◦ Análisis de Requisitos	20/03/14	25/03/14
◦ Repaso y Comparación de objetivos iniciales con actuales	26/03/14	26/03/14
◦ Diseño UML	26/03/14	1/04/14
◦ Especificación de Base de Datos	2/04/14	8/04/14
◦ Sugerencia de Interfaces	9/04/14	14/04/14
◦ Elaboración y Entrega PAC 2	15/04/14	23/04/14
◦ Profundizar sobre HTML 5	20/03/14	23/04/14
◦ Profundizar sobre AJAX	20/03/14	23/04/14
◦ Fase 3	24/04/14	23/05/14
◦ Codificación	24/04/14	13/05/14
◦ Integración de Interfaz	14/05/14	19/05/14
◦ Elaboración y Entrega PAC 3	20/05/14	23/05/14
◦ Fase 4	26/05/14	30/06/14
◦ Redacción Memoria Final	26/05/14	20/06/14
◦ Elaboración Video Final	26/05/14	20/06/14
◦ Elaboración y Entrega PAC 4	23/06/14	30/06/14

Ilustración 3: Planificación

Diagrama Gantt

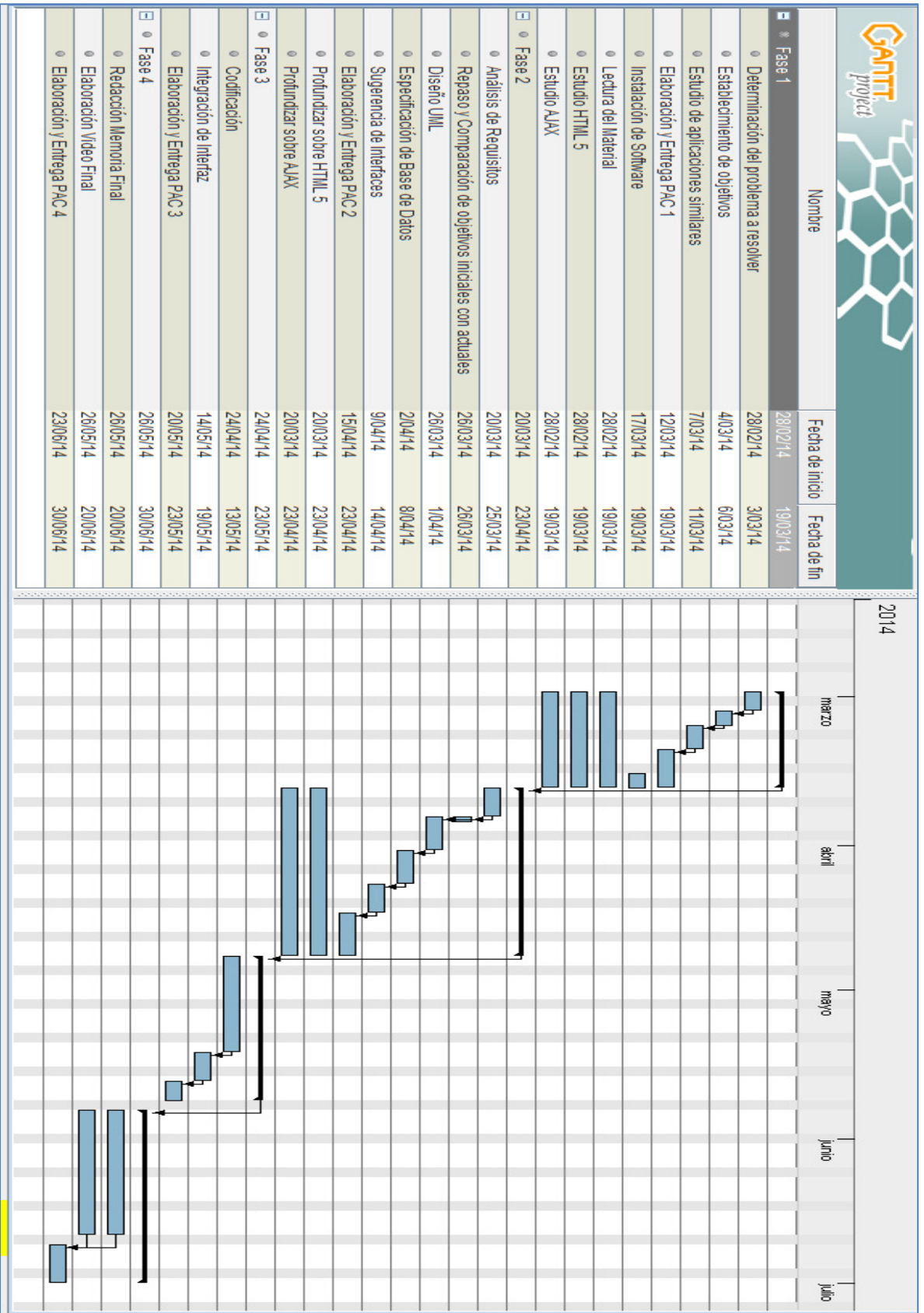
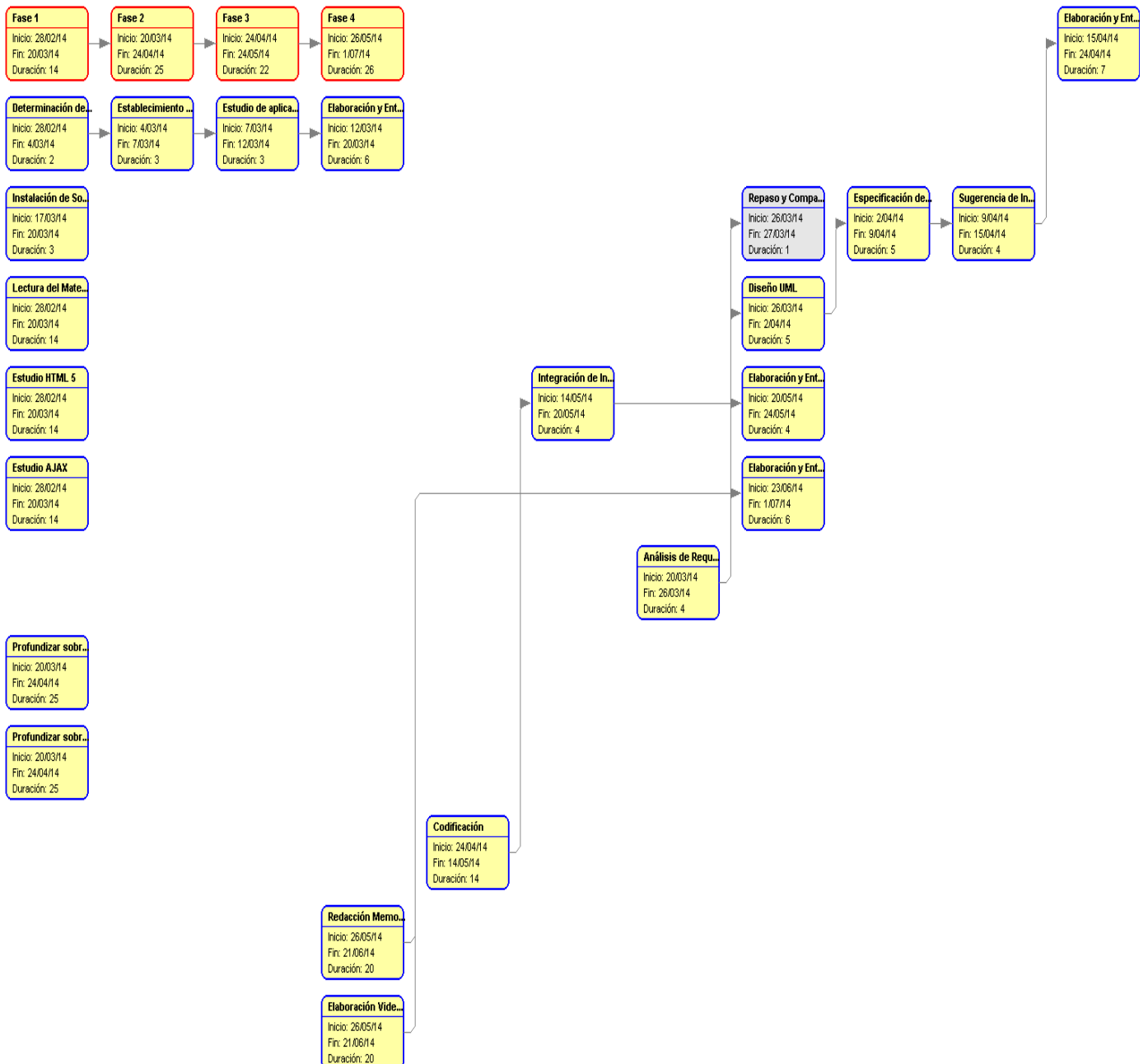


Diagrama PERT

A partir de lo hitos y el diagrama de Gantt anteriores se obtuvo el siguiente diagrama PERT:



1.5 Breve resumen de productos obtenidos

Como resultado a la elaboración de este proyecto se ha obtenido los siguientes productos.

- Una aplicación web orientada a la gestión del conocimiento en la empresa. El nombre clave del proyecto ha sido denominado MAGICO de los acrónimos de Mejora Automatizada para la Gestión de la Inteligencia y la Comunicación en la Organización.
- La aplicación en sí se puede considerar una herramienta para aplicar un sistema de gestión del conocimiento, por lo que podemos considerar que no sólo se ha obtenido la aplicación web, si no que adicionalmente ahora se cuenta con un sistema susceptible de ser aplicado de forma distinta a la reflejada por la aplicación web.
- Derivado del desarrollo web, también hemos obtenido el código fuente de la aplicación el cual se anexa a la entrega de la memoria.

1.6 Breve descripción de los otros capítulos de la memoria

A continuación, se detalla brevemente un listado del resto de capítulos de la memoria y la materia sobre la que versan.

2.1 Especificación de requisitos	Descripción completa de los requisitos funcionales y de definición de la aplicación a desarrollar.
2.2 Especificación de la base de datos	Detalle de como se ejecutó la creación de la base de datos y cada una de sus tablas.
2.3 Arquitectura del sistema	Estructura de la solución tecnológica presentada tanto a nivel hardware como a nivel software.
2.4 Diseño Web	Relata la creación de aspectos gráficos como la creación del logotipo, el interfaz final o las hojas de estilo.
2.5 Implementación	Ofrece información sobre el despliegue e implementación de la aplicación en el servidor web contratado. Asimismo, detalla los datos de acceso a la herramienta.
2.6 Valoración económica	Desglose de posibles costes asociados al desarrollo del proyecto por parte de una empresa.

2. Resto de capítulos

2.1 Especificación de Requisitos

Justo antes de comenzar con la especificación de proyecto comenté acerca del mismo con la dirección de mi empresa la cual se mostró interesada en implementar el sistema una finalizado y evaluado.

Por ese motivo, de cara a realizar un análisis de requisitos más exhaustivo, planteé el sistema al jefe de personal y administración de mi empresa, el cual se mostró muy interesado en participar del análisis de requisitos y disponer de la aplicación una vez desarrollada.

Se analizaron todos los requisitos de la propuesta inicial, su viabilidad de ponerlos en marcha en una empresa de servicios, la necesidades particulares de la empresa en cuestión y los requisitos propios de un proyecto de gestión del conocimiento, ámbito al que pertenece esta trabajo final de grado.

Se establecieron los siguientes requisitos de definición y de funcionalidad:

REQUISITOS DE DEFINICIÓN

RD1: Aplicación web. Las aplicaciones web para Internet e Intranet presentan una serie de ventajas con respecto al software de escritorio, con lo cual lograremos aprovechar y acoplar los recursos empresariales de una forma mucho más práctica que el software tradicional. (Conseguido)

RD2: Contraseña encriptadas. Los usuarios tendemos a estandarizar una contraseña para todas nuestras cosas en internet, pues resulta molesto tener 20 contraseñas distintas para 20 servicios distintos, por ejemplo. Por este motivo y como medida adicional de seguridad, se establece como requisito el realizar una encriptación de las contraseñas almacenadas en la base de datos por si se produjera un ataque capaz de tomar el control de los datos de acceso. (Conseguido).

RD3: Autenticación de usuarios. Debido a que el producto en el que trabajamos consiste en una herramienta empresarial, deberemos mantener una relación de identidades personales (usuarios) asociadas a un perfil de seguridad, roles y permisos. La autenticación de usuarios permite a estos sistemas asumir con una seguridad razonable que quien se está conectando es quien dice ser para que luego las acciones que se ejecuten en el sistema puedan ser referidas posteriormente a esa identidad y aplicar los mecanismos de autorización y/o auditoría oportunos. (Conseguido)

RD4: Roles de usuario. Cada usuario podrá acceder a una determinada información y funcionalidades en función del departamento al que pertenezca y a su rol de usuario en el sistema. (Conseguido)

REQUISITOS FUNCIONALES

RF1: Alta de actividad. Esta función registra una nueva actividad asociada a una tarea en el sistema. Los datos a registrar se recogen a través de un formulario relleno por el creador de la tarea. (Conseguido)

RF2: Modificación de actividad. Función a través de la cual el creador de la tarea puede modificar el contenido de las actividades

asociadas a la misma. Este requisito se logra cancelando la actividad y creando una nueva.

RF3: Eliminación de actividad. Esta función permite eliminar actividades de una tarea por parte del creador de la misma. (Conseguido)

RF4: Cambio de estado de actividad. Función que permite modificar el estado de una actividad. (Conseguido)

RF5: Alta de Usuarios. Función a través de la cual el administrador de la plataforma podrá dar de alta nuevos usuarios en el sistema, gracias a los datos suministrados a través de formulario. (Conseguido)

RF6: Modificación de Usuarios. Esta función realiza la modificación de un usuario de la plataforma a petición del administrador. (Conseguido)

RF7: Eliminación de Usuarios. Función que realiza la eliminación de un usuario a petición del administrador. (Conseguido)

RF8: Alta de Recursos. Esta función registra un recurso empresarial en el sistema a partir de los datos suministrados por el administrador. (Conseguido)

RF9: Modificación de Recursos. Función a través de la cual el administrador de la plataforma podrá modificar un recurso empresarial. (Conseguido)

RF10: Eliminación de Recursos. Esta función permite al administrador del sistema eliminar un recurso empresarial. (Conseguido)

RF11: Alta de Departamentos. Esta función registra un departamento empresarial en el sistema a partir de los datos suministrados por el administrador. (Conseguido)

RF12: Modificación de Departamentos. Función a través de la cual el administrador de la plataforma podrá modificar un departamento empresarial. (Conseguido)

RF13: Eliminación de Departamentos. Esta función permite al administrador del sistema eliminar un departamento empresarial. (Conseguido)

RF14: Asignación de Trabajadores a Departamentos. Función a través de la cual se asigna trabajadores de la plataforma a los departamentos existentes por parte del administrador. (Conseguido)

RF15: Editar Perfil de Usuario. Cada trabajador podrá editar su perfil de usuario a través de esta función. El Administrador puede editar cualquier perfil. (Modificado). Sólo el Administrador puede editar perfiles.

RF16: Creación de Nueva Tarea. Función que crea una nueva tarea en el sistema. (Conseguido)

RF17: Cambiar estado de Tarea. Función que permite determinar si una tarea está pendiente, en curso o finalizada. (Conseguido)

RF18: Modificar Tarea. Función que permite al creador de la tarea modificar el contenido de la misma. (Conseguido)

RF19: Crear Nuevo Documento. Función que registra un documento en el sistema de gestión documental y sube el fichero asociado al mismo a la plataforma. (Conseguido)

RF20: Crear Nueva Versión de Documento. Función que registra una nueva versión de un documento en el sistema de gestión documental y sube el fichero asociado al mismo a la plataforma. (Conseguido)

RF21: Crear Nueva Carpeta. Función que permite al administrador crear una nueva carpeta en el sistema de gestión documental. También permite a un trabajador crear una carpeta dentro de la carpeta asociada a su departamento. (Conseguido)

RF22: Asignar Recurso a Trabajador. Función que asigna un recurso a un trabajador durante un periodo tiempo determinado. (Eliminado)

RF23: Asignar Recurso a Departamento. Función que asigna recursos empresariales a departamentos. (Conseguido)

RF24: Mostrar Tareas pendientes. Esta función muestra en el panel principal las tareas pendientes del usuario. (Conseguido)

RF25: Mostrar Tareas cerradas. Función a través de la cual se pueden visualizar las tareas que un usuario ha cerrado. (Conseguido)

RF26: Mostrar Logros. Esta función muestra los logros de un usuario acorde al sistema de gamificación. (Eliminado)

RF27: Establecer Logros. Esta función sirve para que el administrador establezca los distintos tipos de logros que se pueden obtener en la plataforma. (Eliminado)

RF28: Enviar Mensaje. Función que envía un mensaje a un usuario, bien por parte de otro usuario o bien por parte de un proceso. (Conseguido)

RF29: Leer Mensaje. Función que permite a un usuario leer un mensaje recibido. (Conseguido)

RF30: Hacer comentarios. Función que permite a los usuarios hacer comentarios sobre un documento, tarea o actividad. (Modificado)
Los comentarios se hacen directamente en el muro de actualidad.

RF31: Eliminar Documentos. Función a través de la cual el administrador elimina un documento del sistema. (Eliminado)

RF32: Eliminar Carpetas. Función a través de la cual el administrador elimina una carpeta del sistema. (Eliminado)

RF33: Mostrar mensajes del muro. A través de la función se muestran mensajes en el muro de los usuarios. (Conseguido)

RF34: Insertar mensajes en el muro. Función que permite a un usuario publicar en el muro. (Conseguido)

RF35: Buscar documentos. Permitirá buscar documentos bajo el criterio de las etiquetas. (Conseguido)

RF36: Visualizar carga de trabajo de trabajadores. Función que permite visualizar la carga de trabajo a la que están sometidos los trabajadores. . (Conseguido)

Comparación de los requisitos iniciales con los resultados finales

Contraponiendo los requisitos establecidos en la propuesta inicial con los requisitos actuales nos encontramos con que prácticamente

todos los requisitos iniciales Se mantuvon intactos o si han sufrido alguna modificación, esta ha sido de carácter menor, por regla general.

Volviendo a la lista de requisitos inicial, en la tabla representada a continuación se detalla los cambios producidos del requisito inicial al actual.

- **Almacenamiento centralizado.** Se mantuvo. Se conseguirá al estar todos los documentos de la empresa almacenados en la base de datos de la aplicación a desarrollar.
- **Clasificación y búsqueda por varios criterios.** Cambia. Los documentos estarán clasificados en carpetas las cuales se crearán dentro de otras carpetas creadas por el administrador para cada departamento. La búsqueda se realizar
- **Control de versiones.** Se mantuvo. Los documentos no se borrarán de la plataforma si no que se crearán versiones.
- **Listas de distribución.** Se suprimió.
- **Registro de actividad.** Se mantuvo. Las actividades quedarán registradas en la plataforma y serán visibles por los coordinadores.
- **Base de datos de recursos.** Se mantuvo.
- **Reserva de recursos mediante asignación libre.** Se suprimió.

- **Reserva de recursos mediante autorización.** Se suprimió.
- **Asignación de recursos a tareas.** Se suprimió.
- **Seguimiento de Tareas.** Se mantuvo.
- **Asignación de Tareas a Trabajadores.** Se mantuvo.
- **Asignación de Tareas a Proyectos.** Se suprimió. Se establece como tarea lo que se describió como proyecto y se establece como actividad lo que se describió como tarea.
- **Notificaciones.** Se suprimió.
- **Monitor de Carga de Trabajo.** Se mantuvo.
- **Perfiles de trabajadores.** Se suprimió.
- **Muro de actualidad.** Se mantuvo.
- **Ranking.** Se mantuvo.
- **Sistema de puntuación en base a tareas realizadas.** Se mantuvo.

Como se puede comprobar en la tabla anterior y al listado final de requisitos, los requisitos iniciales apenas se vieron alterados y se añadieron funcionalidades adicionales como la definición de roles de

usuario y la gestión de departamentos. Ambos fruto de las reuniones mantenidas con la dirección de mi empresa actual.

Hubo que simplificar el sistema de red social y de notificaciones por el escaso tiempo de desarrollo para un proyecto de tal envergadura y posibilidades.

Encriptación de contraseñas

En la descripción del segundo requisito de definición se estableció la necesidad de realizar una encriptación de las contraseñas almacenadas en la base de datos, por si se produjera un ataque capaz de tomar el control de los datos de acceso.

De cara a cumplir con dicho requisito, se optó por el uso de MD5. **MD5** es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (*Massachusetts Institute of Technology*, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese una debilidad.

Los resúmenes MD5 se utilizan extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores. Esto protege al usuario contra los 'Caballos de Troya' o 'Troyanos' y virus que algún otro usuario malicioso pudiera incluir en el software. La comprobación de un archivo descargado contra su suma MD5 no detecta solamente los archivos alterados de una manera maliciosa, también reconoce una descarga corrupta o incompleta.

En sistemas UNIX y GNU/Linux se utiliza el algoritmo MD5 para calcular el *hash* de las claves de los usuarios. En el disco se guarda el resultado del MD5 de la clave que se introduce al dar de alta un usuario, y cuando éste quiere entrar en el sistema se compara el *hash* MD5 de la clave introducida con el *hash* que hay guardado en el disco duro. Si coinciden, es la misma clave y el usuario será autenticado. Los sistemas actuales GNU/Linux utilizan funciones de hash más seguras, como pueden ser SHA-2 o SHA-3.

El MD5 también se puede usar para comprobar que los correos electrónicos no han sido alterados usando claves públicas y privadas.

Al tratarse de un algoritmo tan extendido y probado lo hacen idóneo para el cometido de encriptar las contraseñas almacenadas en la base de datos del proyecto. La forma de encriptarlas la obtendremos gracias a la función md5() que PHP incorpora dentro de sus funciones nativas.

Solución al RD3

El tercer requisito de definición instaba al uso de cuentas de usuario para la validación y autenticación de los usuarios en la plataforma.

Normalmente, cuando trabajamos en entornos de Intranet, la web comprende una parte pública en donde encontraremos en alguna sección de la misma un formulario de acceso a la parte privada. Esta parte privada puede constar de una o varias páginas. La validación se obtiene comprobado si el nombre de usuario existe en una tabla de la base de datos y si la contraseña facilitada procesada con algún algoritmo de encriptación coincide con el hash de contraseña almacenado y asociado al nombre de usuario introducido.

Ahora bien, al tratarse de páginas webs, al fin y al cabo y debido al uso del protocolo HTTP para su visualización, requieren de una URL, alguien podría conocer las urls de las páginas internas y acceder a ellas

escribiendo la URL en el navegador sin pasar por el formulario de acceso. Para evitar esto lo que se hace es comprobar que usuario ha introducido el nombre de usuario y contraseña correctamente a través de un formulario y hacer la comprobación en cada página interna. Esto, efectivamente, solucionaría el problema, pero obligaría al usuario a estar validándose constantemente para cada página a visitar y destrozaría la experiencia de usuario y la funcionalidad de la aplicación. Sin embargo podemos aprovechar las sesiones de usuario que podemos crear con PHP y guardar en variables de sesión el usuario validado en el formulario con lo que dotaríamos a la aplicación de la seguridad requerida sin alterar la funcionalidad y experiencia de usuario deseada para la web.

Profesionalmente en los últimos años he tenido la oportunidad poner en práctica soluciones de autenticación de usuarios basados en sesiones y he encontrado de gran utilidad un script con licencia GPL llamado Autenticator, script que debido a mi experiencia previa utilicé para la realización de la aplicación que atañe al proyecto.

Autenticator se trata de un Script desarrollado Pedro Noves para la Gestión de Páginas restringidas sólo a Usuarios registrados, con nivel de acceso y gestión de errores en el Login. Ideal para zonas de Administración ó zona "solo usuarios" que requieran autenticación para acceder a las mismas.

Incluye administración de usuarios (altas/bajas/modificaciones), pero sin embargo esta funcionalidad no será utilizada en el proyecto, ya que he encontrado otras sistemas de gestión más eficientes para nuestro cometido.

El gestor de usuarios está programado en lenguaje PHP (optimizado para versiones $\geq 4.1.x$), usa motor de Base de datos Mysql y se basa en el uso de sesiones; los cuales deben estar habilitados en el servidor que se instale.

Solución al RD4

Cada usuario se marcará como Administrador dotándole del valor 0 en la base de datos, Coordinador, estableciéndolo como tal en la tabla de asignación a departamentos o Trabajador, asignándole el valor 5 en la tabla de usuarios de la base de datos.

En función del rol establecida podrá acceder a distintas secciones y funcionalidades del sistema.

2.2 Especificación de la base de datos

Tras la realización del diagrama de clases y llegada la hora de la realización del Diagrama de Entidad Relación (DER), debido a que en el diagrama de Clases presentado no ha representado las operaciones a realizar por cada clase y no haber aplicado Herencia, Polimorfismo, Asociación directa ni haber indicado los privilegios Público, Privado o protegido de los atributos, podemos afirmar que la representación de las entidades relacionales del DER coincidiría con la del diagrama de clases y por tanto, se obviará el mismo y se procederá a la especificación de la base de datos directamente.

Por lo tanto, en base a las entidades y relaciones encontradas se especifica la base de datos con las siguientes tablas:

```
-- phpMyAdmin SQL Dump
-- version 3.4.11.1
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 26-06-2014 a las 11:31:47
-- Versión del servidor: 5.0.96
-- Versión de PHP: 5.2.6
```

```

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `appnware_magico`
--
CREATE DATABASE `appnware_magico` DEFAULT CHARACTER SET
latin1 COLLATE latin1_swedish_ci;
USE `appnware_magico`;

-----

--
-- Estructura de tabla para la tabla `actividades`
--

CREATE TABLE IF NOT EXISTS `actividades` (
  `id` int(11) NOT NULL auto_increment,
  `titulo` varchar(255) character set latin1 NOT NULL,
  `descripcion` text character set utf8 collate
utf8_spanish_ci NOT NULL,
  `fecha_inicio` date NOT NULL,
  `fecha_limite` date NOT NULL,
  `fecha_cierre` date default NULL,
  `estado` int(11) NOT NULL,
  `tarea` int(11) NOT NULL,
  `usuario` int(11) NOT NULL,
  `puntos` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_spanish_ci AUTO_INCREMENT=15 ;

-----

--Estructura de tabla para la tabla `asignaciones_recursos`
-----

CREATE TABLE IF NOT EXISTS `asignaciones_recursos` (
  `id` int(11) NOT NULL auto_increment,
  `recurso` int(11) NOT NULL,
  `usuario` int(11) NOT NULL,
  `fecha_inicio` date NOT NULL,
  `fecha_final` date NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

-----
-- Estructura de tabla para la tabla `carpetas`
-----

CREATE TABLE IF NOT EXISTS `carpetas` (
  `id` int(11) NOT NULL auto_increment,
  `creadapor` int(11) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

-----
-- Estructura de tabla para la tabla `departamentos`
-----

CREATE TABLE IF NOT EXISTS `departamentos` (
  `ID` int(11) NOT NULL auto_increment,
  `departamento` varchar(255) NOT NULL,
  `coordinador` int(11) NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=14 ;

-----
-- Estructura de tabla para la tabla `documentos`
-----

CREATE TABLE IF NOT EXISTS `documentos` (
  `id` int(11) NOT NULL auto_increment,
  `titulo` varchar(255) NOT NULL,
  `ruta` varchar(255) NOT NULL,
  `etiquetas` varchar(255) NOT NULL,
  `fecha` date NOT NULL,
  `autor` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=24 ;

-----
-- Estructura de tabla para la tabla `documentosasociados`
-----

CREATE TABLE IF NOT EXISTS `documentosasociados` (
  `id` int(11) NOT NULL auto_increment,
  `idtarea` int(11) NOT NULL,
  `iddoc` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

-----
-- Estructura de tabla para la tabla `mensajes`
-----

CREATE TABLE IF NOT EXISTS `mensajes` (
  `id` int(11) NOT NULL auto_increment,
  `texto` varchar(255) NOT NULL,
  `autor` int(11) NOT NULL,
  `destinatario` int(11) NOT NULL,

```

```

    `fecha` datetime NOT NULL,
    `leido` smallint(6) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=13 ;

```

```

-----
-- Estructura de tabla para la tabla `mensajes_muro`
-----

```

```

CREATE TABLE IF NOT EXISTS `mensajes_muro` (
  `id` int(11) NOT NULL auto_increment,
  `texto` varchar(255) NOT NULL,
  `autor` int(11) NOT NULL,
  `fecha` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=81 ;

```

```

-----
-- Estructura de tabla para la tabla `perfiles`
-----

```

```

CREATE TABLE IF NOT EXISTS `perfiles` (
  `id` int(11) NOT NULL auto_increment,
  `idusuario` int(11) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  `apellidos` varchar(255) NOT NULL,
  `iddepartamento` int(11) NOT NULL,
  `email` varchar(255) NOT NULL,
  `telefono` varchar(9) NOT NULL,
  `foto` blob NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

```

```

-----
-- Estructura de tabla para la tabla
`pertenencia_departamento`
-----

```

```

CREATE TABLE IF NOT EXISTS `pertenencia_departamento` (
  `id` int(11) NOT NULL auto_increment,
  `departamento` int(11) NOT NULL,
  `usuario` int(11) NOT NULL,
  `coordinador` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

```

```

-----
-- Estructura de tabla para la tabla `recursos`
-----

```

```

CREATE TABLE IF NOT EXISTS `recursos` (
  `ID` int(11) NOT NULL auto_increment,
  `recurso` varchar(255) NOT NULL,
  `departamento` int(11) NOT NULL,
  PRIMARY KEY (`ID`)
)

```

```
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
-----  
-- Estructura de tabla para la tabla `tareas`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tareas` (  
  `ID` int(11) NOT NULL,  
  `titulo` varchar(255) NOT NULL,  
  `creadapor` int(11) NOT NULL,  
  `fechainicio` date NOT NULL,  
  `fechafin` date NOT NULL,  
  `estado` int(11) NOT NULL,  
  `descripcion` varchar(255) NOT NULL,  
  `puntos` int(11) NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
-----  
-- Estructura de tabla para la tabla `usuarios`  
-----
```

```
CREATE TABLE IF NOT EXISTS `usuarios` (  
  `ID` smallint(6) unsigned NOT NULL auto_increment,  
  `usuario` tinytext character set latin1 collate  
latin1_spanish_ci NOT NULL,  
  `pass` tinytext character set latin1 collate  
latin1_spanish_ci NOT NULL,  
  `nivel_acceso` smallint(4) unsigned NOT NULL default '0',  
  `nombre` varchar(255) character set latin1 collate  
latin1_spanish_ci NOT NULL,  
  `apellidos` varchar(255) character set utf8 collate  
utf8_spanish_ci NOT NULL,  
  `iddepartamento` int(11) default NULL,  
  `email` varchar(255) character set latin1 collate  
latin1_spanish_ci default NULL,  
  `telefono` varchar(15) character set latin1 collate  
latin1_spanish_ci default NULL,  
  `foto` blob,  
  `color` varchar(7) character set latin1 collate  
latin1_spanish_ci NOT NULL,  
  `puntos` int(11) NOT NULL,  
  PRIMARY KEY (`ID`),  
  UNIQUE KEY `ID` (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 PACK_KEYS=1  
AUTO_INCREMENT=8 ;
```

```
-----  
-- Estructura de tabla para la tabla `versiones`  
-----
```

```
CREATE TABLE IF NOT EXISTS `versiones` (  
  `id` int(11) NOT NULL auto_increment,  
  `ruta` varchar(255) NOT NULL,  
  `fecha` date NOT NULL,  
  `autor` int(11) NOT NULL,
```



```
`id_documento` int(11) NOT NULL,  
PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;  
  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT  
*/;  
/*!40101 SET  
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION  
*/;
```

2.3 Arquitectura del sistema

El sistema quedó constituido tanto por elementos hardware como software.

2.3.1 Arquitectura Hardware.

Al tratarse de una aplicación web empresarial, el diseño del sistema variará dependiendo de las particularidades propias de la web y de las necesidades empresariales de la empresa que lo implemente. En este sentido nos podemos encontrar con distintos escenarios:

- **Empresa monosede con acceso local y servidor web propio.** Representa el caso de una empresa con una única sede y cuyos empleados accederán al sistema únicamente desde los equipos dispuestos en dicha sede. Además la empresa opta por alojar la aplicación web en un servidor propio.
- **Empresa monosede con acceso mixto y servidor web propio.** En esta opción la empresa dispondrá, además de los trabajadores que acceden de forma local a la aplicación, un grupo de trabajadores con la necesidad de acceder de forma remota al sistema.
- **Empresa monosede con acceso remoto y servidor web externalizado.** Este escenario es similar al anterior, salvo por el hecho de que se decide utilizar un servidor externo a la empresa como alojamiento de la aplicación web. Al estar el servidor externalizado, los trabajadores,

independientemente de su localización, accederán de forma remota.

- **Empresa multisede con acceso mixto y servidor web propio.** Representa el caso en que la empresa disponga de distintas sedes. Para esta situación se contempla un acceso local para los usuarios ubicados en la misma sede en la que se encuentre el servidor y acceso remoto para los trabajadores de otras sedes o en movilidad.
- **Empresa multisede con acceso remoto y servidor web externalizado.** En esta opción todos los trabajadores, independientemente de la sede en la que se encuentre, accederán de forma remota a la aplicación ya que el servidor será externalizado.

Tal y conforme se comentó en la PAC3 del desarrollo de este TFG, en función de la estructura elegida tendremos las distintas especificaciones técnicas del sistema que influirían en la arquitectura del mismo. En concreto debido a las particularidades de proyecto, mi empresa y mi proveedor de hosting se optó por un **Arquitectura para Empresa multisede con acceso remoto y servidor web externalizado**

Esta arquitectura necesitará los siguientes elementos:

- Conexión a Internet. Conectará a los equipos remotos de cada sede con el servidor externalizado. Todas las sedes deberán disponer de conexión a Internet.

- Switch. Sirve para conectar varios elementos dentro de una red. En este caso, se necesitará un switch en cada sede.
- Cableado de red. Realiza la conexión física de los elementos de la red.
- Equipos Cliente. Supondrán el terminal de acceso de los usuarios a los servicios prestados por el servidor.
- Router. Gestionará las conexiones a internet de los equipos clientes. En este caso, se necesitará un router por cada sede que se quiera conectar.

El diagrama de red quedará representado por la siguiente ilustración:

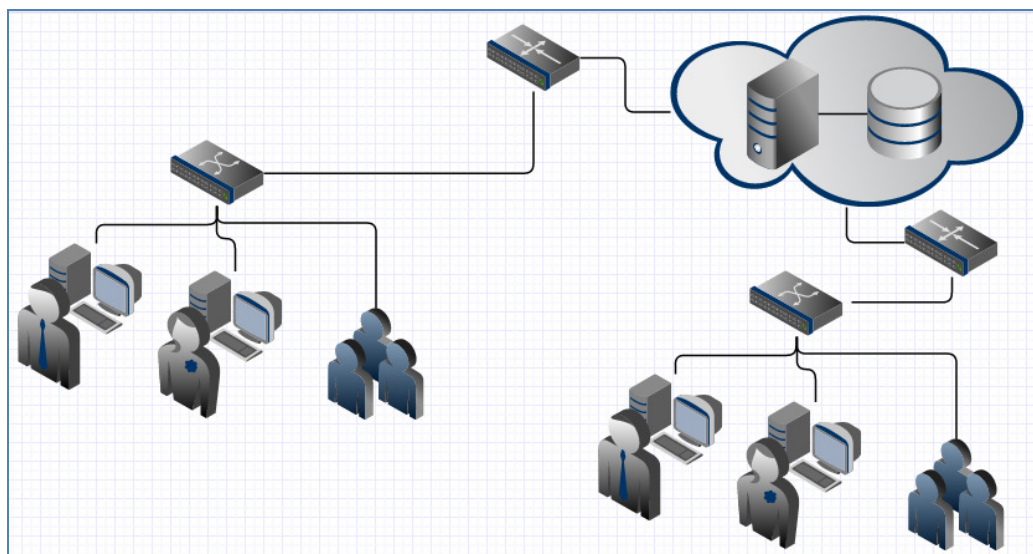


Ilustración 4: Diagrama de Red

Según la anterior ilustración, todos los usuarios se conectarían de forma remota. Además cualquier trabajador podría acceder desde cualquier ubicación bien de la empresa o ajena a ésta.

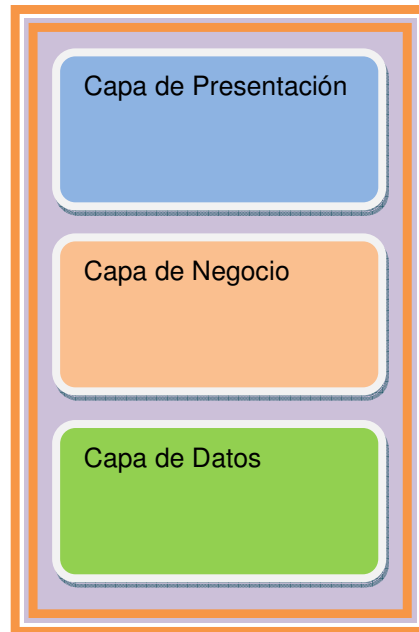
2.3.2 Arquitectura Software

La arquitectura software del sistema seguirá las pautas de un MVC. El MVC (**modelo–vista–controlador**) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

MVC propone la construcción de tres componentes distintos que son el **modelo**, la **vista** y el **controlador**. Por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. De hecho, MVC es considerado un patrón de diseño de software ampliamente comprobado que convierte una aplicación en un paquete modular fácil de mantener, a la vez que mejora la rapidez del desarrollo. La separación de las tareas de la aplicación en modelos, vistas y controladores hace que su aplicación sea además muy ligera. Ofrece la posibilidad de añadir nuevas funcionalidades a código viejo de manera sencilla. El diseño modular también permite a los desarrolladores y los diseñadores trabajar simultáneamente, incluyendo la capacidad de hacer prototipos rápidos.

Pero sin duda, el mayor beneficio que ofrece este modelo es el hecho de que la separación también permite a los desarrolladores hacer cambios en una parte de la aplicación sin afectar a las demás.

Basándonos en este patrón, la arquitectura software del sistema se desarrolló con la siguiente forma:



La capa de presentación comprendería a la interfaz de usuario y representaría la vista de la arquitectura MVC. La vista hace una presentación de los datos del modelo. Estando separada de los objetos del modelo, es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de cualquier petición que se presente.

En este nivel, al tratarse de una aplicación web las tecnologías HTML5 y CSS3, las cuales ya han sido mencionadas durante el desarrollo del presente Trabajo Final de Grado. En concreto, HTML5 sirvió para mostrar elementos como capas, tablas, párrafos, listas o imágenes. Por su parte, CSS3 se encargó de dar formato y estilo a los elementos HTML anteriores bien a través de definiciones de estilo insertadas en la misma página o a través de las hojas de estilo.

En la capa de negocio, la cual equivaldría a la negociación de la arquitectura MVC, se gestionan todas las peticiones que los usuarios realizan a la aplicación a través de la capa de vista. Es decir, es

responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista.

En esta capa se espera peticiones de los clientes, se comprueba su validez de acuerdo a las normas de autenticación o autorización, se delega la búsqueda de datos al modelo, y se selecciona el tipo de respuesta más adecuado según las preferencias del cliente. Finalmente, se delega este proceso de presentación a la capa de la Vista.

En el proyecto se utilizaron las tecnologías JQUERY y AJAX las cuales hicieron de enlace entre la interfaz y el modelo de datos de la aplicación.

Por último, el modelo representa la parte de la aplicación que implementa la lógica de negocio, lo que significa que es responsable de la recuperación de datos, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.

Los objetos del modelo suelen ser considerados como la primera capa de la interacción con cualquier base de datos. Pero, en realidad, representan los principales conceptos en torno a los cuales se desea implementar la aplicación.

En nuestro caso, la capa de modelo se hizo cargo de tareas como guardar el almacenamiento de asignaciones de recursos, el almacenamiento y la recuperación de fotos de los usuarios, encontrar documentos, etc. Mientras que los objetos del modelo pueden ser considerados como "Tarea", "Departamento" y "Documento", por ejemplo.

Como representación del ciclo de una petición a esta arquitectura podemos observar la siguiente figura:

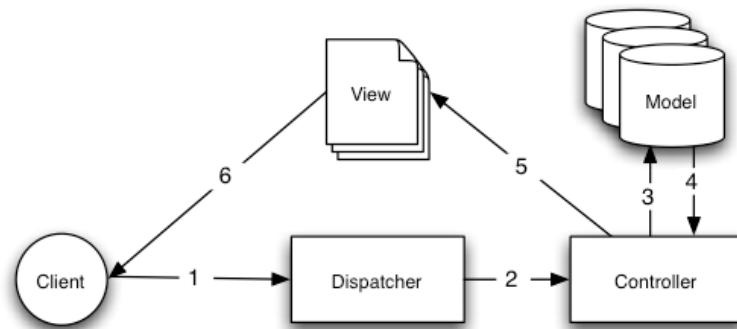


Ilustración 5: Ciclo de petición

El ciclo comienza cuando un usuario (Client) solicita una página o un recurso de la aplicación. Esta solicitud es procesada por un despachador (dispatcher) que selecciona el controlador correcto para manejarlo.

Una vez que la solicitud llega al controlador (Controller), este se comunicará con la capa del Modelo (Model) para cualquier proceso de captación de datos o el guardado de los mismos según se requiera. Una vez que finaliza esta comunicación, el controlador procederá a delegar en el objeto de vista correcto la tarea de generar una presentación (View) resultante de los datos proporcionada por el modelo.

Finalmente, cuando esta presentación se genera, se envía de inmediato al usuario.

Cualquier petición que se haga al MVC conllevará una consulta a la base de datos de la aplicación para la cual utilizaremos la tecnología propuesta por MySQL, la cual ya ha sido comentada con anterioridad junto con la especificación de las tablas propuestas derivadas del Diagrama Entidad Relación generado a partir del Diagrama de Clases UML propuesto.

2.4 Diseño Web

2.4.1 Logotipo

Se utilizó el siguiente logotipo para la aplicación:



El nombre MAGICO vino determinado por las siglas de Mejora Automatizada de la Gestión de la Información y Conocimiento en la Organización, concepto sobre el que versa el presente trabajo.

Se ha elijió el color azul ya que el azul es un color fresco, tranquilizante y se le asocia con la mente, a la parte más intelectual de la mente, al igual que el amarillo, tono al que nos acercamos con el naranja utilizado, que a su vez representa la alegría, hace sentir menos inseguro y estimula la mente fomentando la creatividad.

2.4.2 Interfaz

Finalmente se mantuvo el diseño propuesto como sugerencia de interfaz:

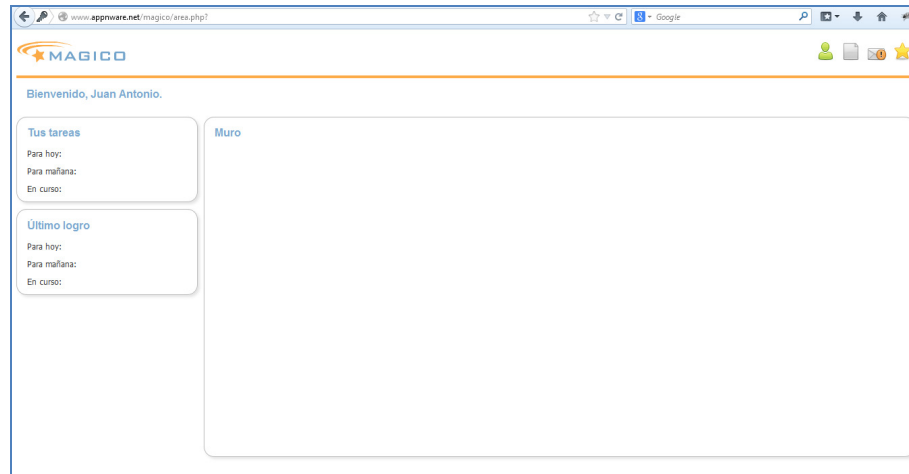


Ilustración 6: Ejemplo de Interfaz

El diseño, por tanto, mantuvo el predominio del color blanco que destacué en la PAC2 y que representaba el color más protector de todos, aporta paz y confort, alivia la sensación de desespero y de shock emocional, ayuda a limpiar y aclarar las emociones, los pensamientos y el espíritu.

También representa el optimismo y la perfección dos cualidades muy demandadas en la empresa.

2.4.3 Hojas de Estilo

Las Hojas de Estilo (o **CSS**, por Cascading StyleSheets) son un mecanismo que permiten aplicar formato a los documentos escritos en HTML (y en otros lenguajes estructurados, como XML) separando el contenido de las páginas de su apariencia. Para el diseñador, esto significa que la información estará contenida en la página HTML, pero este archivo no debe definir **cómo** será visualizada esa información. Las

indicaciones acerca de la composición visual del documento estarán especificadas en el archivo de la CSS.

A modo de ejemplo, ya que el contenido total de la especificación CSS para el proyecto es demasiado extensa, detallo el CSS final utilizado para la página de Login al sistema.

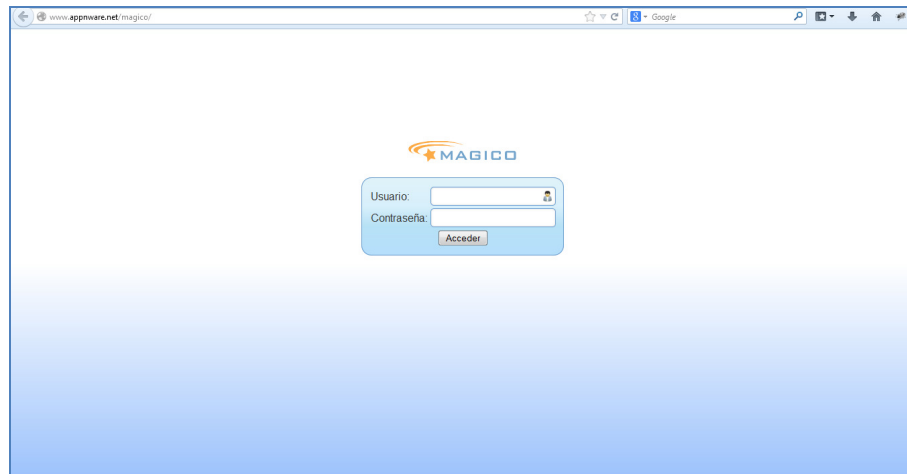


Ilustración 7: Página de Login

```
<style type="text/css">
```

```
.Estilo1 {  
    font-family: Arial, Helvetica, sans-serif;  
    color:  
;    color: #333;  
}  
#form1 {  
    text-align: center;  
}  
html, body  
{  
    height: 100%;  
    margin: 0px;  
    padding: 0px;  
}
```

```

#cuerpo{
  background: linear-gradient(0deg, #98C0FC, white 50%);
  height: 100%;
  margin: 0px;
}
#login{
background: rgb(224,243,250); /* Old browsers */
background: -moz-linear-gradient(top,  rgba(224,243,250,1)  0%,
  rgba(216,240,252,1)  37%,      rgba(184,226,246,1)  62%,
  rgba(182,223,253,1) 100%); /* FF3.6+ */
background: -webkit-gradient(linear, left top, left bottom, color-
stop(0%,rgba(224,243,250,1)),   color-stop(37%,rgba(216,240,252,1)),
color-stop(62%,rgba(184,226,246,1)),           color-
stop(100%,rgba(182,223,253,1))); /* Chrome,Safari4+ */
background: -webkit-linear-gradient(top,  rgba(224,243,250,1)
0%,rgba(216,240,252,1)          37%,rgba(184,226,246,1)
62%,rgba(182,223,253,1) 100%); /* Chrome10+,Safari5.1+ */
background: -o-linear-gradient(top,  rgba(224,243,250,1)
0%,rgba(216,240,252,1)          37%,rgba(184,226,246,1)
62%,rgba(182,223,253,1) 100%); /* Opera 11.10+ */
background: -ms-linear-gradient(top,  rgba(224,243,250,1)
0%,rgba(216,240,252,1)          37%,rgba(184,226,246,1)
62%,rgba(182,223,253,1) 100%); /* IE10+ */
background: linear-gradient(to  bottom,  rgba(224,243,250,1)
0%,rgba(216,240,252,1)          37%,rgba(184,226,246,1)
62%,rgba(182,223,253,1) 100%); /* W3C */
filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#e0f3fa', endColorstr='#b6dfffd',GradientType=0 ); /* IE6-9
*/
}
.cuadrouuario {
  background: url('img/usuario.jpg') no-repeat scroll right center
#FFFFFF;
}

```

```
border: 1px solid #69C;
border-radius: 5px;
box-shadow: 0 0 2px #DDD inset;
color: #666;
float: left;
padding: 5px 10px;
width: 165px;
outline: none;
}
.cuadro {
    background: #FFFFFF;
border: 1px solid #69C;
border-radius: 5px;
box-shadow: 0 0 2px #DDD inset;
color: #666;
float: left;
padding: 5px 10px;
width: 165px;
outline: none;
}
</style>
```

2.5 Implementación

Con el fin de mostrar el trabajo realizado y para servir de apoyo durante la etapa de desarrollo se alojó la aplicación en la carpeta “mágico” de un dominio de mi propiedad llamado appnware.net. La aplicación se encuentra, por tanto, en la ruta <http://www.appnware.net/magico>

El dominio apunta hacia un servidor virtual compartido contratado a la empresa HostingZoom con el plan Reseller Budget. Dicho plan de Hosting ofrece la siguientes características:

- 10 Gb de espacio en disco con tecnología RAID
- 100GB de transferencia mensual de ancho de banda.
- 25 cuentas cPanel. (sirven para administrar hasta 25 dominios en el espacio contratado)
- 15 tickets de soporte mensuales gratuitos.
- Scripts Softaculous, whm/cpanel (Sirven para añadir funcionalidades al servidor y administrar el mismo).
- Rvsitebuilder, un creador instantáneo de páginas web basado en plantillas.

Gracias a esta configuración obtuve un escenario similar al planteado como arquitectura para una empresa multisede con acceso remoto y servidor web externalizado, de forma que cualquier persona involucrada en la tutorización, evaluación o desarrollo de este proyecto pueden acceder a la aplicación independientemente de su ubicación.

2.5.1 Configurando el Hosting

De cara a administrar el servidor, crear la base de datos en el mismo y comenzar a desarrollar la aplicación se procedió a configurar el Hosting a través del script cPanel.

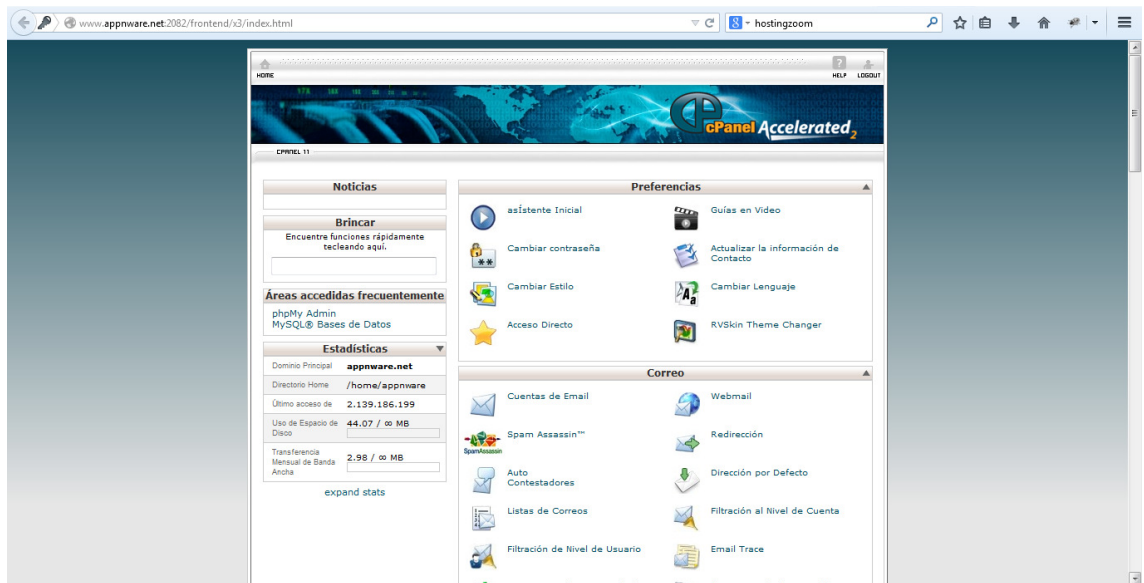


Ilustración 8: Pantalla de Cpanel

2.5.1.1 Creando la base de datos

En cPanel nos encontramos con el menú de base de datos siguiente:



Ilustración 9: Herramientas de Bases de datos de Cpanel

El sistema nos da opción a crear bases de datos Mysql o PostgreSQL. En nuestro caso seleccionaremos la opción Mysql Bases de Datos.

MySQL Usuarios [↑ Brincar a las Bases de Datos MySQL](#)

añadir Nuevo Usuario

Nombre Usuario: ✓

Contraseña: ✓

Contraseña (Otra vez): ✓

Fuerza (por qué?): Fuerte (72/100)

añadir Usuario a Base de Datos

Usuario: ▼

Base de Datos: ▼

Ilustración 10: Formulario de MySQL

Gracias al formulario ofrecido, creamos el usuario `appnware_uoc` al que le asignaremos la base de datos `appnware_magico`. De esta manera obtuve la base de datos preparada para comenzar a crear las tablas de la aplicación y accesible gracias al usuario creado.

A continuación, a través del script PHPMyAdmin, que se encuentra en el menú de bases de datos de cPanel y utilizando el mismo login creado para el usuario `appnware_uoc`, podemos administrar la base de datos creada y se crearon las tablas, según lo establecido en el apartado de especificación de la base de datos.

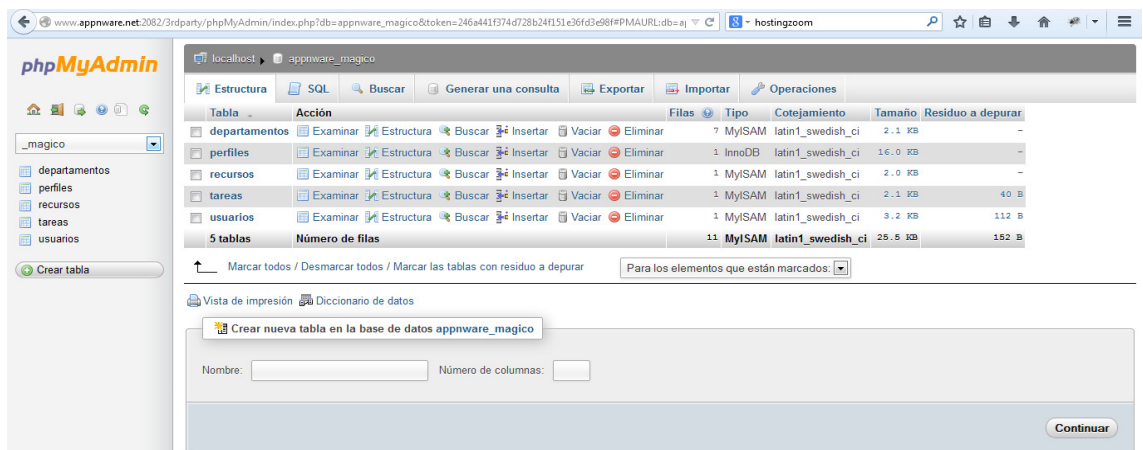


Ilustración 11: Ejemplo de PHPMyadmin

Una vez creada la base de datos bastó con subir los ficheros de la aplicación al hosting a través del cliente de FTP Fillezilla para tener la aplicación funcionando al completo.

Para acceder a la aplicación se debe entrar en <http://www.appnware.net/magico> y utilizar el Login “Admin” con contraseña “admin”, accediendo de esta manera en modo Administrador.

2.6 Valoración económica del proyecto

Llevar a cabo un proyecto de estas características no debe suponer un gran esfuerzo económico para una empresa. Como costes directos tenemos los siguientes:

- Hosting: Un servidor virtual como el utilizado para alojar el proyecto supone unos 11€/mes si se contrata en empresas norteamericanas.
- Puestos de trabajo: No se exige unos requerimientos elevados para poder trabajar con Magico, por lo que cualquier ordenador con navegador de internet actual puede ser válido. El coste aproximado puede rondar unos 300 euros/puesto.
- Línea de Internet: Existen líneas de Internet en el mercado con velocidades aceptables por unos 60€/mes, las cuales pueden ser adecuadas para trabajar con Magico. La elección de la línea dependerá del uso que se le de finalmente a la misma, el número de usuarios conectados a ella y la localización del punto de acceso.

- Salario del personal de desarrollo: Este coste irá en función del personal a contratar (Jefe de proyecto, Programador, Diseñador Gráfico y otro personal implicado). En este caso ha bastado un Programador durante 6 meses para conseguir la consecución del proyecto. El coste medio de un programador en España ronda los 25500 euros brutos anuales.

El proyecto no conlleva gastos adicionales en licencias ya que se han escogido herramientas gratuitas y software libre por lo que, como podemos comprobar, el grueso del coste lo representa el propio salario del personal.

Por este motivo, no considero necesario realizar estudio de viabilidad, ya que básicamente es como si se comprara la aplicación al coste del salario de programador ya que el resto de los costes no son representativos.

3. Conclusiones

El Trabajo Final de Grado ha supuesto una oportunidad única de poner en práctica gran parte de las competencias adquiridas a lo largo de la carrera sobre todo las adquiridas gracias al itinerario que escogí sobre Sistemas de Información como Integración de Sistemas de Información o Gestión de Proyectos Informáticos y las adquiridas en las asignaturas relativas a Ingeniería del Software y Programación. Ha supuesto investigación, gestión, planificación, depuración.

Me ha resultado bastante didáctico como en diversas ocasiones había que volver al análisis de requisitos, o incluso la etapa de diseño, conforme se iba desarrollando el proyecto y encontrabas nuevas funcionalidades o simplemente porque, a la hora de la implementación, la decisión tomada inicialmente no era la más óptima para llevar a término en el proyecto. Todo ello a pesar de que durante las fases de análisis y diseño se intentó reflejar todos los requisitos de definición y funcionales con el fin de obtener un desarrollo fidedigno a la planificación especificada. Como suele ser habitual en los proyectos informáticos los cambios surgieron aunque, gracias a los estudios realizados en las dos primeras fases, fueron minimizados y no tuvieron repercusión apenas en la etapa de desarrollo y, por tanto, no influyeron en la planificación final y ejecución del proyecto.

Por otro lado, realizar el trabajo en el área de la Gestión del Conocimiento me ha enseñado que hay mucho trabajo por hacer a nivel empresarial en este sentido y, tal y conforme he podido comprobar con mi propia empresa, las necesidades existen y por lo tanto herramientas de este tipo podrían suponer una buena oportunidad comercial.

En este sentido me gustaría destacar que se trata de un área muy extensa y que puede crecer hasta límites insospechados. En el caso de este TFG me centré en la gestión documental, el seguimiento de tareas,

la red social dentro de la empresa y la gamificación, pero perfectamente el proyecto podría haber crecido hacia otros áreas como facturación, contabilidad, business intelligence, elaboración de informes dinámicos gestión de recursos humanos, agendas compartidas y sin fin de funcionalidades las cuales no desentonarían en un proyecto de este tipo.

En base a esto, la reflexión crítica que hago acerca del logro de los objetivos planteados inicialmente es bastante positiva, ya que se ha conseguido la esencia del objetivo del proyecto que era el de conseguir una gestión del conocimiento aplicada a la empresa. Se han conseguido todos los objetivos a excepción de la gestión de recursos empresariales que, si bien con la aplicación podemos asignar recursos a los departamentos con el fin de tener un listado/inventario de los mismos, si que me hubiese gustado poder profundizar sobre la asignación de los mismos a las tareas empresariales y por tanto controlar su disponibilidad. A pesar de ello, sigo considerando la consecución del proyecto como positiva, ya que desde el principio era consciente, junto con mi consultora Margarita Hospedales, de que mi planteamiento acerca del área del proyecto era bastante ambicioso, aunque no obstante nos pareció muy interesante llevarlo a cabo según la planificación inicial.

Por este motivo, la razón de que características como la gestión de recursos, los perfiles de usuario o el sistema de comentarios de tareas y documentos no hayan sido logrados al cien por cien, no ha sido otra si no el tiempo de desarrollo que implicaban. Debido a esto se decidió “sacrificar” estas funcionalidades a favor de la consecución global del proyecto. Escribo entre comillas sacrificar porque a pesar de que no se haya implementado completamente esas características, si que se han desarrollado las soluciones susceptibles de ser utilizadas para su consecución, por ejemplo el muro podría reutilizarse en las secciones de documentos y tareas para el sistema de comentarios, y/o el sistema de asignación de recursos puede lograrse de la misma forma que la asignación de personal a las actividades de las tareas.

En relación a ello, considero acertada la planificación y metodología empleada, teniendo en cuenta que ya de por sí, venía determinada por la propia planificación de la asignatura TFG en el semestre académico.

Las líneas de trabajo futuro que no se han podido explorar en este proyecto han sido las características no conseguidas de las que he hablado anteriormente y las funcionalidades adicionales también comentadas como integración de la facturación, contabilidad, gestión de recursos humanos y similares. Este área es susceptible de incorporar muchas y diversas líneas de trabajo. Por ejemplo, sistemas como JYRA de Atlassian para la gestión de proyectos podría ser una pequeña sección de Magico, sin embargo, cuando hablamos de JYRA estamos hablando de un superventas, lo cual denota el grado de escalabilidad del proyecto.

A nivel técnico se podría trabajar en mejorar el diseño, que aunque está bastante logrado, no soy especialista en dicha competencia. Podríamos mejorar la usabilidad, adaptar la aplicación a dispositivos móviles lo que dotaría de movilidad a las labores de gestión del conocimiento desarrolladas y otras mejoras como optimización de los tiempos y similares.

Por último, expresar que el área del TFG ofrece muchas posibilidades y recomendaría seguir ofreciéndola ya que entre los productos desarrollados por los estudiantes se puede llegar a crear una aplicación bastante interesante.

Igualmente, no podría finalizar esta exposición sin mencionar la calidad de los materiales de la UOC que he podido consultar, tanto en biblioteca como en apuntes, para el desarrollo de este proyecto, lo aprendido durante estos años gracias a la metodología UOC y las orientaciones sobre el TFG por parte de mi consultora.

4. Glosario

- **Aplicación Web:** En la ingeniería de software se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.
- **Dominio:** Un dominio de Internet es una red de identificación asociada a un grupo de dispositivos o equipos conectados a la red Internet.
- **FTP:** (siglas en inglés de *File Transfer Protocol*, 'Protocolo de Transferencia de Archivos') en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.
- **Hosting:** El alojamiento web (en inglés *web hosting*) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web. Es una analogía de "hospedaje o alojamiento en hoteles o habitaciones" donde uno ocupa un lugar específico, en este caso la analogía alojamiento web o alojamiento de páginas web, se refiere al lugar que ocupa una página web, sitio web, sistema, correo electrónico, archivos etc. en internet o más específicamente en un servidor que por lo general hospeda varias aplicaciones o páginas web.
- **Información:** En sentido general, la información es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.
- **Intranet:** Una intranet es una red informática que utiliza la tecnología del Protocolo de Internet para compartir información, sistemas operativos o servicios de computación dentro de una organización

- **Red social:** Un servicio de red social es un medio de comunicación social que se centra en encontrar gente para relacionarse en línea. Están formadas por personas que comparten alguna relación, principalmente de amistad, mantienen intereses y actividades en común, o están interesados en explorar los intereses y las actividades de otros.

5. Bibliografía

- Wikipedia: <http://www.wikipedia.org> (Diversas consultas desde Febrero de 2014 hasta Junio de 2014)
- Biblioteca de la UOC: Visualización y Lectura de distintos TFG de otros alumnos de la Universidad.
- Mis Materiales de la UOC: Apuntes de TFG, Integración de Sistemas Informáticos, Gestión de Proyectos Informáticos, Ingeniería del Software.
- API JQuery: <http://api.jquery.com/> (Diversas consultas desde Febrero de 2014 hasta Junio de 2014)
- Javascript and HTML5 Now. Kyle Simpson (30 de Junio de 2012). Versión Kindle.
- Desarrollo web con PHP y MySQL. Juan Diego Gutiérrez Gallardo (20 de Julio de 2012).
- "Enterprise Integration", Kent Sandoe, Gail Corbitt y Raymond Boykin (2 de abril de 2001).

6. Anexos

- PAC 1
- PAC 2
- PAC 3

- Url de la aplicación: <http://www.appnware.net/magico>
- Código fuente de la aplicación
- Vídeo presentación